



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 21 de marzo de 2023

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

JUAN DAVID TRUJILLO SUAREZ, con C.C. No. 1075315335,

JOSE THOMAS PERES RIVERA, con C.C. No. 1075319499,

Autor(es) de la tesis y/o trabajo de grado titulado PROPUESTA DE STARTUP NEEDER EN TORNO A PROTOTIPO WEB QUE ENLACE COMPONENTES DE RED SOCIAL presentado y aprobado en el año 2023 como requisito para optar al título de INGENIERO DE SOFTWARE; Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales "open access" y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Firma:

EL AUTOR/ESTUDIANTE:

Firma:

Vigilada Mineducación



**UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS**



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

2 de 2

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 4
---------------	---------------------	----------------	----------	-----------------	-------------	---------------	---------------

TÍTULO COMPLETO DEL TRABAJO: PROPUESTA DE STARTUP NEEDER EN TORNO A PROTOTIPO WEB QUE ENLACE COMPONENTES DE RED SOCIAL Y MARKETPLACE

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
TRUJILLO SUAREZ	JUAN DAVID
PEREZ RIVERA	JOSE THOMAS

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
ZAMBRANO SANJUAN	JUAN PABLO

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
N/A	N/A

PARA OPTAR AL TÍTULO DE: INGENIERO DE SOFTWARE

FACULTAD: INGENIERIA

PROGRAMA O POSGRADO: PROGRAMA DE INGENIERIA DE SOFTWARE

CIUDAD: NEIVA

AÑO DE PRESENTACIÓN: 2023






NÚMERO DE PÁGINAS: 348

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías ___ Grabaciones en discos ___ Ilustraciones en general Grabados ___ Láminas ___
Litografías ___ Mapas ___ Música impresa ___ Planos ___ Retratos ___ Sin ilustraciones ___ Tablas o Cuadros

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS					   		
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						SC 7384-1	SA-CERE 197326
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 4	

SOFTWARE requerido y/o especializado para la lectura del documento: N/A

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>
1. Emprendimiento	Entrepreneurship
2. Contenido Multimedia	Multimedia Content
3. Prototipo de Software	Software Prototype
4. Patrón de Arquitectura	Architectural Pattern
5. Ciclo de vida del Software	Software Life Cycle

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El presente documento describe el proceso de desarrollo de la propuesta de creación de empresa tecnológica (en formato de startup), presentando un modelo de negocio a través de productos ofrecidos mediante un prototipo web. Combinando las características de un Marketplace de contenido multimedia y una red social.

Para la ideación del emprendimiento, se usó la guía metodológica del centro de emprendimiento. A partir de ella, se estructuraron los elementos organizacionales, legales y financieros. Por otro lado, el proceso de ingeniería se realizó en la construcción del prototipo inicial. En el alcance del proyecto se contemplaron 5 etapas, de las cuales solo fue desplegada la primera, y las dos siguientes diseñadas. Las 2 etapas restantes se planearon para un futuro lejano. Gracias al uso de Scrum y Kanban, se organizaron los requerimientos efectivamente, llevando a cabo el ciclo de vida del software de forma óptima. Por otra parte, mediante la aplicación de patrones de diseño, herramientas de modelado UML y el uso del



CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 4
---------------	---------------------	----------------	----------	-----------------	-------------	---------------	---------------

patrón de arquitectura (MVT) se estructuró un prototipo escalable, a través del framework Django, el cual permitió desarrollar el producto sin complicaciones. Así mismo, por el uso de Bootstrap el desarrollo frontend se realizó rápidamente y se logró optimizar gracias a PurgeCSS.

Para concluir gracias a los índices económicos encontrados, como la tasa interna de Oportunidad y la relación costo beneficio, se obtuvo que el proyecto será sustentable en el tiempo y su ejecución es viable, por lo que se espera que se siga desarrollando en un futuro.

ABSTRACT: (Máximo 250 palabras)

This document describes the development process of the proposal for the creation of a technological company (in startup format), presenting a business model through products offered from a web prototype. Combining the characteristics of a multimedia content marketplace and a social network.

For the ideation of the business model, the methodological guide of the entrepreneurship center was used. From it, the organizational, legal, and financial elements were structured. On the other hand, the engineering process was carried out during the construction of the initial prototype. The scope of the project contemplated five stages, of which only the first was deployed and the following two were designed. The remaining two stages were planned for the distant future. Thanks to the use of Scrum and Kanban, the requirements were organized effectively, carrying out the software life cycle in an optimal way. On the other hand, through the application of design patterns, UML modeling tools, and the use of the architecture pattern (MVT), a scalable prototype was structured through the Django framework, which allowed the product to be developed without complications. Likewise, through the use of Bootstrap, the frontend development was carried out quickly and optimized thanks to PurgeCSS.

To conclude, thanks to the economic indexes found, such as the internal rate of opportunity and the cost-benefit ratio, it was determined that the project will be sustainable over time and that its execution is viable, so it is expected to be further developed in the future.

APROBACION DE LA TESIS

Nombre Presidente Jurado: FERNANDO ROJAS ROJAS

Vigilada Mineducación



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	4 de 4
---------------	---------------------	----------------	----------	-----------------	-------------	---------------	---------------

Firma: *ferley*

Nombre Jurado: FERLEY MEDINA ROJAS

Firma: *Ferley Medina Rojas*

Nombre Jurado: ALVARO JAVIER GOMEZ LAMILLA

Firma: *Alvaro Gomez Lamilla*



UNIVERSIDAD
SURCOLOMBIANA

**UNIVERSIDAD SURCOLOMBIANA
PREGRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INGENIERÍA**

**PROPUESTA DE STARTUP NEEDER EN TORNO A
PROTOTIPO WEB QUE ENLACE COMPONENTES
DE RED SOCIAL Y MARKETPLACE**

**PRESENTADO PARA OPTAR EL TÍTULO DE:
INGENIERO DE SOFTWARE**

PRESENTA:

**JUAN DAVID TRUJILLO SUÁREZ
JOSE THOMAS PEREZ RIVERA**

**Tutor: Juan Pablo Zambrano
Colombia, Neiva - Huila**

Resumen:

El presente documento describe el proceso de desarrollo de la propuesta de creación de empresa tecnológica (en formato de startup), presentando un modelo de negocio a través de productos ofrecidos mediante un prototipo web. Combinando las características de un Marketplace de contenido multimedia y una red social.

Para la ideación del emprendimiento, se usó la guía metodológica del centro de emprendimiento. A partir de ella, se estructuraron los elementos organizacionales, legales y financieros. Por otro lado, el proceso de ingeniería se realizó en la construcción del prototipo inicial. En el alcance del proyecto se contemplaron 5 etapas, de las cuales solo fue desplegada la primera, y las dos siguientes diseñadas. Las 2 etapas restantes se planearon para un futuro lejano. Gracias al uso de Scrum y Kanban, se organizaron los requerimientos efectivamente, llevando a cabo el ciclo de vida del software de forma óptima. Por otra parte, mediante la aplicación de patrones de diseño, herramientas de modelado UML y el uso del patrón de arquitectura (MVT) se estructuró un prototipo escalable, a través del framework Django, el cual permitió desarrollar el producto sin complicaciones. Así mismo, por el uso de Bootstrap el desarrollo frontend se realizó rápidamente y se logró optimizar gracias a PurgeCSS.

Para concluir gracias a los índices económicos encontrados, como la tasa interna de Oportunidad y la relación costo beneficio, se obtuvo que el proyecto será sustentable en el tiempo y su ejecución es viable, por lo que se espera que se siga desarrollando en un futuro.

Palabras claves: *Modelo de negocio, startup, prototipo, requerimientos, diseño, desarrollo, Python, Django.*

Abstract:

This document describes the development process of the proposal for the creation of a technological company (in startup format), presenting a business model through products offered from a web prototype. Combining the characteristics of a multimedia content marketplace and a social network.

For the ideation of the business model, the methodological guide of the entrepreneurship center was used. From it, the organizational, legal, and financial elements were structured. On the other hand, the engineering process was carried out during the construction of the initial prototype. The scope of the project contemplated five stages, of which only the first was deployed and the following two were designed. The remaining two stages were planned for the distant future. Thanks to the use of Scrum and Kanban, the requirements were organized effectively, carrying out the software life cycle in an optimal way. On the other hand, through the application of design patterns, UML modeling tools, and the use of the architecture pattern (MVT), a scalable prototype was structured through the Django framework, which allowed the product to be developed without complications. Likewise, through the use of Bootstrap, the frontend development was carried out quickly and optimized thanks to PurgeCSS.

To conclude, thanks to the economic indexes found, such as the internal rate of opportunity and the cost-benefit ratio, it was determined that the project will be sustainable over time and that its execution is viable, so it is expected to be further developed in the future.

Keywords: *Business model, startup, prototype, requirements, design, development, Python, Django.*

TABLA DE CONTENIDO

TABLA DE CONTENIDO	4
I. INTRODUCCIÓN	12
II. PRESENTACIÓN DEL PROYECTO	14
2.1 Planteamiento/Identificación del problema	14
2.1.1 Planteamiento del problema	14
2.1.2 Formulación del problema	15
2.2 Objetivos	15
2.2.1 Objetivo general	15
2.2.2 Objetivos específicos	15
2.3 Alcance	16
2.4 Justificación	17
III. MARCO TEÓRICO	18
3.1 Modelo de negocio	18
3.1.1 Propuesta de valor	18
3.1.2 Mercado Potencial	20
3.1.3 Entorno competitivo	20
3.1.4 Canales de distribución y comercialización	21
3.1.5 Sistema de fidelización	22
3.1.6 Inversiones y Costos	23
3.1.7 Rentabilidad	24
3.2 Estructuración del proyecto	24
3.2.1 Estudio de mercado	25
3.2.1.1 Estudio de demanda	25
3.2.1.2 Estudio de oferta	26
3.2.1.3 Estudio de comercialización	26
3.2.1.4 Diseño de marca	26
3.2.2 Análisis organizacional	28
3.2.2.1 Misión	28
3.2.2.2 Visión	29
3.2.2.3 Organigrama	30

3.2.2.4	<i>Manual de funciones</i>	31
3.2.2.5	<i>Requisitos de contratación</i>	32
3.2.2.6	<i>Objetivos estratégicos</i>	34
3.2.2.7	<i>Valores corporativos</i>	35
3.2.3	Análisis Legal	36
3.2.3.1	<i>Camara de Comercio de Colombia</i>	36
3.2.3.2	<i>Homonimia</i>	37
3.2.3.3	<i>Figura Juridica S.A.S</i>	38
3.2.3.3	<i>Responsabilidades Tributarias de la SAS</i>	40
3.2.3.4	<i>Leyes de la plataforma</i>	41
3.2.4	Módulo de desarrollo	42
3.2.4.1	<i>Servicio de Red Social en Internet</i>	43
3.2.4.2	<i>Marketplace de Contenido Multimedia</i>	46
3.2.4.3	<i>Ciclo de vida del software</i>	47
3.2.4.4	<i>Agile</i>	49
3.2.4.5	<i>SCRUM</i>	51
	<i>Equipo en Scrum</i>	51
	<i>Artefactos</i>	52
	<i>Sprint</i>	53
3.2.4.6	<i>Analisis de las Historias de Usuario</i>	54
	<i>Épicas</i>	54
	<i>Historia de Usuario</i>	55
	<i>Priorización de las Historias de Usuario</i>	56
3.2.4.7	<i>Organizacion de los requerimientos: Kanban</i>	57
3.2.4.8	<i>Diseño de los requerimientos</i>	59
	<i>Diseño enfocado a objetos</i>	59
	<i>Diagrama de clases</i>	60
	<i>Diagrama de secuencia</i>	61
	<i>Diagrama de casos de uso</i>	62
	<i>Diseño UX/UI</i>	62
	<i>Metodología de diseño UX/UI</i>	63
	<i>Esquemas de organización</i>	63
	<i>Estructuras organizativas</i>	64

Patrones de diseño UI	65
<i>Diagrama de despliegue</i>	73
3.2.4.9 <i>Arquitectura de la aplicacion</i>	73
<i>Patrón MVC</i>	74
3.2.4.10 <i>Desarrollo Web</i>	75
<i>Desarrollo Frontend</i>	76
La estructura básica: HTML	76
Estilos: CSS	76
Librerías	76
Bootstrap librería para los estilos	77
PurgeCSS Optimización de CSS	77
Lenguajes de Programación	77
Animación visual: Javascript	77
JQuery	78
<i>Desarrollo Backend</i>	79
Python	79
Clean code: Flake8	79
Frameworks	80
Framework web: Django	80
Patrón: MVT	80
Base de Datos	83
Bases de datos no relacionales	83
Bases de datos relacionales	83
Sistema Gestor de Base de Datos	85
PostgreSQL	86
APIs de Terceros	87
3.2.4.11 <i>Etapas del Testing</i>	87
Pruebas de Unitarias	88
3.2.4.12 <i>Despliegue</i>	89
3.2.4.13 <i>Control de versiones</i>	89
3.2.5 <i>Estimación de ingresos, costos y gastos</i>	90
3.2.5.1 <i>Analisis de Ingresos</i>	90
3.2.5.2 <i>Proyección de gastos</i>	91

3.2.5.2.1 <i>Gastos de Personal</i>	91
3.2.5.2.2 <i>Gastos Depreciación de Bienes Fijos</i>	91
3.2.5.2.3 <i>Gastos Generales</i>	91
3.2.5.3 <i>Inversiones Iniciales</i>	92
3.2.5.4 <i>Flujo de Caja</i>	92
3.2.5.5 <i>Estado de Resultado</i>	92
3.2.5.6 <i>Balance</i>	92
3.2.6 Evaluación social, ecológica y financiera	92
3.2.6.1 <i>Análisis de Rentabilidad</i>	93
3.2.6.1.1 <i>Tasa Interna de Retorno (TIR)</i>	93
3.2.6.1.2 <i>Valor Presente Neto (VPN)</i>	93
3.2.6.1.3 <i>Tasa Interna de Oportunidad (TIO)</i>	93
3.2.6.1.4 <i>Relación Beneficio/Costo (RBC)</i>	93
3.2.6.1.5 <i>Tasa Verdadera de Rentabilidad (TVR)</i>	94
3.2.6.2 <i>Impacto Social</i>	94
3.2.6.3 <i>Impacto Ambiental</i>	94
IV. RESUMEN DE METODOLOGÍA	95
4.1 <i>Propuesta de Creación de Empresa</i>	95
4.2 <i>Prototipo de Software</i>	97
V. MODELO DE NEGOCIO	103
5.1 Propuesta de valor	103
5.1.1 Desarrollo de la idea	103
5.1.2 Productos o Servicios	104
5.1.2.1 Creador de Contenido	104
5.1.2.2 Usuario Consumidor	105
5.1.3 Diferenciación del producto	106
5.1.4 Experiencia del equipo	106
5.2 Mercado Potencial	107
5.2.1 Sector económico de la propuesta	107
5.2.2 Investigación del sector	107
5.2.3 Clientes o beneficiarios	108
5.2.4 Tamaño del mercado	109
5.3 Definición del entorno competitivo	110

5.3.1 Competidores	110
5.3.2 Alianzas claves	111
5.3.3 Precio de venta del producto o servicio	111
5.4 Planteamiento de los canales de distribución y cobertura geográfica	112
5.4.1 Canal de distribución	112
5.4.2 Cobertura geográfica	112
5.4.3 Sistema de fidelización	112
5.5 Estimación de inversiones, ingresos y costos	113
5.5.1 Tipo de inversiones	113
5.5.2 Magnitud de las inversiones	113
5.5.2.1 Inversiones diferidas	114
5.5.2.2 Capital de Trabajo	114
5.5.2.3 Inversión Fija	115
5.5.2.4 Total Inversión Inicial Estimada (Plan Inicial)	115
5.5.3 Financiamiento de las inversiones	115
5.5.4 Ingresos estimados iniciales	116
5.6 Impacto	117
5.6.1 Impacto en el sector	117
5.6.2 Generación de empleos	118
5.7 Experticia del equipo	118
5.7.1 Disciplinas del equipo de trabajo	118
5.7.2 Experiencia del equipo	118
VI. EJECUCIÓN DEL PROYECTO	119
6.1 Investigación de mercados	120
6.1.1 Estudio de demanda	120
6.1.1.1 Estudio cualitativo	120
6.1.1.1.1 Introducción	120
6.1.1.1.2 Objetivos de la entrevista	121
6.1.1.1.3 Justificación de la encuesta	121
6.1.1.1.4 Metodología a aplicar	121
6.1.1.1.5 Resultados Esperados	122
6.1.1.1.6 Procedimiento y resultados	122
6.1.1.1.7 Conclusiones	132

6.1.1.2 Estudio Cuantitativo	133
6.1.1.2.1 Introducción	133
6.1.1.2.2 Objetivos de la encuesta	133
6.1.1.2.3 Justificación de la encuesta	134
6.1.1.2.4 Metodología a aplicar	134
6.1.1.2.5 Resultados Esperados	134
6.1.1.2.6 Procedimiento y resultados	134
6.1.1.2.7 Conclusiones	143
6.1.2 Estudio de oferta	143
6.1.2.1 Análisis de competencia	144
6.1.2.2 Conclusiones, semejanzas y diferencias	146
6.1.2.3 Diferenciación del producto	148
6.1.3 Estudio de comercialización	150
6.1.4 Desarrollo de la marca	151
6.1.4.1 Definición del nombre	151
6.1.4.2 Diseño del logo	151
6.1.4.3 Análisis cromático	152
6.1.4.4 Analisis tipografico	152
6.1.4.5 Slogan	153
6.2 Componente Organizacional	153
6.4.1 Misión	153
6.4.2 Visión	154
6.4.3 Organigrama	154
6.4.4 Manual de funciones	155
6.4.5 Requisitos de contratación	156
6.4.6 Objetivos estratégicos	157
6.4.7 Valores corporativos	158
6.3 Componente Legal	159
6.3.1 Homonimia	159
6.3.2 Modelo de Constitución de la Sociedad	160
6.3.3 Políticas de Privacidad y Términos y Condiciones	166
6.4 Componente Técnico	167
6.4.1 Análisis	167

6.4.1.1	<i>Definición de Epicas</i>	167
6.4.1.2	<i>Desarrollo de las Historias de Usuario</i>	169
6.1.4.3	<i>Planeación de Sprints</i>	172
	<i>Etapa 1</i>	174
	<i>Etapa 2</i>	176
	<i>Etapa 3</i>	178
6.1.4.4	<i>Story Points y Sprint Burndown</i>	180
6.1.4.4	<i>Tareas</i>	192
6.4.2	<i>Diseño de la aplicación</i>	193
6.4.2.1	<i>Diagrama de Caso de Uso</i>	193
6.4.2.2	<i>Diagrama de Secuencias</i>	195
6.4.2.3	<i>Diagrama de Clases</i>	197
6.4.2.4	<i>Diseño de Base de Datos</i>	199
6.4.2.5	<i>Arquitectura del sistema</i>	210
6.4.2.6	<i>Diagrama de despliegue</i>	212
6.4.2.7	<i>Diseño de Historias de Usuario</i>	213
	<i>Etapa 1</i>	215
	<i>Etapa 2</i>	221
	<i>Etapa 3</i>	230
6.4.3	<i>Desarrollo</i>	238
6.4.3.1	<i>Preparacion del entorno de trabajo</i>	238
6.4.3.1	<i>Sprint 1</i>	241
6.4.3.1	<i>Sprint 2</i>	254
6.4.3.1	<i>Sprint 3</i>	261
6.4.4	<i>Testing</i>	272
6.4.4.1	<i>Plan de Pruebas</i>	272
6.4.4.2	<i>Pruebas Unitarias</i>	277
6.4.4.2.1	<i>Pruebas Unitarias U-1</i>	277
6.4.4.2.2	<i>Pruebas Unitarias U-2</i>	282
6.4.4.2.3	<i>Pruebas Unitarias U-6</i>	285
6.4.4.3	<i>Resultados</i>	288
6.4.5	<i>Despliegue</i>	293
6.4.5.1	<i>Preparación del proyecto para el despliegue</i>	294

6.4.5.1.1 <i>Purge CSS</i>	294
6.4.5.1.2 <i>Configuración de la base de datos y variables de entorno</i>	296
6.4.5.1.3 <i>Configuración del CDN</i>	298
6.4.5.2 <i>Ejecución del despliegue</i>	299
6.5 Proyección de ingresos, costos y gastos	306
6.5.1 Análisis de Ingresos	306
6.5.2 Proyección de gastos	308
6.5.2.1 <i>Gastos de Personal</i>	308
6.5.2.2 <i>Gastos Depreciacion de Bienes Fijos y Amortizacion de Inversiones Diferidas</i>	308
6.5.2.3 <i>Gastos Generales</i>	309
6.5.3 Inversiones Iniciales	309
6.5.4 Flujo de caja	311
6.5.5 Estado de Resultado	312
6.5.6 Balance	313
6.6 Evaluación financiera, social, ambiental	315
6.6.1 Análisis Financiero de Rentabilidad	315
6.6.2 Impacto Social	316
6.6.3 Impacto Ambiental	316
VII. CONCLUSIONES Y FUTUROS TRABAJOS	317
Referencias bibliográficas	318

I. INTRODUCCIÓN

En el último siglo, el internet se ha vuelto una de las tecnologías más decisivas para el ser humano moderno, sin embargo esta herramienta no es algo del milenio actual, debido a que ya había sido desarrollada y fue presentada inicialmente como *Arpanet* en la década de los 70 (*Abbate, 1999*). Esta constaba de redes interconectadas a partir de *nodos* situados en distintas universidades de la costa occidental de los Estados Unidos, siendo luego usada por el sistema de seguridad de este mismo país. No obstante, esta no fue comercializada hasta 1990, cuando el ministerio de comercio liberó su uso. Desde este momento el crecimiento fue exponencial, en 1995 ya se contaba con aproximadamente 84 países conectados y 48 millones de usuarios. Posteriormente con el desarrollo de la *WWW* (World Wide Web), las cifras incrementaron notoriamente (*Cañedo Andalia, 2004*). Según datos del Banco Mundial cerca del 60% de la población global cuenta al día de hoy con internet, presentando un incremento de más del 300% en menos de 20 años (*Banco Mundial, 2021*).

Esto conllevó a que se crearan diferentes empresas y proyectos como las redes sociales y sitios de ventas en línea. A principios del año 2000, apareció una popular empresa llamada MySpace, esta permitía la creación de perfiles individualizados y personalizables, que reflejaban los gustos y pasiones de cada usuario, y que además permiten el intercambio de opiniones y contenidos [4]. Asimismo, Facebook surgió como producto de este auge de las redes sociales, la plataforma “azul” que en un inicio solo permitía realizar publicaciones y enviar mensajes como una especie de *blog*, hoy en día da la posibilidad de vender productos físicos o multimedia en línea desde su *marketplace*, hacer *streaming* en vivo o hasta jugar dentro de su plataforma.

La aparición de estos entornos atrajo a personas y figuras públicas que buscan tener un grupo de seguidores que quieran invertir tiempo o dinero en ellos, ya sea por apoyarlos o por sentirse notados por estos mismos. Esto consiguió que surgieran servicios digitales similares a las redes sociales enfocados en ofrecer perfiles para que cada *creador de contenido* pueda vender *contenido multimedia* a sus comunidades. Páginas como *Only Fans* o *Patreon* se dedican a ofrecer un catálogo de perfiles, en los cuales cada usuario puede seguir o apoyar monetariamente mediante una *suscripción* a su creador favorito, y de esta forma consumir el contenido de estos perfiles. Sin embargo este modelo implementado puede ser esclavizante para el usuario y genere resistencia a la hora de registrarse, por lo que es necesario que surjan

diferentes ofertas en el mercado en las que el cliente pueda registrarse sin la necesidad de agregar métodos de pago, para interactuar y comprar por unidad lo que desee y con la exclusividad que se espera.

Gracias a esto surgió **Needer**. Esta es una startup que ofrece una plataforma que fusiona una red social y un marketplace. Mediante la cual los usuarios pueden registrarse e interactuar con sus creadores de contenido favoritos, ya sea siguiéndolos o adquiriendo sus productos por *unidad*, transformando el modelo de negocio tradicional a partir de suscripciones, adquiriendo una filosofía en pro del consumidor y facilitando la interacción entre las dos partes, creando así una comunidad sólida en torno a los “**Needers**”.

II. PRESENTACIÓN DEL PROYECTO

2.1 Planteamiento/Identificación del problema

2.1.1 Planteamiento del problema

El término contenido multimedia, se utiliza para describir cualquier objeto digital que quiera expresar o comunicar información. En el mercado del entretenimiento, el contenido multimedia es una clave importante, pues abarca desde texto, vídeos, videojuegos, diseños, fotos, libros, sonidos, música, entre otros. Actualmente existen plataformas que se enfocan en ofrecer servicios a través de los cuales las personas pueden crear perfiles para realizar publicaciones o interactuar con ellas, o también para adquirir o comercializar contenido multimedia (específicamente fotos, vídeos, audios, o realizar Streaming en vivo), con el fin de entretener y crear una comunidad en torno a ellos.

Hay empresas que brindan este tipo de herramientas, para todo público. Un ejemplo es Only Fans, empresa líder en el segmento, enfocada en ofrecer perfiles tanto de pago como gratuitos para las personas, por medio del marketplace de contenido multimedia o red social, junto a otras como Facebook, Patreon y Ko-fi, diseñadas para ofrecer este tipo de servicios.

Algunas de estas compañías cuentan con un modelo de negocio simple pero esclavizante, el cual consiste en comercializar por medio de una **suscripción**, garantizando el acceso al usuario a todo el contenido de un perfil (en caso de que el creador tenga el perfil de pago). Para poder ingresar y suscribirse para consumir el contenido, deben en un principio asociar al menos una tarjeta de crédito a la plataforma para utilizarla y realizar los pagos, este ha sido un punto clave del descontento para muchos usuarios, por múltiples razones.

En primera medida, los usuarios deben asociar una tarjeta para hacer uso de la plataforma y consumir los perfiles, además, así los usuarios no hayan consumido el contenido de un perfil por más de un mes, si tienen activa la suscripción de dicho perfil, se les seguirá cobrando por el mismo. En segunda instancia, no hay forma de consumir el contenido por unidad, evitando la suscripción mensual a un perfil, causando la falta de variedad en el consumo. Cómo adicional, se ha descubierto una necesidad extra de los usuarios, ya que estos requieren un sistema en el cual se pueda adquirir contenido multimedia personalizado exclusivamente para ellos, sin la necesidad de recurrir a un intermediario.

En resumen, no existe una plataforma que, además de ofrecer los mismos servicios presentes en una red social y marketplace para los usuarios, pueda brindar esa garantía que estos necesitan a la hora de realizar una compra, evitando el pago de contenido que no consumen, permitiendo la compra por unidad y brindando la posibilidad de personalizar los productos, siendo más asequible y llamativo para los usuarios.

2.1.2 Formulación del problema

¿Cómo desarrollar una propuesta de creación de empresa, que ofrezca un prototipo web que enlace elementos de red social y marketplace, para la publicación y venta de contenido multimedia en base al modelo de negocio de venta sobre demanda?

2.2 Objetivos

2.2.1 Objetivo general

Plantear una propuesta de creación de empresa que diseñe y desarrolle un prototipo inicial que enlace elementos de una red social y marketplace, aplicando un modelo de negocio de venta por unidad el cual supla la necesidad de los usuarios.

2.2.2 Objetivos específicos

- Diseñar un modelo de negocio que garantice la rentabilidad del proyecto y resuelva las necesidades del cliente, mediante la retrospectiva de la información recopilada en los estudios de mercado.
- Estructurar una propuesta de creación de empresa que ofrezca los servicios generados en base al modelo de negocio, empleando la metodología para la propuesta de creación de empresa.
- Analizar las posibles funcionalidades que se encontraran en el servicio a ofrecer, estimando los requerimientos necesarios para su construcción, haciendo uso de las metodologías ágiles para la gestión de proyectos.

- Diseñar los elementos técnicos del producto, implementando los requerimientos planteados en el análisis de los servicios, aplicando los patrones y metodologías de diseño requeridos.
- Desarrollar un prototipo inicial que contenga todos los elementos de red social y que sirva como atractor para los usuarios, en base a los diseños generados a partir de los requerimientos, haciendo uso de frameworks y tecnologías enfocadas al desarrollo web.

2.3 Alcance

Para el presente proyecto se espera diseñar un modelo de negocio para el producto, que integre elementos de una red social y marketplace de contenido multimedia. Permitiendo la venta de contenido multimedia por unidad y la creación de comunidades. Así mismo, se estructurará una propuesta de StartUp tecnológica alrededor del modelo, la cual contará con componentes organizacionales, legales y financieros, así como el diseño de marca y la filosofía de la empresa. Por último, se estima diseñar los aspectos técnicos del producto y desarrollar un prototipo inicial que atraiga a los usuarios, implementando los análisis generados a partir de los requerimientos.

La ejecución técnica del proyecto se dividirá en 5 etapas, la etapa inicial será la única presentada en el producto mínimo viable como proyecto de grado, puesto que la falta de miembros en el equipo representan una limitante para el desarrollo del proyecto. Esta etapa, contará con los componentes esenciales de red social, los cuales se espera atraigan a los usuarios para que se registren en la plataforma y de esta forma crear una base de personas para ofrecerles las demás funcionalidades. Para la segunda y tercera etapa, se llevará el ciclo hasta realizar los diseños correspondientes.

La segunda etapa contiene las funcionalidades iniciales de la plataforma de venta de contenido multimedia en la cual se aplicará el modelo de negocio encontrado, como lo es el registro y verificación como creador de contenido, creación de contenido para la venta, agregar productos a un carrito de compra y añadir métodos de pago. Por otro lado, el apartado secundario del marketplace y la moderación, harán parte de la tercera etapa. Por último, las 2 etapas finales se piensan aplicar en un futuro lejano, por lo que solo se fijarán como

características en el modelo de negocio, pero no serán analizadas ni diseñadas a nivel técnico por el momento; estas comprenden: Integración de pago con criptomonedas, y chat en vivo.

2.4 Justificación

Actualmente, el sector del entretenimiento se encuentra en constante crecimiento, permitiendo ser una gran fuente de ingresos para los creadores de contenido, los cuales son esenciales para atraer un público en específico, dispuestos a seguir a sus figuras, adquirir productos digitales o apoyar proyectos mediante donaciones. Cabe resaltar que, en este sector, existen plataformas que cuentan con servicios que dan la posibilidad de crear publicaciones, interactuar con ellas u ofrecer perfiles de pago por suscripción. Este modelo, permite al usuario poder interactuar con sus creadores de contenido preferidos o con el contenido digital realizado por ellos. Pero, obliga a las personas a registrar sus datos de pago al registrarse en la plataforma, no ofrece los precios ideales para los usuarios, y tampoco garantiza una comunicación directa entre la comunidad y los creadores de contenido.

En base a esto, se presenta la propuesta de creación de empresa en torno a la plataforma llamada **Needer**. Una plataforma que implementará un modelo de negocio diferente al tradicional, ya que permitirá a los usuarios la posibilidad de consumir contenido multimedia por unidad, siendo esta forma de venta la base del modelo de negocio a implementar. Además, integrará los componentes de red social, garantizando la creación de comunidades y atrayendo masas a la plataforma. Por otro lado, los productos en venta podrán ser ofrecidos como genéricos o personalizados, presentando estos últimos un precio estipulado de antemano por el creador y una descripción explicando a los usuarios que podrán personalizar del mismo.

III. MARCO TEÓRICO

Teniendo en cuenta el objetivo del proyecto, el cual se centra en el diseño de una propuesta de creación de empresa en torno al modelo de negocio, se plantean diferentes teorías de cómo desarrollar y estructurar tanto el modelo, el prototipo tecnológico inicial y la empresa misma. En base a la “*Guía metodológica para argumentar la idea primera aproximación de la idea al modelo de empresa*” del centro de emprendimiento, a los contenidos propuestos en su formato para la alternativa de grado “**Propuesta de Creación de Empresa**” y las asesorías recibidas, se extrae el modelo de desarrollo y base fundamental para toda la ideación empresarial y de mercado [5], se presentan dos secciones principales para la creación y ejecución de la propuesta: La ideación del modelo de negocio y la estructuración de la propuesta junto al desarrollo del proyecto.

3.1 Modelo de negocio

Se define modelo de negocio como: El plan de una empresa para generar ingresos y obtener beneficios. Según el Centro de Emprendimiento, para poder articular el modelo de negocio, es necesario analizar y plantear los siguientes aspectos:

- Propuesta de valor.
- Mercado potencial.
- Entorno competitivo.
- Canales de distribución y comercialización.
- Estimación de inversiones, ingresos y costos.

3.1.1 Propuesta de valor

Este comprende a un grupo de productos o servicios que satisfacen los requisitos de un segmento de mercado determinado [6].

Para delimitar la propuesta de valor, según la metodología definida, se debe primero identificar las necesidades, problemas o situaciones que se buscan satisfacer con la idea a desarrollar. Después, es preciso definir la idea central del modelo de negocio; Luego, se establecen cuáles serán los productos o servicios que se van a ofrecer dentro del marketplace.

Y por último, se deben encontrar las diferencias de los productos en la propuesta de valor con respecto a lo que se ofrece en el mercado.

Un producto, según la American Marketing Association (A.M.A por sus siglas en inglés), es especificado como: *"Un conjunto de atributos (características, funciones, beneficios y usos) que le dan la capacidad para ser intercambiado o usado. Usualmente, es una combinación de aspectos tangibles e intangibles. Así, un producto puede ser una idea, una entidad física (un bien), un servicio o cualquier combinación de los tres. El producto existe para propósitos de intercambio y para la satisfacción de objetivos individuales y de la organización"* [7].

En un principio es importante conceptualizar los términos de Red Social y Marketplace de Contenido Multimedia, esto dado que el entregable principal del presente proyecto es un prototipo inicial con funciones de red social, así como los diseños que enlazan estos elementos con el marketplace de contenido multimedia.

Una red social es un sitio web que tiene como finalidad ofrecer a los usuarios la oportunidad de poder comunicarse, relacionarse y compartir información con otros usuarios alrededor del mundo, generando comunidades en torno a gustos, personas e intereses en común [8].

Por otro lado, un marketplace es un sitio para realizar negocios en Internet, siendo un punto de encuentro entre empresas o personas compradoras y vendedoras de productos o servicios [9]. Por otro lado, contenido multimedia hace referencia a los archivos o programas digitales que representan información, tal como, imágenes, fotos, videos, audios, etc [10]. De tal forma se podría plantear que, un marketplace de contenido multimedia es un sistema en línea que permite comercializar archivos o programas digitales. Para el proyecto se estima que se intercambiarán únicamente imágenes, videos o audios, los cuales pueden tener diferentes fines y temáticas, como por ejemplo: Podcast Informativos, Imágenes de Cosplays, Videos exclusivos jugando videojuegos, Arte digital, Música, etc.

Para definir la interacción que habrá dentro de la plataforma, se debe categorizar esta según los agentes involucrados en cada interacción, a estos comúnmente se les hace mención a partir de una sigla:

- La sigla “A” corresponde a “Administración”, este engloba a los gobiernos y administraciones públicas.
- La sigla “B” hace referencia a las empresas por su traducción “Business” en inglés, este engloba a todo el sector empresarial.
- La sigla “C” se refiere a “Customer” o “Consumer”, señala a todos los clientes finales, en otras palabras, a quién va destinado el producto.

La interacción entre los usuarios dentro de la plataforma se espera sea C2C [11], es decir, la plataforma conectará a los clientes entre sí, donde cada quien podrá ofrecer, adquirir, publicar o interactuar con los productos multimedia que deseen.

3.1.2 Mercado Potencial

Una vez definida la idea principal, es necesario reconocer las posibles necesidades de los usuarios consumidores y creadores de contenido multimedia, y posteriormente cuantificar el mercado que potencialmente hará uso de la plataforma. Para esto, se debe realizar un estudio del sector económico al cual apunta el proyecto.

Según Douglas da Silva, Senior Web y SEO Especialista, el mercado potencial es el mercado total que tiene la capacidad de comprar y adquirir un producto o servicio específico. Se habla de mercado potencial, porque los usuarios tienen el potencial para comprarlo, sin embargo, esto no implica que todas las personas que conforman ese mercado vayan a hacerlo [12].

Con el fin de encontrar el mercado potencial, la guía menciona que: Se debe identificar el sector económico al que pertenece la idea principal, después, continúa el proceso realizando una investigación sobre este, reconociendo qué tan grande es y cuáles son los clientes probables que se encuentran en el mismo.

3.1.3 Entorno competitivo

La relación que hay entre la empresa con su entorno, es definido como entorno competitivo. Conocer este aspecto ayuda a identificar las fortalezas y debilidades de la empresa, así como las oportunidades y amenazas que le afectan dentro de su sector de mercado [13]. Este proceso

es la base sobre la que se diseñará la estrategia de mercado, para ello se debe de conocer o intuir:

- La naturaleza y el éxito de los cambios probables que pueda adoptar el competidor.
- Como el competidor responde a los posibles movimientos estratégicos que otras empresas puedan iniciar.
- La reacción y adaptación a los posibles cambios del entorno que puedan ocurrir de los diversos competidores.

Basándose en la metodología del centro de emprendimiento, para encontrar el entorno competitivo, se debe reconocer a los competidores en el mercado, también, es necesario comprender la diferencia entre los productos o servicios de estos, con el producto o servicio que se planea ofrecer. La guía propone además, la posibilidad de crear una alianza estratégica con algún competidor.

Esto tiene sus pros y sus contras, puesto que en algunos casos los competidores podrían tener una disminución marcada de compradores al realizar la alianza, o podrían tener un aumento marcado de compradores. Estas situaciones dependen mucho del entorno competitivo. Adicionalmente, se deben estimar los precios de venta de los posibles productos o servicios que se ofrecerán en la plataforma, definirlos es muy importante con el fin de contrastar los precios ofrecidos en las plataformas de los competidores directos, con los propuestos.

3.1.4 Canales de distribución y comercialización

Ahora bien, el paso a seguir está enfocado en conocer los canales de distribución y comercialización que tendrán los productos o servicios de la red social y el marketplace. Los canales son el conjunto de medios que utiliza la empresa para hacer llegar el producto desde el fabricante hasta el cliente final. Según indica el docente en Máster Oficial Online de Dirección y Gestión en Marketing Digital y Social Media, Nacho Somalo [14], el producto debe atravesar tres etapas antes de que el consumidor pueda hacerse con el:

- **Conocimiento:** El cliente debe de saber de la existencia de la empresa y saber que la empresa está dispuesta a venderle el producto que esa persona quiere.

- **Consideración:** La empresa debe conseguir que el cliente considere su producto cómo una de las mejores opciones para suplir sus necesidades.
- **Visita:** El cliente debe de tener la facilidad de poder adquirir el producto, ya sea por medio de una sucursal física o virtual.

El centro de emprendimiento plantea unas estrategias que son claves para identificar los canales de distribución y comercialización. Lo primero, es entender cómo llevar los productos o servicios a los clientes, esto es importante con el fin de comprender la facilidad que tienen ellos de acceder a los productos, ya sea de manera física o virtual.

También, se debe conocer la cobertura geográfica inicial de la empresa, tener este factor presente es de suma importancia, puesto que muchas personas no conocen la cobertura geográfica que su empresa pueda tener. Es decir, hasta dónde es capaz de cubrir la empresa un territorio para poder enviar u ofrecer sus productos o servicios.

3.1.5 Sistema de fidelización

Siguiendo con la metodología definida, ahora se deben presentar las estrategias de fidelización con los clientes, usuarios o beneficiarios. En este punto es necesario realizar un análisis de consumo de los clientes, con el fin de asegurar la lealtad de estos en un futuro.

Rosario Peiró expone al sistema de fidelización como, aquel sistema que se ofrece a los consumidores para potenciar las ventas de la empresa [15]. En la actualidad, muchas empresas utilizan estos sistemas de fidelización para poder crear una relación estrecha con sus clientes, aumentando de manera significativa las ventas por parte de la empresa y entregando ciertas recompensas a los clientes que cumplan con los estándares establecidos por el sistema. También, se genera una necesidad por parte de los clientes, puesto que ellos empezarían a realizar dichas actividades con la condición de obtener una remuneración a futuro.

Según Lorena Agüera, en su trabajo de grado final, titulado “*Estrategia de fidelización de clientes*”, habla sobre el sistema de fidelización a partir de “*Puntos por compra*”. Se detalla como un programa de acumulación de puntos tras la realización de compras de un importe

determinado. Estos se acumulan en una cuenta del usuario, y le permiten obtener ciertas ventajas a la hora de realizar una nueva compra [16].

En los últimos años las empresas han decidido aplicar este sistema mencionado por Lorena, pero conceptualizado de manera diferente, generando puntos a partir de la cantidad de ventas que cada usuario haya realizado dentro de sus perfiles. Esto con el fin de entregar un reconocimiento para aquellos que concreten una cierta cantidad de ventas, ya sean puntos que se puedan redimir por productos o servicios, o, reconocimiento al usuario a través de insignias y premios.

Un ejemplo de este modelo es el aplicado por la plataforma de ventas Mercado Libre, en la cual los usuarios reciben niveles tanto por sus compras como por sus ventas, donde estas últimas definen la reputación que tiene un vendedor. Creando un sistema de referencia en el cual los compradores puedan sentirse más seguros, adquiriendo productos en perfiles con mejor puntaje. Este sistema es uno de los más óptimos para aquellas empresas que están empezando a crecer, en vista que necesitan crear una relación óptima y confiable dentro de su plataforma.

3.1.6 Inversiones y Costos

Posteriormente, el paso a seguir para la generación del modelo de negocio, planteado por la guía metodológica, es la estimación de inversiones y costos. Estos se deben contrastar con las ganancias de la plataforma, para determinar si el modelo de negocio es viable o no. Cabe apuntar que, de los servicios, el único que generará ingresos será el marketplace, esto dado que, no se piensa incluir anuncios dentro de la plataforma, por lo que la Red Social tendrá un enfoque más directo con la comunicación entre los usuarios y no producirá ganancias.

Una inversión se describe como una actividad que consiste en dedicar recursos con el objetivo de obtener beneficios de cualquier tipo. *Francisco Lopez en su artículo Inversion (2018)*, menciona que, estos “recursos” se podrían identificar como cualquier factor indispensable para una persona ya sea tiempo, tierra, trabajo o capital [17].

Existen diferentes tipos de inversiones, que se realizan en momentos claves para obtener una ganancia óptima. Es el caso de las inversiones fijas, inversiones que corresponden a los bienes

intangibles de la empresa. Son adquiridas al inicio del proyecto una vez y presentan una vida útil mayor a un año.

También se encuentran las inversiones diferidas, conocidas como inversiones efectuadas en el periodo previo a la operación del proyecto. Están relacionadas a todas los bienes intangibles de la empresa, y su inversión es recuperada a largo plazo.

Por último, se menciona al capital de trabajo, conocido como el monto de recursos económicos necesarios para empezar la fase de producción de la empresa. Como por ejemplo, el salario de los empleados, el inventario, y deudas por cobrar [18].

3.1.7 Rentabilidad

Finalmente, para culminar el proceso de la creación del modelo de negocio, se deben contrastar las inversiones y las posibles ganancias de la empresa, esto con el fin de determinar si el modelo de negocio es rentable o no. La rentabilidad económica de una empresa puede definirse como la capacidad o aptitud de una empresa de generar excedentes a partir de un conjunto de inversiones y transacciones generadas (*Jesus Lizcano Alvarez, 2004*) [19]. En pocas palabras, cuando las ganancias generan dividendos en una empresa y esta se encuentra en un punto de equilibrio, se puede plantear que el proyecto es viable.

Sin embargo, el tiempo para que una startup tecnológica pueda ser viable tiende a ser volátil. Por lo que para determinar esta variable será necesario utilizar los datos encontrados en el estudio de mercado, para prever la cantidad de usuarios que interactúan en el sistema y predecir las posibles ventas que se concretarán. De esta forma en el análisis financiero se determinan las ganancias anuales, se comparan con los costos fijos, y así se evaluará en cuantos años el proyecto será viable.

3.2 Estructuración del proyecto

Una vez planteado el modelo de negocio y con la seguridad de que este sea rentable, se podrá iniciar la ejecución del proyecto. Para esto, según el centro de emprendimiento, se debe realizar inicialmente un estudio de mercado más a profundidad, en el cual se evaluarán elementos como, el análisis de la demanda y oferta, las vías de comercialización y el diseño de la marca. Posteriormente, se dará paso al análisis organizacional y legal de la empresa.

Después se ejecutará el desarrollo técnico del proyecto. Luego, en base a toda la información recopilada en el análisis de oferta y demanda, se realizan los estudios financieros necesarios para estimar si el proyecto es rentable o no. Y por último, se definirán los posibles impactos que generará la empresa a su entorno, a nivel financiero, ambiental y social.

3.2.1 Estudio de mercado

En este paso se recopila y analiza la información del mercado, con el fin de estimar óptimamente las estrategias comerciales [20]. Estos estudios suelen dividirse en estudio de necesidad o demanda y estudio de oferta. Es necesario realizar estos análisis para afianzar el éxito de los productos y servicios a desarrollar, y asegurarse que la empresa será viable.

3.2.1.1 Estudio de demanda

Enrique Nunes Jimenez, en su libro “*Guía para la Preparación de Proyectos de Servicios Públicos Municipales 1a Ed*” menciona en el capítulo 6, llamado “*Estudio de mercado*” que, el estudio de demanda, es aquel cuyo objetivo es determinar cuál es la demanda del servicio o producto considerado, que puede esperarse sea atendida por el proyecto al entrar en operación [21]. Este estudio puede enmarcarse en un análisis cuantitativo o cualitativo.

La investigación cuantitativa es aquella que se realiza para poder obtener datos, cifras, porcentajes o estadísticas de una situación en particular. Estos datos son, en la mayoría de los casos, fáciles de interpretar y analizar, puesto que brindan el respaldo necesario para llegar a la conclusión de la investigación. Por otra parte, se cuenta también con la investigación cualitativa, esta última hace uso de datos particulares para describir aspectos sin la necesidad de medirlos. Estos datos se caracterizan por ser opiniones, criterios y perspectivas que, a pesar de que aportan mucha información, son muy complejos de analizar y resumir en una sola respuesta.

Para el estudio de mercado se espera hacer uso de ambos estudios, el estudio cuantitativo se aplicará debido que se desea conocer las necesidades puntuales de los usuarios frente a las redes sociales y marketplaces actuales recopilando información sobre sus fortalezas y debilidades en el mercado. Por otro lado, el análisis cuantitativo ayudará a determinar la disposición de consumo de las personas del común.

3.2.1.2 Estudio de oferta

Este analiza los distintos competidores en el ámbito geográfico de interés, así como la cantidad de productos en el mercado. Esta exploración puede realizarse para el producto planteado o sobre el sector de mercado al que pertenece.

Para realizar este estudio, según el centro de emprendimiento, se debe realizar un análisis de competencia, comprendiendo los productos o servicios que estos ofrecen y encontrando semejanzas con el propio. Además, se requiere denotar las diferencias principales que tendrá el producto con la competencia, con el fin de entregar un producto superior y con diferenciación en el mercado.

3.2.1.3 Estudio de comercialización

El análisis de comercialización consiste en entender el proceso que debe ocurrir para entregar un producto al cliente. Es decir, deben instaurarse los procesos e instrumentos que posibiliten la interacción de los usuarios con el producto [22].

Para concretar la comercialización es necesario plantear aspectos como: Qué tipo de producto es, cuales son los canales de distribución, el tipo de cliente y la forma en que se publicitaria el mismo. Este estudio se debe llevar a cabo desde el estudio de la competencia, en pocas palabras, comprender cómo estos distribuyen y publicitan su producto. De esta manera, se plantean los canales de comercialización del marketplace.

3.2.1.4 Diseño de marca

Para finalizar con el estudio de mercado, es necesario plantear el diseño de marca. Para esto, el primer paso a realizar es una investigación interna de la compañía, con el fin de definir los principios de la empresa y visualizar la imagen que se quiere proyectar hacia los clientes. También, se debe realizar una revisión a todos los competidores más fuertes que se encuentren presentes en el mercado. En base a esa investigación, se explorará el desarrollo de marca e identidad corporativa que tendrá la empresa, con el fin de utilizarlos como base de inspiración, evitando el desarrollo de una imagen que refleje confusiones y que no aproveche un diseño único.

El primer punto clave para el diseño de marca, es la definición del nombre comercial, puesto que será el identificador de la empresa o de los productos que se ofrezcan. El nombre debe representar los valores de la empresa y generar una sinergia con los productos o servicios.

También, se debe desarrollar un logotipo que deba seguir ciertos aspectos para que sea atractivo en el mercado, de estos se destacan los siguientes:

- Máximo 3 colores.
- Tipografía sea fácil de leer.
- Representa los productos ofrecidos.
- Simple y fácil de entender.
- Puede ser visto en múltiples dispositivos.

Una vez sean conceptualizados los bocetos del logotipo, se comienza a darle color y brindarle vida a la marca de la empresa. Para esto, se escoge una paleta para cada propuesta y se juega con ella hasta llegar a los resultados que mejor se vean estéticamente, pero siempre cuidando los lineamientos de la esencia de la empresa.

Cada color tiene una función y es capaz de transmitir sensaciones o representar ideas, es por eso que la selección de estos es muy importante, ya que deben cumplir ciertos aspectos, como lo mencionan *Garber, Burke, & Jones (2000)* [23]:

- Identificar la categoría al que pertenece el producto.
- Contrastarse con respecto a sus competidores.
- Identificar a la marca.
- Entregar un significado.

Por último toda empresa debe contar con un slogan acorde a la filosofía de la misma. Este debe contar con ciertas características, haciéndolo memorable para quien lo lea. El slogan tiene que ser corto y contener un verbo o acción. Además, es necesario que perdure en el tiempo, es decir, que este quede en la memoria de la audiencia. También, debe generar una incitación, provocación o invitación, ya que busca tocar la fibra sensible del consumidor. Por

último y como aspecto más importante, está en la obligación de representar la propuesta de valor de la empresa.

Una vez encontrados todos los elementos profundos en el estudio de mercado, se indagan los aspectos organizacionales de la empresa. En este caso se decide plantear este aspecto y la temática legal, antes de la ejecución técnica del proyecto, debido a que estos elementos serán necesarios para seguir los lineamientos de la compañía en el desarrollo del prototipo, para posteriormente en base al producto construido y diseñado realizar un análisis financiero.

3.2.2 Análisis organizacional

Teniendo en cuenta las definiciones del modelo de negocio, la propuesta de valor y el estudio de mercado, el siguiente análisis a realizar, teniendo en cuenta los lineamientos del centro de emprendimiento, es el estudio de los elementos organizacionales que conformarán la estructura de la empresa. Estos se encargan de ordenar y representar al conjunto de normas y elementos de la compañía, y contienen lo siguiente:

- Misión.
- Visión.
- Organigrama.
- Manual de funciones.
- Requisitos de contratación.
- Objetivos estratégicos.
- Valores corporativos.

3.2.2.1 Misión

El primer componente organizacional que se define es la misión, esta representa el motivo o la razón por la cual se crea una empresa. Es el objetivo principal que pretende alcanzar para satisfacer las necesidades de los clientes, del personal de la empresa, de la competencia y otro público. En concreto, determinar la misión orienta y ayuda al personal y a los socios estratégicos a conocer sobre la naturaleza de la organización y el rumbo que piensa tomar.

En la misión se define la posición de la organización en su campo de acción y las ventajas que tendrá frente a la competencia, incluyendo las estrategias que se van a implementar y la filosofía que se va a seguir.

La misión puede ser generada a partir del análisis mental realizado por el propietario, o por un proceso de trabajo gerencial en equipo. Esta contiene normalmente tres elementos esenciales [24]:

- El propósito de la organización (para qué existe y cuáles son sus metas).
- Los valores Morales y Normas de Conducta de la organización (en qué cree la organización, cuáles son sus valores morales y cómo son las políticas y prácticas de conducta).
- Hacia dónde va la organización (metas a cumplir, estrategias y tácticas a utilizar para alcanzar las metas).

3.2.2.2 Visión

La visión es otro elemento organizacional importante para el desarrollo de la startup, esta representa una situación imaginaria y desafiante en el tiempo, capaz de satisfacer el sentido de existencia de la organización a mediano o largo plazo, para atender con eficiencia las cambiantes necesidades y gustos del público destinatario.

La visión invita a la transformación de la empresa, exponiendo la necesidad de un cambio positivo y progresivo. Se conceptualiza de manera estática en el tiempo, pero su alcance tiene una proyección dinámica.

Esta formulación no es necesariamente un proceso de opinión, pero se puede identificar un proceso grupal en el desarrollo de esta. Es establecida por el gerente o grupo responsable de administrar la empresa. Por lo general, la visión va dirigida hacia una mejor condición que la que presenta actualmente en la empresa.

En base a esto, se construye una idea relativamente estable de los objetivos que la organización quiere alcanzar, en tanto las condiciones y estrategias de la empresa puedan

modificarse con el tiempo, pero a diferencia de la misión, la visión permanece inalterable en su esencia [25].

3.2.2.3 Organigrama

Continuando con los componentes organizacionales para la generación de la startup, se menciona al organigrama. Es representada como una guía que esboza mediante una gráfica, la estructura interna y formal de una entidad administrativa de manera objetiva o global (**Figura 1**). Esto es importante, puesto que ayuda a dimensionar mejor la organización y contribuye a monitorear el crecimiento estructural de la misma.

Con el fin de realizar un organigrama limpio y ordenado, se debe conocer realmente la estructura de la empresa y todas sus funciones internas. A continuación, se mostrarán todos los pasos que se deben tener en cuenta para realizar de manera correcta un organigrama [26]:

- 1. Clasificar trabajadores:** Es decir, identificar las obligaciones de todo el personal en sus respectivos sectores laborales. También, se deben reconocer a los líderes encargados de cada área de trabajo y las personas responsables de cumplir con las tareas establecidas por los líderes.
- 2. Delimitar departamentos:** Se deben catalogar los departamentos de trabajo que conforman la empresa. Con el fin de establecer planes y metas acordes a los objetivos de negocio, permitiendo segmentar a los trabajadores dependiendo de sus competencias y habilidades.
- 3. Establecer tareas y responsabilidades a cada departamento:** Esto refiere a definir las tareas y responsabilidades de cada grupo de trabajadores. Esto ayudará a aclarar las funciones de cada empleado y evitará el duplicado de tareas.
- 4. Determinar el manejo de equipos:** Se deben precisar la cantidad de personas que están bajo el cargo del líder de cada división. Esto es esencial para los líderes, dado que facilita la asignación de tareas y mejora la logística laboral.



Fig 1. Ejemplo de un organigrama.

3.2.2.4 Manual de funciones

Asimismo, un componente importante a tener en cuenta es el manual de funciones. El cual se representa como un documento que ayuda a orientar a los miembros de cada organización en los procesos administrativos y operativos. También, respalda la evaluación de los procesos al interior de la corporación por parte de los organismos de control.

Es importante que todas las organizaciones tengan documentado el trabajo que realizan cada unos de los funcionarios, con el fin de llevar un seguimiento de las actividades que se llevan a cabo. Por otro lado, las personas encargadas de la elaboración de este manual, asumen el reto de transmitir a todos los miembros de la empresa, de manera clara y concisa, la forma en que deben desarrollar su trabajo y la importancia de realizar estos procesos de forma correcta.

Existen diversas maneras para realizar el manual de funciones. A continuación, se planteará un método ideal para empresas con enfoque de startup:

En primer lugar, se requiere identificar todos los roles que hay dentro de la empresa. Luego, se deben establecer las áreas de trabajo al cual pertenecen cada uno de estos roles. Después, se realizará una pequeña descripción del cargo, esto es importante para poder identificar las tareas y responsabilidades de cada miembro de la empresa. Posteriormente, se plantean los requisitos mínimos necesarios para que una persona pueda ocupar ese cargo, en esta parte se encontrarán los niveles educativos, experiencia de trabajo y habilidades que la persona debe

tener para desempeñar un buen trabajo. Por último, se encuentran las funciones y procedimientos del rol, en otras palabras, son todas aquellas tareas y responsabilidades que va a desarrollar la persona en el cargo [27].

1. IDENTIFICACIÓN DEL PUESTO		
Denominación del Puesto	Categoría Laboral	
DIRECTOR TÉCNICO	DIRECTOR TÉCNICO	
Departamento	Nº. Titulares	Jornada laboral
Técnico	1	Mañana y tarde
2. FINALIDAD		
<ul style="list-style-type: none"> • La captación de clientes y elaboración de presupuestos. • Ajustar el desarrollo de las obras a los presupuestos consignados. • La dirección del Departamento de Producción, así como la gestión de los recursos humanos y materiales aplicados en los proyectos de trabajo, asignando el personal a los distintos trabajos. 		
3. ÁREAS DE EFICACIA / RESULTADOS		
<ul style="list-style-type: none"> • Clientes captados. • Presupuestos ajustados a un margen comercial eficaz. • Gestión de recursos humanos y materiales de su departamento. • Ajuste de resultados al presupuesto establecido en cada proyecto de trabajo. • Control de presupuestos. 		

Fig 2. Ejemplo de un manual de funciones

3.2.2.5 Requisitos de contratación

Para que una persona pueda pertenecer a una corporación, debe pasar por un proceso de contratación realizado por la empresa. Este proceso se encarga de medir a los oferentes sus capacidades jurídicas, financieras, organizacionales y su experiencia en un cargo en específico. Para que este proceso se pueda llevar a cabo de manera óptima, debe cumplir con unos requisitos mínimos.

El propósito de los requisitos de contratación es establecer una condición mínima aceptable para los proponentes, de manera que, la entidad sólo se encargue de evaluar las ofertas que cumplan las condiciones del proceso de contratación. Cabe resaltar que, los requisitos siempre se refieren a las condiciones de un oferente y no de la oferta [28].

Las exigencias de contratación mínimas para continuar con el proceso son las siguientes:

- **Hoja de vida:** Es el currículum presentado por el aspirante, refleja la experiencia que la persona ha obtenido desde su primer trabajo hasta el último. Las empresas deben

revisar de manera minuciosa si la hoja de vida de las personas cumplen o no, con la experiencia solicitada para el cargo.

- **Entrevista:** Es una reunión de dos o más participantes, en la cual se evalúa a la persona solicitante del puesto, con el fin de saber si es apto para ser contratado o no. Este requisito es importante, puesto que al ser uno de los medidores de selección, ayuda a identificar a las personas que cumplen correctamente con el perfil solicitado.
- **Prueba técnica/teórica:** Son test realizados con el fin de observar los métodos que usan los aspirantes para solucionar un problema en específico o el conocimiento que tienen de la empresa, o sobre el cargo al cual aspiran. Dependiendo de los resultados obtenidos, se realizará un listado de las personas ideales para el cargo aplicado.
- **Entrevista final:** En esta reunión, los entrevistadores no realizarán pruebas a los aspirantes, puesto que solamente se hablará de los términos laborales y del contrato, como lo es negociar el salario y mencionar los horarios establecidos.

CONTRATO INDIVIDUAL DE TRABAJO

_____, de _____ años de edad, casado, GUATEMALTECO, vecino del municipio de _____, departamento de Guatemala, quien se identifica con el documento personal de identificación con código único de identificación número _____ (_____), extendido en el Registro Nacional de Personas de la República de Guatemala, actúa en representación de la entidad _____, con sede social ubicada en _____ de la ciudad de Guatemala, calidad que acredita con el acta notarial de su nombramiento autorizada en esta ciudad el ____ por el Notario _____, el cual se encuentra debidamente inscrito en el Registro Mercantil General de la República bajo el número _____, folio ____ del libro _____, a quien en el presente instrumento se le denominara como "El Patrono"; y _____, de _____ años de edad, _____ (estado civil), guatemalteco, ejecutivo, de este domicilio y vecindad, se identifica con el documento personal de identificación con código único de identificación número _____ (_____), extendido por el Registro Nacional de las personas de la Republica de Guatemala, a quien en el presente instrumento se le denominara como "El Trabajador". **CELEBRAN** por este acto **CONTRATO INDIVIDUAL DE TRABAJO** contenido en las siguientes cláusulas:

PRIMERA. Antecedentes. La relación de trabajo se inició con la entidad _____ y por sustitución patronal efectuada el _____ debidamente autorizada ante el Ministerio de Trabajo y Previsión Social, ocupa el puesto de _____.

SEGUNDA. Inicio de la relación de trabajo. La relación de inicio el _____, de conformidad con lo expuesto en la cláusula primera.

TERCERA. De los servicios a prestar. El trabajador se obliga a desempeñar el puesto de _____, Específicamente en las actividades siguientes: _____.

Además, el trabajador se obliga a prestar los servicios que le sean requeridos y que directa o indirectamente se relacionen con la naturaleza, características y condiciones de su puesto y a desempeñar las labores de cualquier otra naturaleza que sean del mismo género de los que formen el objeto del negocio, actividad o industria del patrono, giro o actividad que manifiesta el trabajador que es de su conocimiento, de conformidad con el Reglamento Interior de Trabajo y/o Descripciones de Puesto, los cuales incluyen pero no limitan las actividades asignadas al puesto de trabajo y son de su conocimiento.

CUARTA. Lugar de prestación. Los servicios serán prestados por el trabajador en Los servicios serán prestado en la sede de la entidad _____, o en cualquier otra dependencia de la Empresa o en cualquier otro lugar o centro de trabajo dentro de la República de Guatemala, o bien en el territorio que indique Patrono proporcionando para el efecto los gastos respectivos al trabajador.

QUINTA. Duración del contrato. La duración del presente contrato es indefinido.

SEXTA. Jornada de trabajo.

La jornada de trabajo será de lunes a viernes, y de conformidad con el artículo 124 del Código de Trabajo, no se encuentra sujeta a los límites de jornada por ser un puesto de _____.

1

Fig 3. Ejemplo de un contrato laboral.

3.2.2.6 Objetivos estratégicos

Todas las empresas proyectan sus logros sobre un periodo de tiempo determinado, reflejando lo que es realmente importante dentro de la corporación. A estos se les denomina objetivos estratégicos, que a pesar de estar basados en la visión, misión y los valores de la organización, son ellos mismos los que se encargan de determinar cuáles serán los procesos y métodos para poder cumplirlos.

Por otro lado, es importante tener claridad sobre la naturaleza de la empresa, no es necesario tener los mismos objetivos que otras organizaciones, incluso si ellas se encuentran dentro del mismo sector económico. Estos objetivos permiten al equipo idear una solución óptima que encaje con el perfil de la empresa, solución que deberá ser proyectada en un tiempo determinado por el equipo de trabajo.



Fig 4. Ejemplo de objetivos estratégicos.

En la **figura 4**, se puede observar un claro ejemplo de un objetivo estratégico por parte de la empresa Apple. Esta se enfoca en proveer soluciones para el sector tecnológico. Sin embargo, ha sido el foco de muchas críticas por utilizar materiales no reciclables en sus productos. Para eliminar esta mala imagen, una de las soluciones planteadas por la empresa fue, **“crear campañas de concientización y diseñar productos con materiales reciclados”**, enfocados en un sólo objetivo puntual, ayudar al medio ambiente. En este ejemplo, la naturaleza de la empresa no se ve afectada por la solución propuesta, puesto que no se fija un tiempo estimado de implementación del objetivo. Esto es importante, porque da garantías a la empresa de poder adaptar medidas no inmediatas para alcanzar ese objetivo.

3.2.2.7 Valores corporativos

Para finalizar con el análisis organizacional para la creación de la startup, se suman los valores corporativos. Según Julia Martins, son definidos como los principios fundamentales con los que la empresa precisa sus criterios frente a la forma de trabajo, la contribución interpersonal y el bienestar de los empleados. Estos valores marcan los procesos a realizar y las responsabilidades por cada labor dentro de la organización .

Los valores corporativos son importantes porque generan una cultura corporativa saludable. También, establecen los estándares para la colaboración entre los diferentes departamentos de la empresa. Además, aumenta la participación y la retención de los empleados, aumentando la satisfacción dentro del trabajo [29].

3.2.3 Análisis Legal

Una vez planteada la organización de la empresa y sus valores. Un componente importante a tener en cuenta, es la formalización de la compañía y sus aspectos legales, tanto en términos de tributación y, las leyes que rigen y amparan a la plataforma ya sea en sus elementos de marketplace o de red social. En primera instancia la empresa para ser definida como una persona jurídica, debe ser inscrita en la Cámara de Comercio de Colombia, bajo una figura predeterminada y con una razón social (nombre).

3.2.3.1 Camara de Comercio de Colombia

Las Cámaras de Comercio en Colombia son organizaciones municipales privadas de carácter gremial y sin ánimo de lucro presentes en 57 zonas del país. Encargadas de funciones públicas como lo es la administración de registros públicos empresariales [30]. Las funciones de las cámaras de comercio según el **artículo 86** del Código de Comercio son [31]:

- Servir de promotor de los intereses generales del comercio ante el gobierno representando a los comerciantes.
- Realizar estudios económicos sobre aspectos del comercio interior y exterior, para formular recomendaciones a los organismos encargados de la ejecución de los planes en la materia.
- Llevar el registro mercantil y certificar sobre los actos, así como los documentos inscritos en él.
- Dar noticia en sus boletines de las inscripciones hechas en el registro mercantil y de toda modificación, cancelación o alteración de dichas inscripciones.
- Certificar las costumbres mercantiles de los lugares correspondientes a su jurisdicción.
- En caso de ser solicitados, designar un árbitro o árbitros o los amigables compondores.

- Servir de tribunales para resolver las diferencias presentes entre los contratantes, en el cual el tribunal se conformará por todos los miembros de la junta.
- Prestar sus buenos oficios para hacer arreglos entre acreedores y deudores.
- Ofrecer exposiciones y conferencias, editar o imprimir estudios relacionados con sus objetivos.
- Definir su reglamento interno que deberá ser aprobado por el Superintendente de Industria y Comercio.
- En el mes de enero dar un informe al Superintendente de Industria y Comercio, acerca de las labores realizadas en el año anterior y las situaciones económicas de sus zonas, así como el detalle de ingresos y egresos.
- Se agrega como funciones las demás leyes que les atribuyen las leyes y el Gobierno Nacional.

En síntesis las cámaras cumplen por encargo del Gobierno Nacional, la administración y gestión del Registro Público Mercantil de las empresas y entidades sin ánimo de lucro. Ahora bien, para poder conformar una empresa en un inicio es necesario establecer qué figura jurídica tendrá la misma, así como cumplir una serie de requisitos y suministrar documentos para el registro en la Cámara de Comercio. Además, su razón social no podrá ser homónima a la de otra empresa.

3.2.3.2 Homonimia

Existe homonimia cuando ya se encuentra registrado un nombre igual al que se pretende registrar o modificar en la Cámara de Comercio dentro del registro mercantil, sin tener en cuenta el tipo de sociedad o entidad [32]. Por lo tanto, es importante que a la hora de establecer los aspectos organizacionales y de marca de la empresa se evite usar un nombre que no se pueda inscribir.

Para conocer si hay homonimia en el nombre ideado, basta con entrar a la plataforma de la Cámara de Comercio y realizar una búsqueda de homonimia en el registro mercantil (cuando se trate de empresas comerciales).

3.2.3.3 Figura Jurídica S.A.S

Para poder ejercer ventas de forma legítima, la startup y su plataforma debe estar respaldada por una figura jurídica que cumpla las obligaciones estipuladas en el código de comercio. Una persona jurídica, es una persona ficticia, que al igual que cualquier persona puede contraer obligaciones y tiene derechos civiles, la cual puede ser representada judicial y extrajudicialmente. Estas pueden figurar como corporaciones y fundaciones de beneficencia pública. En Colombia, existen diferentes tipos de sociedades representativas como lo son la Sociedad Limitada (LTDA), Empresa Unipersonal, Sociedad Colectiva, Sociedad por Acciones Simplificada (S.A.S) entre otras.

En concreto para la constitución de la startup se puede optar por la forma más fácil de constitución y formalización, mediante la Sociedad por Acciones Simplificada S.A.S. Esta figura jurídica tiene responsabilidad limitada, y su naturaleza siempre será comercial, independientemente de las actividades prescritas en su objeto social. Caracterizada por tener una estructura que garantiza la libertad de autonomía contractual en el diseño del contrato social, sino también la posibilidad de que los asociados decidan las pautas sobre las relaciones jurídicas. Para su conformación según la **ley 1258 de 2008** expedida por el congreso de la república y el **código de comercio** en su **artículo 110**, se debe suscribir los siguientes documentos [33]:

- Documento autenticado, de carácter privado, reconocido o con presentación personal por sus signatarios.
- Acta de constitución pública, cuando hay aporte de inmuebles, o en caso de que los accionistas así lo consideren.
- Cualquiera que sea el documento debe contener la siguiente información:
 - Nombre, documento de identidad y domicilio de los accionistas.

- Razón social, seguido de las palabras Sociedad por Acciones Simplificada o SAS.
 - Domicilio principal de la empresa, en el cual se desarrollará la actividad de la sociedad. En caso de establecerse sucursales se debe indicar el municipio donde están ubicadas.
- Término de duración: Podrá tener duración de término indefinido.
 - Definición de las actividades principales.
 - Capital autorizado, suscrito y pagado.
 - El capital, deberá ser pagado en los plazos establecidos en los estatutos. No podrá exceder dos años.
 - Definir de forma clara y precisa la forma de administración de los negocios sociales. Con indicación de las atribuciones y facultades de los representantes legales y administradores.
 - Nombre, identificación y facultades de los administradores.
 - También deben definirse los nombres, apellidos e identificación de los representantes legales, miembros de la junta directiva, revisores fiscales, según el caso.

El registro en la cámara de comercio puede realizarse vía internet o solicitar cita para el trámite presencial, ya sea en la página de la cámara del municipio o departamento según la dirección de domicilio de la empresa. Otro aspecto importante son los beneficios que trae tener esta figura jurídica. Según lo resume el Consultorio Contable y Financiero de la Universidad ICESI algunas de las ventajas de una S.A.S son:

- **Simplificación de los trámites:** Como se mencionó anteriormente, la inscripción puede realizarse por un documento privado. Sin embargo, para el caso de la Sociedad por acciones Simplificada, no es necesariamente estricto dar fe pública mediante el

documento, a menos de que se posea bienes inmuebles. Además, puede conformarse con una sola persona en la sociedad.

- **Limitación de responsabilidad:** En la SAS se debe delimitar la responsabilidad de los accionistas acorde al monto del capital aportado. Esto en casos de fraude, deudas, obligaciones laborales o tributarias que reciba la sociedad.
- **Autonomía para estipular las normas del negocio.**
- **Estructura flexible de capital:** Se puede constituir así no se pague en el momento de crearla. Es decir, el capital de aporte puede ser 0.

3.2.3.3 Responsabilidades Tributarias de la SAS

Un punto clave a la hora de tener una empresa son sus responsabilidades. Para las SAS existen documentos que deben ser reportados periódicamente para llevar registro de las ganancias y responder a las obligaciones tributarias que se tiene [34]. Al igual que otras sociedades las S.A.S tienen pagos de impuestos y sus obligaciones son las siguientes:

- Son organismos de retención en la fuente, como el Iva, Ica y demás. En el caso del proyecto se aplicaría solo a compradores dentro de Colombia.
- Deben responder por el cobro del Iva en caso de realizar una actividad que grabe ese impuesto.
- Son contribuyentes del impuesto de renta.
- Se está obligado a expedir facturas para llevar registro de las ventas y ganancias.
- Son requeridos de aportar la información exógena.
- Deben pagar los impuestos de industria y comercio.

- Están obligadas a pagar impuesto predial (en caso de tener la dirección física a nombre de la empresa).
- Están obligadas a llevar contabilidad.

3.2.3.4 Leyes de la plataforma

En primera medida se estipula los mensajes de datos como legítimos. Siendo estos definidos como una información generada, para ser enviada por el emisor y recibida por un receptor. Esta puede ser almacenada vía electrónica y debe cumplir los requisitos legales. En pocas palabras se plantea que toda la información generada dentro de una página web representa a los mensajes. Por otro lado, establece que el uso de mensajes de datos es válido en los procesos de transportes de mercancía. Implementando métodos para garantizar su veracidad. Además, las firmas digitales se toman como válidas, estas deben pertenecer a una persona y son representadas por aspectos numéricos. Por último habla de los encargados de la creación de firmas digitales, estableciendo los deberes sobre aquellos relacionados a estas actividades.

Otro aspecto importante de las leyes que protegen a los usuarios y rigen a las plataformas web en Colombia es la ***Ley de Protección de Datos Personales o Ley 1581 de 2012***. La norma reconoce y protege el derecho a conocer, actualizar y rectificar la información que esté sobre ellas en bases de datos o archivos que puedan tratar entidades de naturaleza pública o privada [36]. Estos datos pueden ser personales de cualquier tipo, ya sea de identificación, nombres, direcciones, datos bancarios, etc. Además, estos datos deben ser tratados como el mayor bien de una empresa, puesto que no pueden ser repartidos a entidades externas.

Por último tenemos el beneficio de las empresas comerciales para ejercer leyes y normas dentro de sus productos. A esta herramienta se le conoce como términos y condiciones, comúnmente aplicada en las redes sociales. Estas dictan las cláusulas legales que definen la forma en la que se puede suministrar, usar y acceder a la información de una aplicación o página web [37]. Además, establecen lo que el dueño de la página hará con los datos personales y los que se generen dentro de la misma. Así mismo, al dar consentimiento a estos términos, se da permiso a almacenar, analizar y utilizar la información personal, datos bancarios, imágenes, videos, las interacciones, etc. Es por eso que será de utilidad para dar

conocimiento del uso que se le hará a los datos a las personas, a la hora del registro en la plataforma y además definir las limitaciones que abran a la hora de vender el contenido, tanto en términos de derechos de autor, como en el contenido que ofrecen.

3.2.4 Módulo de desarrollo

Una vez planteados los aspectos del estudio de mercado, el modelo de negocio, los componentes organizacionales y legales. Por último, es necesario conocer los conceptos técnicos del proyecto.

Dado que los entregables a presentar son el prototipo de red social y diseños del Marketplace de Contenido Multimedia que unificarán ambos conceptos. Se deben conocer primero las teorías que engloban esta construcción, como el Internet, el desarrollo y las aplicaciones web. Así como las características técnicas que se requieren para crear el prototipo y los diseños posteriores, esto con el fin de comprender e identificar las posibles soluciones que se aplicaran para la construcción.

Internet es una red global de computadoras que provee acceso a variedad de información y, garantiza entornos y funciones de comunicación, a partir de protocolos de comunicación estandarizados como lo es el TCP/IP [38]. Estos proveen de servicios de comunicación básica que transmiten paquetes, los cuales representan unidades de información.

Cuando se realiza una búsqueda a través de un navegador (aplicación para navegar a través de internet), por lo general se puede encontrar páginas y aplicaciones web. Una página web es un archivo de hipertexto que forma parte de la WWW (World Wide Web) que puede contener imágenes, videos o texto. Por lo general, es un archivo estático, el cual se visualiza al hacer una petición. Por otro lado, una aplicación web es un conjunto de páginas web dentro de una misma dirección. Usualmente, son dinámicas, dado que usan una base de datos para administrar la información y contienen variedad de funciones. Las páginas y aplicaciones web, hacen uso del protocolo HTTP (Protocolo de Transferencia de Hipertexto) para enviar y recibir archivos o información.

3.2.4.1 Servicio de Red Social en Internet

Ahora bien, una vez comprendidos los aspectos básicos de cómo funciona una aplicación web. Es necesario entablar los elementos necesarios que deben estar presentes en la Red Social.

Según lo define *Claudia Patricia Flórez y Harold Mendez* en su trabajo de especialización llamado “*Estudio de Ingeniería Social en el Uso de Las Redes Sociales*”. Un servicio de red social en internet como por ejemplo “Facebook”, es un producto, que brinda herramientas como blogs, chats, canales de mensajería, grupos de discusión y marketplaces, para crear comunidades en torno a intereses, personas o actividades [39]. Estos elementos son fundamentales para el correcto funcionamiento e interacción entre los usuarios, además de otros aspectos básicos como lo es el registro de usuario, log-in, personalización del perfil entre otros.

El **registro de usuario** es el proceso en el cual una persona a través de un formulario en un servicio web, ingresa sus datos personales con el fin de ser guardados en la base de datos, para posteriormente ingresar en la plataforma. Por otro lado, la **autenticación de usuario**, es la forma en la que se provee de identidad al cliente dentro del servidor web, y con esto acceso a funciones dentro de la aplicación [40]. Para que un usuario sea autenticado, es necesario que ingrese sus datos de acceso a través de un formulario o botón de ingreso. En la siguiente figura se muestra un ejemplo del proceso de autenticación en un sistema.

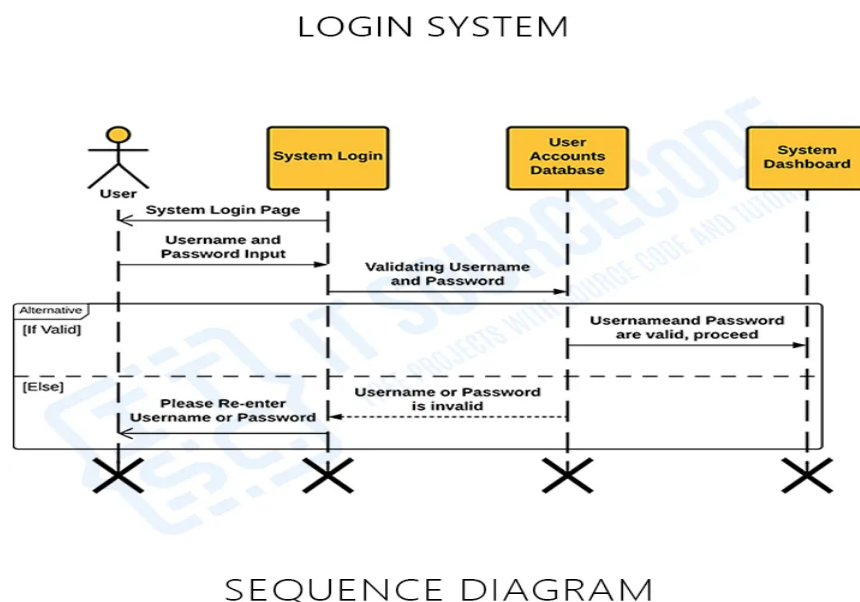


Fig 5. Proceso de autenticación en un sistema ficticio

Una vez autenticado en la plataforma, se podrá hacer uso de las funcionalidades básicas de la red social, teniendo como referencia la guía básica para construcción de redes sociales realizado por la compañía *MLSDev* [41] algunas de las herramientas que debe ofrecer una red social para brindar un ambiente de comunicación entre los usuarios son las siguientes:

- **Perfil de Usuario:** Según lo expuesto en el trabajo titulado como “*Social Network Sites: Definition, History, and Scholarship*”, las SNSs o mejor conocido como Social Network Sites (Sitios de Redes Sociales), tienen unas funcionalidades básicas para la identificación y expresión, como lo es el perfil del usuario. Este elemento se presenta como una vista o página en la cual, después de haberse registrado e ingresado los datos personales, el usuario puede incluir descriptores definiendo su edad, ubicación e intereses. Asimismo, se brinda la posibilidad de agregar foto de perfil, esta puede ser del propio usuario o también, de algún gusto o interés del mismo [42].
- **Feed:** El feed se presenta como la vista inicial de la red social, es decir la que todo usuario visualiza al iniciar sesión. A través de ella se pueden visualizar todas las publicaciones o interacciones realizadas, también se ofrece la posibilidad de encontrar personas y páginas con gustos similares a los designados por la persona.
- **Publicaciones:** Las publicaciones o “*Posts*”, son artículos publicados por los usuarios dentro de una página web, un blog, un foro o una red social. Las plataformas, por lo general, cuentan con herramientas que permiten a los usuarios poder personalizar estas publicaciones, ofreciendo a los usuarios la oportunidad de expresar sus ideas de manera libre. Cuenta con elementos que ayudan a la interpretación de la idea, en algunos casos se pueden observar títulos que permiten a los lectores poder interpretar dicha idea sin necesidad de hacer lectura de la publicación. También, se pueden visualizar textos, imágenes, vídeos o figuras que ayudan a la interpretación del mensaje. No es necesario que se cuente con muchos elementos para poder expresar una idea, puesto que la calidad de la publicación depende mucho del mensaje que se quiere entregar y del lugar en donde es publicado [43].
- **Amigos o Conexiones:** Es la función más importante de la red social, puesto que permite la interacción constante con personas en específico, dado que las

publicaciones realizadas por los miembros de lista de amigos/seguidores aparecerán siempre de primero en el feed. A través de esta herramienta los usuarios pueden escoger quienes seguir o con quienes interactuar a través de la lista mencionada, las solicitudes de amistad o seguimiento pueden almacenarse en ella y la persona puede decidir si aceptarla o rechazarla, o por el contrario podrían no almacenarse y aceptarse automáticamente. Estas funcionalidades garantizan la posibilidad de extraer funciones adicionales como la mencionada por *Alexander Richter* y *Michael Koch* en su trabajo “*Functions of Social Networking Services*”, dado que podría dar paso a la Administración de contactos (“Contact Management” en inglés), a través del cual el usuario puede restringir acceso a su perfil o solo permitirlo a personas dentro de su lista de contactos.

- **Comentarios:** Los comentarios son definidos como opiniones cortas y son realizadas usualmente dentro de una publicación por parte de los usuarios. Las plataformas diseñan una sección dentro de las publicaciones, permitiendo a los usuarios realizar dichos comentarios. Estos comentarios son personalizables y no necesariamente tienen que estar relacionados con el contenido de la publicación, tampoco deben de tener un fin en específico.
- **Likes:** Conocido también como “*me gusta*”, es una cualidad implementada en las redes sociales que permite a los usuarios la oportunidad de entregar un feedback positivo a cualquier tipo de contenido. En algunas plataformas, esta funcionalidad permite dar un feedback diferente al positivo, disponiendo de reacciones que encajan y representan las emociones del usuario, además son utilizados para dar credibilidad a los autores de las publicaciones, permitiendo compartir su contenido con más usuarios [45].
- **Servicio de Mensajería Instantánea:** Es un servicio que utiliza como herramienta un canal de comunicación por medio de una red (puede ser local o por internet), a través del cual dos o más personas pueden enviar y recibir mensajes de texto instantáneamente. En otras palabras, los mensajes son emitidos y recepcionados en tiempo real, por lo que ambos actores hacen el papel de emisor y receptor dentro del sistema. La arquitectura utilizada en estos canales suele ser Cliente/Servidor, donde los clientes envían el mensaje al servidor y estos lo remiten a un receptor [46]. Los

servicios de mensajería instantánea más conocidos actualmente son: Whatsapp, Telegram, Facebook Messenger y QQ Mobile. Este servicio es de suma importancia en la red social, por lo que garantiza una interacción tanto pública como privada de forma inmediata entre los usuarios. En el proyecto a realizar, se estima que los servicios de mensajería sean útiles para la comunicación entre usuarios y, entre usuario y moderador. Además, es importante que estos canales sean seguros para preservar la privacidad de los clientes.

3.2.4.2 Marketplace de Contenido Multimedia

Por otro lado, ya entablados los elementos esenciales de la red social, es menester conocer los requerimientos fundamentales dentro de un *marketplace de contenido multimedia*, con el fin de que estén presentes en el prototipo a realizar.

Para que un usuario pueda realizar compras a través de la plataforma, primero debe estar registrado y autenticado en esta. Por ese motivo, se define como requisito dentro del marketplace contar con un sistema de registro y autenticación. Esta funcionalidad será la misma funcionalidad implementada en la red social, puesto que el prototipo a realizar es una fusión de ambos sistemas. Además, se contarán con otros elementos a implementar dentro del sistema:

- **Carrito de Compras:** Es una característica que permite almacenar los productos que los usuarios desean comprar en una web. Permite al cliente conocer la totalidad de su compra, modificar los productos almacenados y poder realizar dicha compra.
- **Pasarela de Pagos:** Una pasarela de pagos, según lo planteado en el libro “*Pasarela de Pagos para la Seguridad de Transacciones Bancarias en Línea*” es “Una página web que representa un servicio intermediario entre una página de comercio electrónico y un banco cuando se ejecutan transacciones bancarias online. Se integran a la tienda virtual y almacenan información del banco que maneja las cuentas de compradores y vendedores. En el pago con tarjeta, la pasarela de pagos valida la veracidad de la tarjeta y organiza la transferencia del dinero de la cuenta del comprador a la cuenta del vendedor” [47].

- **Inventario en línea:** Un inventario representa los elementos que componen el patrimonio de una empresa o persona. En él se especifican de forma detallada las características de los elementos, se puede agrupar por orden en específico y expresa el valor de cada elemento en unidad monetaria [48]. Por lo tanto, un inventario en línea es un sistema alojado en la red en el cual una o más personas pueden administrar objetos tangibles e intangibles. Para el desarrollo del presente proyecto, esta funcionalidad se aplicará para el inventario de los creadores de contenido y los usuarios que hayan adquirido productos, a través del cual estos actores podrán visualizar el contenido multimedia propio y administrarlo a su antojo. Es importante mencionar que, todo el producto que se compre no debe ser reproducido, filtrado, ni revendido, por lo que será importante la labor de los moderadores en prevenir estas situaciones.

Definidos ya los aspectos técnicos básicos del proyecto, ahora se explicará el proceso que debe seguir un producto de software a la hora de ser creado, teniendo en cuenta que el prototipo a realizar es un software.

3.2.4.3 Ciclo de vida del software

En todo proceso de Ingeniería de software, este debe seguir ciertos pasos a la hora de ser creado. Antes de que sean esbozados se debe plantear que es la Ingeniería de Software. Primero, se entiende como software a los programas e información operativa utilizada por un computador. No obstante, software no solo son los programas, si no, todo lo asociado a la documentación y configuración de los datos para que los programas funcionen correctamente [49]. Por lo tanto, se podría definir la Ingeniería de Software como una rama de la ingeniería que se centra en la producción de software de calidad [50].

Siguiendo con lo anterior, el ciclo de vida del software es el proceso que debe seguir este para ser producido [51]. Los pasos presentes en el ciclo tradicional son:

- Análisis
- Diseño
- Desarrollo
- Testing

- Despliegue
- Mantenimiento

Según lo describe el Dr. Hemant Verma en su documento “*Software Development Life Cycle: Factors affecting Software Quality*”, el primer paso, también llamado ingeniería de requerimientos, consiste en capturar los detalles de la necesidad de los usuarios [52]. Es importante que el administrador del proyecto entienda los objetivos principales y la necesidad puntual de los clientes. En resumen, esta fase establece que es lo que el sistema debe hacer y tener.

Seguido a este, se presenta la etapa de diseño. Es un proceso en el cual el equipo revisa los requerimientos generados en la etapa anterior y plantea posibles soluciones. El diseño debe ser lo más explícito posible para que el desarrollo sea más efectivo[53]. En este proceso surgen preguntas tanto del software, como el hardware en el cual estará predispuesto el programa. Algunas de las posibles incógnitas que se deben plantear son:

- ¿Qué arquitectura tendrá el sistema?
- ¿Con qué tecnologías será desarrollada?
- ¿En qué servicios será desplegada?
- ¿Cómo se verá el sistema (En caso de tener apartado gráfico)?

Además, los documentos que soportan esta etapa pueden ser, documentación Técnica donde se explique el propósito y descripción de cada método/función, Modelo de datos en el cual se distinguen cuáles son las entidades involucradas en el sistema, así como diseños de interfaces y modelado de las mismas.

El tercer paso a seguir es la codificación o desarrollo del proyecto, en él se deben seguir las tareas definidas en el diseño y utilizar la tecnología planteada para realizar este proceso. El equipo que ejecute estas labores, debe ser experto en programación con el fin de realizar culminar la etapa satisfactoriamente[54].

El siguiente proceso es el testing o pruebas, consiste en revisar todas las funcionalidades del programa con el fin de encontrar errores y asegurarse que funcione acorde a lo planteado. En

caso de encontrar errores es necesario volver a realizar todos los pasos anteriores, hasta que en el sistema no se encuentre ninguno.

Una vez garantizada la fiabilidad del producto se puede seguir a la fase de despliegue. En esta se hacen los procesos para que el usuario pueda utilizar el sistema. Estos procesos suelen consistir en: Subir la aplicación a un servidor web , distribuir el sistema compilado y realizar y entregar los manuales de uso.

Finalmente ya con el sistema en funcionamiento, los usuarios empezaran a entregar retroalimentación del sistema, en este paso se requerirá realizar mantenimientos o actualizaciones según las necesidades nuevas que tengan los usuarios. Además, según surjan nuevos requerimientos pueden realizarse mejoras en el sistema, ya sea de infraestructura, optimización de código o nuevas funcionalidades.

Estos procesos pueden realizarse en diferentes momentos según la metodología de desarrollo que se escoja.

3.2.4.4 Agile

La metodología ágil de desarrollo es un marco que permite crear productos en un corto periodo de tiempo. Este sigue los procesos del ciclo de vida del software pero con diferentes prácticas, dado que la estructura se presenta cíclica para entregar mejoras continuas al sistema como se observa en la siguiente figura.

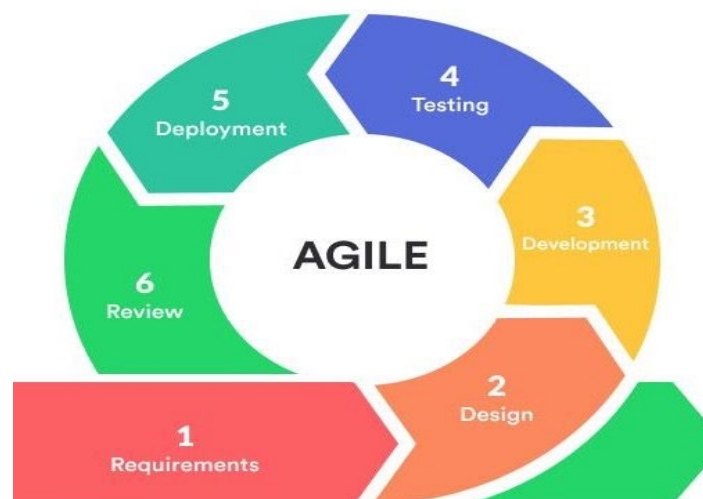


Fig 6. Ciclo de vida del software desde la metodología ágil de desarrollo

El modelo se basa en la disciplina y el trabajo en equipo, puesto que cada persona debe autoasignarse tareas [55]. Hay ciertos principios en los que se fundamenta esta metodología, los más importantes a tener en cuenta son:

- **El valor de la interacción y los individuos por encima de los procesos y las herramientas:** Se valora el recurso humano como principal factor para el éxito. Es importante contar con personas calificadas, que sepan trabajar en equipo y puedan interactuar con el usuario, ya que brinda mayor garantía de éxito.
- **Funcionalidad por encima de la documentación:** El manifiesto ágil respeta la producción de documentación como parte del diseño del sistema. Sin embargo, hace énfasis en que solo se deben producir los documentos necesarios y estos deben ser cortos y limitarse a lo fundamental.
- **Colaboración del cliente por encima de la negociación del contrato:** El desarrollo ágil incluye directamente al cliente o contractor al equipo de trabajo, con el fin de que se conozca todos los requerimientos del proyecto de primera mano y tener mejor relación con el cliente. En este caso, se puede relacionar este proceso con las entrevistas a realizar.
- **Adaptabilidad de los requerimientos al cambio:** A la hora de crear software pueden surgir cambios inesperados ya sea por parte del cliente o por restricciones funcionales. Por lo anterior, en la metodología ágil la planeación no es estricta, de esta manera, los cambios que puedan surgir a lo largo del proyecto pueden ser adaptables a la construcción del mismo.

El aplicar esta metodología tiene ventajas a la hora de realizar proyectos de desarrollo de software como por ejemplo [56]:

- Mantiene motivado al equipo de desarrollo.
- Mejora la satisfacción del cliente.
- Ahorro de tiempos y costes.

- Mejor velocidad y eficiencia.
- Mejora la calidad del producto.
- Facilita la gestión del riesgo
- Optimiza los resultados

3.2.4.5 SCRUM

Scrum es un framework para el desarrollo de proyectos basado en las metodologías ágiles, que ayuda a personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos [57]. Este será implementado para el desarrollo del prototipo inicial junto a Kanban, dado que es útil para la realización de proyectos grandes que pueden dividirse en etapas y requieren incrementos en poco tiempo, aplicando los principios de las metodologías ágiles. Este tiene como base 3 principios que permiten el control empírico de los procesos [58].

- **Transparencia:** La información que se genere en el proyecto debe ser conocida por todos los involucrados.
- **Inspección:** Se debe realizar inspección continua de los requerimientos y el progreso de cada Sprint, con el fin de encontrar posibles fallas en el camino.
- **Adaptabilidad:** Este principio permite realizar ajustes a lo largo del proyecto, por lo que todo el equipo debe acoplarse a estos en caso de haberlos.

Equipo en Scrum

El grupo de trabajo en Scrum, se encuentra confirmado por: Scrum Master, Product Owner y el equipo de desarrollo o Team.

El primer miembro mencionado, actúa como intermediario entre el Product Owner y el equipo de trabajo, filtrando las actividades para el alcance de cada entregable y reorganizando los requerimientos nuevos que surjan. Además, el **Scrum Master** es responsable de velar porque se cumplan los principios establecidos de Scrum, transmitiendo la teoría y las prácticas de la metodología [59].

El **Product Owner** es el encargado de controlar los procesos para la creación del producto, este representa al equipo de trabajo ante los interesados o Stakeholders. Además, debe mantener el flujo de trabajo y priorizar las actividades de mayor necesidad [60].

Por último, el **Team o equipo de trabajo** está compuesto por desarrolladores, diseñadores y demás involucrados en la construcción del Software [61]. Este equipo es autoorganizado y multidisciplinario, preocupado por la productividad y flexibilidad. Ellos trabajan por iteraciones o Sprints, generando incrementos al entregable por cada ciclo que se cumpla.

Como adicional se hace mención a los **Stakeholders**, estos no hacen parte del grupo Scrum en sí, dado que son los interesados del proyecto. Algunos ejemplos de personas que pueden ser Stakeholders serán los usuarios finales, gerentes patrocinadores y cualquier persona interesada en el proyecto.

Artefactos

Los artefactos en Scrum son los elementos utilizados por el equipo, los cuales garantizan transparencia del estado del Sprint, dado que todos los miembros pueden revisar y adaptarse en base a la misma información [62]. Los artefactos principales mencionados en el libro “The Scrum Guide” son [63]:

- **Product Backlog:** Conocida también como cartera del producto, es la lista de lo que necesita el producto y sus requerimientos. El Product Owner es el encargado de su priorización e inclusión de contenido.
- **Sprint Backlog:** Son todos los elementos seleccionados del Product Backlog para ser realizados en el Sprint, todos los requerimientos deben contar con un plan para ser cumplidos. Este artefacto hace posible visualizar todo el trabajo necesario para alcanzar la meta dentro del Sprint.
- **Planning Poker:** Es la herramienta implementada para calificar las historias de usuario. A cada una se le dará un punto de historia el cual simboliza la dificultad o tiempo que requiere la solución de cada requerimiento.

Sprint

El Sprint es el contenedor de todos los eventos de Scrum. Básicamente, todos los pasos de ingeniería a realizar se integran dentro del Sprint. Su estructura es cíclica, por lo tanto, cada vez que se realiza un Sprint, se vuelven a desarrollar todos los procesos que lo conforman. Es por eso que, se piensa implementar este componente, con la finalidad de dividir las etapas del proyecto en diferentes sprints, contando cada uno con una duración máxima de 4 semanas, donde al final de cada uno, se presentan mejoras constantes de la etapa en curso. A continuación se podrán observar los procesos que conforman un Sprint:

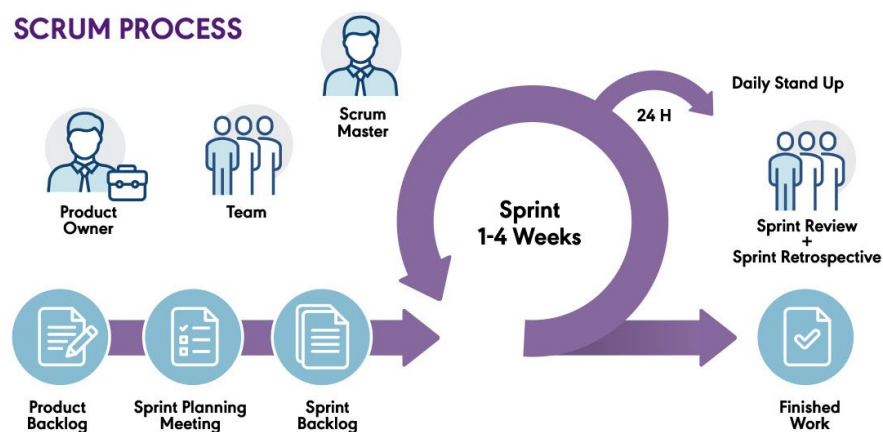


Fig 7. Ciclo del Sprint en la metodología Scrum

Los eventos a seguir en el Sprint son los siguientes [64]:

1. **Sprint Planning:** En esta etapa inicial se establece que trabajo se realizará durante el Sprint. Es decir, se definen las historias de usuario a realizar y cual es el alcance del mismo. Es común que antes de la realización de un proyecto se definan todos los Sprints y sus historias de usuario, para agilizar el proceso de análisis de los requerimientos.
2. **Daily Scrum:** El fin de la reunión diaria es conocer el progreso que hay para cumplir el Objetivo del Sprint y adaptar las historias de usuario en caso de ser necesario. Este evento tiene una duración de 15 minutos en la cual el equipo de desarrollo intercambia información entre ellos.

3. **Sprint Review:** Este proceso se basa en inspeccionar el resultado de la totalidad del Sprint y determinar futuros cambios. Aquí todo el Scrum Team presenta los resultados del trabajo a los Stakeholders y se ve el progreso que ha tenido con respecto a los objetivos.
4. **Sprint Retrospective:** El propósito del Sprint Retrospective es planificar formas para aumentar la calidad y efectividad del siguiente Sprint. En este el equipo analiza cómo funcionó el incremento realizado y las interacciones que se dieron entre las personas y los procesos. El Scrum Team puede sugerir cambios útiles para mejorar la efectividad, e incluso pueden agregarse al Product Backlog para que esté presente siguiente Sprint.

3.2.4.6 Analisis de las Historias de Usuario

Antes de definir los Sprint y cargar las historias de usuario al Product Backlog, primero se deben definir las épicas, de las que surgirán las historias de usuario:

Épicas

Las Épicas se caracterizan por ser una historia de usuario muy grande, permitiendo ser desglosada en historias de usuario pequeñas. Para poder identificar una épica, se deben conocer las necesidades puntuales que tienen los usuarios frente a una situación en específico. Para el presente proyecto, se decide implementar una teoría propia, que permita identificar de manera adecuada las épicas del proyecto. Esta teoría se basa en representar a cada involucrado como una épica, y de esta forma dividir cada historia de usuario según a la persona a quien va dirigida. Para el desarrollo de la épica, se deben encontrar los elementos descriptivos que definen al involucrado, el producto o solución a desarrollar en base a las necesidades y el valor que genera la solución:

- **Identificación del involucrado:** Se debe identificar a todos los involucrados en el proyecto, con sus respectivas necesidades.
- **Productos:** Se deben presentar todas las soluciones frente a las necesidades presentadas por los involucrados.

- **Valor:** Es identificado como el valor que pueden generar las soluciones presentadas a los involucrados.

Historia de Usuario

Las historias de usuario son definidas como frases cortas, que se encargan de describir una funcionalidad desde la perspectiva del usuario [65]. Estas frases deben ser lo más precisas posibles, para ser realizadas de manera correcta en la fase desarrollo. Para este proyecto, se piensa utilizar una plantilla para poder distinguir todos los componentes de cada historia de usuario, cabe resaltar que, conforme se vaya desarrollando el proyecto, se pueden agregar componentes nuevos a la tabla de ser necesario:

Desarrollo ágil: Historias de usuario y criterios de aceptación

Elaborado por: oficinaproyectosinformatica.blogspot.com

Identificador (ID) de la Historia	Enunciado de la Historia				Criterios de Aceptación		
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento
XX-XXXX-XXXX	Como un [Rol]	Necesito [Descripción de la Funcionalidad]	Con la finalidad de [Descripción razón o resultado]	1	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				2	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				3	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				4	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
XX-XXXX-XXXX	Como un [Rol]	Necesito [Descripción de la Funcionalidad]	Con la finalidad de [Descripción razón o resultado]	1	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				2	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				3	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]
				4	[Título del escenario]	En caso que [Contexto] y/ o [Contexto]	cuando [Evento]

Fig 8. Plantilla de los componentes de las historias de usuario

- **Identificador:** Es un ID que se encarga de, como su nombre lo indica, identificar a la historia de usuario. Usualmente está conformado por la letra inicial de la épica a la que pertenece la historia de usuario y el número que tiene dentro de la épica (Orden).
- **Rol:** Es el nombre asignado a la épica.
- **Funcionalidad:** Como su nombre lo indica, es la funcionalidad asignada de la historia de usuario.
- **Número de escenarios:** Es la cantidad de escenarios que puede presentar la funcionalidad de la historia de usuario. Está relacionado con el criterio de aceptación y el contexto.

- **Criterio de aceptación:** Son las características que la historia de usuario debe cumplir en su desarrollo.
- **Contexto:** Son las situaciones que puede tener un usuario o el sistema frente a la funcionalidad presentada anteriormente. La cantidad de contextos presentes en una historia de usuario, dependen de la cantidad de escenarios definidos anteriormente.
- **Evento:** Es una explicación resumida que facilita la comprensión del contexto.

Priorización de las Historias de Usuario

La priorización de las historias de usuario corresponde al nivel de importancia otorgado a las historias de usuario dentro de un Sprint y tiene como finalidad permitir al grupo de trabajo poder identificar el orden en el que tienen que ser desarrolladas dichas historias de usuario. Esta priorización se puede manejar de diferentes maneras, ya sea por colores, por números o símbolos. Para este proyecto, se planea el uso de colores y nombres, facilitando la identificación de estas historias de usuario dentro del Sprint:

- **Highest:** Identificado con el color rojo, hace referencia a las historias de usuario más importantes para el desarrollo del proyecto.
- **Critical:** Se identifica con el color anaranjado y representa a las historias de usuario que tienen un nivel de importancia crítico dentro del sprint.
- **High:** Representado con el color amarillo, este es el nivel intermedio de todos los 5 niveles. Se caracteriza por referenciar a las historias de usuario que no cuentan con una necesidad de desarrollo tan alta ni tan baja.
- **Medium:** Referenciado con el color verde, este nivel representa a las historias de usuario que tienen una prioridad baja en referencia al estándar.
- **Low:** Identificado con el color azul, este nivel hace referencia a las historias de usuario que tienen un nivel de prioridad muy bajo y que no estarán en la fase de desarrollo del proyecto, pero si en el diseño.

3.2.4.7 Organización de los requerimientos: Kanban

Las historias de usuario deben ser dispuestas de forma que se facilite su autoasignación y desarrollo. Kanban es una metodología útil que resuelve la necesidad planteada, utilizada para la organización del trabajo la cual se basa en formas visuales a través de tarjetas para la priorización y gestión del esfuerzo de las historias de usuario. En Kanban se producen y realizan los trabajos que el equipo de desarrollo puede hacer, no hay necesidad de sobrecargar de tareas al equipo.

Esta propuesta metodológica ágil basada en la productividad tiene algunas prácticas a la hora de gestionar productos como se definen en “La Guía oficial de Kanban” [66]:

- **Visualización:** Las tareas deben ser visualizadas para una buena colaboración dentro del equipo, estas pueden representarse a través de tarjetas en un tablero, el cual divide las tareas en progreso, por realizar o terminadas.
- **Limitar el trabajo en progreso:** Debe haber un límite de tareas en progreso en un periodo de tiempo con el fin de no agotar recursos.
- **Manejar el flujo de trabajo:** El fin de manejar el flujo de trabajo a través del tablero es completar el proyecto en los lapsos definidos y de la mejor manera posible.
- **Crear políticas explícitas:** En cada tarea creada se debe definir quien realizará la tarea, en cuanto tiempo, cada cuanto se harán reuniones y demás aspectos con el fin de que todo el equipo entienda sus roles y responsabilidades.
- **Implementar ciclos de retroalimentación:** Con el fin de mejorar el producto continuamente es necesario retroalimentar las mejoras realizadas al servicio con el equipo.
- **Mejorar la colaboración, evolucionar experimentalmente:** Como se define en uno de los fines principales de Kanban, se debe buscar el mejoramiento del sistema a través de la adaptabilidad del equipo.

Ahora bien, aplicar Scrum y Kanban es posible dado que se pueden seguir los ciclos de Scrum a través del tablero de organización de requerimientos Kanban. Para esto se pueden utilizar herramientas como Jira o Trello, en este caso se aplicará Trello, puesto que cuenta con una interfaz muy intuitiva, facilitando el trabajo en equipo. En Trello se pueden utilizar diferentes marcos de desarrollo, para este proyecto se utilizara la base para proyectos de Ingeniería, puesto que en el tablero se mostrarán todos los procesos de Scrum para el ciclo de vida del software, como se observa en la siguiente figura.

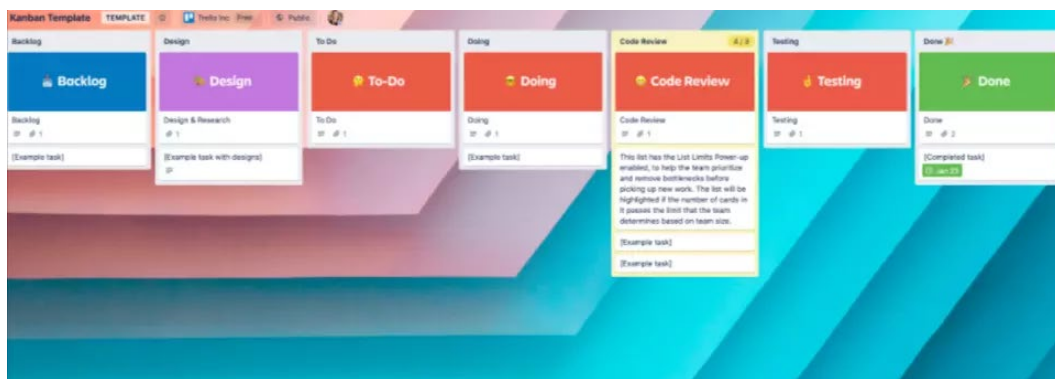


Fig 9. Tablero Kanban para proyectos de Software.

En este tablero se introducen nuevamente conceptos como el Product Backlog de Scrum, pero con algunos cambios significativos. Sin embargo, para el desarrollo del prototipo se aplicarán las reuniones diarias de Scrum así como las divisiones por Sprints y las retrospectivas de los mismos para el incremento del Producto Mínimo Viable.

En la tarjeta de Backlog se cargarán las historias de usuario ya predefinidas a partir del grupo de análisis teniendo en cuenta los requerimientos encontrados, estas historias ya deberán tener una puntuación definiendo su complejidad o tiempo de realización y una priorización.

Los elementos en el apartado de diseño serán como dice su nombre diseñados por el equipo de diseño, en este paso cada tarea tendrá unos modelos propios que serán entregados al equipo de desarrollo.

Una vez planteados los diseños el equipo de desarrollo recibe la historia de usuario en el apartado de To-Do. Aquí, cada miembro tendrá la autonomía de escoger la historia de usuario que tenga la capacidad de realizar sin problemas, es importante que el trabajador escoja teniendo en cuenta sus habilidades técnicas. Una vez escogida la historia de usuario deben

pasar la historia de usuario a “Doing” en este momento ya el equipo sabrá quien está desarrollando dicha historia de usuario.

Una vez finalizada la historia de usuario deberá pasar a revisión en esta etapa el equipo se reúne y revisa el incremento realizado en este paso, posteriormente ya una vez revisada la historia de usuario podrá pasar al testing. En el testing en caso de encontrar errores la tarea deberá contener un apartado de “Bugs” en la cual se definirán qué errores debe corregir el equipo de esa tarea. Por otro lado, si no se encuentran errores en la tarea esta pasará a Done y el encargado de esta podrá pasar a otra que se encuentre en To-Do.

3.2.4.8 Diseño de los requerimientos

Un factor importante a la hora de desarrollar un software, es el diseño de cada requerimiento en base a los análisis realizados. Para estructurar el proceso de diseño existen múltiples metodologías como el Diseño Enfocado en el Usuario, Diseño Enfocado en el flujo de datos y el Diseño Enfocado a Objetos.

Diseño enfocado a objetos

La metodología de diseño enfocada a objetos se basa en tres conceptos principales: Abstracción, Esconder Información y Modularidad. El objeto es una estructura de datos privada con operaciones relacionadas que pueden transformar la estructura de datos [67]. En otras palabras, el diseño orientado a objetos es un método de análisis y diseño que examina las necesidades desde la perspectiva de Clases y Objetos encontrando una solución de nivel lógico. Aprovechando conceptos del paradigma como la **Herencia, Polimorfismo, Encapsulación y Abstracción.**

Estos requerimientos pueden ser modelados mediante el Lenguaje de Modelado Unificado (UML). Este lenguaje diseña y proporciona los planos de un software, para poder visualizarlo mejor, ayudando a inspeccionar a fondo todas las funciones que tiene o puede tener. Además, proporciona todas las vías de comunicación que el software tendrá. Existen diferentes tipos de modelos UML, los cuales se explicaran a continuación:

Diagrama de clases

Proporcionan un panorama estático del sistema, muestran las clases, sus relaciones y sus atributos. Las clases cuentan con atributos, estos a su vez poseen un nombre, un tipo y un valor.

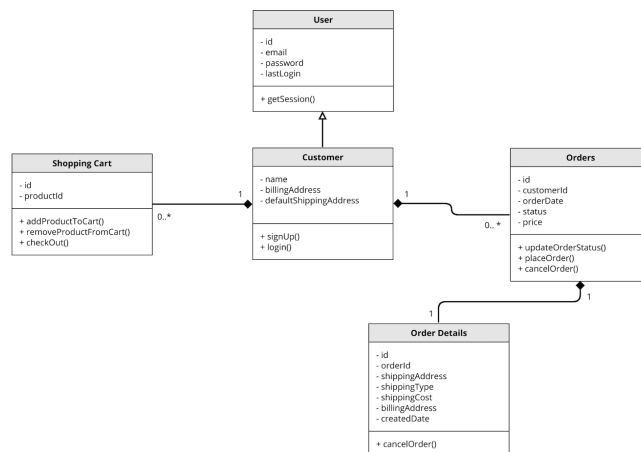


Fig 10. Ejemplo de un diagrama de clases de un e-commerce.

En este ejemplo se puede observar un diagrama de clases de un e-commerce, cuenta con las clases de: Usuario, Cliente, Carrito de compras, Pedidos y Detalles de Pedido.

El Usuario cuenta con los atributos de id, email, password y último acceso, además cuenta con una acción, la cual es iniciar sesión. La relación que hay es de tipo sucesión, esto quiere decir que, la clase Usuario es hija de la clase Cliente.

También se encuentra la clase Cliente, esta tiene los atributos de nombre, dirección de facturación y dirección de envío, también cuenta con dos acciones, registrarse e iniciar sesión. La clase Cliente cuenta con dos relaciones de composición, hace referencia a las clases que no pueden existir sin la otra por sí solas, están conectadas por una línea que apunta a la clase compuesta con una flecha en forma de diamante rellena, en este caso la clase Cliente.

Ahora bien, la primera clase que se encuentra relacionada con la clase Cliente es la clase Carrito de compras, sus atributos son el id y el id del producto, igualmente tiene acciones a realizar, como el agregar productos al carrito, remover productos del carrito y verificar.

Por otro lado, se cuenta con la clase Pedidos, cuenta con los atributos de id, id del cliente, fecha de pedido, estado y precio, sumado a esto, cuenta con acciones cómo actualizar estado de pedido, realizar pedido y cancelar pedido. Algo puntual es que, además de tener una relación de composición con la clase Cliente, también tiene otra relación de composición con la clase Detalles de pedido.

Por último, se encuentra la clase Detalles de pedido, tiene una relación de composición con la clase Pedido, sus atributos son id, id de pedido, dirección de envío, tipo de envío, costo de envío, dirección de envío y fecha de creación. Su única acción es cancelar el pedido [68].

Diagrama de secuencia

El diagrama de secuencias describe cómo y en qué orden, un grupo de objetos funcionan en conjunto. También, se usa para comprender los requisitos de un sistema nuevo o documentar un proceso existente.

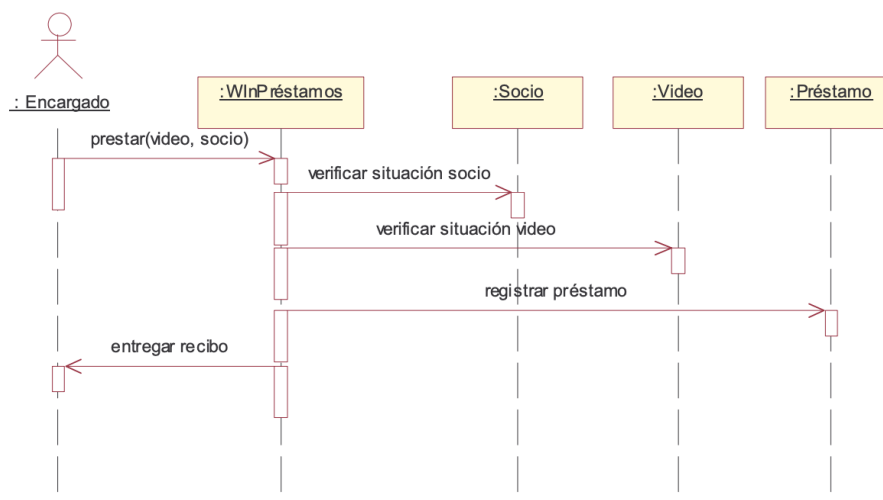


Fig 11. ejemplo de un diagrama de secuencia.

Este diagrama presenta un ejemplo de una tienda de vídeos. El encargado presta un vídeo a un socio, este proceso verifica el estado del socio, también debe verificar el estado del vídeo, posteriormente, se registra el préstamo en el sistema y se entrega el recibido. La interpretación que se puede extraer de este diagrama no es complicada, pero su creación es muy compleja si se cuentan con muchos procesos y usuarios [69].

Diagrama de casos de uso

Utilizado para mostrar las interacciones entre los actores y el sistema, el diagrama de casos de uso, documenta el comportamiento que tiene un sistema, desde el punto de vista del usuario. Es por eso que se determinan los requisitos funcionales del sistema, en otras palabras, representa las funciones que un sistema puede ejecutar.

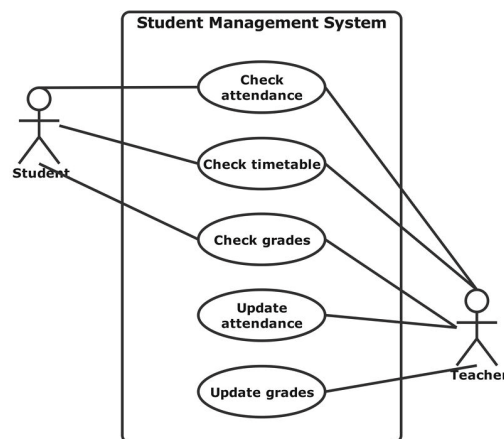


Fig 12. Ejemplo de un diagrama de casos de uso.

Este es un claro ejemplo de un diagrama de casos de uso de un sistema de gestión de estudiantes y dentro se encuentran dos actores, los estudiantes y los profesores. Los estudiantes pueden realizar tres acciones dentro del sistema, comprobar la asistencia, consultar el horario y verificar las calificaciones. Pero los profesores pueden realizar dos acciones más que los estudiantes no tienen permitido hacer, actualizar la asistencia y actualizar las calificaciones.

Diseño UX/UI

El diseño UX/UI está conformado por los diseños de la experiencia de usuario (UX) y la interfaz del usuario (UI). El diseño UX, se encarga de percibir todo lo que el usuario piensa, siente e intuye al momento de interactuar con el producto. Por otro lado, el diseño UI, se encarga de “vestir” el producto, con la finalidad de hacerlo más llamativo para los usuarios, reflejando la marca de la empresa en él. Estos diseños son indispensables para el desarrollo de un producto, porque permiten estudiar y analizar al usuario, al punto de conocer sus necesidades y buscar una manera de solucionarlos, mediante la planificación y el diseño de un producto que se ajuste al gusto de ellos [70].

Metodología de diseño UX/UI

La metodología de diseño UX/UI está compuesta por un conjunto de métodos que se encargan de estandarizar, planificar y desarrollar los diseños de interfaz y experiencia de usuario, realizando un trabajo óptimo con principios de calidad. Estos métodos se dividen por componentes de la aplicación, ya sea la manera de presentar la información, la estructuración de la aplicación o los elementos para crear o interactuar con la información de la aplicación (Formularios, botones, etc). Los métodos que conforman la metodología son los siguientes:

Esquemas de organización

Se caracterizan por la manera de categorizar su contenido, creando diversas relaciones entre cada uno de sus componentes, estos esquemas se dividen en exactos y subjetivos:

Esquemas exactos

Dividen la información en secciones mutuamente excluyentes, facilitando su creación y permitiendo categorizar su contenido para crear la arquitectura del producto. Es importante conocer al usuario, puesto que él debe entender cómo está compuesto el producto, para poder encontrar lo que busca dentro de este.

Existen diferentes tipos de esquemas organizacionales exactos, como los esquemas alfabéticos, los cuales organizan los componentes con etiquetas, las cuales contienen las 26 letras del abecedario, con el fin de que el contenido coincida con lo que el usuario busca. En ocasiones, los esquemas alfabéticos componen una navegación secundaria complementaria en el producto, facilitando y organizando su contenido.

También, se encuentran los esquemas cronológicos, los cuales organizan la información por medio de fechas. Para asegurar el éxito de estos esquemas, se debe tener total registro y disponibilidad del contenido almacenado. Por otro lado, están los esquemas geográficos, caracterizados por organizar la información por lugares, permitiendo un fácil diseño y desarrollo del producto.

Esquemas subjetivos

Se caracterizan por ordenar la información, mediante la organización del producto o el campo que ocupa. Son difíciles de diseñar, pero en ciertos casos, pueden llegar a ser mucho más útiles que los esquemas exactos. Existen varios esquemas subjetivos, como los esquemas por

temas, los cuales organizan la información en función a algo en específico o los esquemas de audiencia, los cuales permiten organizar el contenido del producto, por medio de segmentos de usuarios distintos. Este esquema permite al usuario la navegación entre diferentes tipos de audiencia, facilitando su búsqueda y permitiendo la identificación del usuario con su audiencia [71].

Estructuras organizativas

Se define como la forma de identificar las relaciones entre los contenidos del producto. Estas estructuras permiten al usuario predecir en dónde se encuentra la información que buscan dentro del producto. Para diseñar estas estructuras, se deben tener en cuenta las expectativas de los usuarios e implementar métodos organizacionales coherentes, presentando la información para que los usuarios puedan hacer uso de ella. En algunos casos puede ser muy útil escoger una estructura organizativa por sobre otra, pero en ciertas ocasiones, puede llegar a ser muy útil el implementar las tres formas complementarias de estructuración organizativas, estas estructuras son:

Estructura Jerárquica

También conocidas como estructuras de árbol o estructuras de eje y radio, cuentan con un enfoque de relaciones padre/hijo entre los componentes del producto. Por lo general, permite a los usuarios el acceso a información muy amplia, al punto de empezar a profundizar más en la estructura, para luego encontrar la información que buscaban. Los usuarios están muy familiarizados con la estructuración de la información en jerarquías, puesto que este método es utilizado en la planificación y diseño de muchos productos de uso diario.

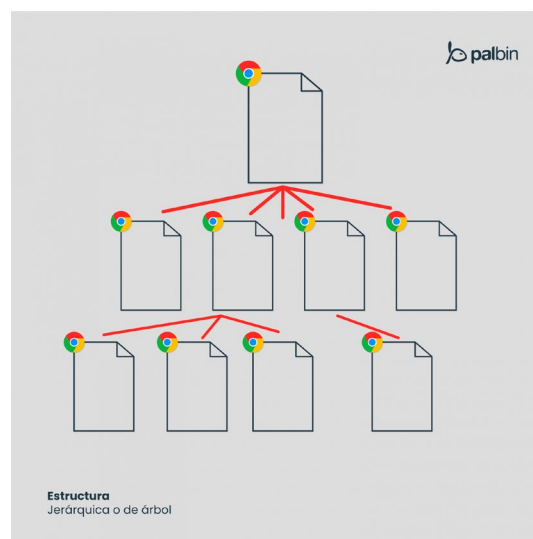


Fig 13. Ejemplo de la Estructura Jerárquica.

Estructuras Matriciales

Permite determinar el camino de cada usuario dentro del producto, puesto que el contenido está vinculado de múltiples maneras. Esta estructura aprovecha al máximo los principios del HTML, permitiendo al usuario escoger la navegación dentro del producto, por medio de contenido organizado por fechas o un tema en específico.

Aparte de estas 2 estructuras complementarias, existen otras estructuras que destacan también en el diseño del producto:

Estructura Modelo de base de datos

Denominada modelo de base de datos, la cual adopta un enfoque de estructura ascendente. Su contenido está soportado por los vínculos creados a través de los metadatos del contenido, facilitando la experiencia del usuario, haciéndola más dinámica y permitiendo realizar búsquedas o filtrados más avanzados, proporcionando información relacionada de interés [72].

Patrones de diseño UI

Los patrones de diseño son definidos como soluciones a problemas comunes de usabilidad de una aplicación web. Su eficiencia se refleja gracias a la popularidad y adaptación, ya que entre más es usado un patrón de diseño, se vuelve más potente, permitiendo su adaptación en distintas aplicaciones. Existen diferentes tipos de patrones de diseño [73]:

Inputs Hints

Los Inputs Hints (Sugerencias de entrada), son un patrón de diseño utilizado, generalmente, en las etiquetas de los campos de los formularios, aclarando al usuario lo que debe ingresar. Este patrón es de fácil implementación, puesto que el HTML5, facilita la inclusión de texto marcador, dentro de los campos de entrada [74].

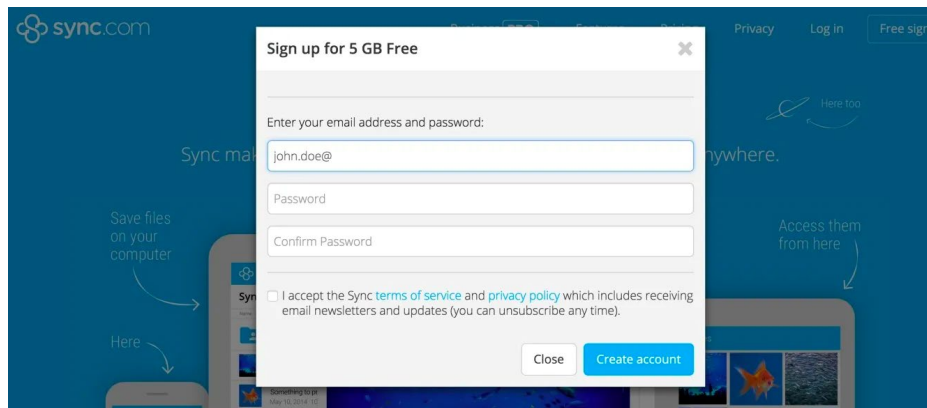


Fig 14. Ejemplo de Inputs Hints.

Forgiving Formats

Este patrón de diseño se caracteriza por aceptar múltiples formatos, esto quiere decir que, el sistema acepta todos los formatos relevantes que el usuario puede ingresar en una sección en específico, ahorrando demasiado tiempo en el diseño, puesto que evita la inserción de diferentes campos en el formulario.

Esto ofrece una gran libertad y comodidad al usuario, puesto que el usuario puede llegar al mismo lugar de la aplicación por diferentes circunstancias. Usualmente, este patrón se utiliza de la mano con los Inputs Hints, puesto que ayudan a especificar al usuario que puede ingresar información relevante, entregando siempre la misma respuesta [75].

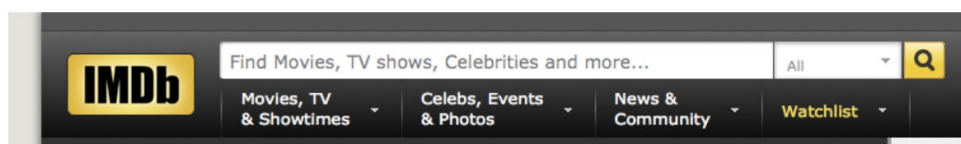


Fig 15. Ejemplo de Forgiving Formats.

Undo

Se caracteriza por permitirle al usuario deshacer acciones que se han realizado, como eliminar contenido, peticiones, solicitudes, entre otros. Esta opción de deshacer permite crear una interfaz de usuario segura, tolerable e intuitiva, reflejando confianza al usuario a la hora de experimentar [76].

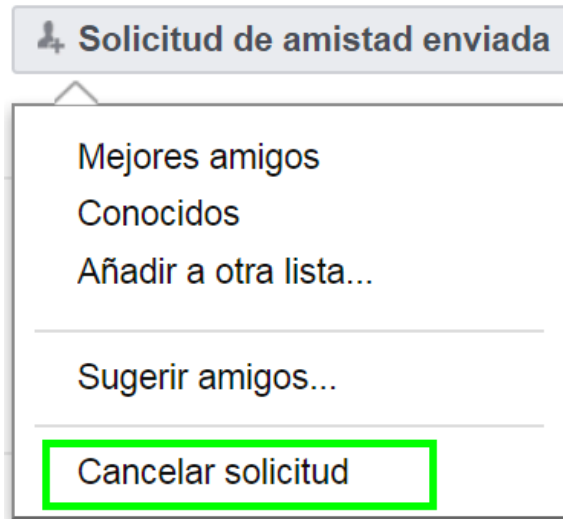


Fig 16. Ejemplo de Undo

Inline Validation

El patrón de diseño Inline Validation, proporciona información instantánea al usuario sobre el estado de los campos del formulario, en los cuales se ingresa información, con la finalidad de reflejar si esa información es válida o no. Usualmente se utilizan mensajes de color rojo, para guiar al usuario hacía ese campo en el formulario que, posiblemente, tenga algún error de digitación. Este patrón ayuda a diseñar una interfaz de usuario más interactiva, facilitando la identificación de los errores y solución de estos, por parte del usuario [77].

VALIDACION FORMULARIOS

Usuario <input type="text" value="john123"/> ❌ <small>El usuario tiene que ser de 4 a 16 dígitos y solo puede contener números, letras y guion bajo.</small>	Nombre <input type="text" value="Carlos Arturo"/> ✅
Contraseña <input type="password" value="....."/> ✅	Repetir Contraseña <input type="password" value="....."/> ❌ <small>Ambas contraseñas deben ser iguales.</small>

Fig 17. Ejemplo de Inline Validation.

Social Login

Este patrón de diseño propone aumentar las opciones en las que el usuario puede iniciar sesión en la aplicación, permitiendo el acceso mediante una red social o un correo electrónico. Estas opciones de ingreso se encuentran en el formulario de inicio de sesión de la aplicación, ofreciendo una gama de posibilidades al usuario. En ocasiones, los usuarios prefieren realizar el inicio de sesión o registro de la aplicación, mediante estas opciones, con la finalidad de ahorrar tiempo [78].

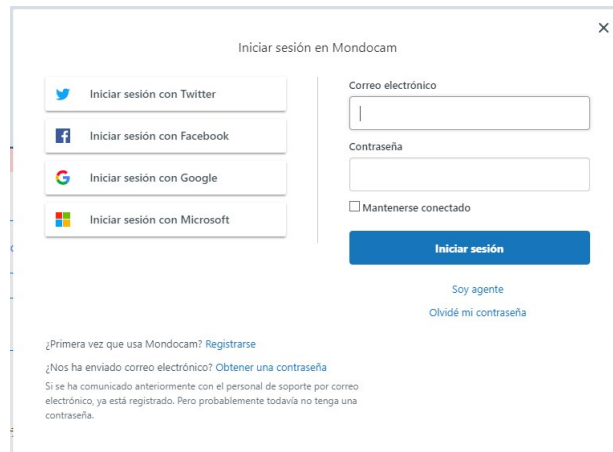


Fig 18. Ejemplo de Social Login.

Flagging/Reporting

El patrón Flagging/Reporting, se caracteriza por ofrecer un diseño, en el cual los usuarios podrán crear reportes de conducta inapropiados, realizados dentro de la plataforma. Esto es importante para poder tener un control sobre los comportamientos de los usuarios, ya sea en una red social o en un marketplace. Plantea la opción de poder realizar reportes por contenido obsceno, ofensivo, mal etiquetado o publicado, spam, violaciones a las políticas del sitio, derechos de autor, entre otros. Este patrón de diseño permite aumentar el control sobre los usuarios, ofreciendo un entorno saludable y seguro [79].



Fig 19. Ejemplo de Flagging/Reporting

WYSIWYG Text Editor

Se caracteriza por ofrecer al usuario la posibilidad de conocer con precisión, el aspecto que tendrá su texto al momento de estar terminado, eliminando todas las preguntas sobre los otros formatos que puede tomar el texto. Además, le permite al usuario poder utilizar otras herramientas para editar su texto a gusto, permitiendo la personalización completa de este [80].

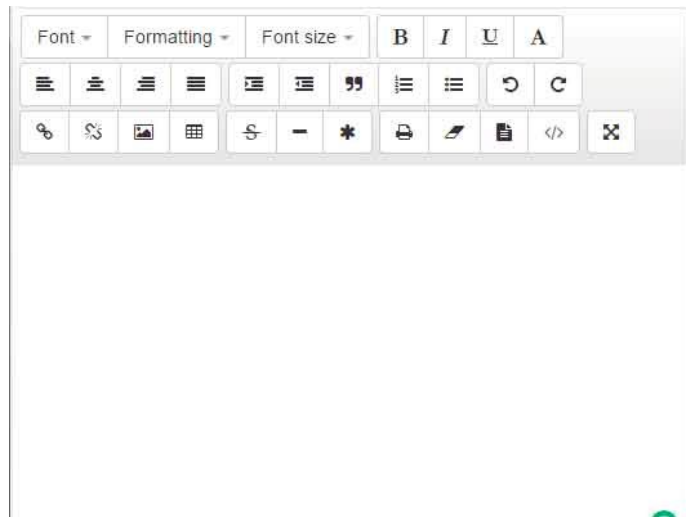


Fig 20. Ejemplo de WYSIWYG Text Editor.

Searches

Conocido por ser uno de los patrones más simples que hay, puesto que solamente cuenta con un elemento y es la opción de búsqueda. Está representado como una caja de entrada de texto, ubicada en la parte superior de la aplicación, permitiendo al usuario encontrar contenido relacionado dentro de la aplicación, es por esto que, este patrón es uno de los más populares actualmente [81].

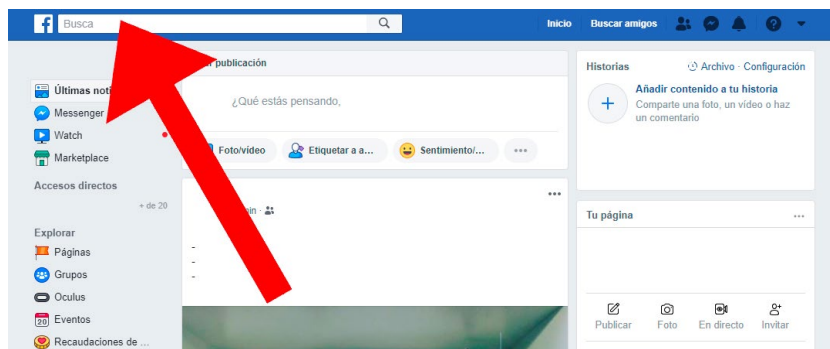


Fig 22. Ejemplo de Searches.

Notifications

Se popularizó en el diseño de todas las redes sociales, puesto que es usado como una señal para comunicar al usuario sobre su interacción con otros usuarios. En algunas ocasiones, también es usado para comunicar al usuario sobre nuevos contenidos, productos, noticias, entre otros. Este patrón de diseño puede ser personalizado por el usuario, permitiéndole decidir sobre qué quiere recibir notificaciones. Es representado por un icono de una campana y cuando contiene una notificación, se representa la cantidad de estas por medio de un número y un círculo de color rojo, esto con el fin de resaltar en toda la interfaz del usuario [82].

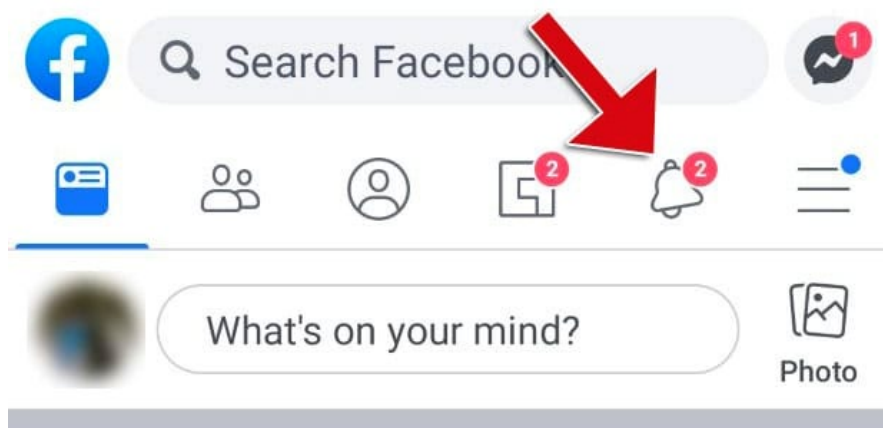


Fig 22. Ejemplo de Notifications.

Sticky (Fixed) Navigation

Es implementado en aplicaciones que cuenten con un feed de desplazamiento infinito, puesto que cuenta con un menú de navegación estático, ubicado en la parte superior o una parte lateral del diseño. Esto permite al usuario la oportunidad de cambiar de rumbo, sin necesidad de volver al inicio del feed [83].

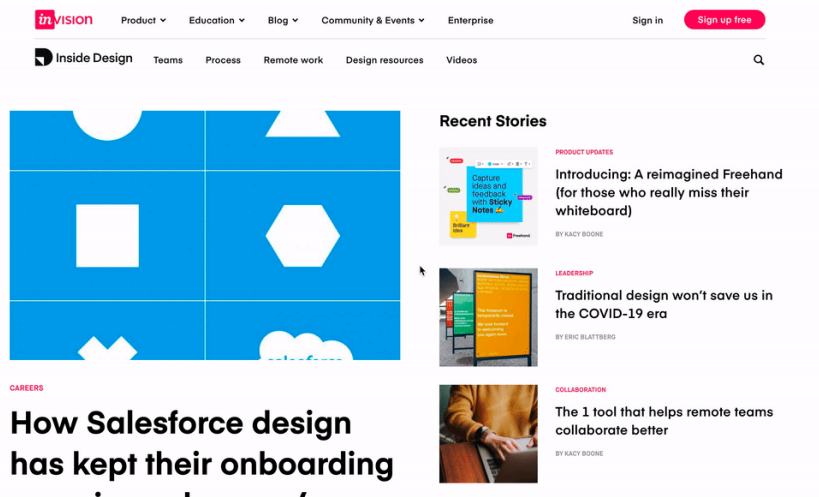


Fig 23. Ejemplo de Sticky (Fixed) Navigation.

Vertical Navigation

Como su nombre lo indica, implementa un menú de navegación vertical, ubicado en una parte lateral de la aplicación. Permite numerar las secciones importantes, teniendo en cuenta la importancia de estas. Este patrón es utilizado por muchas aplicaciones, que quieren ofrecer a los usuarios todas las funcionalidades posibles de ejecutar. Esta navegación es estática, por lo que el usuario podrá acceder a ella, sin importar en qué sección se encuentre de la aplicación [84].

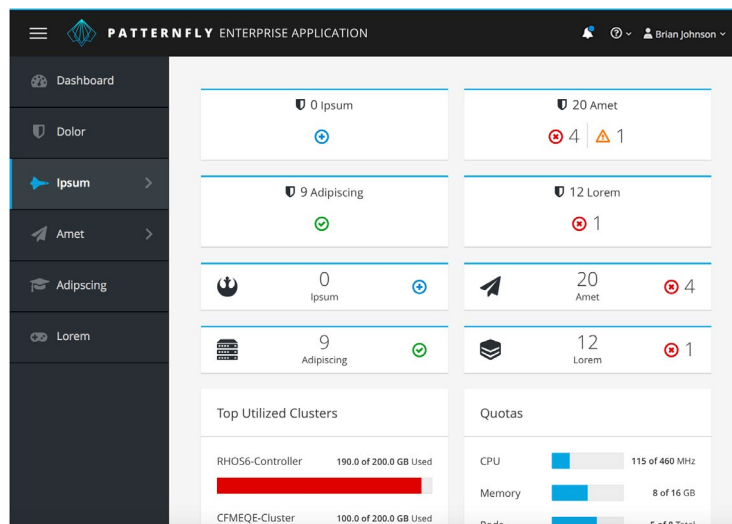


Fig 24. Ejemplo de Vertical Navigation.

Modals

Se caracteriza por funcionar mediante una interacción, por parte del usuario, permitiendo desempeñar su función. Tiene como finalidad el comunicar información importante al

usuario, al realizar click en algún componente importante. Está conformado por la información que se entregará al usuario y una serie de botones, los cuales serán modificados, dependiendo de la utilidad que tenga el modal [85].

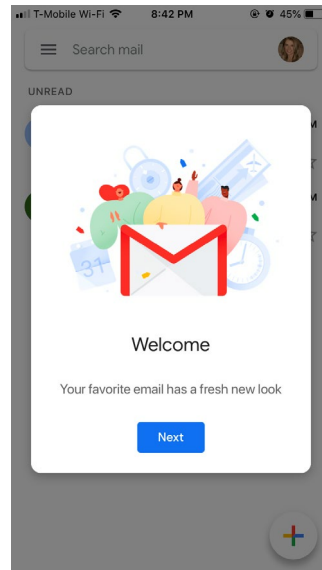


Fig 25. Ejemplo de Modals.

Recommendations

Se enfoca en ofrecer a los usuarios, productos o servicios que sean de su preferencia. Estas recomendaciones se adaptan específicamente a los gustos individuales de cada uno de los usuarios. Este patrón es implementado en aplicaciones de comercio electrónico o redes sociales, puesto que estas disponen de información necesaria, para crear un patrón de preferencias para cada uno de sus usuarios [86].

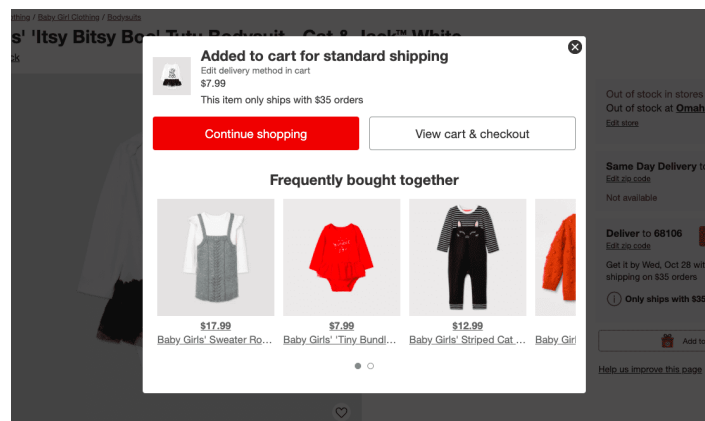


Fig 26. Ejemplo de Recommendations.

Diagrama de despliegue

Este diagrama se caracteriza por mostrar las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Representa la disposición de los componentes que se ejecutan en los nodos conectados por enlaces de comunicación. Los nodos son los recursos de ejecución, como lo son los computadores, un dispositivo o memoria [87].

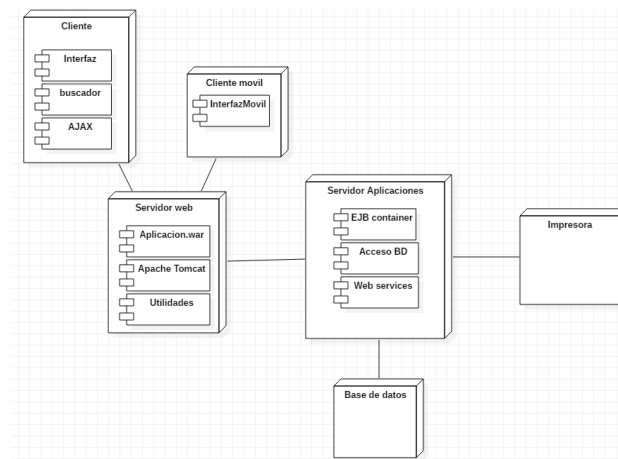


Figura 27. Ejemplo de un diagrama de despliegue.

En este ejemplo se puede observar un diagrama de despliegue de una aplicación web, cuenta con los nodos de Cliente, Cliente móvil, Servidor Web, Servidor Aplicaciones, Base de datos e Impresora. El nodo de Cliente, cuenta con subnodos como Interfaz, buscador y AJAX, además cuenta con una conexión al nodo de Servidor Web.

El nodo Cliente móvil cuenta con el subnodo de Interfaz móvil, además, tiene una conexión al nodo Servidor Web. Luego, se encuentra el nodo Servidor Web, el cual tiene tres subnodos, Aplicación.war, apache Tomcat y Utilidades, este nodo tiene una conexión al nodo Servidor Aplicaciones. Dentro del nodo Servidor Aplicaciones, se encuentran tres subnodos, EJB container, Acceso BD y Web Services. Por último, se encuentran dos conexiones a dos nodos diferentes, Bases de datos e impresora.

3.2.4.9 Arquitectura de la aplicación

Otro aspecto importante a la hora de diseñar cualquier sistema es tener en cuenta la arquitectura del mismo. La Arquitectura de Software se define en el libro “Software

Architecture in Practice 2nd ed” como la estructura o estructuras internas de un sistema, las cuales comprenden elementos de software, así como los elementos externos visuales de esos elementos y las relaciones entre ellos [88]. En otras palabras, la arquitectura de un sistema determina cómo se estructura una aplicación, como se comunicara el servidor y el cliente entre sí, y la interacción entre otros sistemas. Algunos de los elementos importantes a la hora de definir la arquitectura son: clientes, servidores, bases de datos, filtros, niveles jerárquicos y las interacciones entre estos.

Algunos de los diseños más populares a la hora de estimar la arquitectura de un sistema son: Microservicios, Cliente-Servidor, Por capas, etc. En este caso se implementará el patrón de arquitectura MVC.

Patrón MVC

El patrón Modelo Vista Controlador (MVC) es un patrón de diseño de arquitectura que simplifica el desarrollo y mantenimiento de la aplicación [89]. Esto dado que separa el funcionamiento del sistema en 3 capas principales:

- **Modelo:** Esta capa es la responsable de manejar la lógica de la aplicación. En esta se encapsulan las clases para el acceso a los datos. Usualmente, en el modelo, se encontrarán las abstracciones para el acceso a la base de datos, el manejo de correos electrónicos, etc.
- **Vista:** Es la capa encargada de manejar el diseño visual de la aplicación. En esta se controlan los archivos HTML, CSS o Javascript que se envían al cliente a través del controlador, al igual que el manejo de las entradas por parte del mismo.
- **Controlador:** El controlador es el encargado de integrar las dos últimas capas en una sola. En este se definen los permisos en cada página a redireccionar, los datos que se envían y controla el flujo de información entre las mismas.

El proceso de interacción entre el cliente y el servidor en el patrón MVC se resumiría de la siguiente forma: El cliente hace una petición a través del navegador al servidor web, este recibe la solicitud y el **controlador** define internamente que debe retornar al usuario. En caso de ser información de la base de datos, hace la petición al modelo para que acceda a la base

de datos y retorna la información solicitada, posteriormente el controlador envía la información a la vista y esta la renderiza al cliente. En caso de no necesitar información de la base de datos, envía directamente la página a ser renderizada por el navegador.

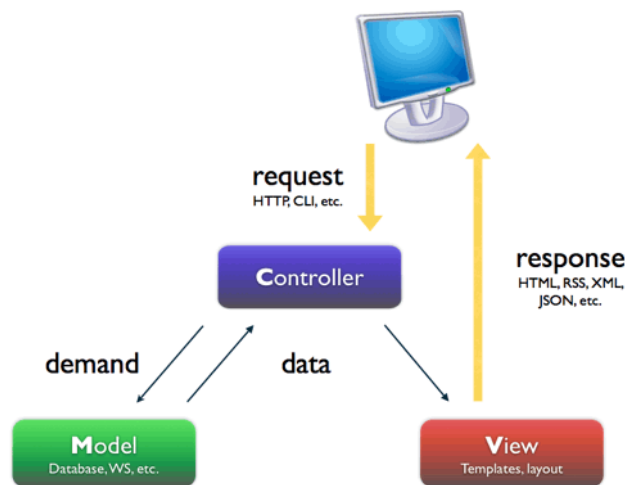


Figura 28. Interacción entre el cliente y el servidor en el patrón Modelo Vista Controlador.

Algunas de las ventajas principales al desarrollar con esta arquitectura son:

- Se puede dividir la complejidad de la aplicación en las 3 capas principales.
- Los formularios en la aplicación no se manejan desde el servidor completamente, por lo que se podrá tener total control de la información que ingresa al sistema.
- El controlador maneja todas las peticiones realizadas por el usuario. Además, este controla todas las rutas dentro de la aplicación.
- El testeo es más simple al poder encontrar errores en cada módulo por aparte.

3.2.4.10 Desarrollo Web

Siguiendo con la metodología planteada, una vez determinada la arquitectura y definidos los diseños a partir de los requerimientos, se puede dar paso a la etapa de desarrollo del prototipo. Esta corresponde a la codificación, programación y mantenimiento de un sitio web. El desarrollo web se divide en dos áreas principales, el desarrollo frontend y el desarrollo backend, ambas son de suma importancia puesto que cada una compone herramientas de desarrollo totalmente independientes de su área [90].

Desarrollo Frontend

Se define el desarrollo frontend como la parte visual de la página web, mediante la cual el usuario podrá interactuar directamente por medio de un navegador. El desarrollo frontend está compuesto por el lenguaje de etiquetas HTML, el cual brinda una estructura básica de la página o aplicación web. Los estilos son dados por el CSS, el cual puede aplicar librerías como Bootstrap, la cual facilita el diseño estético del sitio web. En cuanto a las animaciones, el lenguaje de programación Javascript, con su librería JQuery se encarga de implementarlas de ser necesario. A continuación se explican a fondo cada uno de los componentes mencionados anteriormente:

La estructura básica: HTML

Hypertext Markup Language (HTML por sus siglas) o mejor conocido como lenguaje de etiquetas, es utilizado para la creación, estructuración y organización visual de una página web. Está compuesto por comandos que son interpretados por el navegador del usuario, cuando éste accede a una página o aplicación web. El navegador ejecuta todas las órdenes contenidas en el código html, mostrando los elementos de dicha página, ya sean títulos, cuerpo, encabezado, párrafos, entre otros [91].

Estilos: CSS

El lenguaje CSS (Cascading Style Sheets), es un lenguaje de estilos desarrollado para controlar los estilos y presentaciones de una página web estructurada con HTML. Se encarga de separar el contenido y la apariencia, puesto que definen los colores, tamaños, formas y en algunos casos animaciones que se van a presentar en dicha página web. También, mejora la interacción, reduciendo la complejidad de uso para el usuario, facilitando el mantenimiento por parte del administrador y permitiendo que pueda ser visualizada en cualquier dispositivo [92].

Librerías

Gracias al desarrollo web, muchas personas crearon conjuntos de archivos denominados librerías. Las librerías están compuestas por códigos o datos en específico, tienen como función el solucionar un problema en concreto o facilitar el desarrollo de un proceso. En el caso del desarrollo web, más puntualmente en el frontend, se pueden encontrar librerías de CSS que facilitan el desarrollo y la implementación de estilos, como es el caso de Bootstrap.

Bootstrap librería para los estilos

Bootstrap, es un conjunto de librerías de CSS desarrolladas por Twitter. Se caracteriza por facilitar el diseño y estilo de una página web. Cuenta con una función llamada “Responsive”, esta ayuda a que la interfaz gráfica de la página o aplicación web se adapte al tamaño de los diferentes dispositivos. Su diseño es simple, limpio e intuitivo y además, cuenta con elementos predefinidos que son fáciles de modificar, como botones, barras de navegación, formularios, entre otros [93].

PurgeCSS Optimización de CSS

Una forma sencilla de reducir código CSS inutilizado y optimizar librerías robustas como Bootstrap, es usar PurgeCSS. Este es una herramienta escrita en NodeJs (entorno de tiempo de ejecución de JavaScript para back-end) que remueve CSS que no está siendo utilizado [94]. Cuando se utilizan librerías de CSS como TailwindCSS, Bootstrap o MaterializeCSS, en muchas ocasiones solo se utiliza una fracción de todos los estilos y elementos que ofrecen los frameworks, sin embargo, así no se usen los archivos se siguen incluyendo todos los estilos de CSS.

Por esta razón PurgeCSS es de gran utilidad, dado que, analiza todo el contenido de CSS y compara las clases usadas en el HTML, eliminando todas las clases de estilos que no se utilizan en el CSS. De esta manera, se disminuye el tamaño de los archivos y se obtienen tiempos de carga más rápidos, sin perder la utilidad de los Frameworks de CSS.

Lenguajes de Programación

Un lenguaje de programación es una agrupación de símbolos y reglas, sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. También, permite especificar los datos que se operan en un computador, la forma de almacenar los datos y la manera de transmitirlos [95].

Animación visual: Javascript

JavaScript es un lenguaje de programación utilizado principalmente para crear páginas web dinámicas. Estas se caracterizan por incorporar efectos en los elementos HTML que la conforman. En otras palabras, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos [96].

Javascript ayuda al diseño de la experiencia (UX) de una página o aplicación web, puesto que optimiza la interacción de los usuarios con el producto. Para poder realizar esto, cumple con los siguientes fundamentos:

- **Usabilidad:** Se encarga de evaluar la facilidad de los usuarios al utilizar una interfaz determinada. Una vez se conoce el funcionamiento de la interfaz, se revisa el tiempo en el que pueden completar ciertas tareas dentro de la misma. También, se estudia el tiempo que les toma a los usuarios el volver a reconocer las funcionalidades de la interfaz. Por último, se revisa el estilo de la interfaz, percibiendo que tan agradable es para los usuarios.
- **Interacción:** Desarrolla una guía que planifica y crea los puntos de contacto entre la interfaz y el usuario. El usuario debe ser capaz de determinar qué hacer o cómo utilizar el producto.

JQuery

JQuery es una librería de JavaScript utilizada para simplificar de manera rápida y concisa documentos HTML. También integra eventos, animaciones e interacciones con Ajax, para obtener un rápido desarrollo web. Está compuesta por un conjunto de herramientas de JavaScript, diseñadas para simplificar tareas escribiendo menos código. Estas son las características con las que cuenta jQuery [97]:

- **Manipulación del DOM:** JQuery facilita la selección de elementos del DOM, modificando su contenido, utilizando un motor de selección de código abierto llamado Sizzle.
- **Manipulación de eventos:** JQuery cuenta con una amplia variedad de eventos que se pueden integrar de manera elegante en una página web, sin necesidad de repetir las líneas de código.
- **Soporte AJAX:** Gracias al uso de tecnologías como AJAX, jQuery facilita el desarrollo de páginas o aplicaciones web “responsive”.

- **Soporte para todos los navegadores:** jQuery es compatible con todos los navegadores y funciona bien en IE 6.0+, FF 2.0+, Safari 3.0+, Chrome y Opera 9.0+.

Desarrollo Backend

La segunda área principal del desarrollo web, es el desarrollo backend. Se caracteriza por contener toda la parte lógica de una página web, la cual almacena, asegura, procesa, y analiza datos de forma constante, mostrando sus resultados en el frontend. Algunos procesos que se realizan en el backend son: conexión a la base de datos, autenticación de usuario, manejo de peticiones, brindar seguridad al sistema, gestión del flujo de información, etc. En el backend se encuentra el servidor, el cual contiene la base de datos del sistema. También, cuenta con un lenguaje de programación, el cual ayuda a manejar los procesos que se realizan, frameworks que ayudan a optimizar el desarrollo web y gestores de base de datos, para el almacenamiento de la información [98].

Python

Python es un lenguaje de programación interpretado de tipo dinámico. Al ser un lenguaje interpretado, Python puede ser ejecutado sin necesidad de ser procesado por un compilador, además de detectar los errores en el tiempo de ejecución. También es considerado multiparadigma, esto quiere decir que, a pesar de ser fuerte en la programación orientada a objetos, Python se puede aplicar para la programación imperativa, funcional, entre otros [99].

Además, cuenta con una amplia variedad de librerías y frameworks que facilitan la codificación y flujo de información de los proyectos, a continuación se hablará a fondo sobre las librerías y frameworks a utilizar para el desarrollo del prototipo:

Clean code: Flake8

PEP8 es una guía de estilos de Python que define cómo escribir código legible y de calidad en este lenguaje. La guía de estilos de python sugiere la indentación de código, la elección de espacios, la cantidad de caracteres por línea, organización de los imports, nomenclatura de los módulos, clases, métodos, variables, entre otros [100]. Una herramienta para corregir automáticamente errores de la convención PEP8 es la librería Flake8.

Flake8 revisa localmente el código realizado automáticamente y sugiere cambios para mantener los estándares mencionados. Para poder hacer uso de la librería, se debe ajustar la

versión correspondiente con respecto al proyecto a desarrollar, puesto que de no ser compatible con las versiones, no podrá comprobar los errores en el estilo del código [101].

Frameworks

El desarrollo web cuenta con frameworks, estructuras de software compuestas por elementos personalizables e intercambiables para el desarrollo ágil de una aplicación. Se caracteriza por acelerar el proceso en el desarrollo, además, reutiliza código ya existente y promueve buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por lo tanto, se puede definir como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas [102].

Framework web: Django

Django es un framework web de código abierto, gratuito y de alto nivel escrito en Python. Se basa en la aplicación de un ORM y la arquitectura MVT como patrón de diseño. Cuenta con una muy buena documentación, facilitando a los desarrolladores la creación de aplicativos web, de manera rápida, limpia y segura. Además, gracias a sus librerías de autenticación prediseñadas, ayuda al ahorro de código al crear registros de usuarios [103]. Utiliza una interfaz HTTP Wsgi, por medio de la cual el servidor y la aplicación web se comunican [104].

Patrón: MVT

Django cuenta con la arquitectura Modelo Vista Template MVT, este patrón es similar al patrón MVC planteado anteriormente, pero difiriendo sobre todo en la capa de la vista y el controlador. Este patrón se basa en los cuatro componentes principales, los cuales se explicarán a continuación:

- **URLs:** Las URLs (Uniform Resource Locator) o mejor conocidas como dirección electrónica. Son una secuencia de caracteres utilizados para identificar la localización de los recursos de una página web. Crean la conexión entre los recursos de la página web y el servidor, entregando la ubicación de la información solicitada y la manera de acceder a esta. Gracias a esta cadena de caracteres, se asigna una dirección de carácter único a cada recurso de información de la página web, permitiendo que el navegador encuentre y visualice la información solicitada [105].

Django utiliza un mapeador URL para redirigir las peticiones HTTP a las vistas apropiadas, utilizando la URL de la petición entregada. El mapeador URL empareja también, los patrones de cadenas o dígitos específicos que se reflejan en la URL, los entrega a la función de visualización como datos.

- **Vistas:** Una vista, es una función que recibe y devuelve peticiones HTTP. Las vistas acceden a los datos solicitados para satisfacer las peticiones por medio de modelos y delegan la respuesta a las plantillas denominadas “Templates”.
- **Modelos:** Los modelos son objetos de Python que definen la estructura de los datos en una aplicación y proporcionan mecanismos para gestionar (Añadir, modificar y borrar) y consultar registro a la base de datos.
- **Template:** El template es un fichero que define la estructura de una página HTML, utilizando marcadores de posición que representan el contenido total de las páginas a visualizar. Las vistas pueden crear dinámicamente una página web, utilizando un template, por medio de los datos de un modelo.

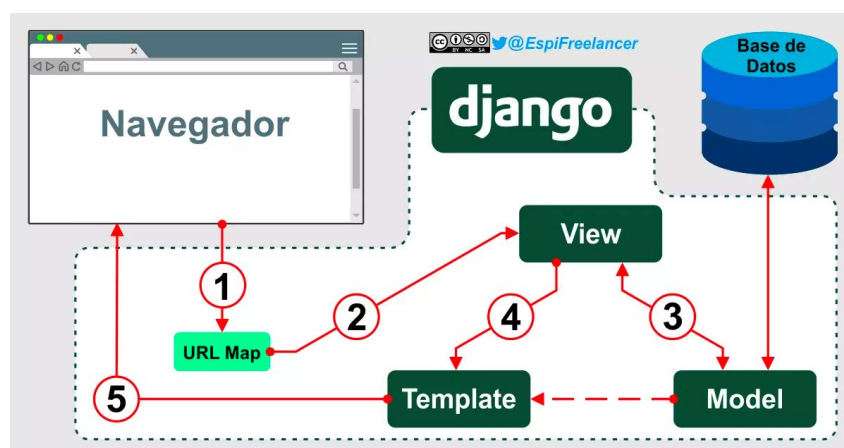


Figura 29. Arquitectura MVT (Modelo Vista Template)

En la **Figura 29**, se puede observar un diagrama que refleja la arquitectura MVT, el proceso que se realiza dentro de esta arquitectura es el siguiente: El primer paso es el manejo de las URLs, Django maneja todos los ficheros de las URLs de manera automática, haciendo que los desarrolladores solamente tengan que definir la ruta a las vistas.

Cuando se encuentra la URLs dentro del fichero, la vista se encarga de procesar la petición, ya sea para enviar, recibir o para reescribir datos. En caso de que la petición recibida solicite elementos de la base de datos, el Modelo se encarga de manejar la petición para crear, actualizar, eliminar o modificar elementos de la base de datos, mediante el mapeador de objetos relacionales (ORM).

Un ORM es un modelo de programación que permite asignar las estructuras de datos de una base de datos relacional a una estructura lógica orientada a objetos. Todos los elementos de la base de datos se vinculan con los objetos de la base de datos virtual, de tal modo que, se podrán realizar las acciones CRUD (Create Read Update Delete) de manera rápida y sencilla sobre la base de datos real a través de los modelos designados. El ORM hace la petición a la base de datos de manera automática, retornando la información a la vista. Una vez la Vista tenga la información, se renderiza en un template o plantilla para poder ser visualizada por el usuario.

Una funcionalidad importante y llamativa que tiene Django, es el modelo de autenticación. Permite verificar las credenciales del usuario y permite definir todas las acciones que el usuario puede realizar dentro de la página web. Cuenta con modelos como Users o Groups, los cuales permiten aplicar de forma genérica permisos a más de un usuario o un grupo de usuarios. También cuenta con indicadores y permisos, los cuales designan si el usuario puede realizar una tarea, visualizar cierta información, iniciar sesión, etc. Y un view tools que restringe el contenido al usuario [106].

Por último, se definen algunas ventajas de usar Django como framework para el desarrollo web:

- El modelo MVT separa las responsabilidades de cada función en diferentes capas, haciendo más simple el desarrollo y su mantenimiento.
- Django provee de un token de seguridad en los formularios que envíen datos al servidor, por lo que evita CSRF o inyecciones SQL.
- Django al estar soportado en Python el proceso de desarrollo es más sencillo y rápido de codificar.

- El framework ofrece por defecto un panel de administración, por lo que visualizar la base de datos y la administración de ella puede hacerse desde el principio.
- El ORM ofrece una forma efectiva para interactuar con la base de datos a través de los Modelos (Clases predefinidas representando cada tabla).
- Ofrece una gran escalabilidad en la aplicación, por su estructura modular.
- Ofrece buena documentación dado que es uno de los frameworks más populares para el desarrollo web.

Base de Datos

Una base de datos, es un conjunto de elementos relacionados que permiten el flujo de información de un sistema. Cada uno de los elementos es representado por un formulario de registro y cada formulario está compuesto por una serie de campos específicos. Los campos del formulario permiten recopilar información de un atributo de una entidad del mundo real. Otra manera de representar a los elementos relacionados es por medio de una tabla, con cierta cantidad de renglones y columnas [107]. Existen dos tipos de bases de datos, las bases de datos no relacionales y relacionales, a continuación se dará una breve descripción de cada una de estas:

Bases de datos no relacionales

Las bases de datos no relacionales, también conocidas como NoSQL, no necesitan especificar la estructura del elemento para poder almacenar los datos. No requiere de un identificador que verifique las diferentes relaciones que existen entre los elementos, ya que la información está organizada en documentos en formato JSON, tampoco utilizan un lenguaje estructurado para manipular los datos [108].

Bases de datos relacionales

Por otro lado, una base de datos relacional es una base de datos que cumple con el modelo relacional. Permite establecer conexiones entre los datos y gracias a estas conexiones, puede relacionar los datos de ambos elementos. Los elementos se representan por medio de tablas, las cuales cuentan con filas llamadas tuplas y columnas llamadas atributos [109].

Este tipo de bases de datos cuentan con una herramienta de diseño llamada el modelo entidad relación, este método de representación permite plasmar las actividades y modelo de negocio de la empresa, representado a través de entidades, atributos y relaciones [110]. En primera instancia, solamente contenía los conceptos de entidad, relación y atributo. Con el paso del tiempo, se fueron añadiendo otros conceptos, como los atributos compuestos y las jerarquías de generalización.

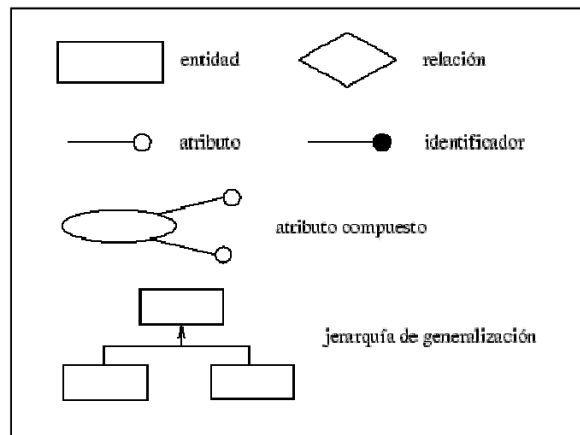


Figura 30. Conceptos del modelo entidad relación.

Para estructurar el diseño de la base de datos relacional, lo primero que se realiza es un esquema conceptual. Por lo general, se construyen varios esquemas conceptuales, representando las distintas visiones que los usuarios tienen de la información. Estas visiones suelen definir las diferentes áreas funcionales que tiene una empresa, como las ventas, producciones, recursos humanos, entre otros.

Las visiones son llamadas vistas, las cuales se pueden identificar de dos formas. La primera forma, es mediante los diagramas de flujo de datos, los cuales se realizan para poder identificar cada una de las diferentes áreas funcionales de la empresa. La segunda forma, es realizar entrevistas a los usuarios, examinar los procedimientos, los informes y formularios. Los esquemas conceptuales de cada vista, se les denomina esquemas conceptuales locales. Están compuestos por entidades, relaciones, atributos, dominios de atributos e identificadores. Cuentan con una documentación, la cual se irá realizando conforme se va desarrollando el sistema. El diseño conceptual cuenta con unas tareas específicas a desarrollar, las cuales son:

1. Identificar entidades.

2. Identificar relaciones.
3. Identificar atributos y asociarlos a entidades y relaciones.
4. Determinar dominios de los atributos.
5. Determinar identificadores.
6. Determinar jerarquías de generalización.
7. Dibujar diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario

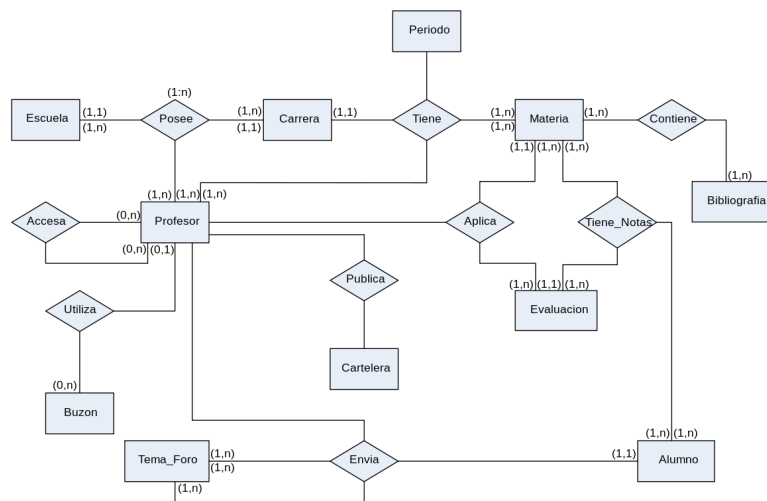


Figura 31. Ejemplo de un diagrama entidad relación.

Sistema Gestor de Base de Datos

Un sistema gestor de bases de datos (SGBD) es una aplicación que permite a los usuarios, mediante el lenguaje de definición de datos (DDL - Data Description Language), crear, definir e interactuar con una base de datos. Este lenguaje, permite especificar la estructura y el tipo de dato de cada atributo de la tabla, además de poder restringir el uso de ciertos datos de ser necesario.

También, mediante el lenguaje de manejo o manipulación de datos (DML - Data Manipulation Language), permite realizar inserciones, actualizaciones, eliminaciones y consultas de datos a las tablas. Los SGBD proporcionan un acceso controlado a las bases de datos, gracias a un sistema de seguridad, el cual restringe a los usuarios que no están autorizados para interactuar con la base de datos, mediante el lenguaje de control de datos (DCL - Data Control Language).

Por otra parte, los usuarios pueden acceder a un diccionario de datos o catálogo, el cual contiene la descripción de cada uno de los datos que conforman la base de datos. Por último y no menos importante, la interfaz del SGBD, componente que cuenta con muchas funcionalidades importantes para los usuarios, permite al usuario interactuar con el servidor mediante un lenguaje sencillo denominado SQL (Structure Query Language), lenguaje estandarizado por la ISO 1, esto quiere decir que, todas las bases de datos que soportan SQL deben de tener la misma sintaxis a la hora de aplicar el lenguaje [111].

PostgreSQL

PostgreSQL es un avanzado gestor de bases de datos relacionales de Open Source. Esto quiere decir que, el código fuente de PostgreSQL está disponible a cualquier persona, libre de cargos directos. Se caracteriza por ser un sistema estable, de alto rendimiento y flexible, puesto que puede funcionar en la mayoría de los sistemas Unix, además tiene características que extienden el funcionamiento del sistema. Puede ser integrado al ambiente Windows, permitiendo generar aplicaciones nuevas, o mantener las ya existentes. También, permite desarrollar o migrar aplicaciones desde Access, Visual Basic, Foxpro, C++, Visual C/C++, etc, permitiendo que lo usen como servidor de Base de datos.

PostgreSQL usa el modelo Cliente/Servidor, en el cual las sesiones consisten en la ejecución de dos procesos. El primero es el servidor, se encarga de manejar archivos de la base de datos, también acepta conexiones a las aplicaciones del cliente y realiza las modificaciones en la base de datos. El programa servidor se conoce como Postmaster. El segundo proceso es la aplicación cliente, la cual necesita realizar operaciones en la base de datos. Las aplicaciones cliente pueden ser aplicaciones de texto en una consola, gráficas, un servidor web que solicita el acceso a la base de datos para mostrar datos en una página web o simplemente herramientas de mantenimiento de la base de datos.

Las aplicaciones Cliente/Servidor pueden estar en diferentes equipos. En estos casos, se comunican mediante conexiones de red TCP/IP. El servidor PostgreSQL, puede gestionar múltiples conexiones concurrentes de los clientes. Para esto, inicia procesos denominados “fork”, estos se encargan de almacenar los datos de la relación [112]. Gracias a esto, el cliente y los nuevos procesos, se podrán comunicar sin la intervención del postmaster, así, el postmaster podrá estar siempre en ejecución, esperando por la conexión de los clientes [113].

APIs de Terceros

Las APIs (Interfaz de programación de aplicaciones) son un conjunto de definiciones y protocolos encargados de diseñar e integrar diferentes funciones entre aplicaciones. Comúnmente, estas son utilizadas para permitir la comunicación entre diferentes servicios o productos, ya sean internos o externos [114].

Utilizar herramientas externas garantiza ahorrar tiempo de desarrollo, ya que se pueden aplicar APIs como Google Captcha para crear formularios más seguros, OAuth para acceder y registrarse en plataformas a través de cuentas externas o herramientas enfocadas al marketplace como pasarelas de pago utilizando la API de Paypal.

3.2.4.11 Etapas del Testing

El siguiente proceso en el ciclo de vida del software es el de encontrar defectos o errores llamado “Testing” o fase de pruebas. La metodología de este proceso es iterativa, por lo que si se encuentran fallos en la tarea desarrollada esta será devuelta al involucrado para que vuelva a realizarla hasta no encontrar errores [115]. Algunas de las actividades principales que se desenvuelven en la metodología planteada son:

- **Planeación de pruebas:** En esta fase el manager de testing decide qué componentes serán probados y por qué.
- **Diseño:** Para el diseño de los tests se tendrán en cuenta los test cases obtenidos en los análisis. Además, se evaluará los tipos de pruebas se realizaran y en qué ambiente se ejecutarán.
- **Creación de test:** Los Ingenieros de Pruebas crean los scripts a ser ejecutados, teniendo en cuenta la información que recibirán y el ambiente de pruebas.
- **Ejecución de los tests:** Al ejecutar las pruebas se documentará las salidas de las pruebas y el comportamiento del sistema probado.
- **Evaluar las salidas de los tests:** Como último paso se evalúan los resultados obtenidos y en caso de haber errores se pasa el reporte a los desarrolladores.

En el caso de Django hay múltiples librerías que se pueden utilizar para realizar pruebas unitarias o de integración tanto para las vistas, como los modelos, métodos, funciones o URLs.

Pruebas de Unitarias

El objetivo de las pruebas unitarias es mantener el proyecto en constante mejora. Básicamente su principio se basa en la sostenibilidad del proyecto [116]. Esta forma de pruebas se centra en validar que la unidad más pequeña del sistema cumpla con las especificaciones definidas en la documentación [117]. En otras palabras, en el paradigma Orientado a Objetos el testing unitario válida la funcionalidad de clases y métodos que modifiquen o interactúen con instancias de esa clase. Un ejemplo podría ser la creación de una nueva Persona en el sistema, en este caso el testing unitario válida la funcionalidad del método crearPersona y que el comportamiento sea correcto.

El framework a utilizar en el desarrollo de este proyecto, aplica las pruebas unitarias mediante librerías basadas en Python. Django basa su librería de testing unitario en la librería “unittest” de Python, sobre la cual se extiende su propia librería “django.test.TestCase” heredando las funciones de la clase “TestCase” de unittest. Esta define las pruebas utilizando una metodología basada en Clases [118].

```
from django.test import TestCase
from myapp.models import Animal

class AnimalTestCase(TestCase):
    def setUp(self):
        Animal.objects.create(name="lion", sound="roar")
        Animal.objects.create(name="cat", sound="meow")

    def test_animals_can_speak(self):
        """Animals that can speak are correctly identified"""
        lion = Animal.objects.get(name="lion")
        cat = Animal.objects.get(name="cat")
        self.assertEqual(lion.speak(), 'The lion says "roar"')
        self.assertEqual(cat.speak(), 'The cat says "meow"')
```

Figura 32. Ejemplo de Testing Unitario a el modelo Animal en Django

En el ejemplo anterior se observa un ejemplo aplicado al modelo Animal en Django. Para iniciar la prueba primero se debe crear una Clase para realizarla. Posteriormente, se preparan las variables de prueba en el método **setUp**, estas variables se borran una vez finalizada la prueba. Dado que el modelo a probar es Animal en este caso se crean instancias nuevas del mismo, las cuales serán almacenadas en una base de datos temporal. Seguido a esto se prueba

el método **speak()** de cada instancia almacenada. Para esto se hace búsqueda de ellas a través de la base de datos haciendo uso del ORM y finalmente se prueba el método seleccionado.

Esta metodología de prueba es útil dado que se comprueba el comportamiento de cada Modelo y método creado dentro del sistema. Además, puesto que en esta forma de realizar pruebas se tiene total control de las entradas y salidas de datos, el desarrollador entiende de forma inmediata el comportamiento de la funcionalidad.

3.2.4.12 Despliegue

Como última etapa se encuentra la implementación, para este momento el Producto Mínimo Viable debe haber finalizado. El sistema será subido al servidor en el que estará alojado y se realizan las configuraciones para que el público pueda acceder a él, e interactuar con la base de datos. Seguido a esto, los usuarios interactúan con la aplicación web y se obtendrá una retrospectiva de qué tan bien está el producto [119]. Para esta última fase existen múltiples páginas que ofrecen servidores en los cuales se pueden alojar las aplicaciones (en caso de ser servicios web), como lo son DigitalOcean, Heroku, Amazon Web Services, Google Cloud Platform, etc.

3.2.4.13 Control de versiones

Un proceso adicional importante en los proyectos de software que utilicen la metodología ágil es la integración continua. Esta hace referencia a los incrementos frecuentes y actualizaciones realizadas por el equipo de desarrollo a partir de las nuevas necesidades o requerimientos encontrados [120]. Para que este proceso sea realizado de manera rápida y efectiva se aplican herramientas de control de versiones, las cuales permiten la creación de múltiples versiones del mismo sistema y moverse entre ellas.

Git es una herramienta para el control de versiones, cuyo objetivo es mantener una gran cantidad de código en un mismo lugar de diferentes desarrolladores. A través del sistema se pueden subir, eliminar o modificar archivos tanto local o remotamente, ya sea también interactuar con cambios de otras personas o poder juntar fragmentos de código en un solo archivo. Algunos de los principales aportes que git ofrece son:

- Conocimiento de cuándo, cómo y quién añade, modifica o elimina archivos.

- Control sobre todas las versiones realizadas del proyecto.
- Regresar sobre los cambios realizados.
- Ofrece seguridad al cifrar todas las estructuras internas.

Por otro lado, GitHub ofrece un servidor donde alojar estos proyectos a través de repositorios de Git. A diferencia de este último, Github es comercial por lo que para proyectos grandes es posible que sea necesario pagar por los repositorios.

3.2.5 Estimación de ingresos, costos y gastos

Empezando con la estimación de ingresos, costos y gastos. Para el desarrollo de esta fase, se toma como base fundamental las asesorías impartidas por el asesor financiero del Centro de Emprendimiento, el cual dividió este proceso en 6 etapas.

3.2.5.1 Analisis de Ingresos

Para poder definir el análisis de ingresos, primero se debe conocer el significado de lo que es un ingreso. Se denominan ingresos a todas las entradas económicas que tiene una empresa, así sean por parte de un producto o de un factor extraordinario. Por lo general, los ingresos principales de una empresa deben originarse mediante las ventas normales de los productos o servicios que se ofrecen y que hacen parte de su entorno comercial. A este proceso se le denomina ingresos normales de operación o ingresos operacionales, los cuales son representados por ventas a contado y a crédito que se efectuaron en un tiempo determinado, usualmente en un ciclo anual. En el caso de los ingresos extraordinarios, son denominados como ingresos no operacionales y no hacen parte del entorno comercial de la empresa, como lo pueden ser depósitos a plazo, ventas de un activo fijo, entre otros. [121]

En cuanto al análisis de ingresos, es un estudio realizado a los ingresos obtenidos gracias a la venta de productos o servicios ofrecidos por la empresa. Este estudio se caracteriza por determinar el porcentaje de ganancias en un tiempo determinado y también se encarga de evaluar un posible aumento porcentual en las ganancias de manera gradual.

3.2.5.2 Proyección de gastos

Con el fin de establecer una proyección de gastos, se debe tener presente la definición de un gasto. El gasto es denominado como un desembolso de dinero realizado por una persona natural o una empresa. Es realizado con el objetivo de administrar adecuadamente el negocio o con el propósito de ejecutar ciertas labores de venta, promoción y distribución de productos manufacturados, entre otros. [122]

La proyección de gastos, es un estudio realizado a todos los gastos presentes que tiene una empresa en un tiempo determinado. Permite determinar la capacidad de gastos que puede soportar la empresa y la cantidad de dinero a invertir para su crecimiento, además de establecer el tiempo por el cual la empresa empezaría a tener una rentabilidad. Estos gastos se dividen en gastos de personal, gastos de depreciación de bienes fijos y gastos generales.

3.2.5.2.1 Gastos de Personal

Los gastos de personal son salidas o giros de dinero realizados con el fin de administrar eficientemente la empresa. Son representados como pagos de salarios al personal de la empresa y pagos de otros conceptos, como los servicios públicos, depreciación de equipos, muebles, impuestos, asesorías contables, entre otros. [123]

3.2.5.2.2 Gastos Depreciación de Bienes Fijos

Los gastos de depreciación son denominados como un mecanismo por medio del cual se reconoce el desgaste que han sufrido los bienes fijos, gracias a los usos presentes en un tiempo determinado. Cuando un bien fijo es utilizado para generar algún tipo de ingreso, sufre un desgaste periodico durante su vida útil, este desgaste se ve reflejado en la pérdida del valor de dicho bien fijo. En base a esta depreciación, se realiza un plan de provisión mediante el cual se puede reemplazar el bien fijo sin afectar con la liquidez y el capital de trabajo de la empresa. [124]

3.2.5.2.3 Gastos Generales

Los gastos generales se identifican como todos aquellos rubros que tienen que efectuarse para dar pie al negocio en capacidad de funcionamiento. No tienen una relación directa con el monto de compras o ventas efectuadas en la empresa. Estos gastos generales son representados por servicios generales, impuestos, entre otros. [125]

3.2.5.3 Inversiones Iniciales

Las inversiones iniciales, según lo define el ingeniero en computación y máster en ingeniería industrial *Helmut Sy Corvo*, como la cantidad de dinero necesario para comenzar un negocio. Esta inversión es utilizada para cubrir todos los gastos iniciales de una empresa, ya sea la compra de bienes fijos, gastos generales, contratación de personal, entre otros. Usualmente, estas inversiones provienen de los ahorros personales del dueño de la empresa o de un préstamo bancario o incluso de una subvención por parte del gobierno. [126]

3.2.5.4 Flujo de Caja

Según *Alba Liliana Córdoba García*, en su ensayo para optar por el título de especialista en alta gerencia titulado "*EL FLUJO DE CAJA COMO HERRAMIENTA GERENCIAL PARA EL CONTROL FINANCIERO*", define el flujo de caja como el resultado de comparar un grupo de cobros y pagos que se producen en un tiempo determinado. El flujo de caja ayuda a determinar el estado de liquidez de la empresa, en otras palabras, la cantidad de dinero que la empresa espera tener en un momento determinado, teniendo en cuenta los ingresos recibidos en el momento esperado a ser recibidos y los pagos en el momento de ser efectuados. [127]

3.2.5.5 Estado de Resultado

El estado de resultados es un procedimiento financiero que se caracteriza por ofrecer, de manera ordenada y detallada, un resumen explicando las ganancias netas de la empresa, teniendo como referencia el flujo de caja, la inversión inicial, los gastos generales, entre otros. [128]

3.2.5.6 Balance

El balance describe el estado del patrimonio de la empresa en un tiempo determinado, es decir, los recursos que posee y la forma en la que estos se encuentran financiados. Desde esta perspectiva, es posible establecer que el balance muestra el uso de los recursos por parte de la empresa, además de identificar la forma en la que obtuvo dichos recursos. [129]

3.2.6 Evaluación social, ecológica y financiera

Por último, se encuentra el impacto social, ecológico y económico, los cuales se encuentran presentes en las empresas que buscan tener un fin diferente al principal, ya sea generando

acciones determinantes en una comunidad o generando prácticas ambientales que estén relacionadas con la mitigación al cambio climático. Para poder determinar la importancia de la empresa en cada uno de estos sectores, se realiza un estudio minucioso, por medio del cual se busca obtener los resultados del impacto de la empresa en estos sectores. [130]

3.2.6.1 Análisis de Rentabilidad

El análisis de rentabilidad es definido como un procedimiento que se realiza para poder observar la rentabilidad que tiene o tendrá una empresa. Para poder realizar este procedimiento, se utilizan los índices financieros de la empresa, como la tasa interna de retorno, el valor presente neto, tasa interna de oportunidad, relación beneficio/costo y la tasa verdadera de rentabilidad.

3.2.6.1.1 Tasa Interna de Retorno (TIR)

La tasa interna de retorno se encarga de evaluar el proyecto en función de una tasa única de rendimiento, con la totalidad de los rendimientos actualizados. En otras palabras, la TIR es la tasa de interés que iguala el valor presente de los ingresos con el valor presente de los egresos. [131]

3.2.6.1.2 Valor Presente Neto (VPN)

El valor presente neto toma en consideración la cantidad del dinero a través del tiempo y representa la utilidad que obtiene el inversionista después de haber recuperado la inversión realizada, obteniendo una rentabilidad exigida. Se obtiene del resultado de restar el valor presente de los ingresos y el valor presente de los egresos. Se encarga de medir los resultados obtenidos por el proyecto al valor actual del periodo en que se hace la evaluación. [132]

3.2.6.1.3 Tasa Interna de Oportunidad (TIO)

La tasa interna de oportunidad, es denominada como la tasa mínima de interés de rentabilidad que un inversionista está dispuesto a aceptar como rendimiento en sus nuevas inversiones. Es utilizada con el fin de poder establecer el valor presente neto de los flujos futuros de caja de la empresa. [133]

3.2.6.1.4 Relación Beneficio/Costo (RBC)

La relación beneficio/costo se define como la razón porcentual entre los ingresos y egresos generados por la empresa. Esta relación es el resultado de la sumatoria del valor presente de los ingresos netos, dividido por la suma de los egresos netos de la empresa. Se caracteriza por

mencionar la cantidad de dinero que gana la empresa, en base a cada peso invertido en el mismo. [134]

3.2.6.1.5 Tasa Verdadera de Rentabilidad (TVR)

La tasa verdadera de rentabilidad, es una tasa interna de rentabilidad modificada, puesto que asume que los flujos de efectivo, reinvertidos en la empresa, se hacen a la tasa de oportunidad del inversionista y no a las mismas tasas internas de retorno. [135]

3.2.6.2 Impacto Social

El impacto social, es definido como un cambio, positivo o negativo, que tiene un grupo determinado de personas como consecuencia del desarrollo de una actividad, programa, proceso o política realizada por una empresa y pueden afectar las condiciones humanas a largo plazo. Este impacto puede ser producido de manera directa o indirecta por una intervención, intencional o no y puede estar relacionado con bienes tangibles o intangibles. [136]

3.2.6.3 Impacto Ambiental

El impacto ambiental, es establecido como la alteración del medio ambiente en una zona determinada, provocado por un proyecto, proceso, programa, política o actividad, generada por una empresa en un área determinada y que afecta de manera directa o indirecta las condiciones ambientales de dicha zona. [137]

IV. RESUMEN DE METODOLOGÍA

Una vez conocidos los aspectos teóricos y técnicos del proyecto, es preciso mencionar la metodología a utilizar en las dos ramas principales del proyecto. Es decir, el componente de Emprendimiento englobando el modelo de negocio y la creación de la propuesta de empresa; y por otro lado, el elemento de Ingeniería de Software.

4.1 Propuesta de Creación de Empresa

Para el primer elemento se hará uso de la “Guía Metodológica para Argumentar la Idea Primera Aproximación de la Idea al Modelo de Empresa”, la cual fue dispuesta por el centro de emprendimiento de la Universidad, al igual que asesorías con miembros del mismo. Esto debido a que este componente no es conocido y será de utilidad para conducir esa parte del proyecto.

Esta fase se divide en la estimación del modelo de negocio, el cual se centra en conocer e identificar el producto, el mercado potencial, el entorno competitivo, así como las inversiones necesarias para poner en marcha la empresa. La cual comprende los siguientes aspectos:

- **Propuesta de Valor:** Desarrollo de la idea de emprendimiento, estimación de los productos y servicios, diferenciación del producto y experiencia del equipo.
- **Mercado Potencial:** Realizar estudios de mercado sobre el sector económico de la propuesta, los clientes y el tamaño del mercado.
- **Entorno Competitivo:** Identificar los competidores potenciales, los productos que estos ofrecen, con el fin de realizar un estudio comparativo.
- **Canales de distribución:** Establecer los medios que se van a utilizar para ofrecer los productos y servicios.
- **Estimación de inversiones:** Estimar las inversiones iniciales, que abarcan desde el alquiler del servidor y el dominio. También, se debe estimar el capital de trabajo, el cual estará centrado en la mano de obra y en la compra de equipo de computo. Estas

inversiones deben estar previstas para una duración de un año, sin embargo, el pago de capital de trabajo solo se realizará por los 3 primeros meses.

- **Impacto:** Se deben identificar los beneficios resultantes de los impactos realizados por parte de los productos y servicios propuestos en el proyecto. Además, se debe realizar una evaluación de los posibles impactos a largo plazo, que podrán generar beneficios en un sector determinado.
- **Experticia del equipo:** Conocer e identificar las capacidades del equipo de trabajo.

Y, por otro lado, la ideación de la propuesta de creación de empresa, fase en la cual se estructurará la idea inicial de la empresa formalmente. Haciendo un estudio de mercado más a profundidad, realizando el desarrollo de la marca, los aspectos organizacionales y legales, por último, se realizará una proyección de ingresos, costos y gastos, la cual ayudará a estimar la viabilidad del proyecto, a partir del análisis financiero, para culminar con la evaluación social y ambiental.

- **Estudio de mercado:** Con el fin de entender las necesidades de los usuarios y evaluar la viabilidad del proyecto, se realizará un estudio de oferta y demanda. Se empleará una encuesta cualitativa, mediante entrevistas abiertas, para obtener opiniones de los usuarios sobre el modelo de negocio tradicional, frente al nuevo modelo de negocio propuesto en el presente proyecto, además de identificar fallas en las plataformas actuales con el fin de proponer soluciones. También, se aplicará una encuesta cualitativa, usando una metodología de encuesta cerrada, para conocer la cantidad de usuarios interesados en el prototipo web y los productos o servicios propuestos. Los resultados serán presentados en gráficas y porcentajes para facilitar su análisis.
- **Componentes Organizacionales:** Con el fin de establecer la estructura de una empresa, es importante definir su misión y visión, los cuales representarán su proyección. Asimismo, se debe realizar un organigrama para representar la jerarquía de los cargos. El manual de funciones será útil para definir las tareas y responsabilidades de cada cargo. Los objetivos estratégicos se utilizarán para proyectar una meta a futuro y los valores corporativos servirán para enfocar el rumbo de la empresa.

- **Elementos Legales:** Se deben contemplar los aspectos legales de la empresa, como lo son el Registro en Cámara de Comercio, la Homonimia, los términos y condiciones y las políticas de privacidad.
- **Proyección de ingresos costos y gastos:** Para la proyección de ingresos costos y gastos se seguirá el proceso planteado por el asesor financiero del centro de emprendimiento. Este iniciará con un análisis de ingresos, definiendo las posibles ventas y ganancias que recibirá la plataforma. Posteriormente, se proyectarán los gastos que deberá solventar la empresa para su sostenimiento, así como la inversión inicial necesaria para desarrollar el prototipo inicial. Después, se proyectará el flujo de caja de la empresa y los estados financieros. El balance será el último elemento que se calculará, el cual será utilizado para realizar el análisis financiero que determinará la viabilidad del proyecto.
- **Análisis financiero:** La evaluación financiera evaluará índices como la Tasa Interna de Retorno, el Valor Presente Neto, la Tasa Interna de Oportunidad, la Relación Costo/Beneficio y la Tasa Verdadera de Rentabilidad. Para así determinar cómo se comportará la empresa proyectado a 5 años y conocer si el proyecto será sustentable y rentable en el tiempo
- **Impacto social y ambiental:** Visualizar los beneficios sociales y ambientales que serán generados, de manera directa o indirecta por parte de la empresa.

4.2 Prototipo de Software

Por otro lado, la fase de Ingeniería de Software utiliza la metodología de Scrum y Kanban para llevar registro de todas las actividades referentes al ciclo de vida del software. Además, se utilizan técnicas y metodologías presentadas a continuación:

- **Definición de los requerimientos:** La metodología ágil Scrum, se utilizará para definir los requerimientos del proyecto, repartiendo cada etapa en múltiples Sprints. Las Épicas se utilizarán para identificar los usuarios. Estas Épicas serán divididas en historias de usuario, detallando las funcionalidades necesarias de la aplicación.

Para priorizar el desarrollo de estas historias de usuario, se usará el planning poker, asignando un valor a cada historia para organizarlas por prioridad. Cada Sprint estará compuesto por estas historias y sus elementos, ayudando a definir el progreso del desarrollo. Una vez organizadas las historias de usuario en Sprints, se utilizará una metodología para organizar el entorno de trabajo, denominado Kanban. La cual ayudará a identificar la fase en la que se encuentra cada historia de usuario, ya sea en diseño, desarrollo, testing o finalizado.

- **Diseño de diagramas y mockups:** Para el desarrollo de los requerimientos del proyecto, es necesario crear diagramas que permitan comprender su estructura y los elementos que lo conforman, así como sus valores. Para esto, se utilizarán como base, los modelos de UML, puesto que diseñan y proporcionan planos muy fáciles de interpretar. Estos diagramas serán realizados conforme se establezca la metodología de diseño orientada a objetos.

En base a esto, primero se deben realizar los diseños generales de la aplicación, para comprender la estructura de la misma, vista de manera macroscópica. De esta forma, el primer diseño que se va a realizar es el diagrama de casos de uso, el cual se encargará de identificar a los actores que harán uso de las funcionalidades de la aplicación. Una vez reconocidas las relaciones entre los usuarios y las funcionalidades, se prosigue al desarrollo del diagrama de secuencias, encargado de presentar el orden general en el que interactúan las clases de la aplicación. Después se encuentra el diagrama de clases, el cual será diseñado en base a las necesidades del sistema, estará conformado por Clases, contando con atributos y relaciones, las cuales ayudarán a comprender su enfoque en la aplicación.

Para el almacenamiento de información, se diseñará un diagrama de bases de datos relacional basado en la persistencia de datos, usando como referencias las clases definidas anteriormente y el tipo de atributos que conforman estas clases. Una vez realizados los diagramas anteriores, se maquetará la arquitectura del sistema, la cual será en base al patrón de arquitectura MVT, puesto que permite la modularidad de las diferentes funcionalidades y fácil acceso a la base de datos a través del ORM. Esta

arquitectura contará con un diseño, el cual reflejará la interacción entre el cliente, el servidor y las APIs externas que ayudarán al funcionamiento de la aplicación.

Finalmente, se diseñará un diagrama de despliegue, el cual mostrará las relaciones físicas presentes entre el cliente, el servidor y los diferentes servicios que ayudan al funcionamiento de la aplicación. Además, se realizarán los diseños mockups de las historias de usuario a través de Figma. Para el desarrollo de estos diseños, se utilizarán metodologías de diseño, las cuales ayudarán a establecer la manera en que se representará la información dentro de la aplicación, por medio de estructuras como formularios, barras de navegación, entradas de datos, entre otros. Estos diseños se basarán a la estructuración de la marca que será realizada en la fase de estudio de mercado, obteniendo la esencia de la empresa.

- **Desarrollo del aplicativo web:** Inicialmente se deberá preparar el entorno de desarrollo creando un entorno virtual de Python con la librería venv. Posteriormente se instalará Django y se dará inicio al proyecto. Además, se deberá crear un repositorio para llevar registro y orden del control de versiones. Y un archivo de requerimientos para administrar las versiones de las librerías.

La aplicación será desarrollada con el lenguaje de programación Python (v3.9) haciendo uso del framework Django en su versión 4.1.7, bajo el patrón de arquitectura MVT el cual facilitará el desarrollo y el soporte del aplicativo, puesto que cada función principal estará dividida en aplicaciones representadas en paquetes y estos últimos dividen cada aplicación en Vistas para manejar las peticiones, Modelos para acceder a los datos y plantillas para renderizar la información. El proceso de desarrollo estará dividido en la estructuración de los Modelos, desarrollo de las vistas y maquetación de las plantillas.

Los modelos están almacenados en cada aplicación respectiva y una vez se migren estos modelos, serán almacenados en la base de datos relacional, usando el gestor de base de datos PostgreSQL y el ORM. Este último será de utilidad para realizar peticiones a partir de instancias de objetos de los modelos. Una vez los modelos sean creados dependiendo de cada Historia de Usuario, se dará paso a codificar las vistas, las cuales recibirán las peticiones de los usuarios.

Posteriormente, cuando ya las vistas permiten renderizar información, se podrán desarrollar las plantillas, las cuales serán desarrolladas utilizando herramientas como HTML5 y CSS3. Para agilizar el diseño visual de la aplicación web, se utilizará Bootstrap 5.3, dado que cuenta con clases que facilitan el desarrollo "Responsive" para adaptar la interfaz gráfica a diferentes dispositivos y clases para manejar los componentes rápidamente. Y PurgeCSS, que ayudará a reducir el código CSS innecesario. Así mismo, se hará uso de JavaScript y JQuery, herramientas útiles para agregar animaciones y eventos al Frontend.

Finalmente, se usarán librerías externas para agilizar el desarrollo y aplicar funcionalidades avanzadas, tales como, allauth para facilitar el acceso a través de cuentas externas, reCaptcha para brindar seguridad a los formularios, Pillow para manipular imágenes y por último, django-compressor-purge-css-filter para facilitar la compresión y limpieza de los archivos CSS.

- **Elaboración y ejecución de pruebas:** El proceso de pruebas se llevará a cabo mediante una metodología iterativa. Si se detectan errores en una tarea, se devolverá al responsable para que la realice de nuevo hasta que se resuelvan todos los problemas. El proceso se dividirá en 5 etapas:
 - **Planificación de pruebas:** Se establecerá un plan de pruebas para cada uno de los requerimientos a testear. Esta prueba contará con unos elementos, los cuales ayudarán a identificar los parámetros para que la prueba sea exitosa.
 - **Diseño:** Se diseñará una estructura de los planes de pruebas, con la finalidad de poder establecer el tipo de prueba a realizar y el ambiente de su ejecución.
 - **Creación de los test:** Se crearán los respectivos Scripts que serán aplicados en cada una de las pruebas, también se ajustarán los ambientes en donde se aplicarán dichas pruebas.

- **Ejecución de pruebas:** Se analizará el proceso realizado por el test a la hora de ser aplicado, con el fin de poder ver su comportamiento durante la aplicación.
- **Evaluación de resultados:** Se evaluarán los resultados obtenidos, con la finalidad de poder encontrar errores en estos.

Para mantener el proyecto en constante mejora, se utilizarán pruebas unitarias. Django tiene su propia librería de pruebas llamada "unittest", que se extiende con la librería "django.test.testcase" para heredar las funciones de la clase "testcase" de "unittest".

- **Despliegue del prototipo:** Para poder empezar el proceso de despliegue, primero se realizará la preparación del proyecto mediante los siguientes pasos:
 - **Purge CSS:** Se ejecutará una librería llamada compressor, la cual se encargará de adjuntar todos los estilos, tanto de css como de javascript, y los entregará en un sólo archivo. Esto se debe realizar con el fin de poder limpiar el proyecto de los estilos que no son usados, liberando la mayor cantidad de espacio posible.
 - **Configuración de la base de datos y las variables de entorno:** Se creará la base de datos en Heroku y se extraerán las variables de entorno para poder hacer la conexión entre la base de datos y el proyecto.
 - **Configuración del CDN:** Se creará el CDN en la nube de Google Storage, con el fin de poder almacenar los archivos estáticos del proyecto.

Una vez se haya preparado el proyecto, lo primero que se debe realizar para el proceso de despliegue, es crear un droplets de Digitalocean. Este droplets se alojará en un datacenter en Estados Unidos, en New York y contará con Ubuntu Server versión 22.04, un tamaño de 512MB, 10 GB SSD Disk y 500GB de transferencia. Todo esto tendrá un costo de US \$4 mensuales. Esto será cubierto por los créditos estudiantiles regalados por DigitalOcean.

Creado ya el droplets, se creará el entorno virtual dentro del servidor y se realizará el enlace remoto al repositorio en Github. Luego, se deberá hacer el respectivo pull a la rama, con el fin de descargar el proyecto. Con el proyecto descargado, se inicializará el entorno virtual y se instalarán todos los paquetes necesarios para su ejecución. Después, se realizarán las respectivas configuraciones del Gunicorn y el Nginx, con el fin de establecer el alojamiento del servidor, cifrar conexiones entre el cliente y el servidor, cachear el contenido en la memoria del servidor y configurar el sitio SSL.

V. MODELO DE NEGOCIO

Una vez entablados los aspectos teóricos y el rumbo de trabajo del proyecto, se inicia el proceso de reconocimiento y construcción del modelo de negocio, en base a la metodología del centro de emprendimiento de la universidad. Para después ejecutar el desarrollo de la empresa y el prototipo en base al modelo encontrado.

5.1 Propuesta de valor

5.1.1 Desarrollo de la idea

Needer será una empresa de desarrollo de software que ofrecerá una fusión de una **Red Social** con un **Marketplace de Contenido Multimedia**, dirigido para todo público a través de una aplicación web progresiva (móvil a futuro), la cual se espera pueda servir de intermediario entre consumidores y creadores de contenido.

En esta plataforma, los usuarios podrán utilizar las funcionalidades de la **Red Social**, permitiéndoles realizar publicaciones (Fotos, vídeos o audios), intercambiar mensajes entre los distintos usuarios que utilizan la plataforma, seguir a otros usuarios, ver sus publicaciones, etc. También, los usuarios podrán utilizar las funcionalidades del **Marketplace de Contenido Multimedia**, teniendo acceso a los distintos **Productos Multimedia** (Fotos, vídeos y audios) genéricos o personalizados, ofrecidos por unidad, que los usuarios podrán subir una vez estén verificados en la plataforma como creadores de contenido.

Con el modelo de negocio de venta unitaria, se busca dar una solución a quienes no deseen estar suscritos a una aplicación, siendo más rentable para los clientes. Permitiendo que compren lo que deseen, sin la necesidad de estar atado a la plataforma. Además, se da la libertad de uso de la plataforma sin la obligación de registrar un medio de pago.

Needer espera tener dos tipos de usuarios, los creadores de contenido y los consumidores. La plataforma garantiza la libertad en la creación del material (siempre y cuando no infrinjan las normas de derechos de autor). Esto quiere decir que contará con múltiples categorías según los gustos: Deportes, Streamers, Gameplays, Modelos, Artistas, Danza, entre otros.

En un futuro se espera que **Needer** pueda escalar al punto de ofrecer chat en vivo entre el consumidor y los creadores de contenido, con el fin de acercar a los mismos con su público,

además de garantizar la posibilidad de concretar ventas a través de un sistema de criptomonedas.

5.1.2 Productos o Servicios

Los servicios que se ofrecerán en la plataforma varían con respecto a la necesidad del usuario y la funcionalidad principal que se esté utilizando, ya sea la red social o el marketplace.

Los dos principales actores que interactúan en la aplicación son: Creador de contenido y Consumidor; según la necesidad, se brindará una solución diferente.

5.1.2.1 Creador de Contenido

1. **Needer** en su fase inicial espera entregar los elementos de red social en un prototipo, el cual se estima sea el gancho atractor y genere una base de usuarios. Por lo tanto, el creador de contenido podrá crear publicaciones ya sea a partir de texto, vídeos, audios o fotos. También, podrá seguir otros perfiles de cualquier tipo de usuario, visualizando sus publicaciones desde el feed. Por último, estos podrán interactuar con las publicaciones ya sea a través de “me gusta” o comentarios.
2. Cualquier usuario podrá aspirar a ser **Creador de Contenido**. Para esto, el usuario deberá verificar su identidad y además tendrá que vincular una cuenta bancaria a **Needer**, permitiéndole publicar y vender sus productos.
3. La herramienta de venta de contenido multimedia (Fotos, Videos, Audios, Arte digital, etc) donde el Creador de Contenido ofrece sus productos **genéricos** por unidad en el marketplace por un precio fijo. Este puede personalizar la publicación del contenido que desea vender, permitiéndole categorizar y además limitar el número de compras en la publicación. Los creadores de contenido podrán ver las estadísticas de ventas e ingresos generados en la plataforma, con el fin de llevar un control de sus ingresos. Como punto adicional, la plataforma garantizará la seguridad de cada transacción realizada entre el Creador de Contenido y el Consumidor.
4. El Creador de Contenido podrá publicar a la venta bajo el precio que desee, productos multimedia personalizados por unidad dentro del marketplace. Una vez se realice el

pago del producto, se abrirá el canal de comunicación, donde se podrán definir las solicitudes del contenido personalizado por parte del consumidor al Creador de Contenido.

5. En un futuro se brindará la posibilidad de realizar streams en vivo, donde los creadores de contenido podrán interactuar instantáneamente con su audiencia. Permitiendo recibir donaciones para incentivar su stream.

5.1.2.2 Usuario Consumidor

1. **Needer**, durante su fase inicial, entregará elementos de una red social en su prototipo, los cuales se encargarán de atraer la atención de los usuarios, con el fin de crear un banco de usuarios para la plataforma. Los usuarios podrán realizar publicaciones y editarlas, así como interactuar con múltiples perfiles y reaccionar o comentar a las posts que quieran. En la red social, todos podrán relacionarse como deseen y con quien deseen, manteniendo el respeto y un ambiente sano.
2. El marketplace estará disponible para todos los usuarios que deseen **consumir** por unidad el Contenido Multimedia. En este apartado, el Contenido Multimedia estará organizado por diferentes secciones, ya sea por tipo de contenido (Video, Imagen, Audio, Arte digital, etc), autor (Deportista, Artista, Cantante, Influencer, etc) o categoría (Contenido natural, Pop, Gótico, Futurista, etc). Adicionalmente, podrán ingresar al perfil de cada usuario que esté verificado **Creador de contenido** y tendrán la capacidad de seleccionar y comprar lo que quieran del marketplace.
3. Los usuarios podrán comprar contenido multimedia personalizado, el cual contará con un precio fijo y una descripción que definirá cómo personalizar el producto. Una vez realicen el pago de este contenido, podrán crear solicitudes, al dueño de la publicación, ordenando el tipo de modificaciones que quieren realizar al producto.
4. En el futuro dentro de **Needer** los usuarios consumidores podrán entrar a los perfiles de los **Creadores de Contenido** y si están realizando streams en vivo tendrán la opción acceder a este, donde verán y escucharán a su personaje favorito. Además, si

deseen, se posibilita la opción de realizar donaciones para que estos los noten mediante un mensaje escuchado por toda la audiencia.

5.1.3 Diferenciación del producto

Las principales características de Needer las cuales le permiten diferenciarse del mercado son las siguientes:

- **Needer** acoge un modelo de negocio basado en transacciones por unidad a través del marketplace, por el cual los usuarios podrán comprar lo que deseen sin necesidad de pagar suscripción mensual para consumir el contenido. Por esto, **Needer** será más accesible, debido a que los creadores de contenido ofrecerán productos multimedia, los cuales contarán con un rango de precio razonable para el público.
- **Needer** integra elementos de una red social y por lo tanto permite que se creen comunidades en torno a los creadores de contenido, donde su público pueda comentar y dar “me gusta” a las publicaciones.
- **Needer** recompensará a los usuarios creadores de contenido con insignias por el monto acumulado de ventas realizadas en la plataforma.
- **Needer** manejará, posteriormente, un sistema de contenido personalizado a través del marketplace. El Creador de contenido publicará un producto multimedia personalizable a la venta por unidad. Este al ser comprado por un consumidor, permitirá la apertura del canal de solicitudes, por medio del cual se podrán exigir las modificaciones que el consumidor requiere para dicho producto.
- **Needer** permitirá en un futuro el chat en vivo donde el creador de contenido interactúa con sus seguidores y estos a su vez podrán hacer solicitudes en vivo a cambio de donaciones en el stream.

5.1.4 Experiencia del equipo

El equipo cuenta con **amplia experiencia en desarrollo web** como por ejemplo en sistemas de reservación, sistemas de economía colaborativa y registros médicos veterinarios. Sin

embargo, en el sector de entretenimiento no se cuenta con experiencia previa, frente a esto, se decide asesorarse con Managers y conocedores del mundo del entretenimiento.

5.2 Mercado Potencial

5.2.1 Sector económico de la propuesta

El proyecto hará parte del sector **terciario** de la economía, más concretamente al ocio, esto debido a que **Needer** será una plataforma de contenido multimedia usada como entretenimiento. Por tal razón se realiza una investigación del sector mencionado:

5.2.2 Investigación del sector

Según datos extraídos del ministerio de las TIC y brindado por *Diario la República* en Colombia hay 8,2 accesos a internet mediante una red, de estos los departamentos los que más cuentan con internet son **Bogotá** con 2,15 millones, **Antioquia** cuenta con 1,45 millones y **Valle del Cauca** con 883.939 accesos. Por último, los accesos a internet móvil son de 33,8 millones donde más del 77% tienen acceso a internet 4G [138].

Adicionalmente, según datos de *Global Web Index*, Colombia es el cuarto país que por habitante gasta más tiempo en internet, el promedio es de 9,01 horas siendo facebook la página que más visitan [139].

Ahora bien, hablando más concretamente del consumo de redes sociales y contenido multimedia a través de internet, en latinoamérica se cuentan con más de **4.300** millones de accesos de los cuales el **90%** lo hace a través de internet [140].

De estos países México es el que cuenta con más registros por usuario donde el **92,3%** consume redes sociales, siendo este de los que más dedican tiempo a ver contenido multimedia y redes sociales, seguido por Argentina con un **91,1%** , Colombia, Perú y Chile. Las páginas que más suelen visitar son Facebook, Instagram, Twitter y LinkedIn respectivamente [141] .

Ahora hablando del panorama internacional más concretamente en Estados Unidos, según un estudio de mercado realizado por Business Insider cerca de 245 millones de personas en este

país consumen contenido en línea y de estos el 54% prefieren consumir contenido de marcas o influencers a los que siguen [142].



Figura 33. Uso de internet en EEUU.

5.2.3 Clientes o beneficiarios

Como se mencionó anteriormente, en Needer todos los usuarios podrán acceder a las funcionalidades de la plataforma según su rol, por esta razón se definen a los actores según la necesidad: Quienes producen el contenido multimedia para la venta serán denominados como **Creadores de contenido** y quienes lo adquieren, los **Consumidores**.

- **Creadores de contenido:** Los usuarios **creadores de contenido** se caracterizan por ser grupos de personas o usuarios **mayores de 18 años**. Se asocian a la plataforma debido a que **desean realizar o realizan actividades que son atractivas para los consumidores**. Estos tienen la cualidad de que **pueden estar ejerciendo o no, cualquier profesión**. Podrían ser músicos, modelos, actores o actrices, deportistas, etc, con la intención de **generar o mantener una base de fanáticos a través de una red social y generar ingresos adicionales mediante la venta de contenido digital**.
- **Consumidores:** Los usuarios consumidores se caracterizan por ser usuarios **mayores de 18 años**, los cuales sienten la necesidad de seguir, interactuar o apoyar monetariamente a un creador de contenido, con el fin de ser notado por este o de adquirir sus productos exclusivos. También, pueden simplemente estar buscando hacer parte de una comunidad a través de una red social, ya sea generando publicaciones o interactuando con ellas.

Tomando como base los datos aportados por la **plataforma de transmisiones y chat en vivo Twitch**, dos tercios de los usuarios son hombres (**65% hombres y 35%**

mujeres) y el rango más predominante de edades en la plataforma son los **centennials (1994 - 2004) (41%)** y **millennials (1981 - 1993) (32%)** [143]. Este grupo de personas puede ser de cualquier estrato socioeconómico y suele contar con internet en casa y con un equipo para acceder a esta. Este dato es de suma importancia debido a que el tipo de usuario de Twitch se espera sea el mismo que recurra en **Needer**.

5.2.4 Tamaño del mercado

Como se describió anteriormente, el consumo de contenido multimedia ha crecido en los últimos 10 años, y aún más gracias a la pandemia este ha aumentado un **120%** a comparación de años anteriores. **Según un estudio de la Unión Internacional de Telecomunicaciones (UIT) realizado en 2021**, actualmente **Colombia tiene un 64%** de población que dispone de internet e interactúa frecuentemente con redes sociales, y por medio de estas consumen el contenido multimedia de sus creadores de contenido favoritos [144].

De acuerdo a los datos analizados de Colombia, se podría deducir basado en el porcentaje aportado que, los usuarios que usan estas redes sociales de entretenimiento y ocio en **Neiva** son un aproximado de **228730 personas**.

Por otro lado, según una encuesta realizada en el 2019 por la **Administración Municipal de Neiva**. El 54% de los encuestadas en el municipio invierte entre 3 a 6 horas navegando por internet al día y el 19% gasta más de 6 horas al día en internet, siendo Google la página a la cual acceden más, seguido por Facebook e Instagram con un 28% del tiempo [145].

Finalmente, para ampliar el panorama del mercado, se han tomado cifras a partir de páginas web que prestan un servicio similar a Needer:

- **Patreon** es una de las páginas de venta de contenido digital más grandes a nivel mundial, esta tiene cerca de 200.000 creadores generando entre ellos más de 100 millones de dólares mensualmente [146]. **Fiverr** es otra plataforma que vende contenido digital profesional en el mercado, bajo pedido y personalizable desde su marketplace. Aquí se pueden encontrar desde traducciones, producción musical, videos, fotos, etc. Esta empresa cuenta con 3.5 millones de compradores activos generando un *revenue* anual de **189 millones** de dólares [147].

- Por otro lado, otro competidor que abarca parte del mercado objetivo, pero en un segmento similar, es la página de consumo multimedia por suscripción y a su vez red social **OnlyFans**. Según datos ofrecidos por la propia empresa, los cuales fueron recopilados por el diario *La República* en el 2017. La página contó en ese año con **1,7 millones** de creadores de contenido en el mundo, los cuales generaron en total **US \$3,800 millones**, esto siendo generado por **87 millones** de suscriptores. Tan solo en **Colombia** se contaba con **350 creadores** de contenido y **4 millones** de usuarios inscritos que pagaban activamente, siendo casi un **10%** de la población Colombiana [148].



Figura 34. Consumo de OnlyFans

Según lo anterior se podría hacer un aproximado del consumo local, donde teniendo en cuenta la cifra en Colombia, a nivel Regional en **Neiva** de las **228730** personas el **10%** consume un producto similar que se encuentra en el segmento de **Needer**, representando una cantidad de **22.873** personas, las cuales a nivel local estarían suscritas a plataformas como Only Fans o similares.

5.3 Definición del entorno competitivo

5.3.1 Competidores

Needer tiene competidores directos e indirectos. El primero de estos es el ya mencionado **OnlyFans** [149], uno de los portales más famosos del momento, al cual se han suscrito cerca de **1 millón** de creadores de contenido. **OnlyFans** tiene componentes de marketplace y red social, esto permite a los usuarios suscritos en perfiles de pago o gratuitos interactuar, de cierta forma, con los creadores de contenido vinculados con la plataforma. Facilitando el comercio

dentro de la aplicación, generando así **2 billones** en ventas, siendo infinitamente superior que en el año 2017 y estando a casi el doble de su competidor directo **Patreon**.

Este último, es un sitio web donde los usuarios se suscriben a sus creadores de contenido por una cuota monetaria mensual a cambio de contenido exclusivo. Aunque algunos usuarios usan la plataforma para vender contenido personalizado a sus seguidores.

5.3.2 Alianzas claves

Una de las alianzas y necesidades claves para **Needer** será la búsqueda de creadores de contenido para atraer una base de usuarios a su red social inicial, y posteriormente producir ventas desde el marketplace. Una de las propuestas a ofrecer para los creadores de contenido será el beneficio del 100% de las ventas que ellos realicen por cierto tiempo. Con la finalidad de promover el proyecto cuando se haya lanzado por completo. Posteriormente se plantea hacer alianzas estratégicas con sitios web, managers y agencias que tengan conexiones con creadores de contenido para obtener credibilidad y visibilidad.

5.3.3 Precio de venta del producto o servicio

En **Needer** los creadores de contenido pondrán el precio a productos. Estos se verán reflejados en el marketplace y **Needer** cobrará una pequeña comisión por cada transacción que se realice, a parte de la comisión que se llevará la plataforma de pagos, con la finalidad de mejorar la infraestructura del proyecto y poder escalar a futuro. Por otro lado, la red social será de libre acceso, por lo que no se cobrará por el uso de ella, ni por el acceso a los perfiles de creadores de contenido. Debido a ello, no se estiman precios ni ganancias de esta funcionalidad.

Como se mencionó anteriormente, **Needer** pretende obtener una comisión del **13%** por compra en el marketplace. No se pretende invertir en publicidad por el momento, debido a los recursos económicos que se disponen. Para contrarrestar esto, se propondrá a los creadores de contenido que se registren en etapas tempranas, darles el 100% de las ventas que realicen. Con el fin de que ellos mismos den publicidad al proyecto.

Se realizó un estudio de mercado para estimar el precio del contenido multimedia usando como referencias distintas páginas como **ebay**, **ismygirl**, **fejira** donde se estudiaron los precios presentes para encontrar un promedio de precios de venta del contenido multimedia.

Contenido	Precio promedio
Imagen	\$27.000,00
Video	\$45.000,00
Audio	\$33.500,00
Álbum	\$130.000,00

Tabla 1. Rango de precios promedio de contenido multimedia.

5.4 Planteamiento de los canales de distribución y cobertura geográfica

5.4.1 Canal de distribución

Needer es un servicio web que funciona a través de internet. El cliente podrá hacer uso de la página a través de un computador, tablet o smartphone. Mediante estos tendrá la posibilidad de acceder al marketplace y comprar de manera segura el contenido que desee, el cual podrá visualizar directamente a través de la plataforma. También, tendrá acceso a las funcionalidades de la red social, cómo crear publicaciones, visualizar las publicaciones que han realizado otros usuarios, interactuar con ellas, seguir a otros usuarios, entre otros.

5.4.2 Cobertura geográfica

Needer buscará acercarse inicialmente a los consumidores de la ciudad de Neiva, para luego abrirse a toda la región Colombiana. Posterior a esto, se espera apuntar a las personas hispano hablantes del mundo, donde puedan disfrutar de la red social y el marketplace en español. Más adelante en mayor escala, debido al alto consumo de contenido multimedia en el globo, **Needer** puede llegar a ser usado alrededor del mundo esperando tener una amplia aceptación, incluso en zonas angloparlantes.

5.4.3 Sistema de fidelización

Al ser **Needer** una plataforma gratuita y de libre acceso, los clientes podrán ingresar sin restricción alguna y en cualquier momento, a hacer uso de las funcionalidades del marketplace y la red social. Podrán adquirir los productos que deseen, generando un estilo de vida alrededor del consumo del material, permitiendo la fidelización de los clientes. También,

podrán crear publicaciones, las cuales serán visualizadas desde sus perfiles y contarán con interacciones por parte de otros usuarios (Comentarios o Likes). Sumado a esto, se busca que los creadores de contenido disfruten y se sientan cómodos produciendo o vendiendo su material.

Por último, se definió el uso de sistemas de ranking según la cantidad de ventas que cada creador de contenido haya realizado, para que los que más hayan vendido tengan insignias reconociendo su labor.

5.5 Estimación de inversiones, ingresos y costos

5.5.1 Tipo de inversiones

Inicialmente para entregar el prototipo solo se requiere una inversión monetaria. Para adquirir la anualidad del servidor que alojara el servicio, el almacenamiento y el dominio. También, para pagar el capital de trabajo durante 3 meses, e invertir en los equipos de cómputo a utilizar para el desarrollo.

A futuro, se espera que inversionistas apuesten por la propuesta que **Needer** ofrece. Para que de esta forma se pueda escalar a nivel mundial. Sin embargo, para ello se requeriría mejorar la infraestructura del servidor, más desarrolladores, y moderadores que revisen la información que se suba a la web, y publicidad que le brinde más alcance al portal.

5.5.2 Magnitud de las inversiones

Inicialmente se estima que se deba adquirir un servidor lo suficientemente potente para llevar al menos **23000** usuarios mensuales, esto dado que se proyectan las **inversiones diferidas** del proyecto como si se tuviera la afluencia de personas estimada para el 5 año. Para esto se requiere un **monto de COP \$19.550.000** para el soporte del primer año, teniendo en cuenta que, el pago de los servidores, dominio, almacenamiento y demás, se realiza de manera anual.

Por otro lado, el sueldo se estima únicamente para los 3 primeros meses, estimado para los miembros iniciales del equipo. Una vez el prototipo inicial sea realizado y se empiece a atraer usuarios, los salarios se sustituyen por objetivos y ganancias del proyecto. Por otro lado, el capital de trabajo estará compuesto también por los recibos de energía e internet desde donde los miembros ejercerán sus funciones (debido a que la empresa no contará con instalaciones

físicas y se trabajará remotamente). Para esto se requerirá una inversión de **COP \$18.600.000** destinada a **capital de trabajo** proyectada a 3 meses.

Finalmente se tienen las **inversiones fijas**, las cuales pagarán los equipos de cómputo que se usarán para desarrollar el proyecto, estos deberán ser lo suficientemente potentes como para poder realizar renders y programar en estos. Se tomó como referencia el portátil **ASUS TUF Dash F15**, debido a que cuenta con las características necesarias para solventar las necesidades. Teniendo en cuenta que serán 2 desarrolladores inicialmente, para culminar el producto mínimo viable, la inversión fija inicial será aproximadamente de **COP \$10.000.000**, esta podrá variar según las necesidades de licencias o más equipos de computo.

De esta forma la magnitud de inversión inicial necesaria para poder entregar un producto usable y escalable a los usuarios, es de **\$48.150.000 pesos colombianos**.

5.5.2.1 Inversiones diferidas

Gastos	Inversión Inicial (COP) Anual
Dominio: Hostinger needer.com.co	\$30.000,00
Hosting: Digital Ocean Virtual Machine 2CPUs	\$2.880.000,00
GitHub: Profesional Repo	\$440.000,00
Base de datos: Digital Ocean Managed Database PostgreSQL	\$3.600.000,00
CDN: Beluga CDN 25T b	\$12.600.000,00
TOTAL	\$19.550.000,00

Tabla 2. Inversiones diferidas

5.5.2.2 Capital de Trabajo

INVERSIONES CAPITAL DE TRABAJO	
Gasto	3 Meses (COP)
Sueldos	\$18.000.000,00
Recibos de Energía	\$180.000,00
Recibos de Internet	\$420.000,00
TOTAL	\$18.600.000,00

Tabla 3. Capital de trabajo

5.5.2.3 Inversión Fija

INVERSIÓN FIJA	
Descripción	Inversión Inicial (COP)
Asus Tuf Dash F15 - i7 11370H 8Gb Ram5 12Gb SSD RTX 3050ti	\$10.000.000,00

Tabla 4. Inversión Fija

5.5.2.4 Total Inversión Inicial Estimada (Plan Inicial)

INVERSIÓN INICIAL	
Inversión Diferida Primer año	\$19.550.000,00
Inversión Fija	\$10.000.000,00
Capital de Trabajo 3 Primeros meses	\$18.600.000,00
Total	\$48.150.000,00

Tabla 5. Total inversión inicial

5.5.3 Financiamiento de las inversiones

Teniendo como base la cantidad de dinero que se necesita para financiar la propuesta, se espera que el 10% de esta sea financiada por recursos propios. El resto se estima pueda ser

cubierta por un crédito o por el Fondo Emprender, accediendo a esta posibilidad mediante una convocatoria.

5.5.4 Ingresos estimados iniciales

Needer planea iniciar con una base de usuarios, representando un **5%** de los **22.783** usuarios del mercado local (Neiva) del sector de entretenimiento siendo estas **1143** personas, atraídas a partir del prototipo inicial, es decir la red social. Gracias a esto, para el primer año ya se contará con un número de usuarios base. Y para los siguientes 2 años se espera aumentar los usuarios a un promedio de **150%** de usuarios con respecto a cada año anterior. Por último, en los últimos 2 se estima que el incremento de usuarios sea un poco más conservador, siendo este un **80%** en base a los años anteriores. De esta manera se planea que, para el 4to año la empresa obtenga un ingreso bruto anual de **COP 71 millones de pesos** aproximadamente. De tal forma, se estima que el proyecto será rentable y estará en equilibrio a partir de ese periodo.

Se calcula que por el fuerte crecimiento de la industria del contenido de entretenimiento **Needer** tendrá gran acogida y su crecimiento será exponencial. Por otro lado, la plataforma obtendrá por cada transacción realizada en la página un **13%** de comisión, con respecto al precio del producto vendido, y aparte se cobrará una comisión destinada a la pasarela de pagos que se utilice captando un **2%** restante, para un cobro total del **15%** de comisión por transacción.

# Usuarios	Año	Tasa de Inflación	Imagen	Video	Audio	Álbum	Total Ventas	Ingresos Brutos
1143	1	0,00%	\$13.887.450,00	\$18.002.250,00	\$5.743.575,00	\$7.429.500,00	\$45.062.775,00	\$5.858.161,00
2858	2	9,00%	\$37.849.923,00	\$49.064.715,00	\$15.653.981,00	\$20.248.930,00	\$122.817.549,00	\$15.966.281,00
7145	3	9,00%	\$94.624.808,00	\$122.661.788,00	\$39.134.951,00	\$50.622.325,00	\$307.043.872,00	\$39.915.703,00
12861	4	9,00%	\$170.324.654,00	\$220.791.218,00	\$70.442.912,00	\$91.120.185,00	\$552.678.969,00	\$71.848.266,00
23150	5	9,00%	\$306.587.025,00	\$397.427.625,00	\$126.798.338,00	\$164.017.750,00	\$994.830.738,00	\$129.327.996,00
TOTAL			\$623.273.860,00	\$807.947.596,00	\$257.773.757,00	\$333.438.690,00	\$2.022.433.903,00	\$262.916.407,00

Tabla 6. Ganancias Brutas

Donde el porcentaje de ventas dividido por producto, el precio promedio de este y las ganancias por cada contenido se representan en la siguiente tabla.

Contenido	Precio promedio	% de usuarios que consumen	% en ganancias
Imagen	\$27.000,00	45%	31%
Video	\$45.000,00	35%	40%
Audio	\$33.500,00	15%	13%
Álbum	\$130.000,00	5%	16%
TOTAL		100%	100%

Tabla 7. Porcentaje de ganancias

5.6 Impacto

5.6.1 Impacto en el sector

Needer llegará a revolucionar el consumo de contenido multimedia, debido a su modelo de negocio de adquisición de contenido por unidad y a su red social de libre acceso. El marketplace y la red social abarcan el sector del entretenimiento, uno de los sectores de mayor consumo que se extiende en todas las edades y no discrimina gusto.

Uno de los impactos que se espera **Needer** pueda generar, es el de promover la creación de comunidades en base a un creador de contenido o un gusto en específico. Se espera que los usuarios puedan tener mayor interacción con la plataforma gracias a estas comunidades, además de incitar al consumo de contenido multimedia. Esto puede crear interés en los creadores de contenido, permitiendo el aumento de las ganancias obtenidas por las ventas realizadas dentro de la plataforma. Además, se espera promover la producción artística de cualquier tipo, ya sea arte digital, musical, etc. Brindando un espacio donde los pequeños productores puedan crecer y percibir un ingreso ocasional.

Otro impacto que **Needer** pretende originar, es el de promover y generar el primer empleo para algunas personas. Se podrá contratar jóvenes que aún no cuenten con experiencia laboral, para ser formados y volverse parte esencial del equipo de **Needer**. Promoviendo la familia dentro de la compañía, e incentivando un ambiente laboral flexible y moderno para que todos los que hagan parte de él se sientan cómodos y felices a la hora de realizar sus tareas.

Adicionalmente, se espera que a futuro **Needer** pueda crear o financiar fundaciones o empresas que promuevan el medio ambiente, la educación, el crecimiento y las mujeres.

5.6.2 Generación de empleos

Inicialmente **Needer** generará empleo a los integrantes del proyecto. De ser necesario por el crecimiento en la plataforma se requerirá de moderadores que revisen el contenido publicado en la plataforma o servir como intermediarios en las ventas, en caso de que los integrantes no puedan suplir la carga de resolver las solicitudes.

5.7 Experticia del equipo

5.7.1 Disciplinas del equipo de trabajo

El equipo estará conformado por 2 Ingenieros de Software en formación dedicados a diferentes roles dentro del proyecto. Entre estos están: Diseñador y desarrollador frontend con énfasis en diseño de marca, así como el Ingeniero Backend y Product Manager con enfoque en Desarrollo en el Servidor.

5.7.2 Experiencia del equipo

El equipo cuenta con experiencia desarrollando páginas web de diferentes sectores. Gracias a esto, el grupo cuenta con estándares de calidad, brindando la mejor experiencia posible para el usuario.

Lider Desarrollo Frontend | Jose Thomas Perez Rivera: Cuenta con certificación en Scrum Fundamentals y experiencia en el desarrollo de aplicaciones full stack Web con Django, Javascript y Tailwind.

Líder del proyecto de TI - Líder de Desarrollo | Juan David Trujillo Suarez: Estudiante de Ingeniería con énfasis Python, con experiencia en desarrollo web, cuenta con múltiples certificaciones en Python, Django y Scrum Fundamentals. Dirigió varios proyectos de desarrollo con altos estándares de calidad.

VI. EJECUCIÓN DEL PROYECTO

Una vez definido el modelo de negocio, el siguiente paso a seguir es estructurar la propuesta de creación de empresa. Lo primero será crear un estudio de mercado que permita profundizar en las necesidades de los clientes, con el fin de favorecer en la creación de un producto que se encargue de solucionar dichas necesidades.

En segunda instancia, se deben establecer los componentes organizacionales, legales y técnicos. Los componentes organizacionales se encargarán de definir los aspectos que conformará a la empresa y permitirán establecer los objetivos que la empresa quiere lograr alcanzar. Estos objetivos se verán reflejados en la misión y la visión, puesto que serán utilizados como guía durante todo el proceso de ejecución. Por otra parte, los componentes legales contienen los documentos que la empresa debe entregar a la hora de ser constituida. En estos documentos se menciona la razón social de la empresa y los estatutos que regirán en esta, una vez sea constituida. Un ejemplo claro de esto, son los términos y condiciones, los cuales se encargarán de delimitar los derechos y deberes de los clientes a la hora de usar el producto ofrecido por la empresa, que en este caso será el prototipo web.

De manera similar, se tienen también los componentes técnicos. Estos componentes son los encargados de documentar el proceso de construcción del prototipo web de la empresa, estableciendo fases de análisis, diseño, desarrollo, testeo y despliegue. Estas fases están constituidas por distintas metodologías, las cuales permiten planificar una estructura sólida para la creación del prototipo. Además, se implementarán herramientas y tecnologías enfocadas al desarrollo web, utilizando las metodologías y aplicando los diseños, permitiendo la creación de un prototipo que encaje con el estudio de mercado obtenido anteriormente.

Por último se tienen las proyecciones de ingresos, costos y gastos y las evaluaciones financieras, sociales y ambientales. Por un lado, las proyecciones de ingresos, costos y gastos, se utilizarán para determinar los recursos que el proyecto necesitará para su ejecución y las ganancias que generará durante este tiempo, permitiendo desarrollar un análisis y estudio financiero minucioso en torno a la empresa.

Por otro lado, las evaluaciones financieras, sociales y ambientales, son un factor importante para poder establecer los avances que la empresa tendrá en cada uno de estos sectores. En el

aspecto financiero, en conjunto con el análisis de demanda, se evaluará la viabilidad del proyecto. En el aspecto social, se espera reconocer el impacto social que ha la empresa generará, ya sea brindando empleo, promoviendo el comercio, entre otros. En cuanto al aspecto ambiental, se espera reconocer los beneficios ambientales que la empresa logrará realizar durante su ejecución, ya sea promoviendo campañas del cuidado del medio ambiente o realizando inversiones para la preservación y cuidado del mismo.

6.1 Investigación de mercados

Con el fin de determinar una solución a las necesidades de los clientes, identificados en el Modelo de Negocio, se plantea una estrategia que estudie a profundidad los problemas que tienen los clientes o beneficiarios frente al producto que se ofrecerá en el prototipo. Para esto, se propone realizar un estudio de demanda, con el fin de establecer y solucionar dichos problemas, y además entender la disposición de consumo de los clientes. También se propone realizar un estudio de oferta, en el cual se puedan identificar características como semejanzas, diferencias y cualidades de la competencia frente a la empresa. Esto ayudará a la implementación de las funcionalidades al prototipo a desarrollar, permitiendo que se diferencie y se destaque de ellos.

6.1.1 Estudio de demanda

El estudio de demanda está compuesto por el estudio cualitativo y el estudio cuantitativo. El primer estudio a realizar es el cualitativo, este se encarga de determinar las necesidades puntuales de los actuales clientes. El segundo es el cuantitativo, el cual se espera que determine la disposición del consumo del producto por parte de los clientes.

6.1.1.1 Estudio cualitativo

6.1.1.1.1 Introducción

En vista que se necesita reconocer las necesidades puntuales de los usuarios, se propone la implementación de una encuesta cualitativa, aplicando la metodología de entrevista abierta. Permitiendo obtener las opiniones de los usuarios en base al modelo de negocio tradicional (Por suscripción), comparado al nuevo modelo de negocio que se planea implementar en el prototipo web. Además, se espera conocer las falencias de las actuales plataformas, con el fin de proponer soluciones óptimas en el prototipo a realizar.

6.1.1.1.2 *Objetivos de la entrevista*

Proponer una entrevista no estructurada con el fin de conocer y analizar las necesidades puntuales de los usuarios consumidores o creadores de contenido.

- Diseñar una entrevista no estructurada, a partir de la metodología de análisis cualitativo de la demanda.
- Aplicar la entrevista a un grupo prediseñado de 10 consumidores y 2 creadores de contenido.
- Analizar los resultados obtenidos de la entrevista realizada, organizando las respuestas en grupos, permitiendo determinar las situaciones que la originan.
- Encontrar las posibles soluciones, a partir del estudio de los requerimientos planteados en el análisis.

6.1.1.1.3 *Justificación de la encuesta*

Con la finalidad de estructurar los productos de manera adecuada y que garanticen la satisfacción del cliente, es indispensable identificar las necesidades que presentan cada uno de ellos. Como se mencionó anteriormente, se piensa implementar la metodología cualitativa en entrevistas no estructuradas, determinando las necesidades puntuales de los clientes y permitiendo trazar las posibles soluciones que encajen con estas. Además, esto posibilita la oportunidad de implementar nuevas funcionalidades que los clientes quieran utilizar a futuro.

6.1.1.1.4 *Metodología a aplicar*

Para estructurar el estudio de demanda, se realizará una entrevista a 12 usuarios objetivos, esta será no estructurada (aplicada de forma abierta), permitiendo a las personas expresar sus necesidades como ellos prefieran por medio de un diálogo.

Una vez se tenga recopilada toda la información, se realizará un análisis de coincidencias, permitiendo identificar las respuestas similares entre ellas. Estas respuestas serán organizadas y se determinará una situación que permita reconocerse como detonante para cada caso.

6.1.1.1.5 Resultados Esperados

Como resultado de la entrevista se espera obtener y comprender las necesidades de los usuarios a través de charlas no estructuradas. De estos requerimientos se estima que los más relevantes puedan ser la falta de un modelo de negocio de venta sobre demanda, la necesidad de exclusividad en los productos y un entorno en el cual se puedan crear comunidades en base a una persona o celebridad.

6.1.1.1.6 Procedimiento y resultados

El primer paso realizado fue identificar a las personas que posiblemente consumen este tipo de plataformas regularmente, dado que se requería entender las necesidades puntuales de ellos. En este caso se optó por encuestar a 10 personas en el rango de edad principal estimado dentro del estudio de mercado, es decir, entre 18 a 28 años. Por otro lado, se escogieron 2 creadores de contenido que suelen vender sus productos a través de los marketplaces ofrecidos por la competencia, con el fin de comprender las necesidades y falencias que tienen estas. Se realizó la búsqueda de personas interesadas en participar en la entrevista en distintas redes sociales y plataformas de comercio.

Se realizaron las entrevistas en plataformas como Discord, E-pal y Google Meet las cuales cuentan con una herramienta de comunicación por chat de voz, entrevistando un total de 8 hombres y 4 mujeres, de las cuales 2 mujeres eran creadoras de contenido. Estas entrevistas tuvieron una duración entre 5 minutos a 18 minutos. Por otra parte, se propuso utilizar un nombre diferente al propio de cada uno de ellos, para mantener el anonimato, preservando así su identidad. Las grabaciones de las entrevistas fueron guardadas en una carpeta anexa al proyecto.

A continuación, se mostrarán las preguntas que se realizaron en las entrevistas para los dos tipos de usuario. No se utilizó una estructura cerrada para conllevar la entrevista, puesto que se deseaba implementar como un diálogo:

Consumidor:

- ¿Ha consumido alguna vez contenido multimedia en plataformas de suscripción o pago por unidad a través de marketplaces (EJ: Patreon, Onlyfans, Ko-Fi, Twitch)?
- ¿Qué tipo de contenido suele consumir más?

- ¿Cuáles son sus creadores de contenido favoritos y por qué?
- ¿Cuánto dinero gasta al mes de suscripciones o compras?
- ¿Qué plataformas ha utilizado a la hora de consumir el contenido multimedia?
- ¿Qué le gusta de estas plataformas?
- ¿Qué no le gusta de estas plataformas?
- ¿Qué cambiaría usted de estas plataformas?
- ¿Qué opina del modelo de negocio de compra de contenido multimedia al detal y suscripción?

Creador de Contenido:

- ¿Qué tipo de contenido hace?
- ¿Cada cuánto publica productos multimedia?
- ¿En qué plataformas suele vender sus productos?
- ¿Qué le gusta de estas plataformas?
- ¿Qué no le gusta de estas plataformas?
- ¿Qué cambiaría de estas plataformas?
- ¿Qué opina de que los consumidores compren sus productos al detal o por suscripción?

Una vez realizadas las entrevistas, se identificaron las ideas principales presentadas por cada uno de los usuarios dentro de un documento aparte. Este documento está organizado por entrevista y se utilizó como base para poder identificar los 6 problemas presentados en las plataformas mencionadas, resumido en la siguiente tabla.

#	Problemas identificados
1	Falta de sugerencias por parte de la plataforma en contenidos relacionados a las preferencias de los usuarios.
2	Interfaz gráfica muy simple y poco llamativa para todos los usuarios.
3	No existen diferentes métodos de pago dentro de la plataforma.
4	Falta de seguridad por parte de la plataforma frente a los productos publicados por los

	creadores de contenido.
5	Precios muy elevados en los productos ofrecidos en la plataforma.
6	No existe una herramienta que permita la comunicación directa entre los consumidores y los creadores de contenido.

Tabla 8. Problemas encontrados por parte de los usuarios

Una vez identificados los problemas, se realiza un diagrama de Ishikawa o mejor conocido como diagrama de causa y efecto, por cada problema. Este diagrama permite determinar las causas potenciales de los inconvenientes presentados. Para el desarrollo de los modelos, se utilizaron las dificultades identificadas anteriormente y se plantearon de manera directa con la plataforma. También se identificaron los 4 actores principales del diagrama, el sistema, el consumidor, los creadores de contenido y los desarrolladores. Por otra parte, se decidió colocar mínimo 2 causas por parte de cada actor, facilitando así el reconocimiento de las posibles soluciones frente a estos.

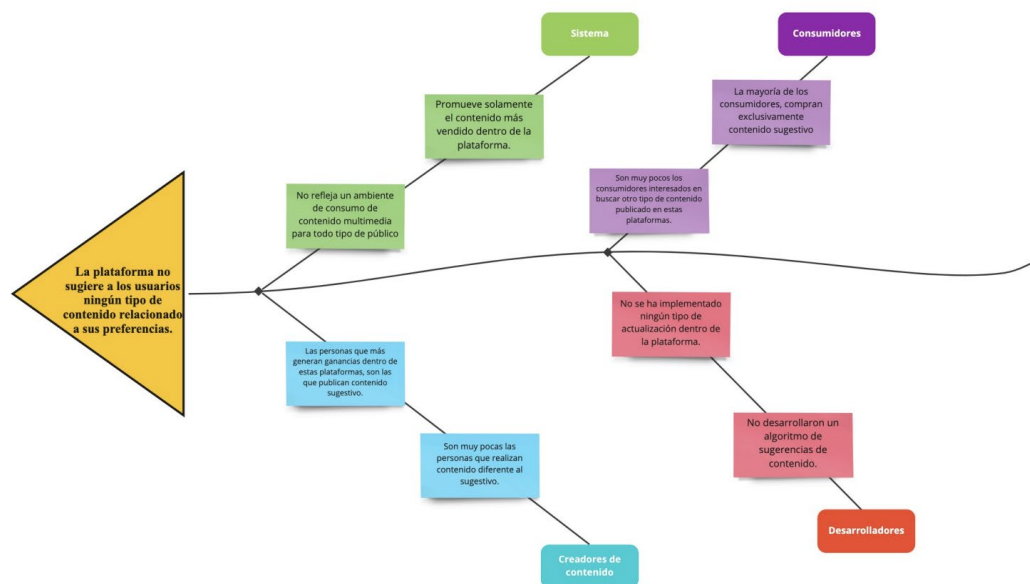


Figura 35. Diagrama No 1. Causa y efecto de "Needer By Juan Trujillo y Thomas Pérez".

Para el desarrollo del diagrama anterior, se utilizó el problema descrito como "Falta de sugerencias por parte de la plataforma en contenidos relacionados a las preferencias de

los usuarios”, este problema fue identificado gracias a las opiniones presentadas por los entrevistados, en dónde reflejan la necesidad de disponer de una plataforma que sugiera contenido relacionado a las preferencias de ellos, evitando así la monotonía en el consumo. En base a esto, se planteó el problema de manera directa con la plataforma, permitiendo identificar las causas por parte de cada actor:

- **Sistema:** Al no estar diseñado para identificar los gustos de cada uno de los usuarios y contar con muy poca variedad de contenido en su marketplace, permite determinar causas relacionadas con su estructura y con los productos que ofrece.
- **Consumidores:** Al adquirir solamente productos de un tipo y no mostrar interés por la variedad de productos que ofrece la plataforma, permite definir causas relacionadas al consumo e interés propio de los consumidores.
- **Creadores de Contenido:** El mercado actual permite generar muchas ganancias a los creadores de contenido que no ofrecen productos para todo público dentro de la plataforma, por esta razón, se presentan causas relacionadas a los intereses del creador de contenido.
- **Desarrolladores:** En vista de que la plataforma cuenta con un modelo de negocio tradicional y está enfocada a un público adulto, impidiendo así la variedad de contenido dentro de esta, logrando así la identificación de causas relacionadas a su desarrollo.

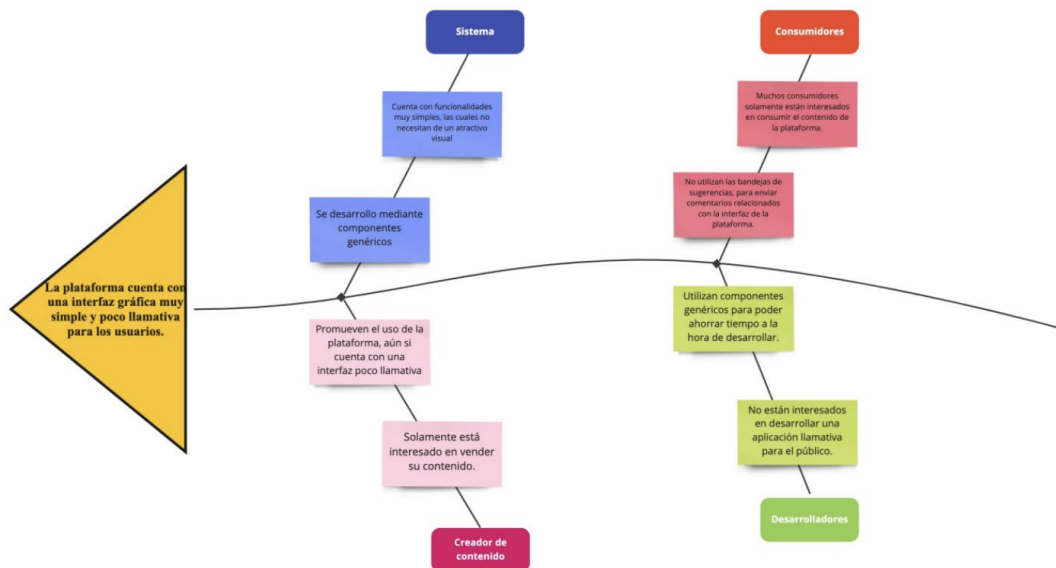
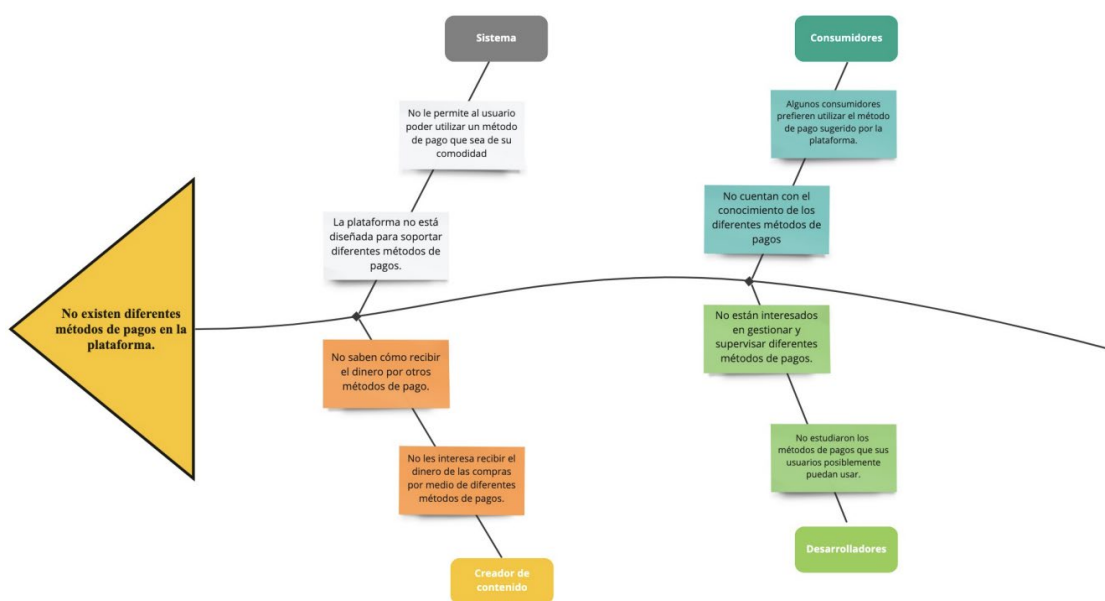


Figura 36. Diagrama No 2. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

En el caso del segundo diagrama (Figura 36), se utilizó el problema descrito como “**Interfaz gráfica muy simple y poco llamativa para todos los usuarios**”, este problema se identificó por medio de las opiniones presentadas por los entrevistados, en dónde mencionan la necesidad de utilizar una plataforma que cuente con una interfaz gráfica más llamativa, generando a los usuarios comodidad a la hora de hacer uso de esta. Gracias a esto, se establecen las causas de cada actor:

- **Sistema:** La plataforma está diseñada para ofrecer productos y permitir al usuario hacer uso de ciertas funcionalidades. Estos factores no se ven afectados por el atractivo de la interfaz gráfica, gracias a esto se pudo identificar causas vinculadas al diseño y desarrollo de la plataforma.
- **Consumidores:** Los consumidores solamente están interesados en adquirir el contenido de la plataforma y no en la manera en la que esta presenta los productos. Por tal razón, se determinan causas enlazadas a la opinión de los consumidores y al consumo del producto.
- **Creadores de Contenido:** No están interesados en el atractivo visual que tiene la plataforma en donde publican sus productos. Es por esto que, se plantean causas vinculadas a los intereses del creador de contenido y a la venta de los productos.

- **Desarrolladores:** Con la finalidad de ahorrar tiempo e implementar solamente funcionalidades que no abarcan más allá de lo necesario, se plasmaron causas enlazadas al diseño y desarrollo de la plataforma.



Figuran 37. Diagrama No 3. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Para la ejecución del tercer diagrama representado en la figura mostrada, se empleó el problema: “**No existen diferentes métodos de pago dentro de la plataforma**”, este se reconoció gracias a las opiniones entregadas por los entrevistados, mencionando la necesidad de utilizar un método de pago más cómodo para ellos, facilitando la adquisición de productos por parte de la plataforma. Gracias a esto, se establecen las causas de cada actor:

- **Sistema:** La plataforma está diseñada para permitir un sólo método de pago, siendo un factor de gran incomodidad para los usuarios que adquieren el contenido publicado en esta. Por esta razón, se determinaron causas relacionadas al diseño de la plataforma.
- **Consumidores:** Al no tener el conocimiento de otros métodos de pago o no contar con otro método de pago, genera un gran impedimento a la hora de adquirir productos. Por consiguiente, se logra identificar causas asociadas al conocimiento de los consumidores.

- **Creadores de Contenido:** No tienen intenciones de recibir los pagos realizados por los consumidores mediante diferentes métodos, puesto que pueden no tener acceso o conocimiento sobre estos métodos de pagos, generando causas relacionadas al interés propio del creador de contenido.
- **Desarrolladores:** No tienen intenciones de implementar herramientas que permitan el uso de diferentes métodos de pago. Por esta razón, se identifican causas relacionadas al desarrollo e interés propio por parte de los desarrolladores.

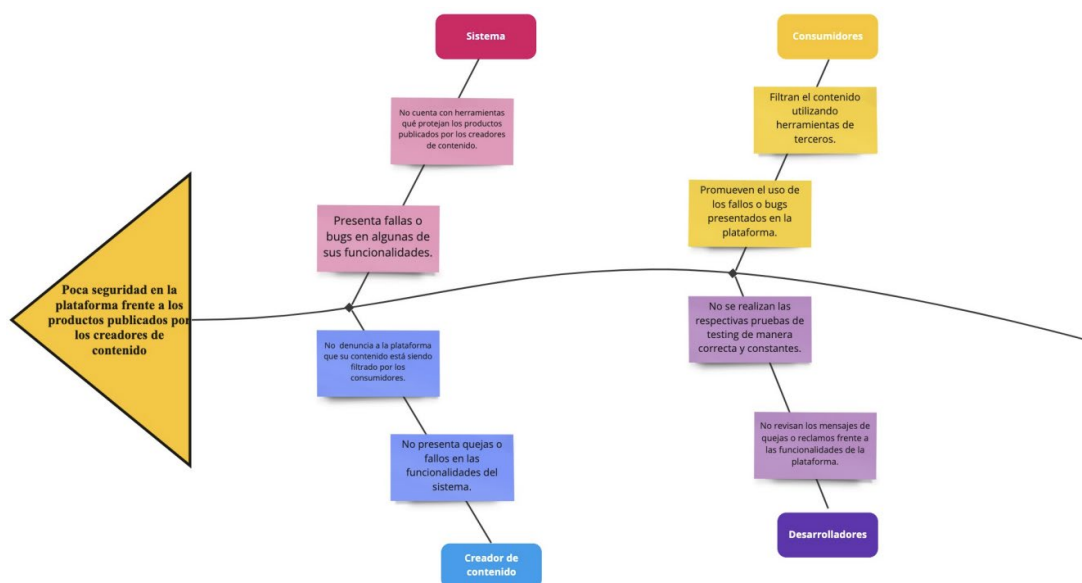


Figura 38. Diagrama No 4. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Para la realización del cuarto diagrama, se empleó el problema expuesto como **“Falta de seguridad por parte de la plataforma frente a los productos publicados por los creadores de contenido”**. Este problema se logró revisar gracias a las opiniones realizadas por los entrevistados, en donde nombran la necesidad de utilizar una plataforma que no cuente con fallas en las funcionalidades e implemente herramientas que ofrezcan seguridad en los productos publicados por los creadores de contenido. En base a esto, se definieron las causas de cada actor:

- **Sistema:** Al estar compuesto por componentes genéricos y no contar con un mantenimiento constante, generando fallas y bugs en el funcionamiento de la plataforma. Se determinaron causas vinculadas al mantenimiento y desarrollo del sistema.
- **Consumidores:** Teniendo como objetivo aprovecharse de las fallas presentadas en el sistema y presentando poco interés en reportar dichos fallos a los desarrolladores. Se generan causas enlazadas al interés propio del consumidor.
- **Creadores de Contenido:** El creador de contenido no presenta ningún tipo de queja frente a las fallas encontradas en el sistema, impidiendo así una pronta identificación y solución de estos. Por esta razón se identificaron causas relacionadas al poco conocimiento o interés de los creadores de contenido por la plataforma.
- **Desarrolladores:** Al utilizar componentes genéricos y confiar sólidamente en estos, se generaron fallas en el sistema y se suma la falta de testings, impidiendo la solución de estos fallos. Esto creó causas relacionadas al interés que tienen los desarrolladores frente a la plataforma.

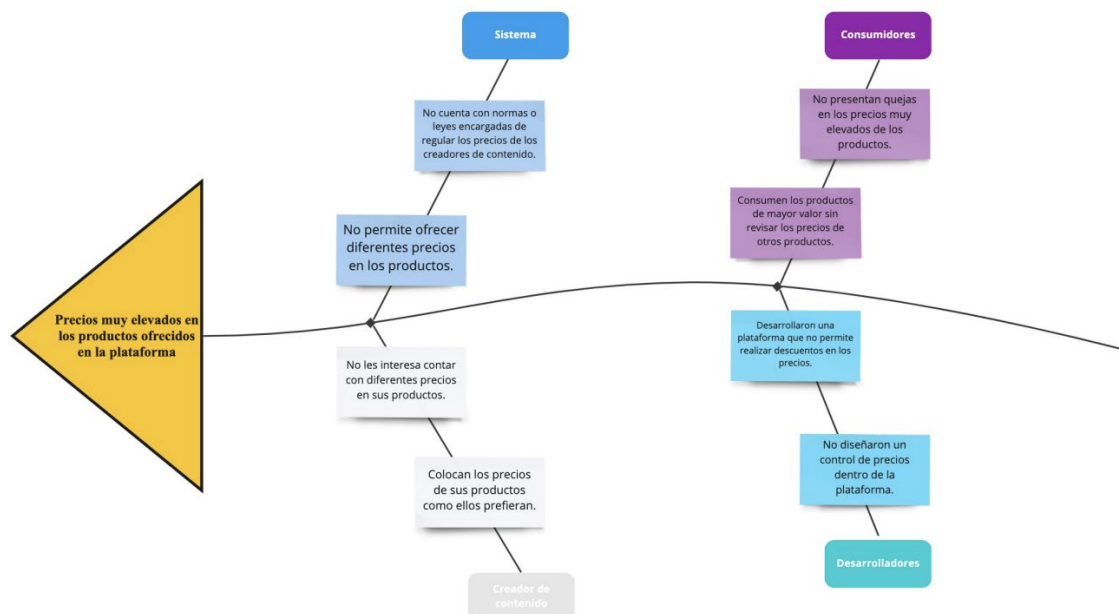


Figura 39. Diagrama No 5. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Para el desarrollo del quinto diagrama, se basó en el inconveniente expuesto como “**Precios muy elevados en los productos ofrecidos en la plataforma**”, este problema se identificó en base a las opiniones presentadas por los entrevistados, mencionando la necesidad de contar con una plataforma que ofrezca productos con precios más accesibles para los usuarios. Gracias a esto, se establecen las causas de cada actor:

- **Sistema:** La plataforma no cuenta con una funcionalidad que permita regular los precios presentados por los creadores de contenido. Esto generó causas en el consumo de los productos.
- **Consumidores:** Los consumidores están enfocados solamente en el consumo del producto, sin tener en cuenta el costo asignado por el creador de contenido. Esto permitió representar causas enlazadas al interés propio del consumidor y al poco conocimiento que tiene frente a los productos.
- **Creadores de Contenido:** No les interesa saber si los consumidores pueden pagar o no su contenido, puesto que el sistema funciona mediante la oferta y la demanda, dando libertad de mercado a los creadores de contenido. En base a esto, se presentaron causas relacionadas al interés propio del creador de contenido.
- **Desarrolladores:** Desarrollaron una plataforma que cuenta con un modelo de negocio tradicional, impidiendo la implementación de herramientas que regulen los precios de los productos, presentando causas enfocadas al desarrollo y diseño de la plataforma.

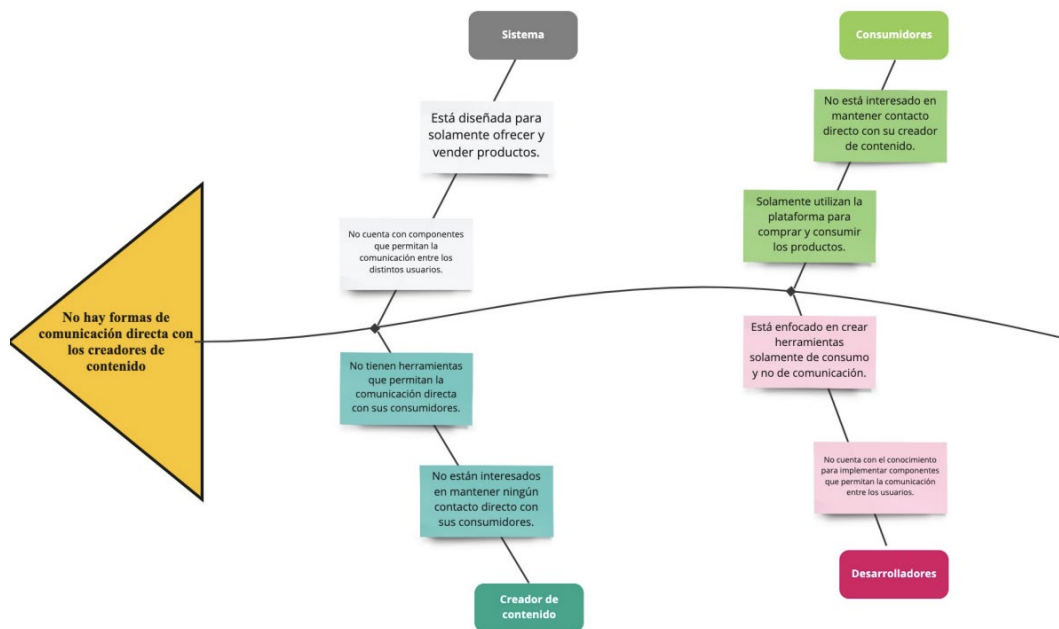


Figura 40. Diagrama No 6. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Para el desarrollo del sexto diagrama, se hizo uso del problema presentado como “**No existe una herramienta que permita la comunicación directa entre los consumidores y los creadores de contenido**”, identificado gracias a las opiniones presentadas por los entrevistados, mencionando la necesidad de contar con una plataforma que implemente una herramienta de comunicación entre los distintos usuarios que hacen uso de esta. Por esta razón, se establecen las causas de cada actor:

- **Sistema:** La plataforma está diseñada únicamente para ofrecer y vender productos, no cuenta con una herramienta que permita la comunicación entre los usuarios. Esta situación presentó causas relacionadas al diseño y desarrollo de la plataforma.
- **Consumidores:** No tienen interés en mantener un contacto directo con sus creadores de contenido favoritos, presentando causas relacionadas en el interés propio del consumidor.
- **Creadores de Contenido:** Desconocen las herramientas de comunicación por parte de la plataforma o no tienen interés en mantener un contacto directo con sus consumidores, permitiendo identificar causas vinculadas al interés propio del creador de contenido y al conocimiento que tiene este frente a la plataforma.

- **Desarrolladores:** No tienen intenciones de implementar herramientas que permitan el contacto entre los usuarios y los creadores de contenido o pueden no tener conocimiento sobre la implementación de estas herramientas, implicando causas dirigidas al conocimiento del desarrollador e intereses propios.

6.1.1.1.7 Conclusiones

Como conclusión del análisis de demanda, se pudo observar que, para conocer las necesidades puntuales de las personas, la metodología de entrevista no estructurada es de gran utilidad. Puesto que se conocen los problemas dentro de la perspectiva de los usuarios y garantiza la comodidad a la hora del diálogo.

Por otro lado, gracias a la ejecución de este análisis, se propusieron soluciones que encajen de manera precisa con el perfil de la plataforma, puesto que estas soluciones serán aplicadas en el diseño y ejecución del proyecto. Además, se espera poder implementar estos procedimientos en etapas futuras, ya que se espera desarrollar un prototipo escalable, el cual permita prevenir problemas similares a futuro, asegurando el bienestar de los usuarios dentro de dicho prototipo. A continuación, se presentarán las soluciones propuestas para los problemas presentados:

- **Sistema de sugerencias de consumo:** Se plantea diseñar un algoritmo de sugerencias a la hora de adquirir un producto del marketplace o reaccionar a una publicación de la red social, permitiendo así analizar el consumo y gustos de los usuarios y registrando referencias para realizar una recomendación en base al consumo y los gustos.
- **Interfaz basada en patrones de diseño modernos:** Se definió el uso de patrones de diseño modernos para realizar una interfaz gráfica más agradable para todos los usuarios que hacen uso del prototipo web, con la finalidad de aumentar y promover el consumo de los productos que se ofrecen en el marketplace o la comunicación entre los usuarios por medio de los componentes de la red social.
- **Implementación de pagos internacionales:** Se propone el uso de una pasarela de pagos internacional, que permita a los usuarios poder realizar transacciones de manera más cómoda. Esto con el objetivo de captar la atención de los usuarios y fomentar el consumo de los productos ofrecidos en el marketplace.

- **Modelo de negocio de venta por unidad:** Se realizó la propuesta de implementar un modelo de negocio, que permita la venta y consumo de contenido por unidad, promoviendo el libre mercado y permitiendo a los usuarios poder adquirir productos más accesibles para ellos.
- **Mensajes entre usuarios y la venta del contenido personalizado con intermediación de los moderadores:** Se planea diseñar un sistema de mensajería entre usuarios, con el fin de promover mayor interacción entre ellos. Además, se enlazará este sistema con un componente del marketplace, el cual permitirá a los comerciantes poder ofrecer productos personalizados por este medio. Esta funcionalidad contará con la presencia de un moderador, que servirá de intermediario frente a las posibles peticiones recibidas por parte de los usuarios y con el fin de que ambos queden satisfechos en el intercambio.

6.1.1.2 Estudio Cuantitativo

6.1.1.2.1 Introducción

Con la necesidad de determinar la viabilidad de la empresa y el proyecto, se propone la implementación de una encuesta cuantitativa, aplicando la metodología de encuesta cerrada. Esto con el fin de establecer la cantidad de usuarios que estarían dispuestos a hacer uso del prototipo web propuesto, consumiendo a su vez los productos multimedia que serán ofrecidos a través de la plataforma. Se aplica la plataforma de Google Forms, para la representación de los datos, mediante gráficas y porcentajes, facilitando así su comprensión y análisis.

6.1.1.2.2 Objetivos de la encuesta

Proponer una encuesta estructurada con el fin de conocer la viabilidad del producto ofrecido por Needer y el interés de los usuarios en el consumo de contenido multimedia.

- Diseñar una encuesta estructurada, a partir de la metodología de encuesta cuantitativa.
- Aplicar la encuesta a un grupo prediseñado de aproximadamente 100 personas, sin importar su género, en un rango de edad entre los 18 - 50 años o superior.

- Analizar los resultados obtenidos de la encuesta realizada, visualizando las respuestas generadas en Google Forms, permitiendo determinar la viabilidad del producto ofrecido por la empresa y el interés de los usuarios frente al consumo de contenido multimedia.

6.1.1.2.3 Justificación de la encuesta

Con la finalidad de determinar la viabilidad de la empresa y de sus productos, es indispensable identificar el interés de los clientes frente al consumo de contenido multimedia. Como se mencionó anteriormente, se piensa implementar la metodología cuantitativa en encuestas estructuradas, permitiendo identificar el interés de los clientes frente a la empresa y sus productos. Una vez recolectada la información de la encuesta, se establecerá la cantidad de personas en Neiva que estén dispuestas a consumir los productos ofrecidos por la empresa y en base a eso tener la posibilidad de abrir el producto a otros mercados

6.1.1.2.4 Metodología a aplicar

Para estructurar el estudio de viabilidad, se planea realizar una encuesta a un aproximado de 100 personas entre los 18 - 50 años y superior, esta será estructurada (contará con opciones múltiples para facilitar la respuestas) permitiendo recopilar información de manera precisa.

Una vez encuestadas las personas, se podrán revisar las respuestas representadas en gráficas y tablas, permitiendo establecer el interés que tienen los usuarios frente a la empresa, sus productos y el consumo de contenido multimedia.

6.1.1.1.5 Resultados Esperados

Como resultado de la encuesta, se espera obtener y comprender los intereses de los usuarios frente a la empresa, sus productos y el consumo de contenido multimedia. Una vez se identifiquen estos intereses, se espera poder determinar la cantidad de usuarios que estarían dispuestos a hacer uso del prototipo web creado por la empresa, permitiendo establecer la viabilidad que se tendrá en el mercado.

6.1.1.2.6 Procedimiento y resultados

En primer lugar, se decidió encuestar una media de 111 personas en la ciudad de Neiva, con el fin de conocer el interés que las personas locales puedan tener frente a la empresa, al prototipo web desarrollado por esta y al consumo de contenido multimedia. Se optó por

encuestar hombres y mujeres entre los 18 años de edad, hasta los 50 años de edad o más, los cuales debieron ingresar a la plataforma de Google Forms para realizar dicha encuesta.

La encuesta se realizó en Google Forms puesto que esta plataforma ayuda a representar las respuestas por medio de gráficas y porcentajes, permitiendo mejorar su interpretación. La encuesta cuenta inicialmente con una breve descripción de la empresa, el prototipo web desarrollado por esta y la finalidad que tiene la encuesta. Posteriormente, se encontrarán 7 preguntas, de las cuales todas serán de opción múltiple. Algo a tener en cuenta, es que para proteger la privacidad de las personas, no se pidió el ingreso de ningún dato personal que permitiera identificar a los encuestados. A continuación, se mostrarán las preguntas realizadas en la encuesta para las 111 personas con sus respectivas opciones:

- ¿Cuántos años tiene usted?
 1. 18 - 24 años.
 2. 25 - 34 años.
 3. 35 años - 44 años.
 4. 45 años - 50 años.
 5. Más de 50 años.
- ¿Con cuál género se identifica?
 1. Hombre.
 2. Mujer.
 3. Otro.
- ¿Usted ha visitado alguna de las siguientes aplicaciones web?
 1. Patreon.
 2. OnlyFans.
 3. Ko-Fi.
 4. Facebook.
 5. Fiverr.
 6. Instagram.
- ¿Cuánto dinero gasta usted mensualmente en el consumo de contenido multimedia?

Nota: Los gastos se miden en Pesos Colombianos.

1. \$0.
 2. \$10000 - \$29000.
 3. \$30000 - \$49000.
 4. \$50000 - \$69000.
 5. Más de \$70000.
- ¿Cuántas horas gasta navegando en las redes sociales diariamente?
 1. Menos de 2 horas.
 2. Entre 3 y 5 horas.
 3. Entre 6 y 8 horas.
 4. Más de 9 horas.
 - ¿Estaría usted dispuesto a utilizar un prototipo web que enlace componentes de red social y marketplace?
 1. Si.
 2. No.
 3. Tal vez.
 - ¿Estaría usted interesado en utilizar una plataforma que ofrezca productos de contenido multimedia (**Fotos, audios y vídeos**) por unidad?
 1. Si.
 2. No.
 3. Tal vez.

Una vez realizadas las encuestas, se analizaron las gráficas y los porcentajes de cada una de las preguntas, con el fin de reconocer los intereses de los usuarios, permitiendo así identificar los puntos claves que ayudarán a conocer la viabilidad de la empresa y sus productos en el mercado.

En la siguiente gráfica se puede reflejar el promedio de edad de los 111 encuestados:

¿Cuántos años tiene usted?

111 respuestas

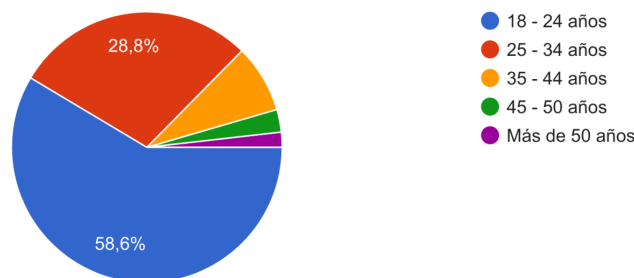


Figura 41. Gráfico de edad de los encuestados de "Needer By Juan Trujillo y Thomas Pérez".

Un análisis de la pregunta sobre la edad de las personas podría enfocarse en las tendencias y patrones demográficos observados en la muestra encuestada. Algunas observaciones generales que se lograron hacer a partir de los datos proporcionados son:

La mayoría de las personas encuestadas se encuentran en la edad joven (58,6% entre 18 y 24 años) y adulta temprana (28,8% entre 25 y 34 años). Esto podría indicar que la muestra encuestada está compuesta principalmente por personas jóvenes y adultos tempranos.

Un porcentaje significativamente menor de personas encuestadas se encuentran en la edad adulta intermedia (8,1% entre 35 y 44 años) y adulta mayor (2,7% entre 45 y 50 años). Esto podría indicar una menor representación de estas edades en la muestra encuestada. El porcentaje restante de personas encuestadas (menos del 2%) se encuentran en edad mayor de 50 años. Esto podría indicar que hay una menor representación de personas mayores en la muestra encuestada.

En la siguiente gráfica se puede observar los porcentajes de género obtenidos por parte de los encuestados:

¿Con cuál género se identifica?
111 respuestas

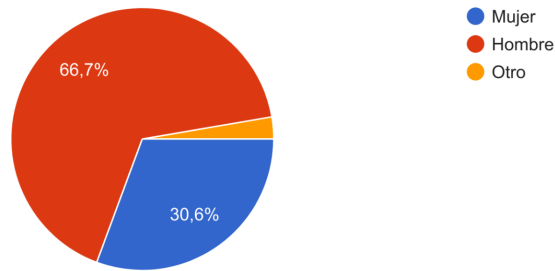


Figura 42. Gráfico de género de "Neder By Juan Trujillo y Thomas Pérez".

Un análisis económico podría enfocarse en los aspectos financieros relacionados con la identidad de género. Sin más contexto sobre el tema específico de la encuesta y la información obtenida, no es posible realizar un análisis económico completo. Sin embargo, se podrían hacer algunas observaciones generales a partir de los datos proporcionados:

La mayoría de las personas se identificaron como hombres (66,7%), seguidas de mujeres (30,6%) y un porcentaje restante de 2,7% seleccionó otra opción. Esto podría indicar que hay una desproporción en la representación de género en la muestra encuestada.

El porcentaje restante de personas que seleccionaron "otro" podría representar una variedad de identidades de género que no se incluyeron en las opciones de la encuesta. Esto podría sugerir que hay una necesidad de considerar una mayor variedad de identidades de género en futuras encuestas. Esto es importante, puesto que la empresa puede conocer el tipo de público al cual deberán estar dirigidos mayormente sus productos, facilitando así la planeación y el desarrollo de dichos productos.

En el siguiente gráfico se muestran la cantidad de personas que han visitado las aplicaciones web mencionadas en la tercera pregunta:

¿Usted ha visitado alguna de las siguientes aplicaciones web?

111 respuestas

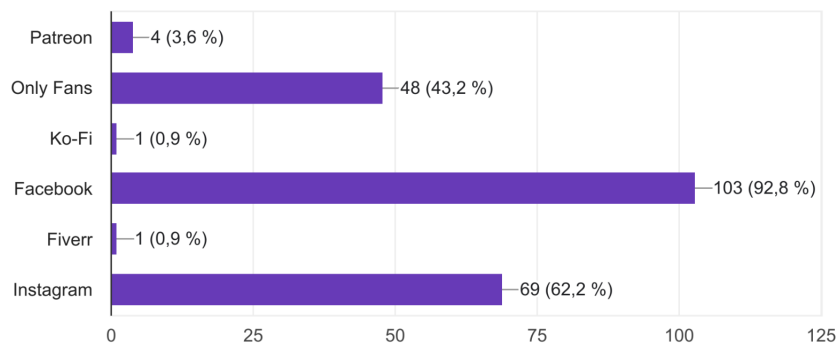


Figura 43. Gráfico de visitantes de las aplicaciones web de “Needer By Juan Trujillo y Thomas Pérez”.

Como se puede observar en el gráfico, un porcentaje mayor de las personas encuestadas (92,8%) han utilizado Facebook, una aplicación web para conectarse con amigos y compartir contenido. Esto podría indicar que hay una alta demanda de conectarse con amigos y compartir contenido en la muestra encuestada. Así mismo, la mayoría de las personas encuestadas (62,2%) han utilizado Instagram, una aplicación web para compartir fotos y videos. Esto podría indicar que hay una alta demanda de compartir contenido audiovisual en la muestra encuestada.

El 43,2% de los encuestados han visto OnlyFans, una aplicación web de contenido para adultos exclusiva. Esto podría indicar que hay una demanda moderada de contenido multimedia exclusivo en la muestra encuestada. Un porcentaje menor de personas encuestadas (4%) han utilizado Patreon, una aplicación web para apoyar a creadores de contenido. Esto podría indicar que hay una baja demanda para apoyar a los creadores de contenido en la muestra encuestada. Para el resto de las aplicaciones, como Ko-Fi y Fiverr, aplicaciones que cuentan con un porcentaje menor al 1%, puede ser a razón de falta de conocimiento de estas aplicaciones o también, porque están enfocadas a un público totalmente diferente al encontrado en la ciudad de Neiva.

En la siguiente tabla, se podrá observar el dinero gastado mensualmente en el consumo de contenido multimedia por parte de los encuestados:

¿Cuánto dinero gasta usted mensualmente en el consumo de contenido multimedia? Nota: Los gastos se miden en Pesos Colombianos.

111 respuestas

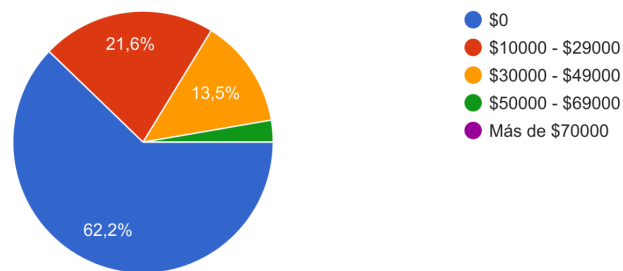


Figura 44. Gráfico de Dinero gastado mensualmente en el consumo de contenido multimedia por parte de los encuestados de “Needer By Juan Trujillo y Thomas Pérez”.

Gracias a este gráfico, se determinó que la mayoría de los encuestados (62,2%) no gastan ninguna cantidad de dinero mensualmente al consumir contenido multimedia. Esto puede ser a causa del desinterés por una parte de la población de Neiva en comprar y consumir contenido multimedia, puesto que la mayoría de las plataformas ofrecen estos productos en dólares, siendo un impedimento para ciertas personas.

Por otra parte, se puede apreciar que, menos del 50% de los encuestados gastan dinero en el consumo de contenido multimedia, ya sea por conveniencia, entretenimiento, prestigio o necesidad. No se puede dar una razón puntual del por qué la gente gasta dinero en el consumo de contenido multimedia, ya que cada persona puede tener muchas razones sobre su consumo.

El siguiente gráfico representa la cantidad de horas gastadas, por los encuestados, navegando en las redes sociales durante el mes:

¿Cuántas horas gasta navegando en las redes sociales diariamente?

111 respuestas

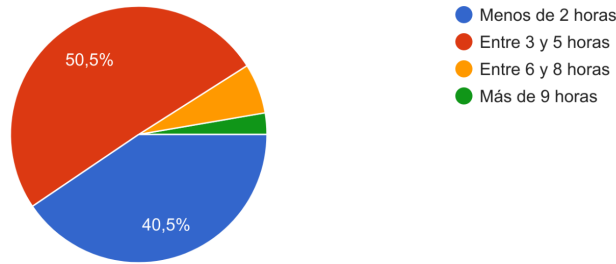


Figura 45. Gráfico de horas gastadas mensualmente en redes sociales de “Needer By Juan Trujillo y Thomas Pérez”.

Como se observó en el gráfico anterior, la mayoría de los encuestados gastan entre 3 y 5 horas navegando en las redes sociales diariamente. Esto significa que la mayoría de las personas, más específicamente un 50,5% tienen un uso moderado de las redes sociales y dedican una cantidad de tiempo a esta actividad. Por otro lado, el 6,3% de las personas encuestadas gastan entre 6 y 8 horas diarias navegando en las redes sociales. Esto significa que una pequeña parte de las personas encuestadas dedican una cantidad de tiempo moderada a esta actividad.

El 40,5% de las personas encuestadas gastan menos de 2 horas al día navegando en las redes sociales. Esto significa que casi la mitad de las personas encuestadas dedican una cantidad ínfima de tiempo a esta actividad. Finalmente, el porcentaje restante (2,7%) gasta más de 9 horas al día navegando en las redes sociales. Esto significa que un pequeño grupo de personas encuestadas dedican una gran cantidad de tiempo a esta actividad.

Teniendo en cuenta el producto explicado dentro del encabezado, ¿Estaría usted dispuesto a utilizar un prototipo web que enlace componentes de red social y marketplace?

111 respuestas

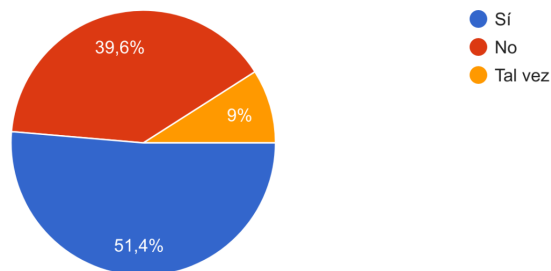


Figura 46. Gráfico de interés de uso de prototipo web “Needer By Juan Trujillo y Thomas Pérez”.

Según los datos proporcionados por la gráfica anterior, la mayoría de los encuestados estarían dispuestos a utilizar un prototipo web que enlace componentes de red social y marketplace, más precisamente un 51,4% de los encuestados. El 39,6% de los encuestados, no estarían dispuestos a utilizar el prototipo web y el 9% de los encuestados pueden que si estén o no interesados en hacer uso de dicho prototipo web. No se sabe con certeza las razones del por qué estarían dispuestos o no a utilizar el prototipo web.

¿Estaría usted interesado en utilizar una plataforma que ofrezca productos de contenido multimedia (Fotos, audios y vídeos) por unidad?
111 respuestas

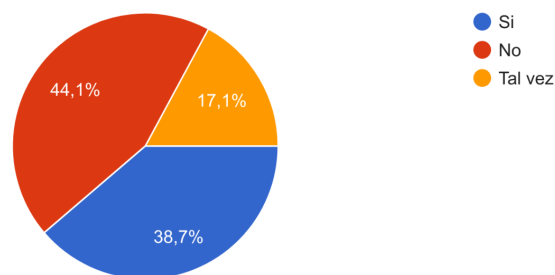


Figura 47. Gráfico de interés de uso de prototipo web con productos multimedia de “Needer By Juan Trujillo y Thomas Pérez”.

Gracias a los datos proporcionados, se puede observar que el 38,7% de las personas encuestadas están interesadas en utilizar una plataforma que ofrezca productos de contenido multimedia por unidad. Esto puede indicar que existe una demanda moderada en el mercado para este tipo de plataforma. Por otro lado, el 44,1% de las respuestas indican que no estarían interesados en utilizar una plataforma de este tipo. Esto puede indicar que existen limitaciones en el mercado, como falta de interés o falta de conocimiento sobre este tipo de plataforma.

Finalmente, el porcentaje restante (9%) indica que pueden estar interesados en utilizar una plataforma de este tipo pero no se ha especificado con certeza. Esto puede indicar que aún hay una cierta incertidumbre en el mercado sobre la demanda de este tipo de plataforma.

En resumen, los datos indican que existe una cierta demanda en el mercado para una plataforma que ofrezca productos de contenido multimedia por unidad, con un 38,7% de las personas encuestadas expresando interés. Sin embargo, también existe una cierta limitación en el mercado, con un 44,1% de las personas encuestadas indicando que no estarían

interesadas en utilizar una plataforma de este tipo. El porcentaje restante puede indicar una cierta incertidumbre en el mercado sobre la demanda de este tipo de plataforma.

6.1.1.2.7 Conclusiones

Gracias al estudio cuantitativo realizado, se puede concluir que, para que la empresa y sus productos tengan una viabilidad en el mercado local, deben estar dirigidos mayormente al público masculino joven y adulto. También, se deberán realizar campañas de marketing, que permitan a los ciudadanos de Neiva, conocer los productos ofrecidos por la empresa.

Una conclusión importante a tener en cuenta, es que se deben implementar nuevos métodos de pago, que faciliten a los usuarios la compra de contenido multimedia en el prototipo web, con el fin de aumentar la cantidad de usuarios que gastan dinero mensualmente en el consumo de contenido multimedia.

Por otro lado, se pudo observar que una cuarta parte de los encuestados no están tan acostumbrados a pasar mucho tiempo en las redes sociales, esto puede reflejar un bajo consumo de contenido multimedia por parte de ellos. Esto ayuda a comprender más los hábitos de consumo que tienen los usuarios frente a ciertos productos o servicios ofrecidos por las aplicaciones.

Por último, se concluye que los usuarios estarían dispuestos a utilizar el prototipo web planteado en el proyecto, mediante el cual se ofrecen productos multimedia. Esto es importante, puesto que permite reconocer la viabilidad frente al mercado local y la viabilidad que tendrán los productos.

6.1.2 Estudio de oferta

Una vez conocidas las necesidades de los usuarios, es importante plantear las semejanzas, diferencias y cualidades de la futura competencia de la empresa. Esto con el fin de construir un prototipo con todas las funcionalidades necesarias para diferenciarse y destacar. Para esto,

se realizará un análisis de la competencia, partiendo desde las funcionalidades al modelo de negocio. Luego, se encontrarán las diferencias y semejanzas, para finalizar con la diferenciación que se espera del producto con respecto a la competencia.

6.1.2.1 Análisis de competencia

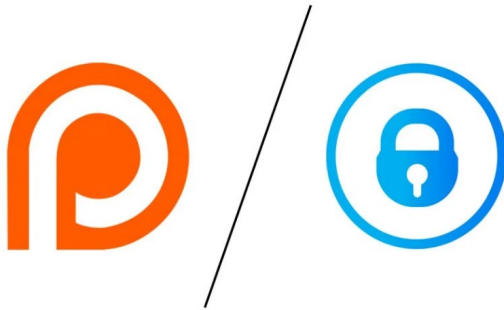


Figura 48. Only Fans y Patreon

Actualmente existe una competencia directa e indirecta con plataformas como **Only Fans, Patreon, Ko-fi, ManyVids, Fansly e Instagram**. El principal contendiente en el mercado es **Patreon**, plataforma que le permite a sus creadores de

contenido monetizar sus creaciones e interactuar con su comunidad a través de la web. También **Only Fans**, donde de igual forma le permite interactuar a los creadores de contenido con su comunidad a partir de una suscripción mensual, aunque en algunos casos hay creadores que ofrecen su contenido de manera gratuita. Estas plataformas tienen como monetización obtener un porcentaje que varía entre el 10% y 20% por cada transacción realizada. A continuación se hablará detalladamente sobre cada una de estas plataformas:

Only Fans: Esta plataforma se enfoca en ofrecer videos y fotos de creadores mediante suscripciones, así como streams, y propinas. Las suscripciones son la principal fuente de ingreso de los creadores de contenido en esta plataforma.

Actualmente cuenta con 1.5 millones de creadores de contenido y 170 millones de usuarios registrados, La forma de monetización de Only Fans es la de tomar un 20% en todas las transacciones realizadas en la plataforma [150], cobrando comisión a través de los siguientes servicios:

- **Suscripción:** Esta consiste en pagar una mensualidad al creador de contenido [151], estos pueden establecer una suscripción entre 5\$ hasta 50\$ dólares. Permitiéndole a

los usuarios pagar suscripciones por 1 Mes, 3 Meses o 1 Año, haciendo posible al usuario ver el contenido publicado por el creador de contenido e interactuar con él a través del feed de la red social.

- **Propinas:** Las propinas son mensajes del usuario consumidor al creador de contenido a cambio de dinero. La plataforma le da la libertad de elegir el monto que desea enviar, haciéndolo una forma de donación pero con la posibilidad de mandar un mensaje el cual el consumidor quiere hacer llegar al creador.
- **Streams:** Los streams le permiten a los creadores de contenido realizar directos en vivo donde el creador de contenido interactúa con su comunidad suscrita, además dentro de los streams los consumidores pueden realizar donaciones del monto que deseen [152].
- **Mensajes por dinero:** Es una forma de permitirle al consumidor intercambiar mensajes con el creador de contenido por un precio fijo por mensaje.

Patreon: Es una plataforma que permite a los creadores de contenido recibir apoyo mediante suscripciones. Este servicio da la facilidad a los creadores de contenido de ajustar distintos tipos de niveles en las suscripciones, donde ellos pueden brindar una variedad de beneficios por cada nivel. Entre los contenidos que ofrece Patreon puede ir desde saludos mediante fotos o videos, a contenido ya predefinido por el creador.

Actualmente cuenta con 6 millones de Patrons (usuarios registrados), de los cuales 4 millones se han registrado en los últimos 3 años. Por otro lado, cuenta con 210.000 creadores de contenido de los cuales estos tienen por lo menos 1 seguidor suscrito [153].

Esta plataforma monetiza de diferentes formas, ya sea por hacer uso de la plataforma, comisión por tramitación de pagos, por cobro o por cambio de moneda [154].

El servicio que genera la mayor parte de los ingresos es el primero. La comisión que se le cobra a los creadores de contenido varía según el plan que quiera adquirir el creador a la hora de mantener su perfil [155].

- **Plan Lite:** Incluye tener un perfil para comunidad, herramientas de comunicación y un sistema de edición. Para este plan se hace un cobro por comisión del 5%.
- **Plan Pro:** Este maneja además de lo anterior por un cobro del 8% de ganancias, niveles para los suscriptores, analíticas para el creador, herramienta para creación de descuentos, integraciones con otras aplicaciones y soporte para el consumidor.
- **Plan Premium:** Por último se tiene el plan más alto de los 3, este abarca los otros dos planes y, planes para venta de mercancía, cuentas para moderadores de la página y un manager de patreon dedicado únicamente al perfil.

Ko-fi: Esta plataforma además de permitir las suscripciones, le permite a los creadores de contenido implementar un marketplace donde pueden fijar un precio a su contenido y le da la posibilidad a los usuarios de enviar propinas a sus creadores de contenido favoritos llamados Kofis (haciendo referencia a como si le estuvieran regalando un café a los mismos). Esto da la posibilidad a los usuarios de tener un mayor acercamiento a sus celebridades favoritas.

Ko-fi posee unos 500.000 creadores de contenido. La plataforma obtiene un 0% de comisión entre transacciones para los usuarios que tengan una suscripción nivel Gold y un 5% para los usuarios que usan la plataforma gratis [156].

- **Suscripción:** Esta consiste en pagar una mensualidad al creador de contenido, cada creador de contenido puede establecer un precio de suscripción.
- **Marketplace:** Le permite a los creadores de contenido vender sus productos digitales y físicos en un apartado especial para ellos. No cobra comisión por listar los productos[157].

6.1.2.2 Conclusiones, semejanzas y diferencias

Como plataforma, Needer brindará mayores herramientas para la interacción entre usuarios a través de la red social. De esta forma, se contará con múltiples herramientas como la creación de perfiles y publicaciones, así como la interacción con las mismas. Ya sea a través de likes o comentarios. Garantizando obtener una base de usuarios antes de

monetizar. Posteriormente brindará oportunidades de monetización a través del modelo de negocio por venta unitaria desde el Marketplace, generando mayor accesibilidad a las compras. También se espera a futuro plantear funcionalidades como la venta de contenido personalizado, chat en vivo y cartera de criptomonedas.

Con esto se espera que Needer pueda abarcar distintas necesidades, con la mejor calidad posible, captando múltiples usuarios de distintas plataformas. En el siguiente cuadro comparativo (**Tabla 9**) se describe de manera gráfica las funcionalidades que Needer abarca, solventando la carencia de ellas en distintas plataformas.

	OnlyFans	Patreon	Ko-fi	Needer
Suscripciones	tiene	tiene	tiene	tendrá
Marketplace	no tiene	tiene	tiene	tendrá
Propinas	tiene	no tiene	tiene	tendrá
Streams	tiene	no tiene	no tiene	tendrá

Contenido personalizado	no tiene	no tiene	no tiene	tendrá
Red social	tiene	no tiene	tiene	tendrá

Tabla 9 - Cuadro comparativo entre la competencia.

Según la investigación realizada donde se evalúa la competencia de Needer se llegó a la siguiente conclusión:

Needer será una plataforma web, que será también accesible y responsiva para los dispositivos móviles al igual que sus competidores, para abarcar el mayor público posible. Uno de los objetivos para ser visibles en la comunidad, es atraer al público de inicial a través de la red social, dado que los usuarios son la base de Needer. Una vez obtenida la base de usuarios, se lanzará el marketplace, sobre el cual se aplicará una estrategia de marketing donde los creadores de contenido que se registren en un tiempo determinado obtendrán el 100% de sus ventas a cambio de publicar la plataforma en redes. Con esto, Needer pretende atraer creadores de contenido que publiciten la plataforma.

6.1.2.3 Diferenciación del producto

Pese a que Needer entrara a un segmento que ya está siendo explotado por plataformas que hacen competencia directa y que estas tienen mayor experiencia en el mercado. Needer pretende entrar al tanque innovando con múltiples herramientas y con su modelo de negocio. Esto debido a que dará la posibilidad de realizar compras por unidad y su servicio girará en torno a una red social, garantizando el libre acceso sin necesidad de suscribirse.

Los servicios que se ofrecerán en la fase 1 del producto serán la implementación de la red social, en la cual los usuarios podrán interactuar con las publicaciones de los creadores de contenido. Sin embargo, la etapa 2 y 3, es decir, el marketplace por unidad y la venta de contenido personalizado se plantearán hasta diseño para su futura implementación. Por último, las últimas 2 etapas serán planteadas a futuro las cuales integrarán el chat en vivo y pago con criptomonedas.

→ **Red social:** En el servicio web se dará la posibilidad de que cada usuario tenga un perfil en el cual tendrá la capacidad de realizar publicaciones, subir fotos o videos, en los que los demás, podrán interactuar ya sea mediante “me gusta” o comentarios. De esta forma se espera que la empresa pueda fidelizar a sus usuarios.

→ **Marketplace:** Este portal tendrá la posibilidad de concretar ventas de contenido multimedia por unidad, entre los consumidores y los creadores de contenido. La principal ventaja de **Needer** será su forma de monetizar, ya que dejará a un lado el modelo de negocio mediante suscripciones para permitir realizar ventas al detal, lo cual dará más flexibilidad de pago a sus usuarios.

→ **Creación de contenido personalizado:** En la aplicación se espera que los creadores de contenido se sientan a gusto y sueltos a la hora de realizar su material, de esta forma su calidad será mayor. Además, con la creación de contenido exclusivo los consumidores podrán sentirse privilegiados a la hora de recibir el producto como ellos esperaban.

→ **Estadísticas del perfil:** A futuro los usuarios podrán ver en tiempo real los ingresos que generan y la cantidad de interacción que reciben con el pasar del tiempo. Para esto, se deberá aplicar técnicas de análisis de datos, de esta forma los needers podrán ver sus estadísticas en gráficas y entender cómo pueden crecer mejor.

→ **Chat en vivo:** En Needer se dará la posibilidad más adelante, de realizar chat en vivo con interacción a través de mensajes y donaciones. Lo que haría que los creadores de contenido puedan expresarse en tiempo real y conversar con sus needers.

→ **Implementación de Cryptos:** Por último, en Needer se desea estar a la vanguardia en métodos de pago. Debido a esto, se espera aplicar en un futuro, pago por medio de criptomonedas utilizando monederos digitales, y con esto darle flexibilidad a los usuarios para hacer sus pagos o recibir las transacciones.

6.1.3 Estudio de comercialización

Dado que todas las plataformas que se han estudiado anteriormente están alojadas en internet se compararon la forma en la que estas ofrecen sus servicios y como lo publicitan.

Only Fans ofrece sus servicios a través de su página web, donde para poder consumir es necesario registrarse y verificar la cuenta. Además, esta empresa ofrece planes de lealtad para que sus propios creadores de contenido hagan publicidad a su página. Así la plataforma garantiza un crecimiento orgánico, publicitando sus servicios de boca en boca.

Por otro lado, **Patreon** es ofrecido a través de internet donde los creadores de contenido tienen perfiles a los cuales los usuarios pueden suscribirse. Asimismo, para publicitar sus servicios brindan herramientas para integrar las redes sociales con la plataforma. De esta forma los creadores de contenido comparten su perfil dentro de sus cuentas vinculadas.

Como último aspecto se evalúa a **Ko-Fi**. A través del marketplace los creadores de contenido ofrecen sus productos por la aplicación web, ya sea físicos o multimedia. Al igual que las plataformas mencionadas anteriormente, son los creadores de contenido quienes publicitan la plataforma, ofreciendo la posibilidad de brindar contenido exclusivo para los consumidores.

Para concluir, se estima que **Needer**, de la misma forma que sus competidores, ofrezca sus servicios a través de la web, y en un futuro desde una aplicación para smartphones. Por otro lado, para publicitar la plataforma se estima que **Needer** haga alianzas con los creadores de contenido, para que los que publiciten la plataforma en el primer año de ser lanzado se les garantice el 100% de sus ganancias. Por último, se evaluará la posibilidad de utilizar servicios publicitarios, permitiendo a **Needer** darse a conocer por otros medios, de manera fácil y rápida.

6.1.4 Desarrollo de la marca

Para llegar al concepto de **Needer** se definieron conceptos como el nombre, el logo, los colores de la empresa, la tipografía, la filosofía y el slogan de la página. Esto fue decidido entre los co-creadores del proyecto, teniendo en cuenta que esta es una startup para público juvenil, con tecnología a la vanguardia.

6.1.4.1 Definición del nombre



El proyecto espera satisfacer a personas con necesidad de usar redes sociales, ya sea para interactuar con personas o celebridades, o también, simplemente necesitadas de contenido digital. Por esta razón, surgió el nombre **Needer** (pronunciado “nider” en español), este proviene del slang en el inglés británico de alguien necesitado, similar a wanted o needed.

Adicionalmente, se escogió el nombre debido a que es simple, de dos sílabas, fácil de recordar y de pronunciar en inglés o español.

6.1.4.2 Diseño del logo



El logo diseñado para la marca de Needer fue creado teniendo en cuenta la simplicidad, con un diseño minimalista con toques de modernidad. En donde la persona que visualiza el logo pueda identificarlo de una forma sencilla.

Además, el símbolo busca evocar mediante los dos triángulos la simetría y perfección, siendo esta última la que se quiere alcanzar en cada línea de código y cada diseño realizado.

Por último, la forma en la cual se dispusieron estos, busca asemejar la forma de la letra N pero sin que el usuario lo note por completo.

6.1.4.3 Análisis cromático



Figura 49. Colores de la marca

El color principal elegido para la marca de Needer es el morado, es un color que transmite sofisticación, lujo, éxito y sabiduría. Generando a las personas un atractivo a la vista con un gusto de elegancia marcando la distinción, con el cual se sientan atraídos.

- **Sofisticacion:** Needer es un proyecto vanguardia, por su idea de red social y su forma de distribución e intermediación de venta de contenido multimedia, debido a que plantea intercambio de contenido por dinero.
- **Lujo:** Las sensaciones al usar Needer deben dar una sentimiento de exclusividad y seguridad.
- **Éxito:** Needer busca ser una de las redes sociales y plataformas de venta de contenido digital más grandes del mundo, compitiendo con empresas ya situadas en el mercado.
- **Sabiduría:** Por último, en Needer se tiene la filosofía de crecimiento personal, tanto para el creador de contenido como para el consumidor. En la aplicación cada quien puede desenvolverse como quiera, publicando lo que desee, estableciendo sus precios, sus metas y se valora el contenido por su unicidad. Además, los consumidores podrán acceder, interactuar y comprar lo que ellos quieran haciendo que su compra e instancia en needer sea inteligente y valga la pena.

6.1.4.4 Analisis tipografico

Inter Source Sans Pro

El tipo de tipografía elegida para representar la marca fue “Inter” y “Source Sans Pro”. Las cuales representan sencillez, siendo agradable a la vista de las personas, mostrando elegancia a la hora de que se presente un texto.

6.1.4.5 Slogan

Needer just what you needed

En needer se busca resolver la necesidad que tienen tanto los consumidores, como los creadores de contenido. Ambos son considerados **needers**.

Los primeros desean, crear o mantener una base de usuarios en su red social, a través de la cual puedan vender su contenido digital a su fandom y grupo de seguidores, generando ingresos, acercándose aún más a su público cumpliendo con la necesidad de mantenerlos enganchados.

Por otro lado, los **needers** consumidores desean interactuar en la red social o convertirse en futuros creadores de contenido. A través de la plataforma buscan la exclusividad en el contenido que adquieren y consumen. Por esta razón Needer lo suple dándoles el marketplace y apartado de red social donde estos podrán interactuar y sentirse más en confianza con su celebridad. En el futuro se espera que **Needer** pueda ofrecerles servicios de chat en vivo, venta de contenido personalizado y donaciones por mensajes.

6.2 Componente Organizacional

Uno de los aspectos importantes en la creación de la propuesta de empresa, fue la definición de todos los componentes organizacionales que la conforman. Gracias a la identificación de estos componentes, se determinó el rumbo que la empresa debe seguir para tener una estabilidad y poder aplicar sus estrategias de mercado.

6.4.1 Misión

El primer componente organizacional que se definió, fue la misión. Para su creación, se tuvieron en cuenta aspectos importantes, como el producto mínimo viable y las necesidades de los usuarios. Además, se analizó el propósito de la organización, sus valores morales y normas de conducta, con el propósito de ser incluidos en la misión. Otro punto importante

que se tuvo en cuenta, fue el porvenir de la empresa, puesto que es importante reconocer las metas que se quieren cumplir a futuro. A continuación se podrá visualizar la misión de la empresa:

“Proveer de un servicio web que ofrezca a los clientes productos multimedia de alta calidad, además de permitir la comunicación entre ellos y garantizar un ambiente de libre de expresión y sin tabúes, permitiendo la creación de comunidades y fortaleciendo las relaciones”.

6.4.2 Visión

Continuando con los componentes organizacionales, se tiene la visión de Needer. Siguiendo los mismos pasos que la Misión, se tuvo en cuenta el propósito de la organización y los valores morales, pero teniendo como referente, una proyección a alcanzar a futuro:

“Para 2030 se espera contar con una versión estable del servicio web, manteniendo y actualizando sus funcionalidades. Además de ayudar a más de 1 millón de usuarios a crear comunidades y ofrecerles una audiencia global”.

6.4.3 Organigrama

Para el desarrollo del organigrama, se analizaron los respectivos encargados de cada área de trabajo, los departamentos que conforman la empresa y las respectivas tareas y responsabilidades de cada departamento. Además, se tomaron como referencias los organigramas de otras empresas, como Onlyfans y Facebook, empresas que tienen estructuras similares y comparten funcionalidades que se implementan en este proyecto:

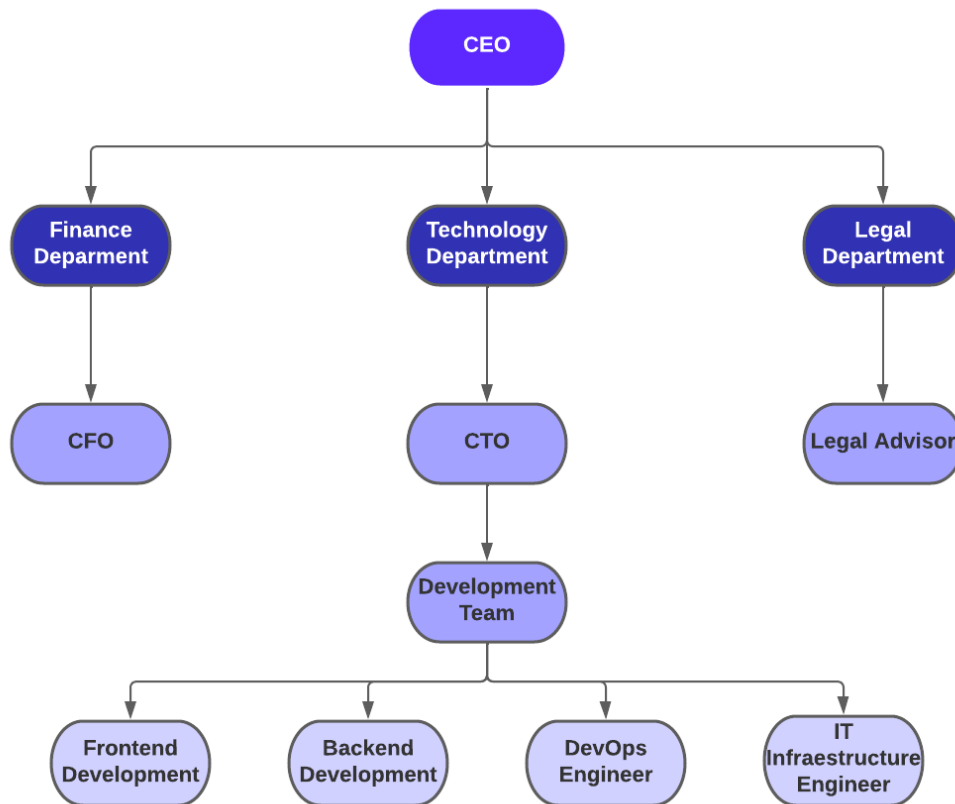


Figura 50. Organigrama Needer.

6.4.4 Manual de funciones

Para el desarrollo de los manuales de funciones, se identificaron los roles que se tendrán dentro de la empresa, gracias al organigrama. Se identificaron los siguientes cargos: CEO, CTO, CFO, Asesor Jurídico, Backend Developer, Frontend Developer, Ingeniero DevOps e Ingeniero de Estructura IT.

Estos manuales fueron definidos en otro documento y , teniendo en cuenta que cada uno de estos cargos se encuentran en diferentes áreas, se identificaron la cantidad de personas requeridas para cada uno de los cargos, sus respectivos jefes, el objetivo general del cargo y las funciones esenciales. A continuación, se podrá visualizar el manual de funciones del CEO, siendo utilizado como referencia para los otros cargos:

<i>Identificación del Cargo</i>	
Nombre del Cargo	CEO (Chief Executive Officer)
Número de Cargos	1

Reporta a	
<i>Objetivo General</i>	
Planificar, dirigir, organizar, coordinar, controlar, analizar, calcular y estimar el trabajo de la empresa. Además de contratar al personal adecuado para el cumplimiento de los procesos planteados para el funcionamiento de la empresa.	
<i>Funciones Esenciales</i>	
<ul style="list-style-type: none"> ● Desarrollar e implementar estrategias para garantizar la rentabilidad del negocio a mediano y largo plazo. ● Supervisar y gestionar las operaciones comerciales, de ventas y marketing alineadas con los objetivos del negocio. ● Analizar la situación financiera de la empresa y gestionar decisiones presupuestarias. ● Dirigir y coordinar las diferentes unidades o departamentos para asegurar su alineación, cumplimiento de proyectos de gestión operativa sobre el día a día. ● Establecer y mantener relaciones efectivas de negocio con Stakeholders y los diferentes actores de sectores públicos y/o privados. 	

Tabla 10. Manual de Funciones CEO.

6.4.5 Requisitos de contratación

Para el desarrollo de los requisitos de contratación, se tuvieron en cuenta los actores identificados en el organigrama y sus respectivas funciones esenciales definidas en los manuales de funciones.

Para el diseño de los requisitos de contratación, se pensó en colocar un salario estimado y las capacidades y condiciones mínimas requeridas de los postulantes al cargo. Es importante mencionar que, el salario inicial será dependiendo de las ganancias de la plataforma, puesto que los puestos iniciales serán cubiertos por los mismos miembros del presente proyecto, a excepción del asesor jurídico.

Por otro lado, cabe resaltar que, estas capacidades y condiciones mínimas varían según el cargo. Los requisitos de contratación realizados, se anexaron en otro documento. Sin

embargo, se inserta como ejemplo los requisitos para el cargo de Chief Executive Officer (Director Ejecutivo de la Empresa):

<i>Identificación del Cargo</i>	
Nombre del Cargo	CEO (Chief Executive Officer)
Salario	De acuerdo a resultados o ganancias
Tipo de Afiliación	Contrato a término indefinido
<i>Requisitos de contratación</i>	
<ul style="list-style-type: none"> ● Título universitario en Administración de Negocios o áreas afines. Grado de Maestría en Administración o áreas relacionadas es preferible. ● 8 años de experiencia ejerciendo funciones como director ejecutivo en compañías multinacionales o nacionales, así como experiencia en juntas directivas, comité ejecutivos o similares. ● Conocimiento y experiencia en desarrollo de nuevos negocios, procesos de fusiones y adquisiciones. ● Comunicación fluida y avanzada con el inglés. Conocimiento avanzado en programas Office. 	

Tabla 11. Requisitos de contratación.

6.4.6 Objetivos estratégicos

Para los objetivos estratégicos, se tuvieron en cuenta la visión, misión y valores corporativos de la empresa. Se lograron identificar 5 objetivos estratégicos, los cuales reflejan los intereses personales de la empresa y se ajustan al perfil de la misma, a continuación se podrán visualizar estos objetivos:

“Ofrecer un espacio que dé a los consumidores y creadores de contenido un ambiente seguro, mediante el cual puedan compartir opiniones y gustos con diferentes personas alrededor del mundo”

“Brindar medios de pago innovadores y seguros, que permitan la venta y consumo fiable de contenido multimedia dentro de la plataforma”

“Posicionar a la marca a nivel global con el fin de ser reconocidos por innovación y libertad de expresión”

“Fidelizar a los clientes creando un estilo de vida sin tabues y con libre opinión ”

“Trabajar utilizando energías limpias para apoyar la lucha contra el cambio climático”

6.4.7 Valores corporativos

Con el fin de trazar una meta a ejecutar de la empresa y definir un camino que la distinga, se extrajeron los valores corporativos. Estos se plantearon en conjunto, dado que se espera estos reflejen la filosofía y cultura de trabajo dentro de la empresa, basados en principios éticos y de innovación. Finalmente, se encontraron los siguientes valores:

- **Calidad e innovación en tiempo record.** *“Nuestro compromiso es ofrecer una plataforma innovadora de calidad en el menor tiempo posible, sin sacrificar excelencia en el proceso”* - Se refiere a que el desarrollo del proyecto debe realizarse lo más pronto posible con el fin de que el público lo conozca, sin sacrificar calidad durante el desarrollo.
- **Equipo autosuficiente y autorregulado.** *“Nuestra metodología de trabajo garantiza libertad a todo el equipo, dado que no existen jerarquías ni tareas preestablecidas para todos”* - Se quiere plantear una forma de trabajar moderna y flexible, donde cada persona es pieza clave del equipo, en el cual todos pueden opinar y decidir.
- **Creatividad.** *“Garantizamos un ambiente sano y seguro con libertad de creatividad, para los clientes y el equipo de trabajo”* - El entorno en Needer se espera pueda generar un ambiente sano y libre de toxicidad, dado que se quiere fidelizar a los clientes por medio de la fraternidad entre usuarios.
- **Salvar al mundo como causa.** *“Nos comprometemos con apoyar en la batalla contra el cambio climático. Fomentaremos el uso de energías limpias y haremos alianzas*

con causas ecológicas” - Needer utilizará energías renovables para sostener sus proyectos e infraestructura. Comprometiéndose en la lucha contra el calentamiento global y fomentando la conciencia ecológica.

- **Nos importas tú como persona.** *“Para nosotros es importante recordar que nos hace humanos. Por ello nos importan tus experiencias, salud y sentimientos”* - Needer espera que tanto clientes como miembros del equipo se sientan queridos, por esto cada persona es importante para la empresa, así como sus experiencias, salud y sentimientos.

6.3 Componente Legal

Para el desarrollo del componente legal, se realizaron dos aspectos importantes ya planteados en el Marco Teórico. El modelo de constitución de la sociedad (incluyendo el proceso de conocer si existen homónimos de **Needer** en Colombia) y el boceto de los términos y condiciones aplicable a los productos que serán ofrecidos por la empresa. El primer elemento se realizó teniendo en cuenta que el tipo de sociedad que se espera constituir es la Sociedad por Acciones Simplificadas (S.A.S), por esta razón se hizo uso de un boceto planteado por la Cámara de Comercio de Cúcuta [158].

6.3.1 Homonimia

El proceso de reconocimiento de homónimo de Needer en Colombia se realizó a través de la página web de la Cámara de Comercio de Bogotá, debido a que se espera inscribir la empresa en el distrito. Para esto, primero se diligenció el nombre o razón social que se desea utilizar, sin el tipo de sociedad (**Figura 51**).

Registro Mercantil - Homonimia nacional

Por identificación

Clase:

No. -

Por nombre o razón social*

Escriba los primeros caracteres del nombre

Búsqueda por palabra clave

Por palabra clave

Registro mercantil

Digite el número de matrícula

Buscar

Figura 51. Búsqueda de Homónimos

Al dar click en el botón buscar, aparecerá una tabla referenciando todos los homónimos encontrados a partir del nombre diligenciado (Figura 52). En este caso, no se encontró ninguno para la razón social Needer y por lo tanto, podrá utilizarse.

Cámara de Comercio	Matrícula	Razón Social	Organización Jurídica	Ultimo año renovado	Estado
1					

Figura 52. Resultados de Homónimos con respecto a Needer

6.3.2 Modelo de Constitución de la Sociedad

Una vez verificado que **Needer** no tiene Homónimo en Colombia y por lo tanto está razón social está disponible para su uso. El modelo se redactó completando la información esencial a partir de la guía utilizada, diligenciando lo mínimo requerido que debe ser constatado a la hora de registrar la empresa en la Cámara de Comercio.

Cabe resaltar que, algunos datos utilizados de los miembros del proyecto son ficticios, dado que es información sensible que no debe ser compartida en el presente documento. A continuación, se expondrá el documento a entregar cuando se vaya a crear la figura jurídica, con todos los artículos correspondientes:

En la ciudad de BOGOTÁ, DISTRITO CAPITAL, República de Colombia, el 27 de Octubre del año 2022, JUAN DAVID TRUJILLO SUAREZ, mayor de edad, identificado con Cedula de Ciudadanía número 11111111 expedida en NEIVA, de estado civil Soltero domiciliado en la ciudad de BOGOTÁ en la dirección Calle 19 y JOSE THOMAS PEREZ RIVERA mayor de edad identificado con Cedula de Ciudadanía número 22222222 expedida en NEIVA, de estado civil Soltero domiciliado en la ciudad de NEIVA en la dirección Calle 20, quien para todos los efectos se denominarán los constituyentes, mediante el presente escrito manifestamos nuestra voluntad de constituir una Sociedad por Acciones Simplificada, que se regulará conforme lo establecido en la ley y en los siguientes estatutos.

ARTÍCULO 1ro. TIPO O ESPECIE DE SOCIEDAD QUE SE CONSTITUYE

La sociedad que se constituye por medio de este documento es por acciones, del tipo Sociedad por Acciones Simplificada, esta se registrará por las disposiciones contenidas en los presentes estatutos, por las normas que de manera especial regulan esta especie de compañía en Código de Comercio y por las generales que en la anterior normativa rigen para las sociedades, teniendo en cuenta que tanto las especiales como las generales sean compatibles con su calidad de una Sociedad por Acciones Simplificada.

ARTÍCULO 2do. DOMICILIO SOCIAL

La compañía tendrá como domicilio principal la ciudad de BOGOTÁ, pero podrá abrir sucursales o agencias en cualquier parte del territorio nacional, para lo cual se procederá como aparece previsto en las normas legales.

ARTÍCULO 3ro. NOMBRE DE LA SOCIEDAD

La sociedad actuará bajo la denominación social NEEDER S.A.S.

ARTÍCULO 4ro. TÉRMINO DE DURACIÓN

La sociedad tendrá un término de duración INDEFINIDO, pero podrá disolverse anticipadamente cuando sus accionistas así lo decidan.

ARTÍCULO 5to. OBJETO SOCIAL

La sociedad tendrá por objeto, el desarrollo de las siguientes actividades: A) Creación, producción, mantenimiento y diseño de productos de software. B) Ofrecer plataforma de Red social, para la comunicación e interacción entre personas. C) Servir de intermediario entre usuarios mediante la plataforma de software para concretar ventas de contenido multimedia D) Cualquier actividad comercial o civil de forma lícita y todas las demás inherentes al desarrollo del objeto social.

ARTÍCULO 6to. CAPITAL AUTORIZADO, SUSCRITO Y PAGADO

El capital autorizado de la sociedad está expresado en PESOS y es de CUARENTA Y OCHO MILLONES, CIENTO CINCUENTA MIL (\$48'150,000), divididos en MIL (1,000) acciones de igual valor nominal, a razón de CUARENTA Y OCHO MIL CIENTO CINCUENTA PESOS (\$48,150) cada una. Los accionistas constituyentes han suscrito MIL (1,000) por un valor nominal total de CUARENTA Y OCHO MILLONES, CIENTO CINCUENTA MIL (\$48'150,000), capital que se encuentra suscrito y pagado en dinero en su totalidad, La clase de acciones aquí suscrita son de clase ORDINARIAS. Parágrafo.- El capital suscrito podrá aumentarse por cualquiera de los medios que admite la Ley, igualmente, podrá disminuirse con sujeción a los requisitos que la Ley señala, en virtud de la correspondiente reforma estatutaria, aprobada por sus accionistas e inscrita en el registro mercantil:

ACCIONISTAS	NO. DE ACCIONES	CAPITAL	PORCENTAJE (%)
JUAN DAVID TRUJILLO SUÁREZ	500	\$24,075,000.00	50.00%
JOSE THOMAS PEREZ RIVERA	500	\$24,075,000.00	50.00%
TOTAL	1000	\$48,150,000.00	100.00%

Tabla 12. Capital suscrito por accionistas

ARTÍCULO 7mo. CARACTERÍSTICAS DE LAS ACCIONES

Las acciones de la sociedad en que se halla dividido su capital son de clase:

OTORGANTES	TIPO ACCIÓN	CANTIDAD
JUAN DAVID TRUJILLO SUÁREZ	Acción Ordinaria	500
JOSE THOMAS PEREZ RIVERA	Acción Ordinaria	500

Tabla 13. Características de las acciones

ARTÍCULO 8vo. TÍTULOS DE LAS ACCIONES

A cada uno de los accionistas se le expedirá un solo título representativo de sus acciones, a menos que prefieran tener varios por diferentes cantidades parciales del total que le pertenezca. El contenido y las características de los títulos se sujetarán a lo preceptuado en las normas legales correspondientes. Mientras el valor de las acciones no hubiere sido pagado totalmente, la sociedad sólo podrá expedir certificados provisionales. Para hacer una nueva inscripción y expedir el título al adquirente, será menester la previa cancelación de los títulos del tradente.

ARTÍCULO 9no. LIBRO DE REGISTRO DE ACCIONES

La sociedad llevará un libro de registro de acciones, previamente registrado en la cámara de comercio correspondiente al domicilio principal de la sociedad, en el cual se anotará el nombre de cada accionista, la cantidad de acciones de su propiedad, el título o títulos con sus respectivos números y fechas de inscripción, las enajenaciones y traspasos, las prendas, usufructos, embargos y demandas judiciales, así como cualquier otro acto sujeto a inscripción según aparezca ordenado en la Ley.

ARTÍCULO 10mo. EMISIÓN DE ACCIONES

Corresponde a los accionistas decidir sobre la emisión de acciones de que disponga la sociedad y que se encuentren en la reserva.

ARTÍCULO 11ro. REGLAMENTO DE EMISIÓN DE ACCIONES:

Corresponde a los accionistas constituyentes expedir el reglamento aplicable a la suscripción de las acciones reservadas, ordinarias y de capital emitidas por la sociedad.

ARTÍCULO 12do. GERENCIA

La representación legal de la sociedad y la gestión de los negocios sociales estarán a cargo de un GERENTE, a su vez la sociedad podrá nombrar un SUBGERENTE, quien reemplazará al gerente en sus ausencias temporales y absolutas contando con las mismas atribuciones que el gerente cuando éste entre a reemplazarlo.

ARTÍCULO 13ro. FACULTADES DEL GERENTE

El gerente está facultado para ejecutar, a nombre de la sociedad, todos los actos y contratos relacionados directamente con el objeto de la sociedad, sin límite de cuantía. Serán funciones específicas del cargo, las siguientes: a) Constituir, para propósitos concretos, los apoderados especiales que considere necesarios para representar judicial o extrajudicialmente a la sociedad. b) Cuidar de la recaudación e inversión de los fondos sociales. c) Organizar adecuadamente los sistemas requeridos para la contabilización, pagos y demás operaciones de la sociedad. d) Velar por el cumplimiento oportuno de todas las obligaciones de la sociedad en materia impositiva. e) Certificar conjuntamente con el contador de la compañía los estados financieros en el caso de ser dicha certificación exigida por las normas legales. f) Designar las personas que van a prestar servicios a la sociedad y para el efecto celebrar los contratos que de acuerdo a las circunstancias sean convenientes; además, fijará las remuneraciones correspondientes, dentro de los límites establecidos en el presupuesto anual de ingresos y egresos. g) Celebrar los actos y contratos comprendidos en el objeto social de la compañía y necesarios para que esta desarrolle plenamente los fines para los cuales ha sido constituida. h) Cumplir las demás funciones que le correspondan según lo previsto en las normas legales y en estos estatutos. Parágrafo.- El gerente queda facultado para celebrar actos y contratos, en desarrollo del objeto de la sociedad, con entidades públicas, privadas y mixtas.

ARTÍCULO 14to. RESERVAS

La sociedad constituirá una reserva legal que ascenderá por lo menos al cincuenta por ciento (50%) del capital suscrito y se formará con el diez por ciento (10%) de las utilidades líquidas de cada ejercicio. Los accionistas podrán decidir, además, la constitución de reservas

voluntarias, siempre que las mismas sean necesarias y convenientes para compañía, tengan una destinación específica y cumplan las demás exigencias legales.

ARTÍCULO 15to. UTILIDADES

No habrá lugar a la distribución de utilidades sino con base en los estados financieros de fin de ejercicio, aprobados por sus accionistas, aprobación que se presume por el hecho de la certificación mientras ocupe el cargo de gerente. Tampoco podrán distribuirse utilidades mientras no se hayan enjugado las pérdidas de ejercicios anteriores que afecten el capital, entendiéndose que las pérdidas afectan el capital cuando a consecuencia de las mismas se reduzca el patrimonio neto por debajo del monto del capital suscrito. Las utilidades de cada ejercicio social, establecidas conforme a los estados financieros aprobados con los que estén de acuerdo sus accionistas, se distribuirán con arreglo a las disposiciones siguientes y a lo que prescriban las normas legales. 1.- El diez por ciento (10%) de las utilidades líquidas después de impuestos se llevará a la reserva legal, hasta concurrencia del cincuenta por ciento (50%), por lo menos, del capital suscrito. Una vez se haya alcanzado este límite quedará a decisión de los accionistas continuar con el incremento de la reserva, pero si disminuyera será obligatorio apropiarse el diez por ciento (10%) de las utilidades líquidas hasta cuando dicha reserva llegue nuevamente al límite fijado. 2.- Efectuada la apropiación para la reserva legal se harán las apropiaciones para las demás reservas que, con los requisitos exigidos en la Ley, decidan sus accionistas. Estas reservas tendrán destinación específica y clara, serán obligatorias para el ejercicio en el cual se hagan, y el cambio de destinación o su distribución posterior sólo podrán autorizarse por los accionistas. 3.- Si hubiere pérdidas de ejercicios anteriores, no enjugadas que afecten el capital, las utilidades se aplicarán a la cancelación de tales pérdidas antes de cualquier apropiación para reservas legales, voluntarias u ocasionales. 4.- Las apropiaciones para la creación o incremento de reservas voluntarias u ocasionales, deberán ser aprobadas por sus accionistas. 5.- El remanente de las utilidades, después de efectuadas las apropiaciones para reserva legal y para reservas voluntarias u ocasionales, se destinarán al pago del dividendo de cada accionista.

ARTÍCULO 16to. DISOLUCIÓN DE LA SOCIEDAD

La sociedad se disolverá por decisión de sus accionistas o cuando se presente alguna de las causales previstas en la Ley y compatibles con la Sociedad por Acciones Simplificada constituida por medio de este documento. La disolución, de acuerdo a causal presentada, podrá evitarse con sometimiento a lo prescrito en la normativa comercial para el efecto.

ARTÍCULO 17mo. LIQUIDADOR

El liquidador y su suplente serán designados por los accionistas y esta designación, una vez ellos manifiesten la aceptación, se llevará a cabo su inscripción en el registro mercantil.

ARTÍCULO 18vo. PROCESO DE LIQUIDACIÓN

Cuando se trate del denominado proceso de liquidación privada o voluntaria, se seguirán las normas que aparecen en los artículos 218 y siguientes del Código de Comercio.

ARTÍCULO 19no. TRANSITORIO - NOMBRAMIENTOS

El cargo de GERENTE será ocupado por, JUAN DAVID TRUJILLO SUAREZ, identificado con Cedula de Ciudadanía número 11111111 de NEIVA y se nombra como SUBGERENTE a JOSE THOMAS PEREZ RIVERA, con Cedula de Ciudadanía número 22222222 de NEIVA. Ambas personas estando presentes en este acto constitutivo, aceptan los cargos para los cuales fueron designados.

ARTÍCULO 20mo. DECLARACIÓN DEL SOCIO CONSTITUYENTE

Los constituyentes de la sociedad NEEDER S.A.S., identificado así: JUAN DAVID TRUJILLO SUAREZ, identificado con Cedula de Ciudadanía número 11111111 de NEIVA y JOSE THOMAS PEREZ RIVERA, con Cedula de Ciudadanía número 22222222 de NEIVA; declaran que la sociedad constituida por medio de este documento privado, reúne los requisitos exigidos por la Ley.

Por último en el escrito se anexarían las firmas e identificación de los socios conformando la sociedad.

6.3.3 Políticas de Privacidad y Términos y Condiciones

El proceso de creación de los términos y condiciones es un paso muy importante, puesto que delimita los derechos y deberes de las personas que harán uso de la plataforma. Además, delimita las responsabilidades legales de la empresa y los usuarios. Para la estructuración de los términos se hizo de varias plantillas similares de términos y condiciones, pero con modificaciones adaptadas al proyecto. Esta normativa se realizó, en base a la ley nacional de

protección de datos personales. Se escogió el español como idioma de las leyes, dado que inicialmente la plataforma va a ser usada por personas hispanohablantes.

Además, los documentos fueron verificados por asesores legales con la finalidad de que cuenten con lo necesario para regir los derechos y deberes de la plataforma, así como definir las obligaciones y términos de uso de los usuarios. Esta normativa estará disponible en el prototipo, desde el apartado de Términos y Condiciones y Políticas de Privacidad, también se anexaron ambos documentos. Sin embargo, las normas podrían estar sujetas a cambio según la regulación nacional.

6.4 Componente Técnico

Considerando la iniciativa técnica del presente proyecto, la cual es, desarrollar un prototipo web que fusione componentes de marketplace de contenido multimedia y elementos de red social. Por lo anterior, el proyecto no contará inicialmente con todas las funcionalidades contempladas, puesto que se entregará un Producto Mínimo Viable con el fin de conocer la retroalimentación del público. Como se mencionó anteriormente en el resumen metodológico, se utiliza la metodología Scrum y Kanban como marco de trabajo.

6.4.1 Análisis

La etapa de análisis es la primera en el ciclo de vida del software. En esta se definieron aspectos como las Épicas, el Producto Mínimo Viable, las Historias de usuario y las tareas que se realizarán para entregar el producto.

6.4.1.1 Definición de Epicas

Las épicas son funcionalidades grandes que se desglosan en historias de usuario. De acuerdo a la idea principal del proyecto, surgieron tres épicas a partir de las etapas, los interesados y sus necesidades. En este caso, el proyecto se enfocó en tres actores principales, Usuario, Creador de Contenido y el Moderador. Para desglosar cada una de las épicas, se estructuró en una tabla para representar la información: La descripción del involucrado y su necesidad, una solución o producto a brindar y el valor que genera la solución a realizar.

Nombre Épicas	Descripción del involucrado	Productos	Valor
---------------	-----------------------------	-----------	-------

Usuario (U)	Los usuarios serán todas las personas que están registradas dentro de la plataforma. Podrán realizar publicaciones, comentar las publicaciones, seguir otros perfiles, comprar cualquier contenido que se ofrezca dentro de la plataforma, crear comunidades en torno a sus preferencias, comunicarse con otros usuarios mediante chats, realizar solicitudes de quejas o inconformidades frente a sus compras realizadas.	<p>Apartado de Red Social implementado a través del prototipo web (Responsive) que permita a las personas registrarse e interactuar entre ellas. Los usuarios podrán realizar publicaciones, ver publicaciones, comentar las publicaciones, reaccionar a las publicaciones y seguir perfiles de preferencias. También podrán comunicarse con otros usuarios mediante chats privados.</p> <p>Se ofrecerá un apartado de Marketplace de contenido multimedia dentro de un prototipo web (Responsive), que ofrezca a los usuarios un amplio listado de contenido multimedia exclusivo, que permita su consumo por unidad. Adicionalmente, el usuario podrá solicitar productos personalizados a los creadores de contenido. El contenido comprado por los usuarios, se verá reflejado en sus perfiles en un apartado especial, en donde podrán encontrar todo lo adquirido.</p>	<p>Los usuarios podrán expresar sus ideas y opiniones dentro de sus perfiles. Además, podrán crear comunidades en torno a ellos y podrán comunicarse con personas de todo el mundo.</p> <p>Ofrece un consumo libre, sin intermediarios y más económico para los usuarios. También brinda una variedad de entretenimiento por parte de los creadores de contenido.</p>
	Creador de Contenido (CD)	<p>El creador de contenido tendrá las mismas funcionalidades del usuario estándar. Pero, deberá verificar su cuenta para poder ser creador de contenido y tener la posibilidad de concretar ventas por unidad, y también podrán ofrecer contenido personalizado. Por otra parte, podrán comunicarse con sus seguidores mediante streaming o chats.</p> <p>Apartado de Marketplace de contenido multimedia dentro de un prototipo web (Responsive), que permita a los creadores de contenido ofrecer su contenido multimedia exclusivo por unidad a los usuarios registrados en la aplicación. También, los creadores de contenido podrán vender productos personalizados por unidad.</p>	<p>Permite al creador de contenido ganar visibilidad, por medio de publicaciones que pueden llegar a ser virales. Además, ofrece la posibilidad de interacción con su comunidad.</p> <p>Garantiza ganancias monetarias a los creadores de contenido a través de la venta de su contenido exclusivo sin la necesidad de intermediarios y la posibilidad de crear una comunidad sólida dentro de la plataforma.</p>

Moderador (M)	El moderador tendrá un apartado diferente dentro del prototipo. Los moderadores revisarán las solicitudes y peticiones enviadas por los usuarios para ser solucionadas.	Apartado dentro del prototipo en el segmento de Marketplace de contenido multimedia, el cual contará con herramientas que permitan la revisión minuciosa de las solicitudes de compras de productos personalizados. Los moderadores podrán también revisar las solicitudes de compras, reembolsos, problemas de pago, quejas del contenido multimedia, quejas con el creador de contenido, etc.	El moderador podrá crear un ambiente fiable para los usuarios a la hora de consumir contenido multimedia exclusivo y personalizado, evitando las estafas, robos o cualquier otro problema que se pueda presentar.
	Además, serán intermediarios durante las ventas de contenido personalizado.	Herramientas dentro del apartado de red social en el prototipo, que permitan el desarrollo de un ambiente sano entre los usuarios. Los moderadores podrán hacer uso de esas herramientas a la hora de revisar las solicitudes generadas por los usuarios, ya sean de quejas, sanciones, baneos, bloqueos, entre otras.	El moderador garantizará una comunidad sana y libre de acoso.

Tabla 14. Epicas

- **Descripción del Involucrado:** Para realizar la descripción del involucrado, se tuvo en cuenta a la persona que tiene las necesidades puntuales, además de estimar y entender su interacción con la plataforma.
- **Producto:** Se planteó la posible solución o soluciones para resolver la necesidad del involucrado en concreto.
- **Valor:** Se definió la posible ganancia que generaría el desarrollo de la solución al usuario.

6.4.1.2 Desarrollo de las Historias de Usuario

Una vez definidas las épicas, se continuó el proceso de análisis mediante la creación de las historias de usuario, a partir de cada una de ellas. Una historia de usuario, es una tarea dividida en procesos, las cuales permiten definir y organizar los requerimientos de un proyecto. Gracias al haber encontrado las épicas y teniendo en cuenta las etapas del proyecto, las cuales se esperan diseñar o desarrollar; Se extrajo una descripción del producto mínimo viable. El MVP se definió como un prototipo web con elementos de red social y componentes de

marketplace de contenido multimedia. Es importante resaltar que, la idea de crear un prototipo inicial en base a la primera etapa, es decir, únicamente con los elementos de red social, es conocer la retroalimentación de los usuarios y adquirir una base de personas registradas.

En base al análisis del producto mínimo viable y a las funcionalidades encontradas a partir de las épicas, se definieron las historias de usuario de este proyecto. Es por eso que, se determinaron los elementos para su estimación, los cuales se representaron en tablas como se presenta a continuación, a partir del ejemplo con la Historia de Usuario U-2.

EPICA USUARIO												
ID HU	ROL	FUN	FIN	# ESCE	C.A	CON	EVEN	D.D.E	E.D	S.D	RES	S.P
U-2	USUARIO	Ingresar a la plataforma	Con la finalidad de hacer uso de la plataforma	1	Es autenticado en la plataforma	En caso de acceder a través de la cuenta de Google por medio de la API	Se accede a través de la API de google y se verifican los datos	1. Email (Max_length = 254, String: "alfanumérico + @ + dominio + . + subdominio") 2. Password (Max_length = 20, Min_length = 8, string: "alfanumérico (obligatorio) + caracteres_especiales")	Datos verificados (Email, Password) de Google	Se visualiza un mensaje de ingreso satisfactorio a la plataforma	Es autenticado en la plataforma y se le brinda sesión en el sistema	8
				2		En caso de que los datos de acceso sean todos correctos y tener una cuenta activa	Se ingresarán los valores solicitados por la aplicación web correctamente		Se ingresan (Email, Password) correctamente			
				3	No es autenticado	En caso de no tener una cuenta activa	Se ingresarán los valores solicitados por la aplicación web correctamente		Se ingresan (Email, Password) correctamente	Se visualiza un mensaje de cuenta no activa, por favor registrarse		
				4		Tiene cuenta activa, pero ingresa de manera errónea o deja vacía la casilla de contraseña	Se ingresará la Password de manera incorrecta o se deja la casilla vacía		Se ingresa correctamente (Email) e incorrectamente o vacío (Password)	Se visualiza un mensaje de error en las credenciales, por favor ingresar correctamente las credenciales		
				5		Tiene cuenta activa, pero ingresa de manera errónea o deja vacía la casilla de email	Se ingresará la Email de manera incorrecta o se deja la casilla vacía		Se ingresa correctamente (Password) e incorrectamente o vacío (Email)	Se visualiza un mensaje de error en las credenciales, por favor ingresar correctamente las credenciales		

Tabla 15. HU U-2

La tabla 15 presenta de forma ordenada la Historia de Usuario U-2 y sus elementos:

- **Identificador:** Se extrajo este elemento de la inicial de la épica a la que pertenece, junto a un código único que identifica a la historia de usuario. En este caso, la **U** representa a la épica **Usuario** y el **#2** representa el código único de la historia de usuario, generando el identificador **U-2**.
- **Rol:** Este elemento representa la persona que usará la funcionalidad generada a partir de la historia de usuario. En este caso, se utilizó el mismo nombre del involucrado de la épica, para definir el rol de esta historia de usuario.
- **Funcionalidad:** Se describió de manera corta la función que suplirá la historia de usuario. En este ejemplo, la necesidad es ingresar a la plataforma, por lo que la función sería esta misma o en otras palabras, hacer Log-in.
- **Finalidad:** Este elemento se determinó desde el punto de vista del involucrado, teniendo en cuenta la finalidad de su acción. Para este ejemplo, se obtuvo como finalidad, hacer uso de la plataforma o de ingresar en ella.
- **Escenario:** Se definieron las posibles circunstancias en las cuales el involucrado podría interactuar con la funcionalidad. En este caso, al haber cuatro contextos diferentes, se fijaron cuatro escenarios distintos.
- **Criterio de Aceptación:** Para el criterio de aceptación, se definió el cómo debería comportarse la plataforma bajo circunstancias idóneas, dependiendo de cada contexto y evento.
- **Contexto:** Para el contexto, se previó bajo qué circunstancias se aplica la funcionalidad. En este ejemplo, para el caso del Log-in, se determinó que el usuario podría ingresar a través de la API o utilizando el formulario de ingreso de la plataforma. Para este último podrían ingresar los datos de forma correcta e incorrecta.
- **Evento:** Se definieron las posibles acciones realizadas por el involucrado, dependiendo de cada contexto.

- **Descripción datos de entrada:** Se definieron los valores que van a tener cada uno de los datos de entrada. Para este ejemplo, se consideraron los valores de los siguientes datos: Email, Password.
- **Entrada de datos:** Se determinaron cuáles iban a ser cada uno de los datos que el usuario debía ingresar dependiendo del evento. Para este ejemplo, se examinaron todos los datos necesarios para que el usuario pueda realizar su respectivo registro.
- **Salida de datos:** Se determinaron cuáles iban a ser cada uno de los datos que el usuario debe recibir, al realizar una acción. Para este ejemplo, se tuvieron en cuenta todos los posibles datos que el usuario puede recibir como resultado de realizar cada uno de los eventos.
- **Resultado:** Se estiman los posibles resultados que se generarían una vez ejecutado el evento.

6.1.4.3 Planeación de Sprints

Ahora bien, una vez desarrolladas las historias de usuario y sus componentes, se estimaron los Sprints a realizar. Sin embargo, antes de estimar la cantidad, orden y la duración de estos Sprints. En primer lugar, se organizaron y priorizaron cada historia de usuario dentro de la plataforma Trello, en el apartado de backlog (**Figura 53**). Con el fin de realizar las historias de usuario con más alta necesidad de primero.

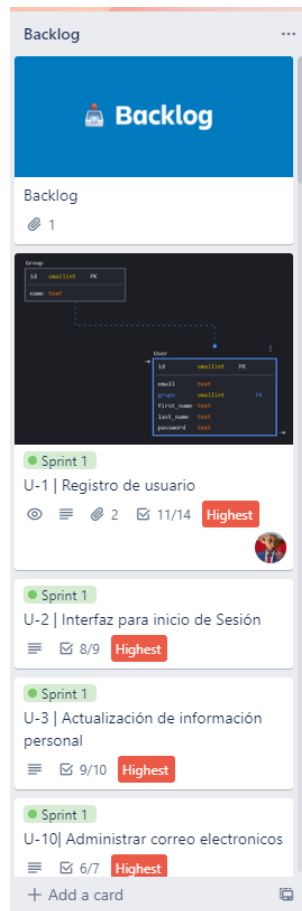


Figura 53. Historias de usuario organizadas en el Backlog de Trello.

Para la priorización, se examinaron las funcionalidades de las historias de usuario que generan más valor y una interacción temprana, entre el usuario y la plataforma. En esta medida, se optó por dividir la priorización en cinco niveles, **Highest, Critical, High, Medium y Low**. Estos niveles de priorización, son identificados con unos colores distintivos y cuentan con unos parámetros de asignación. A continuación, se podrá visualizar la tabla con los niveles de priorización y sus respectivos parámetros:

Highest	La prioridad Highest es asignada a las historias de usuario que tengan definidas las funcionalidades de creación de Usuario en la base de datos, autenticación en la plataforma y la administración de datos personales.
Critical	La prioridad Critical es asignada a las historias de usuario que tengan definidas las funcionalidades relacionadas con la creación y gestión de las publicaciones, además de la visualización de otros perfiles de usuario con sus respectivas publicaciones.

High	La prioridad High es asignada a las historias de usuario que tengan definidas las funcionalidades de interacción con las publicaciones, la posibilidad de seguirse entre usuarios y visualizar dichos seguidores.
Medium	La prioridad Medium es asignada a las historias de usuario que tengan definidas las funcionalidades de vinculación como creador de contenido, creación de contenido multimedia para la venta, canales de comunicación entre usuarios y eliminación de la cuenta. También, están las funcionalidades del carrito de compras, gestión de los métodos de pago, compras y facturación.
Low	La prioridad Low es asignada a las historias de usuario que tengan definidas las funcionalidades relacionadas a la creación y gestión del contenido multimedia personalizado para la venta, y suspensión de contenido. Por otra parte, se encuentran todas las funcionalidades relacionadas al moderador, como la creación y gestión de peticiones, reportes, entre otros.

Tabla 16. Niveles de priorización de las historias de usuario.

Una vez priorizadas las historias de usuario, se estimó la cantidad y duración de los Sprints. En este caso se decidió dividir el desarrollo en Sprints a partir de cada etapa, dada la cantidad de Historias de Usuario. Así mismo, se pactó la duración de ejecución por cada Sprint en 3 semanas. Sin embargo, esto no representa que el tiempo de trabajo debe estar estrictamente ceñido con la duración del proyecto. Adicional a esto, se utilizará este tiempo también, para realizar todas las respectivas etapas de diseño y testing de cada Sprint. En las siguientes tablas, se visualiza la organización de las etapas con sus respectivos Sprints y las historias de usuario por cada Sprint ya priorizadas:

Etapa 1

La etapa inicial del proyecto cuenta con todas las funcionalidades esenciales de la red social, como lo es el registro de usuario, autenticación, creación y administración de publicaciones, galería e interacción con las publicaciones. Esta fase será desarrollada para ser entregada en el prototipo inicial. Está dividida en 3 Sprints presentados a continuación.

SPRINT 1			
#	Nombre	Story Points	Prioridad
U-1	Registro de usuario	15	Highest

U-2	Ingresar a la plataforma	8	Highest
U-3	Actualización de información personal	10	Highest
U-15	Vinculación con Google	8	Highest
U-10	Administrar correo electrónicos	10	Highest
U-17	Cerrar sesión	4	Highest
U-19	Recuperación de contraseña	13	Highest
U-4	Visualización del perfil	7	Critical
U-18	Actualización de contraseña	8	Critical
Total			

Tabla 17. Sprint 1.

En la anterior tabla, se puede observar el **Sprint 1** con todas sus historias de usuario priorizadas y organizadas. Para este Sprint, se buscó realizar las funcionalidades con mayor prioridad, como lo es el registro a la plataforma, por parte del usuario, manejo de sesiones y administración del perfil. Este ciclo busca dar pie al desarrollo de funcionalidades en las cuales el usuario ya debe estar registrado.

SPRINT 2			
#	Nombre	Story Points	Prioridad
U-6	Creación de publicaciones	18	Critical
U-7	Actualización de Publicaciones	15	Critical
U-8	Visualizar detalle de Publicación	12	Critical
U-9	Eliminación de publicación	7	Critical
U-5	Ver perfiles de otros usuarios y su contenido	9	Critical
U-14	Comentar una publicación	15	High
U-16	Eliminación de comentario	6	High
U-13	Dar like a una publicación	6	High
Total			

Tabla 18. Sprint 2 priorizado.

Para el segundo Sprint (**Tabla 18**) se estimó realizar la funcionalidades básicas de la red social, es decir, la creación y administración de publicaciones, ver otros perfiles y la interacción con las publicaciones ya sea con likes o comentarios.

SPRINT 3			
#	Nombre	Story Points	Prioridad
U-20	Búsqueda de contenido en red social	6	High
U-21	Visualización de feed en red social	6	High
U-34	Visualización de contenido multimedia publicado (Galería)	4	High
U-12	Seguir a otros usuarios	12	High
U-39	Visualización de seguidores	6	High
U-40	Visualización de seguidores de otra cuenta	6	High
U-41	Visualización de seguidos propios	6	High
U-42	Visualización de seguidos por otra cuenta	6	High
U-11	Eliminar Cuenta	5	Medium
U-36	Mensajes entre usuarios	21	Medium
Total			

Tabla 19. Sprint 3 priorizado.

Por último para la 1ra etapa, se tiene la **Tabla 19**, la cual desglosa la estructura del **Sprint 3** con sus respectivas historias de usuario. Este último, se planea para integrar las funcionalidades restantes de la red social. Como por ejemplo: La búsqueda de contenido, visualización del feed, seguir a otros usuarios, eliminar la cuenta y los canales de mensajería privados

Etapa 2

Como se mencionó en el alcance del proyecto, las etapas 2 y 3 se planearon ejecutar hasta la fase de diseño, esto con el fin de ser implementadas una vez se despliegue el prototipo inicial. La segunda etapa cuenta con 2 Sprints (Sprint 4 y 5 respectivamente) esenciales, los cuales servirán para construir la base del marketplace.

SPRINT 4			
#	Nombre	Story Points	Prioridad
CD-1	Registro como creador de contenido	21	Medium
CD-2	Registro de datos para el abono a cuenta bancaria	18	Medium
CD-3	Agregar contenido multimedia a la venta	19	Medium
CD-4	Editar contenido multimedia en venta	12	Medium
U-38	Visualización de detalle de producto en venta	12	Medium
Total			

Tabla 20. Sprint 4

La base del marketplace será planteada para el Sprint 4. En este se desarrollará el registro como creador de contenido para que las personas puedan empezar a vender, agregar cuenta bancaria para recibir pagos y el CRUD para el contenido multimedia unitario en venta.

SPRINT 5				
#	Nombre	Story Points	Prioridad	TOTAL
U-22	Ver Marketplace	8	Medium	8
U-23	Filtrar en Marketplace	10	Medium	18
U-44	Visualizar Marketplace de Creador de Contenido	6	Medium	24
U-24	Agregar Método de pago	12	Medium	36
CD-10	Visualización de contenido en venta	4	Medium	40
U-25	Agregar productos a carrito de compra	8	Medium	48
U-43	Administrar Carrito de compras	16	Medium	64
U-26	Pasarela de Pagos	18	Medium	82
U-27	Envío de Factura	6	Medium	88
Total				88

Tabla 21. Sprint 5

Para el 5to Sprint, se ejecutan las funcionalidades que garantizan poder realizar una compra. En este caso sería ver el marketplace, filtrar, agregar métodos de pago, visualización de detalle del producto en venta, carrito de compra, la pasarela de pago y la facturación.

Etapa 3

La tercera y última etapa planeada para diseño permite estimar las herramientas secundarias del apartado de marketplace. Se encuentra dividida en 4 Sprints, siendo el primero el 6to Sprint y el último el 9no. Esta etapa estima las funcionalidades para la venta de contenido personalizado, el apartado para el moderador y la visualización de ganancias en la plataforma.

SPRINT 6			
#	Nombre	Story Points	Prioridad
CD-11	Visualización de ventas realizadas	6	Medium
CD-13	Visualización de detalle venta concretada	6	Medium
U-28	Compra instantánea de contenido	16	Medium
U-29	Visualización de productos adquiridos	11	Medium
CD-6	Agregar producto personalizado a la venta	20	Low
CD-7	Editar Producto Personalizado en venta	13	Low
CD-8	Eliminar producto personalizado en venta	4	Low
Total			

Tabla 22. Sprint 6

El sexto sprint contiene el apartado para el creador de contenido para visualizar las ventas realizadas, también garantiza al usuario poder ver los productos adquiridos y finalmente, da la posibilidad de agregar productos personalizables a la venta.

SPRINT 7			
#	Nombre	Story Points	Prioridad
U-33	Visualización de los detalles de un producto ya adquirido	15	Low
CD-5	Suspender contenido multimedia en venta	12	Low
CD-12	Suspender cuenta propia	4	Low
U-32	Solicitud de contenido personalizado	21	Low
U-35	Mensajes para solicitud de contenido personalizado	21	Low
M-1	Ingresar a la plataforma	8	Low

M-2	Vinculación de cuenta con google	8	Low
Total			

Tabla 23. Sprint 7

El Sprint 7, presentado en la tabla anterior estima las herramientas que permitirán al Creador de contenido suspender un contenido a la venta, también se estima el apartado de mensajes para concretar la venta del contenido personalizado, y por último estima la creación y autenticación del moderador.

Para el penúltimo Sprint, es decir el 8 presentado en la siguiente tabla, se estiman las herramientas para reportar contenido indebido o faltas en la plataforma. Estas serían el contacto con el moderador, visualizar reportes realizados y asignación de moderador para el contenido personalizado

SPRINT 8			
#	Nombre	Story Points	Prioridad
M-5	Actualización de contraseña	8	Low
M-6	Recuperación de contraseña	13	Low
U-30	Contacto con el moderador	18	Low
M-3	Interfaz de lista de reportes y dudas	15	Low
U-31	Visualizar los reportes realizados	11	Low
CD-9	Asignación de moderador para el contenido personalizado	12	Low
Total			

Tabla 24. Sprint 8

Para el último sprint (**Tabla 24**) se da la posibilidad de aprobar el contenido personalizado al moderador, bloquear usuarios, aplicar baneos o faltas por parte del moderador, recibir notificaciones y visualizar las ganancias de la plataforma.

SPRINT 9				
#	Nombre	Story Points	Prioridad	TOTAL
M-7	Aprobación o negación de contenido personalizado	11	Low	11

U-37	Bloquear usuario	15	Low	26
M-4	Cerrar Sesión	4	Low	30
M-8	Bancos o faltas	18	Low	48
M-9	Visualización de usuarios baneados o con faltas	11	Low	59
U-45	Recibir Notificaciones	18	Low	77
M-10	Visualización de ganancias en la plataforma	15	Low	92

Tabla 25. Sprint 9

Por último es importante resaltar que, la etapa 4 y 5 no es estimada en el análisis puesto que fueron planeadas para un futuro lejano.

Una vez estimados los Sprints, fue necesario cuantificar el esfuerzo que tomará realizar cada historia de usuario y Sprint. Por lo tanto se deben prever los story points para medir el trabajo de cada uno de ellos.

6.1.4.4 Story Points y Sprint Burndown

Los Story points, tienen la finalidad de estimar, mediante un número, la magnitud de dificultad de desarrollo de cada historia de usuario, también definen cuántos recursos de tiempo y esfuerzo se necesitan para realizarla. En este caso, por cada historia de usuario dentro de los Sprints, se extrajo un factor de Story Point, teniendo en cuenta la dificultad y los recursos que consumiría. El factor que se tomó fue de 1 a 21, siendo 1 la más fácil y 21 la más difícil.

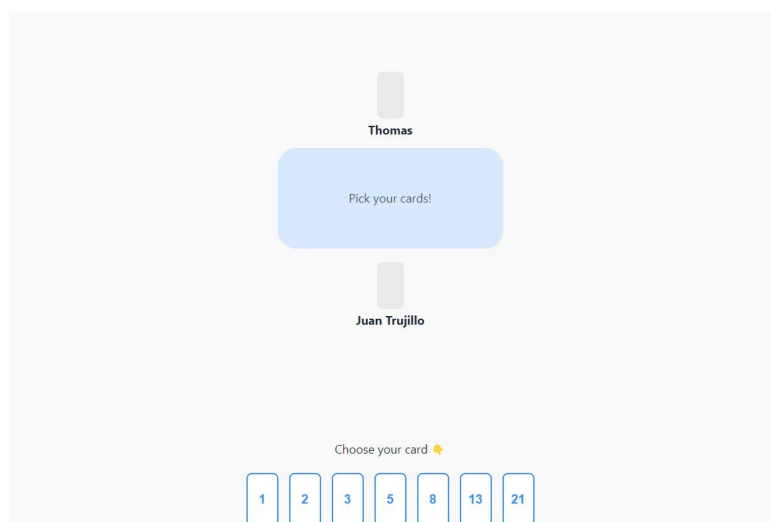


Figura 54. Planning Poker.

Para poder establecer este número para cada Historia de Usuario, se optó por realizar una votación con todos los integrantes del equipo. Dado que es importante conocer las opiniones de cada miembro, y ayuda conocer la capacidad de trabajo de cada uno de los integrantes. Como se puede observar en la **imagen n**, se utilizó una herramienta llamada **Planning Poker**, con la cual, se estimaron todos los Story points de cada una de las historias de usuario.

Para la estimación, se escogía la historia de usuario a analizar y mediante una votación con tarjetas, con los valores entre 1 y 21, se hacía la votación. En algunos casos, se obtuvieron coincidencias en los valores a la hora de votar. Pero en otros casos, se votó con valores totalmente diferentes, por eso se decidió establecer un punto intermedio entre estos valores. A continuación, se observarán las valoraciones de cada historia de usuario en sus respectivos Sprints:

SPRINT 1				
#	Nombre	Story Points	Prioridad	TOTAL
U-1	Registro de usuario	15	Highest	15
U-2	Ingresar a la plataforma	8	Highest	23
U-3	Actualización de información personal	10	Highest	33
U-15	Vinculación con Google	8	Highest	41
U-10	Administrar correo electrónicos	10	Highest	51
U-17	Cerrar sesión	4	Highest	55
U-19	Recuperación de contraseña	13	Highest	68
U-4	Visualización del perfil	7	Critical	75
U-18	Actualización de contraseña	8	Critical	83
Total				83

Tabla 26. Storypoints Sprint 1

SPRINT 2				
#	Nombre	Story Points	Prioridad	TOTAL
U-6	Creación de publicaciones	18	Critical	18

U-7	Actualización de Publicaciones	15	Critical	33
U-8	Visualizar detalle de Publicación	12	Critical	45
U-9	Eliminación de publicación	7	Critical	52
U-5	Ver perfiles de otros usuarios y su contenido	9	Critical	61
U-14	Comentar una publicación	15	High	76
U-16	Eliminación de comentario	6	High	82
U-13	Dar like a una publicación	6	High	88
Total				88

Tabla 27. Storypoints Sprint 2

SPRINT 3				
#	Nombre	Story Points	Prioridad	TOTAL
U-20	Búsqueda de contenido en red social	6	High	6
U-21	Visualización de feed en red social	6	High	12
U-34	Visualización de contenido multimedia publicado (Galería)	4	High	16
U-12	Seguir a otros usuarios	12	High	28
U-39	Visualización de seguidores	6	High	34
U-40	Visualización de seguidores de otra cuenta	6	High	40
U-41	Visualización de seguidos propios	6	High	46
U-42	Visualización de seguidos por otra cuenta	6	High	52
U-11	Eliminar Cuenta	5	Medium	57
U-36	Mensajes entre usuarios	21	Medium	78
Total				78

Tabla 28. Storypoints Sprint 3

Como se puede observar en las anteriores tablas, cada Sprint cuenta con una capacidad planeada de Story points. Este factor explica la disponibilidad del equipo para realizar semanalmente una cantidad de Story Points determinada.

Para el **Sprint 1**, se tuvo una capacidad planeada de 78 Story points, distribuidos en 9 historias de usuario. Con el **Sprint 2**, se estimó una capacidad planeada de 88 Story points, distribuidos en 8 historias de usuario. Y para el **Sprint 3**, se determinó una capacidad planeada de 78 Story points, distribuidos en 10 historias de usuario. Estos datos reflejaron, una capacidad total planeada de 249 Story points distribuidos en 27 historias de usuario para la primera etapa, en otras palabras, para la construcción del prototipo inicial.

Por otro lado, para la segunda y tercera etapa se tuvieron en cuenta de igual forma los story points a pesar que las Historias de Usuario solo se van realizar hasta la etapa de diseño. Sin embargo, se estimaron con la finalidad de que cuando el prototipo esté una vez desplegado se inicie el desarrollo de esos Sprints.

SPRINT 4				
#	Nombre	Story Points	Prioridad	TOTAL
CD-1	Registro como creador de contenido	21	Medium	21
CD-2	Registro de datos para el abono a cuenta bancaria	18	Medium	39
CD-3	Agregar contenido multimedia a la venta	19	Medium	58
CD-4	Editar contenido multimedia en venta	12	Medium	70
U-38	Visualización de detalle de producto en venta	12	Medium	82
Total				82

Tabla 29. Storypoints Sprint 4

SPRINT 5				
#	Nombre	Story Points	Prioridad	TOTAL
U-22	Ver Marketplace	8	Medium	8
U-23	Filtrar en Marketplace	10	Medium	18
U-44	Visualizar Marketplace de Creador de Contenido	6	Medium	24
U-24	Agregar Método de pago	12	Medium	36
CD-10	Visualización de contenido en venta	4	Medium	40

U-25	Agregar productos a carrito de compra	8	Medium	48
U-43	Administrar Carrito de compras	16	Medium	64
U-26	Pasarela de Pagos	18	Medium	82
U-27	Envío de Factura	6	Medium	88
Total				88

Tabla 30. Storypoints Sprint 5

Para la segunda etapa se estimaron que para el **4to Sprint** se realicen 5 historias de usuario con un total de 82 Story Points entre ellas, y para el **5to Sprint** 88 Story Points, teniendo 6 más que el anterior, pero, con 9 Historias de Usuario. Siendo en total 14 historias de usuario para la 2da etapa, con 170 Story Points.

SPRINT 6				
#	Nombre	Story Points	Prioridad	TOTAL
CD-11	Visualización de ventas realizadas	6	Medium	6
CD-13	Visualización de detalle venta concretada	6	Medium	12
U-28	Compra instantánea de contenido	16	Medium	28
U-29	Visualización de productos adquiridos	11	Medium	39
CD-6	Agregar producto personalizado a la venta	20	Low	59
CD-7	Editar Producto Personalizado en venta	13	Low	72
CD-8	Eliminar producto personalizado en venta	4	Low	76
Total				76

Tabla 31. Storypoints Sprint 6

SPRINT 7				
#	Nombre	Story Points	Prioridad	TOTAL
U-33	Visualización de los detalles de un producto ya adquirido	15	Low	15
CD-5	Suspender contenido multimedia en venta	12	Low	27
CD-12	Suspender cuenta propia	4	Low	31
U-32	Solicitud de contenido personalizado	21	Low	52
U-35	Mensajes para solicitud de contenido	21	Low	73

	personalizado			
M-1	Ingresar a la plataforma	8	Low	81
M-2	Vinculacion de cuenta con google	8	Low	89
Total				89

Tabla 32. Storypoints Sprint 7

SPRINT 8				
#	Nombre	Story Points	Prioridad	TOTAL
M-5	Actualizacion de contraseña	8	Low	8
M-6	Recuperacion de contraseña	13	Low	21
U-30	Contacto con el moderador	18	Low	39
M-3	Interfaz de lista de reportes y dudas	15	Low	54
U-31	Visualizar los reportes realizados	11	Low	65
CD-9	Asignación de moderador para el contenido personalizado	12	Low	77
Total				77

Tabla 33. Storypoints Sprint 8

SPRINT 9				
#	Nombre	Story Points	Prioridad	TOTAL
M-7	Aprobación o negación de contenido personalizado	11	Low	11
U-37	Bloquear usuario	15	Low	26
M-4	Cerrar Sesión	4	Low	30
M-8	Baneos o faltas	18	Low	48
M-9	Visualización de usuarios baneados o con faltas	11	Low	59
U-45	Recibir Notificaciones	18	Low	77
M-10	Visualización de ganancias en la plataforma	15	Low	92

Tabla 34. Storypoints Sprint 9

Finalmente, para la última etapa en el **Sprint 6** se encuentran 76 Story Points repartidos en 7 Historias de Usuario. El **Sprint 7** cuenta con 89 Story Points dividido de igual forma en 7 Historias de Usuario. El **Sprint 8 y 9** cuentan ambos con 6 Historias de Usuario y 7 Historias

de Usuario respectivamente, y 77 Story Points para el octavo y 92 para el noveno. De esta forma, la tercera etapa cuenta con 337 Story Points divididos en 27 Historias de Usuario.

Para la representación gráfica del trabajo realizado por semana, en cada Sprint, se creó un gráfico de **Sprint Burndown** junto a una tabla de esfuerzo semanal. Ambas son utilizadas en la metodología Scrum, para poder visualizar el progreso realizado de tareas en un tiempo determinado. El Sprint Burndown se basa en la disponibilidad semanal de Story Points, los cuales son medidos a partir de cuantas horas diarias tiene cada miembro del equipo disponible para trabajar.

SPRINT BURNDOWN - SPRINT 1				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	83	25	28	30
PROGRESO SEMANAL	83	58	30	0

Tabla 35. Sprint Burndown 1

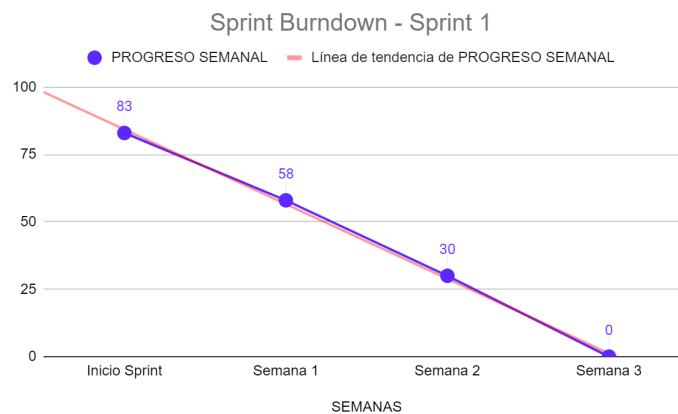


Figura 55. Sprint 1 Burndown

SPRINT BURNDOWN - SPRINT 2				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	88	25	30	33
PROGRESO SEMANAL	88	63	33	0

Tabla 36. Sprint Burndown 2

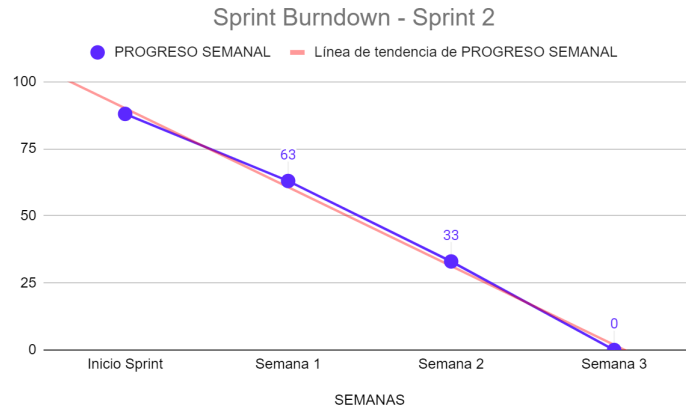


Figura 56. Sprint 2 Burndown

SPRINT BURNDOWN - SPRINT 3				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	78	20	25	33
PROGRESO SEMANAL	78	58	33	0

Tabla 37. Sprint Burndown 3

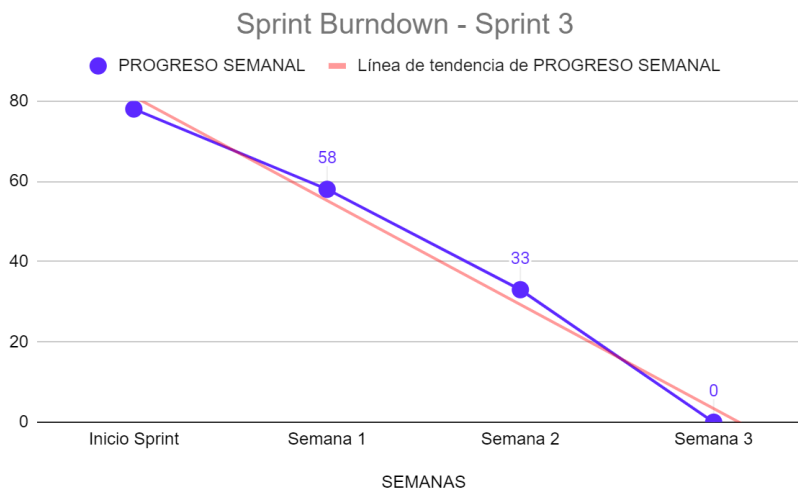


Figura 57. Sprint 3 Burndown

El esfuerzo semanal para la primera etapa se dividió en los 3 Sprints planeados. En este caso en el Sprint 1 Tabla n, conteniendo 83 story points se decidió consumir 25 story points en la primera semana, 28 en la segunda y 30 en la tercera semana, como se puede observar en la gráfica de sprint burndown (**Figura 55**). El segundo Sprint contiene 88 story points, en los cuales a partir del Sprint Burndown representado en la Imagen n, consumirá 25 story points

en la primera semana, 30 en la segunda y 33 en la tercera, para un total de 88 story points para ese Sprint. Para el último Sprint de la etapa inicial (**Tabla 37**), la cual comprende el desarrollo del prototipo se tienen 78 story points, utilizando 20 en la semana inicial, 25 en la semana 2 y 33 en la última semana.

SPRINT BURNDOWN - SPRINT 4				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	82	25	27	30
PROGRESO SEMANAL	82	57	30	0

Tabla 38. Sprint Burndown 4

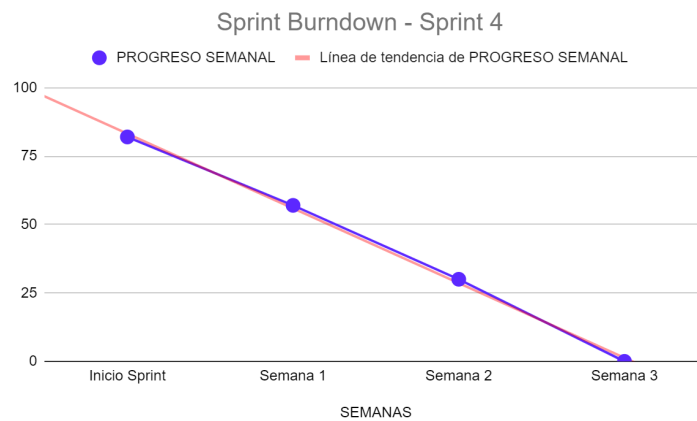


Figura 58. Sprint 4 Burndown

SPRINT BURNDOWN - SPRINT 5				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	88	25	30	33
PROGRESO SEMANAL	88	63	33	0

Tabla 39. Sprint Burndown 5

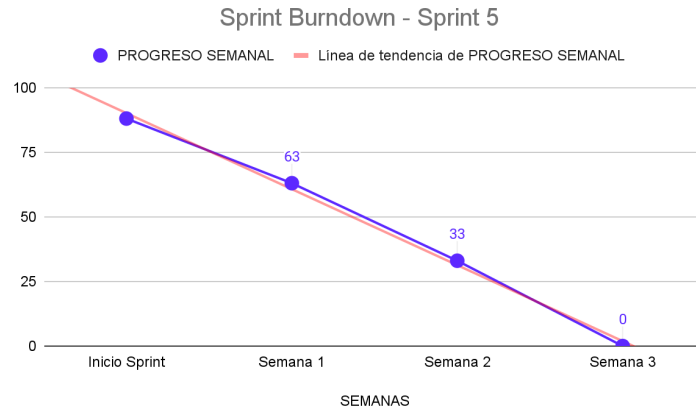


Figura 59. Sprint 5 Burndown

La segunda etapa dividida en los Sprints 4 y 5 a pesar de ser solo de diseño, fue planeada como si se fuera a desarrollar, con el fin de que una vez entregado el prototipo inicie su desarrollo. El Sprint 4 contiene 82 story points los cuales serán divididos en 25 story points en la primera semana, 27 para la segunda y 30 para la última (**Tabla 38**). Por otro lado, el Sprint 5 contiene 88 story points, distribuidos como se observó en la última tabla presentada de esfuerzo semanal (**Tabla 39**).

SPRINT BURNDOWN - SPRINT 6				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	76	20	26	30
PROGRESO SEMANAL	76	56	30	0

Tabla 40. Sprint Burndown 6

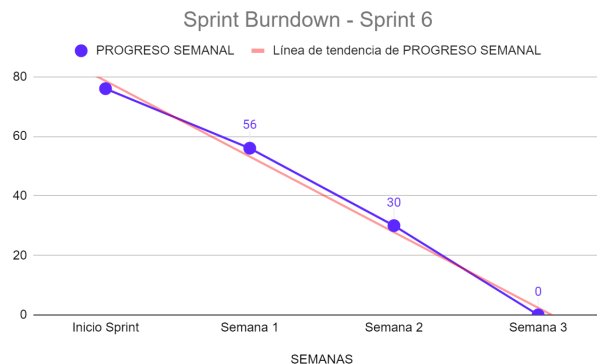


Figura 60. Sprint 6 Burndown

SPRINT BURNDOWN - SPRINT 7				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	89	24	30	35
PROGRESO SEMANAL	89	65	35	0

Tabla 41. Sprint Burndown 7

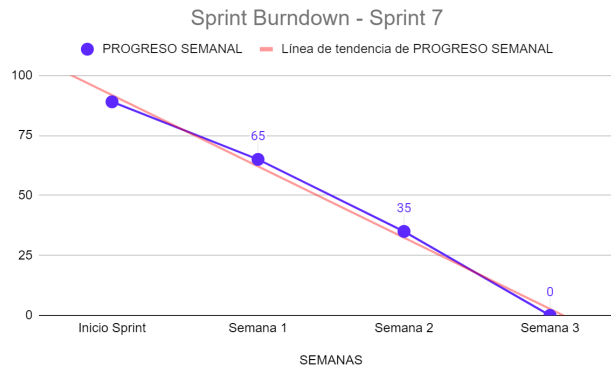


Figura 61. Sprint 7 Burndown

SPRINT BURNDOWN - SPRINT 8				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3
CAPACIDAD PLANEADA	77	20	25	32
PROGRESO SEMANAL	77	57	32	0

Tabla 42. Sprint Burndown 8

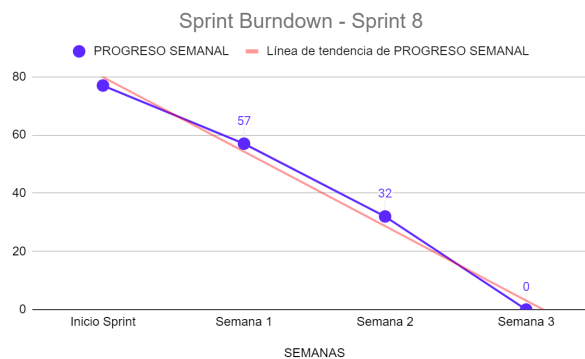


Figura 62. Sprint 8 Burndown

SPRINT 9				
SEMANAS	Inicio Sprint	Semana 1	Semana 2	Semana 3

CAPACIDAD PLANEADA	92	25	30	37
PROGRESO SEMANAL	92	67	37	0

Tabla 43. Sprint Burndown 9

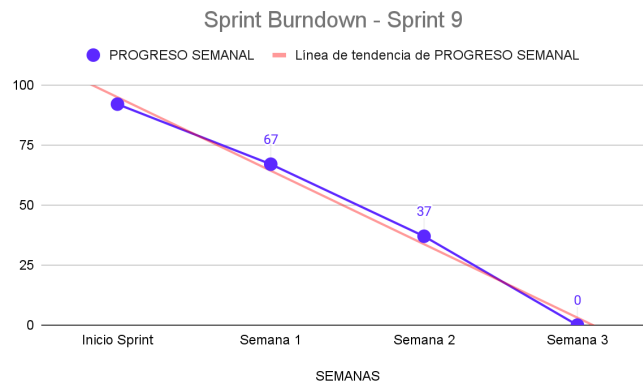


Figura 63. Sprint 9 Burndown

Para la tercera etapa, como se mencionó anteriormente esta etapa al igual que la segunda solo se pensó en diseñar por ahora. El Sprint 6, es decir, el inicial para la etapa, cuenta con 76 story points, los cuales se consumirían según lo expuesto en la tabla n, 20 para la primera semana, 26 para la segunda y 30 para la tercera. El siguiente Sprint contiene un burndown representado en la **Figura (61)** de esfuerzo con 24 story points en la semana 1, 30 en la semana 2 y 35 en la semana 3, para un total de 89 story points. Para el Sprint 8 se tienen 77 story points, distribuidos en 20 para la primera semana, 25 para la segunda y 30 para la tercera. Por último, para el Sprint final se reparte el esfuerzo en 25 story points para la semana 1, 30 para la semana 2 y 37 para la semana 3, teniendo en total 92 story points para el último Sprint.

Story Points por Sprint	
Sprint 1	83
Sprint 2	88
Sprint 3	78
Sprint 4	82
Sprint 5	88
Sprint 6	76
Sprint 7	89
Sprint 8	77

Sprint 9	92
TOTAL	753

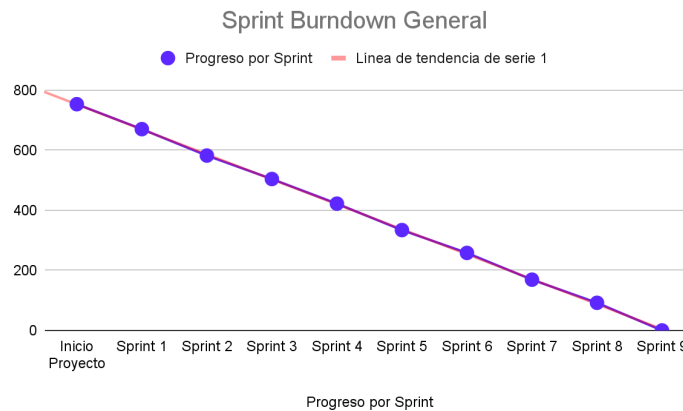


Figura 64. Sprint Burndown General

Finalmente en sumatoria a partir de todos los Sprints se tienen 753 story points, los cuales se distribuyeron en 9 Sprints. El Sprint Burndown General es representado a partir de la Imagen n, se espera que para finales de 2023 o inicios de 2024 se puedan cumplir todos los Sprints correspondientes.

6.1.4.4 Tareas

Como último paso realizado, utilizando la herramienta de Trello, se definieron las tareas de cada una de las historias de usuario, dentro de los Sprints, mediante un checklist a realizar.

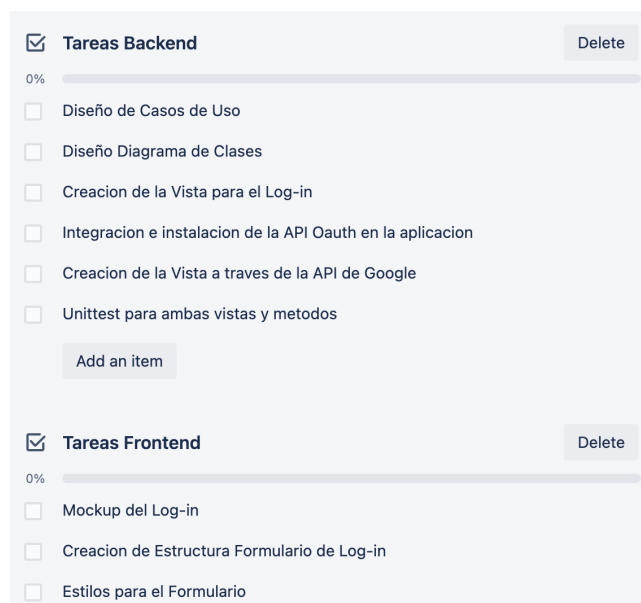


Figura 65. Checklist de una historia de usuario.

En la **Figura (65)**, se puede observar el checklist de la historia de usuario U-2. Para la estructura del checklist, se decidió dividir las tareas en 2 grupos, Frontend y Backend. Para cada división, primero se establecieron las tareas más importantes y complejas, las cuales fueron los diseños de los diagramas, para el backend y los Mockups para el frontend. Posteriormente, se asignaron tareas de desarrollo y testing, principalmente para las funcionalidades. Este proceso fue realizado para las Historias de Usuario que se van a desarrollar, para las que solo se pensaron diseñar, se les asignó sólo tareas de diseño.

6.4.2 Diseño de la aplicación

Una vez realizado el análisis de la aplicación se continuó el proceso con el diseño de los elementos. En este caso se hizo uso de la metodología Orientada a Objetos para definir las funcionalidades lógicas, y los patrones que conforman la metodología del diseño para el diseño frontend. Con el fin de que los desarrolladores comprendan de manera inmediata como ejecutar cada diseño y tarea.

Para realizarlos, se hizo uso de la herramienta **StarUML** para los componentes lógicos, dado que facilita el proceso de creación de los modelos mediante el lenguaje de modelado. Por otro lado, para los elementos visuales y de experiencia de usuario se implementó **Figma** para representar las plantillas a realizar.

El primer paso fue plantear los diseños de la aplicación visualizada de una manera macroscópica, dado que se buscó entender inicialmente el funcionamiento y estructura general de la aplicación, hasta el alcance definido. Para posteriormente diseñar cada elemento de las historias de usuario. Como se sabe, los diseños irán hasta la 3ra etapa, desarrollando solo las historias de usuario de la primera. Sin embargo, los diseños lógicos generales de la aplicación, solo se realizaron teniendo en cuenta la etapa que se va entregar.

6.4.2.1 Diagrama de Caso de Uso

Para el desarrollo del diagrama de casos de uso, se determinaron las principales funcionalidades que el sistema va a ejecutar una vez esté en funcionamiento. Considerando las acciones a las que el usuario accederá con cada una de las herramientas y los permisos que tendrá al realizar cada acción. Para este caso, se tuvo en cuenta el desarrollo de las Etapas

definidas anteriormente, esto quiere decir que, no se realizó ningún diseño de las herramientas de las 4 últimas etapas, puesto que estas no están definidas dentro del producto mínimo viable. Sin embargo, se tienen en cuenta los actores dentro de la aplicación, es decir, el Usuario, el creador de contenido y el moderador.

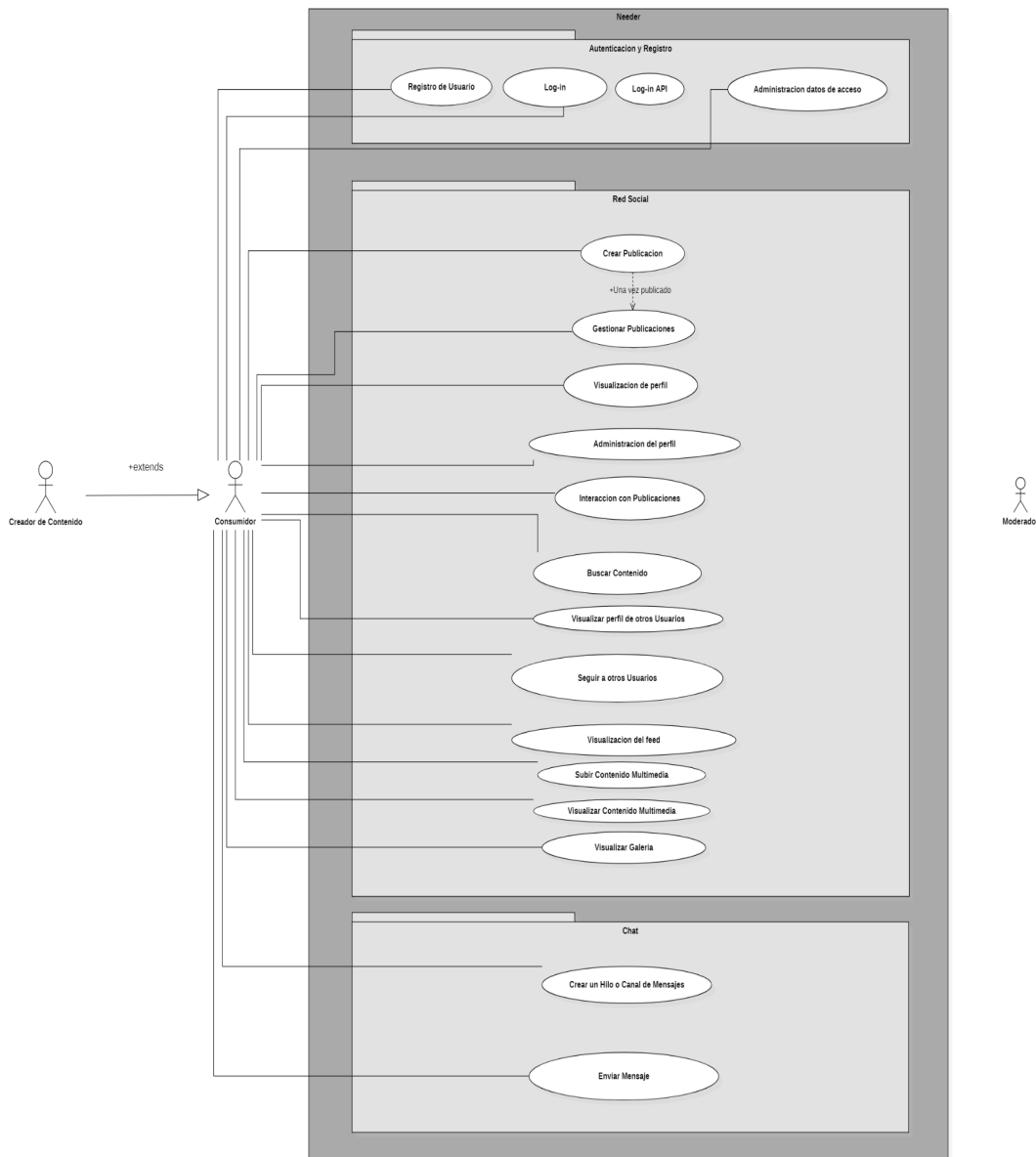


Figura 66. Diagrama de casos de uso Needer.

Empezando con el desarrollo del diagrama de casos de uso, se tuvo en cuenta los tipos de usuarios que harán uso de las funciones del sistema, es decir, el usuario Consumidor, Creador de Contenido y los Moderadores. En base a esto, se asignaron cada una de las funciones que

van a ser ejecutadas por ellos. Estas últimas fueron divididas en grandes módulos diferenciando a qué componente pertenece cada una.

En primera medida se representaron las funcionalidades del consumidor. En el apartado de autenticación y registro, podrá crear una cuenta en la plataforma, autenticarse y actualizar sus datos personales. Por otro lado, desde la app de red social el usuario dispondrá de acceso a crear publicaciones y así mismo gestionarlas (editarlas/listarlas/eliminarlas), también podrá ver a su perfil o el de otras personas, así como buscar elementos e interactuar con las publicaciones ya realizadas (likes/comentarios). En cuanto al chat, el consumidor tendrá la posibilidad de crear un nuevo hilo, en el cual podrá interactuar con otro usuario y a través de ese canal enviar mensajes.

En segundo lugar se plantearon los casos de uso del Creador de Contenido. Es importante resaltar que este actor hereda todas las funcionalidades del Consumidor. En otras palabras, este sería un consumidor con funcionalidades para la venta de contenido multimedia. Las funcionalidades del marketplace y contenido personalizado no se contemplaron puesto que no entran dentro del alcance de desarrollo del proyecto.

El moderador al igual que el creador de contenido, solo se diseña como actor, pero, por el momento, no se contemplan sus funcionalidades.

6.4.2.2 Diagrama de Secuencias

Para el desarrollo del diagrama de secuencias, se definieron procesos que se realizan a la hora de ejecutar una funcionalidad. Este diagrama se desarrolló por clases, puesto que su diseño está basado en el paradigma orientado a objetos.

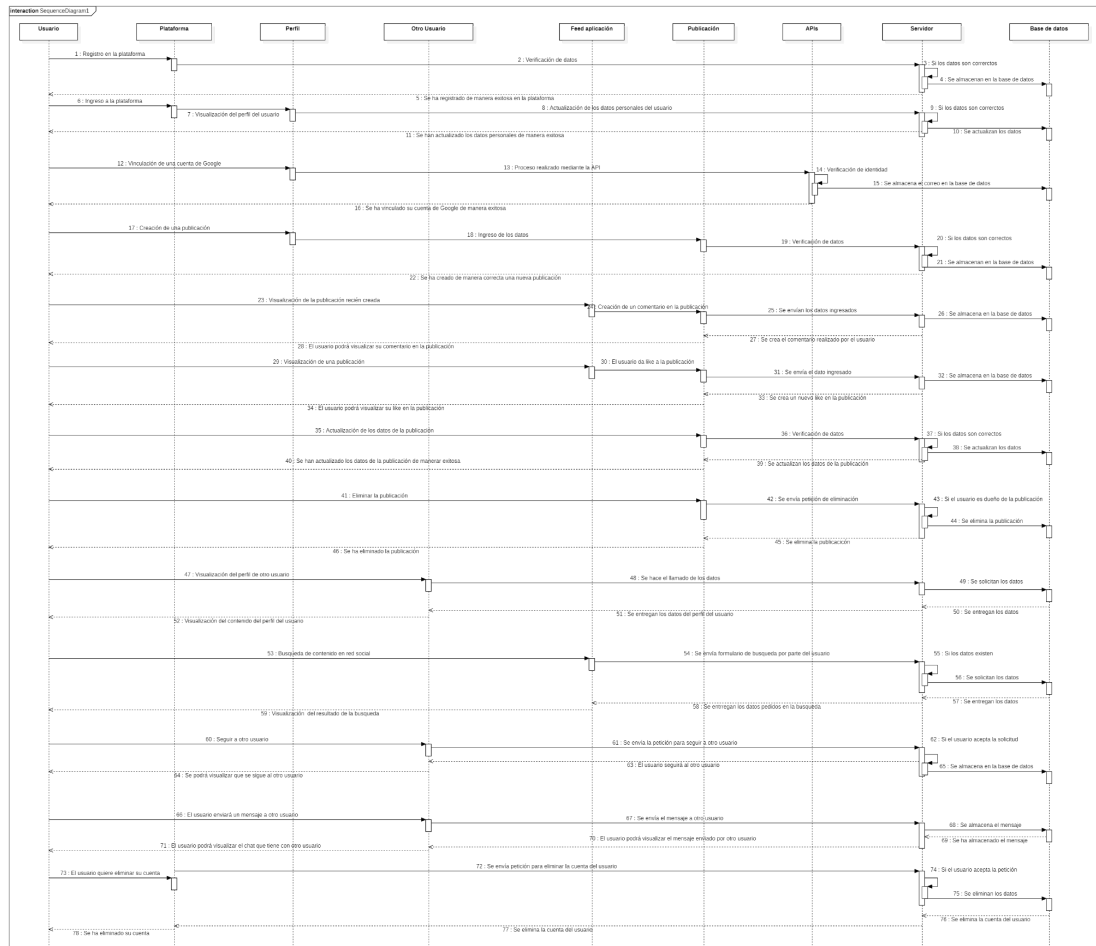


Figura 67. Diagrama de secuencias Needer.

En primera medida, el proceso del diagrama inicia con el registro realizado, por parte del usuario, en la plataforma, para luego autenticar su información mediante el login. Una vez dentro de la plataforma, el usuario podrá actualizar la información de su perfil, estos datos serán actualizados si y sólo si son ingresados de manera correcta. También, se podrán vincular cuentas de Google, mediante la API de Google, la cual se encargará de verificar la identidad del usuario. Una vez registrado en la plataforma, el usuario podrá empezar a utilizar las funcionalidades que Needer le ofrece, como la creación de una publicación. Las publicaciones creadas por el usuario, podrán ser visualizadas en el apartado de perfil, acompañadas de los datos ingresados a la hora de hacer el registro o de actualizar la información.

Los usuarios podrán ingresar al perfil de otro usuario para poder visualizar la información que estos tienen en sus perfiles y las publicaciones que estos han creado. Además, los usuarios podrán interactuar con las publicaciones propias o de otros usuarios, mediante la creación de comentarios o dejando un like en dicha publicación. Las publicaciones podrán ser editadas o

eliminadas directamente por el autor de la publicación. Needer cuenta con un feed de red social, en el cual los usuarios podrán ingresar para observar todas las publicaciones realizadas dentro de la plataforma, también se podrá realizar una búsqueda del contenido publicado hasta el momento.

Otra funcionalidad importante que ofrece Needer, es que los usuarios podrán seguir a otros usuarios de la plataforma, mediante un botón ubicado en el perfil del usuario que quieren seguir. Por otro lado, los usuarios podrán intercambiar mensajes mediante chats, los cuales serán almacenados en la base de datos, permitiendo al usuario poder llevar un registro de la conversación con otros usuarios. Por último, los usuarios podrán eliminar su cuenta de la plataforma, con el fin de dar la oportunidad a esos usuarios que no están interesados en continuar con el uso de esta.

6.4.2.3 Diagrama de Clases

Una vez comprendido el proceso de funcionamiento de la aplicación, se pudo diseñar la estructura de datos. Esta se hizo teniendo en cuenta la etapa a desarrollar y que el **Usuario** será quien ejecute las funciones que generan valor dentro del sistema, representado como el principal actor y centro del diseño (**Figura 68**).

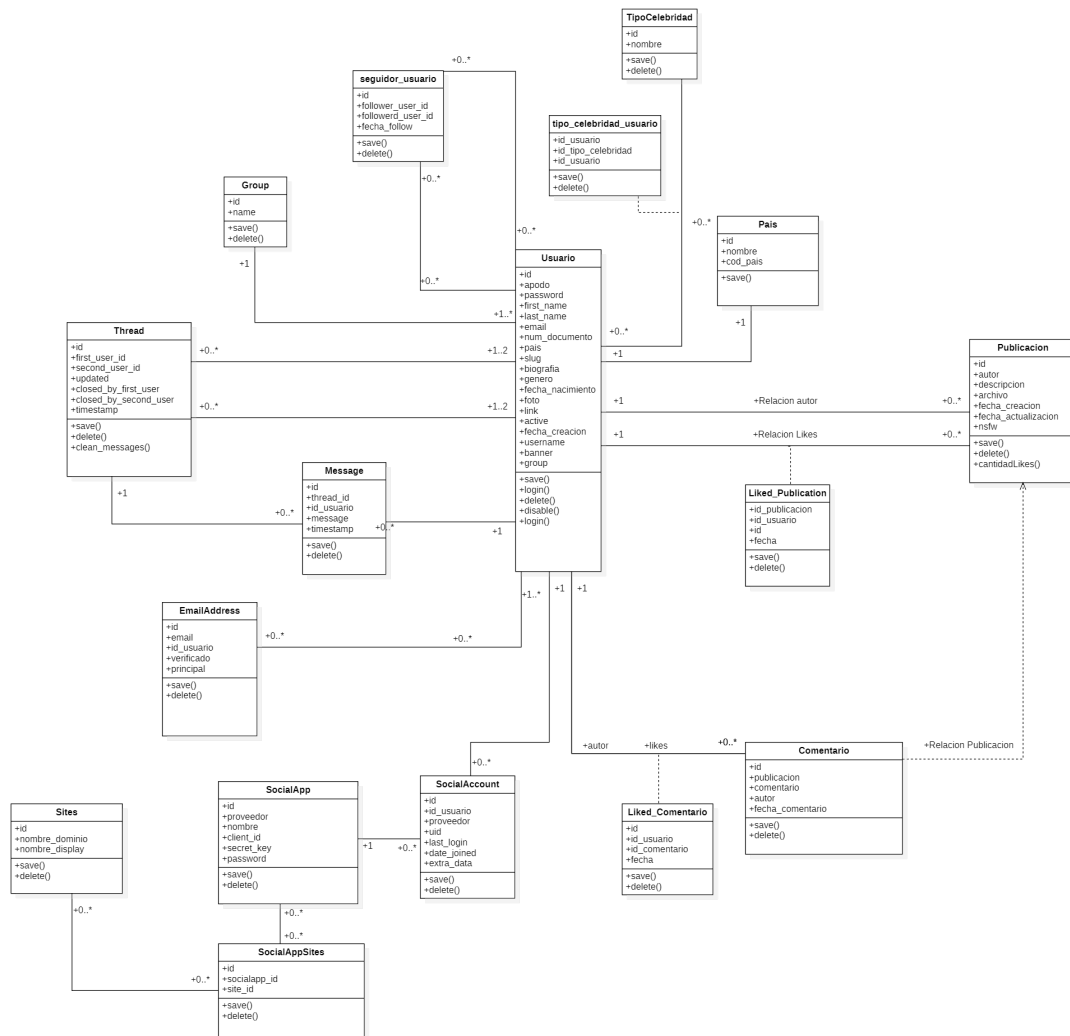


Figura 68. Diagrama de clases Needer.

La figura anterior muestra al nodo central **Usuario** de quien se desprenden las demás clases secundarias y terciarias. Cabe resaltar que los atributos y métodos se especificaron como públicos dado que en Python no existen los modificadores de acceso. Además, las clases planteadas contienen el método `save()` y `delete()` puesto que en Django por el ORM todos los Modelos generados contienen dichos métodos para actualizar, eliminar o crear objetos.

La clase inicial se planteó teniendo en cuenta las necesidades del Usuario, como es tener un identificador personal con el cual los demás usuarios puedan encontrarlo (**apodo y username**), los datos básicos de ingreso que serán el **email** y **contraseña**, un slug para referenciar el perfil del Usuario en las urls, foto y banner para personalizar su perfil, y demás atributos que ayudarán a la experiencia del usuario y a expandir sus funcionalidades. Por otro lado, de los métodos más relevantes definidos es el `login()` para autenticar al Usuario en la

plataforma. El usuario recibe atributos de clases externas como País y grupo. Estas ayudarán a complementar la información del usuario.

Por otro lado se usaron referencias entre dos tablas para representar relaciones de muchos a muchos como lo es la clase tipo_celebridad_usuario, mediante la cual los usuarios tendrán referencia suya y de qué tipo de celebridad son.

Las clases secundarias se dividieron según la característica principal ya sea Red Social o Chat. Para la primera función se definió la clase Publicacion, ella garantizara la interacción en la red social dado que contarán con **descripciones** y **archivos**. A partir de las Publicaciones surgieron los comentarios con la finalidad de que los usuarios opinen en las Publicaciones. Y otro elemento importante son los likes, que serán registros de que usuarios se suscriben y dejan su “me gusta” en cualquier post. Por último en la red social se encuentra una clase que representa a la funcionalidad de seguir un usuario, en ella se almacenará el usuario que se desea seguir y el usuario quien realiza la función, está al igual que el comentario y los likes, representa la relación ManyToMany.

Para la funcionalidad del chat se estructuró la clase **Thread** de la cual se extienden los mensajes. En él se almacenarán los usuarios que pertenecen al hilo, la fecha en que se creó y cuando se actualizó. Por otro lado los mensajes se especificaron con la referencia del hilo, el mensaje y el identificador del usuario que lo envió.

Por último se tienen las clases que serán generadas a partir de las APIs y módulos externos, inicialmente se cuenta con EmailAddress, esta tabla ayudará a crear elementos de correos electrónicos con los cuales los usuarios podrán asociar múltiples correos a sus cuentas. Por otro lado, para el acceso mediante API a la plataforma se tiene la clase Sites, en él se almacenará la información del dominio de Needer, este elemento lo recibe SocialApp, clase que servirá para almacenar la aplicación de autenticación externa, la cual funciona como padre de la clase SocialAccount, que almacenará las cuentas de google de los usuarios.

6.4.2.4 Diseño de Base de Datos

El diseño de base de datos esboza la estructura de persistencia de datos de la aplicación. Para la realización de este se tomaron en cuenta aspectos importantes como independencia lógica,

redundancia, acceso concurrente, integridad entre los datos, etc. Adicionalmente, se representaron las relaciones entre tablas presentes, las cuales son:

- **One-To-One:** Relación que referencia únicamente a 1 elemento de la otra tabla y viceversa. Es representado por una línea con flecha directa, indicando unicidad de elementos.
- **One-To-Many:** Se caracteriza por una entidad que recibe múltiples elementos de otra tabla a un solo objeto. Su diseño es con una línea con múltiples salidas.
- **Many-To-Many:** Es la relación más compleja de todas. Para poder llevarla a cabo se debe crear una tabla que haga de interfaz entre las dos tablas que se desean relacionar. En este caso la tabla enlace recibe múltiples objetos de ambas tablas en una relación one-to-many por ambas puntas.

Las tablas y columnas serán explicadas a continuación en el resumen del diseño representado en la siguiente figura.

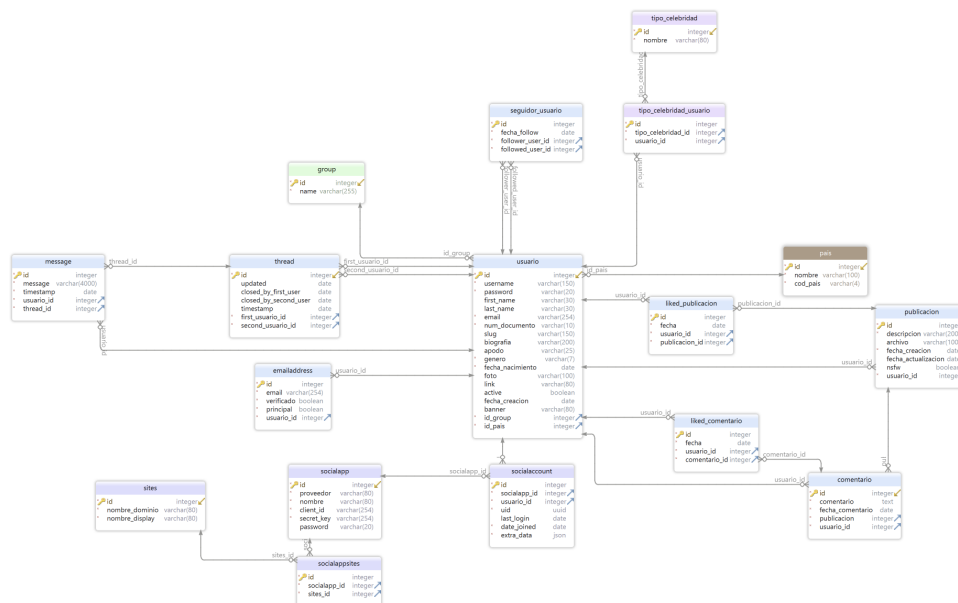


Figura 69. Diseño de base de datos Needer.

Como se expresó en el diseño de Clases, el Usuario será el centro del diagrama y quien utilizará las funcionalidades principales del sistema. Por lo tanto, recibe y envía elementos a diferentes tablas.

A continuación se explica resumidamente como se planteó el bosquejo del diagrama de bases de datos. Para posteriormente presentar el diccionario de datos extraído de los campos divididos por tablas.

- **usuario:** Es la tabla principal del diseño, este cuenta con los datos básicos de acceso (email y contraseña), información para personalizar su perfil como lo es la foto, banner, username y apodo. También presenta columnas para conocer el estado de la cuenta, es decir si está verificado. Información sensible personal como el first_name (nombres), last_name (apellidos) y num_documento. Presenta referencias externas para completar su información como lo es la tabla país y group (para identificar qué tipo de usuario es).
- **país:** Esta tabla almacena información de todos los países del mundo. Es utilizada para referenciar la nacionalidad del usuario. Contiene un id (identificador único), nombre del país y un código único en formato ISO 3166.
- **tipo_celebridad:** Es utilizada para almacenar un identificador y el nombre de tipo celebridad que representa (Streamer, Youtuber, Modelo, etc). Envía datos a tipo_celebridad_usuario para establecer una relación ManyToMany con el usuario a través de esa tabla enlace.
- **tipo_celebridad_usuario:** Representa la tabla enlace o interfaz entre usuario y tipo_celebridad. Su relación con ambas tablas permite que haya una conexión ManyToMany entre estas, creando objetos con pares de elementos de la tabla usuario y tipo_celebridad.
- **group:** Sirve para almacenar los grupos que dividirán los permisos de los usuarios. Guarda el id de cada objeto y un nombre identificando el grupo.

- **publicación:** Representa al componente más importante de la Red Social, contiene un id, descripción, archivo, fecha_creacion para conocer cuando se realizó la publicación y una fecha_actualizacion. Además, presenta un estado de nsfw (“Not Safe For Work”) que define si el post es apto para todo público. Por último, trae referencias externas del autor (Usuario) y se relaciona con la tabla liked_publicacion por medio de una relación ManyToMany para llevar el conteo de “me gusta”.
- **liked_publicacion:** Almacena pares de un objeto publicación y persona que dio like. Permite crear una relación ManyToMany entre ambas tablas y además, guarda una fecha para llevar un histórico de los likes dados por el usuario.
- **comentario:** Es generado por la interacción dentro de la publicación. Contiene una descripción (mensaje a enviar), fecha_comentario, la relación con el Usuario que realizó el comentario (autor) y un conteo de likes que al igual que en las publicaciones, se almacena en la tabla enlace liked_comentario para llevar la relación ManyToMany.
- **liked_comentario:** Permite la relación ManyToMany entre el usuario y el comentario, almacenando el histórico del objeto usuario, el comentario y la fecha en la que se dio like al comentario.
- **seguidores_usuario:** Relaciona a 2 objetos de la misma tabla (usuario), donde un elemento será el usuario que dio follow o seguidor (follower_user_id) y la persona a quien se empezará a seguir (followed+user_id). Finalmente, almacena una fecha de follow con el fin de llevar un histórico de interacciones junto con los likes dentro del perfil de usuario.
- **thread:** Esta tabla es utilizada para crear el hilo de conversación entre dos usuarios. Almacena un id, updated para conocer si el hilo se ha actualizado, closed_by_first_user y closed_by_second_user, para saber si alguno de los dos usuarios ha cerrado el chat con el fin de eliminar los mensajes vinculados al hilo y ahorrar espacio en memoria. Timestamp para almacenar la fecha en que se creó el hilo y por último, el id de ambos usuarios vinculados al thread.

- **message:** Almacena los mensajes enviados por los usuarios. Almacena el identificador único, una fecha de envío del mensaje (timestamp), el id del usuario que referencia a la tabla usuario que envió el mensaje (usuario_id) y finalmente, el id del thread o hilo al que pertenece el mensaje (thread_id).
- **emailaddress:** Se utiliza para guardar múltiples correos asociados a un usuario, con el fin de que cada persona pueda contar con varios correos en su cuenta. Almacena el id, email, estado de verificación del correo (verificado), principal para ajustar el correo como principal y el id del usuario asociando el email a un usuario.
- **sites:** Las siguientes cuatro tablas almacenan información para el manejo de la API de Google Oauth y serán generadas a partir del módulo allauth de Python. La tabla sites almacena la información del dominio que usará el servicio, en este caso needer. Guarda un id único, nombre del dominio y el nombre a mostrar cuando la gente acceda a Google Oauth.
- **socialapp:** La aplicación de red social o social app, almacena la información del proveedor, nombre del proveedor, id del cliente, una clave secreta de la API (en este caso de Google Oauth), y la clave pública.
- **socialappsites:** Guarda que APIs estan almacenadas en el sitio (needer en este caso). Almacena el id, referencia de la aplicación y del sitio.
- **socialaccount:** Por último se almacenan las cuentas de redes sociales, esta tabla será utilizada por los usuarios para almacenar sus cuentas de redes sociales (solo cuentas de gmail en este caso). Guarda el id, la cuenta de red social, referencia del usuario, un uid unico con codigo de la cuenta de red social, último login (last_login), fecha de registro (date_joined) y por último, información adicional almacenada en json (extra_data).

Seguido al resumen de la estructura de las tablas se expone el diccionario de datos encontrado como resultado de analizar las columnas encontradas por tabla.

TABLA USUARIO								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Identificador unico	✓		✓	10		
username	varchar(150)	Nombre de usuario de la persona, minimo 5 caracteres alfanumericos.	✓			mike123		
password	varchar(20)	Contrasena de acceso, debe almacenarse cifrada. Tamano minimo de 8 caracteres alfanumericos				idk12345		
first_name	varchar(30)	Nombres del usuario				Juan		
last_name	varchar(30)	Apellidos del usuario				Trujillo Suarez		
email	varchar(254)	Email de acceso a la plataforma, debe ser verificado antes de almacenarse.	✓			u20172161632@usco.edu.co		
num_documento	varchar(10)	Numero de documento de la persona registrada, debe ser cifrado. Tamano minimo 5 caracteres alfanumericos, contempla identificaciones de todo el mundo.	✓			991245		
slug	varchar(150)	Username formateado para ser usado como url unica. Alfanumerico		✓		mike123		
biografia	varchar(200)	Biografia de la persona. No tiene tamano minimo.		✓		n/a		
apodo	varchar(25)	Apodo utilizado por la persona como referencia publica, tamano minimo de 2 caracteres.		✓		mike		
genero	varchar(7)	Genero basado en 3 opciones (Masculino, Femenino, Otro)	✓			Masculino		
fecha_nacimiento	date/time	Fecha de nacimiento del usuario				07/12/1998		
foto	varchar(100)	Direccion de almacenamiento de la foto subida por el usuario	✓	✓		n/a		
link	varchar(80)	Url de interes almacenado por el usuario		✓		n/a		
verificado	boolean	Estado de la cuenta, True para verificado y False en caso contrario		✓		VERDADE RO		
fecha_creacion	date/time	Fecha registro de la persona				02/04/2022/23:00		
nsfw	boolean	Refiere si la cuenta es apta para todo publico o solo para adultos		✓		VERDADE RO		

falta	int(1)	Numero de faltas cometido por el usuario.		✓		2		
baneo	daterange	Intervalo de tiempo de baneo aplicado al usuario		✓		['02/06/2022-01/01/2024']		
group	int	Id identificando el objeto de la tabla grupo para traer sus atributos.				1	✓	one-to-many
banner	varchar(100)	Link identificando el enlace de almacenamiento de la foto banner. Formatos .jpg, .jpeg, .png.		✓		n/a		
pais	int	Relacion al id de la tabla pais				1	✓	one-to-many

Tabla 44. Diccionario de datos tabla usuario.

TABLA PAIS								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Identificador unico del objeto	✓		✓	23		
nombre	varchar(100)	Nombre del pais almacenado.	✓			Colombia		
cod_pais	varchar(4)	Codigo identificador del pais. Almacenado en formato ISO 3166-1.	✓			245		

Tabla 45. Diccionario datos tabla país.

TABLA GROUP								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	2		
name	varchar(255)	Nombre del grupo	✓			Creador de Contenido		

Tabla 46. Diccionario de datos tabla group.

TABLA TIPO_CELEBRIDAD								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	45		
name	varchar(24)	Nombre descriptivo sobre qué tipo de celebridad representa	✓			Streamer		

Tabla 47. Diccionario de datos tabla tipo_celebridad.

TABLA TIPO_CEBRIDAD_USUARIO								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	45		
usuario	int	Referencia al usuario que ha escogido un tipo celebridad.				6	✓	one-to-many
tipo_celebridad	int	Vincula la tabla tipo_celebridad para definir qué tipo de celebridad es el usuario vinculado anteriormente.				1	✓	one-to-many

Tabla 48. Diccionario de datos tabla tipo_celebridad_usuario.

TABLA SEGUIDOR_USUARIO								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	45		
follower_usuario_id	int	Referencia al objeto usuario que ha dado follow.				8	✓	one-to-many
followed_usuario_id	int	Referencia al usuario que ha sido seguido.				4	✓	one-to-many
fecha_follow	date/time	Fecha y hora de follow.				01/10/2022/ 19:41		

Tabla 49. Diccionario de datos seguidores_usuario.

TABLA PUBLICACION								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	45		
autor	int	Referencia a la tabla usuario, trae el autor de la publicacion.				4	✓	one-to-many
descripcion	varchar(2200)	Texto que el usuario quiere compartir, puede contener cualquier tipo de caracter				n/a		
archivo	varchar(100)	Archivo de imagen, audio o video que el usuario desea compartir.		✓		n/a		
fecha_creacion	date/time	Fecha y hora de creacion de la publicacion				01/10/2022/ 09:41		

fecha_actualizacion	date/time	Fecha y hora de actualizacion de la publicacion, en caso de que sea actualizada		✓		01/10/2022/19:41		
nsfw	boolean	Estado para conocer si la publicacion es apta para todo publico o no				VERDADERO		

Tabla 50. Estructura diccionario de datos tabla publicacion.

TABLA LIKED_PUBLICACION								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos.	✓		✓	45		
id_publicacion	int	Referencia a la publicacion que se desea dar like.				8	✓	one-to-many
id_usuario	int	Vincula al usuario que ha dado like a la publicacion.				3	✓	one-to-many
fecha	date/time	Referencia al usuario que ha sido seguido.				01/8/2022/04:41		

Tabla 51. Diccionario de datos tabla liked_publicacion.

TABLA COMENTARIO								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos	✓		✓	45		
autor	int	Referencia a la tabla usuario, trae el autor de la publicacion.				4	✓	one-to-many
publicacion	int	Conecta al comentario con la publicacion.				80	✓	many-to-one
comentario	varchar(280)	Archivo de imagen, audio o video que el usuario desea compartir.		✓		n/a		
fecha_comentario	date/time	Fecha y hora de creacion de la publicacion				01/10/2022/09:41		

Tabla 52. Estructura diccionario de datos tabla comentario.

TABLA LIKED_COMENTARIO								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	int	Llave primaria de la tabla, identificador de los objetos.	✓		✓	45		
id_usuario	int	Vincula al usuario que ha dado				8	✓	one-to-many

o		like al comentario.						
id_comentario	int	Referencia al comentario que se desea dar like				3	✓	one-to-many
fecha	date/time	Referencia al usuario que ha sido seguido.				01/20/2022/04:41		

Tabla 53. Estructura diccionario de datos liked_comentario.

TABLA THREAD								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, representa el identificador de cada elemento	✓		✓	12		
updated	date	Referencia de la fecha en la que se actualizo el hilo		✓		01/20/2022/04:41		
closed_by_first_user	date	Almacena la fecha en la que uno de los dos usuarios cerro el chat		✓		01/10/2022/09:41		
closed_by_second_user	date	Almacena la fecha en la que uno de los dos usuarios cerro el chat		✓		01/8/2022/04:41		
timestamp	date	Fecha en la que se creo el hilo				01/8/2022/04:41		
first_usuario_id	integer	Referencia a un usuario				3	✓	one-to-many
second_usuario_id	integer	Referencia al segundo usuario				6	✓	one-to-many

Tabla 54. Estructura diccionario de datos thread.

TABLA MESSAGE								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Identificador primario de los objetos	✓		✓	8		
message	varchar(4000)	Texto o mensaje a enviar				n/a		
timestamp	date	Fecha y hora del mensaje				01/8/2022/04:41		
usuario_id	integer	Referencia del usuario que envio el mensaje				3	✓	one-to-many
thread_id	integer	Hilo al que se va a enviar el mensaje				150	✓	one-to-many

Tabla 55. Estructura diccionario de datos message.

TABLA EMAILADDRESS								
--------------------	--	--	--	--	--	--	--	--

NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, identificador de todos los elementos	✓		✓	3		
email	varchar(254)	Email a almacenar	✓			123@gmail.com		
verificado	boolean	Estado de verificación del email		✓		VERDADERO		
principal	boolean	Referencia si el email es el principal de la cuenta		✓		VERDADERO		
usuario_id	integer	Cuenta del usuario a la cual se le vinculara el email				3	✓	one-to-many

Tabla 56. Estructura diccionario de datos emailaddress.

TABLA SITES								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, identificador de todos los elementos	✓		✓	9		
nombre dominio	varchar(80)	Nombre del dominio de la pagina				needer.com		
nombre_display	varchar(80)	Nombre que se mostrara del dominio		✓		needer		

Tabla 57. Estructura diccionario de datos sites.

TABLA SOCIALAPP								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, identificador de todos los elementos	✓		✓	3		
proveedor	varchar(80)	Proveedor de la aplicacion de autenticacion	✓			Google		
nombre	varchar(80)	Nombre de la aplicacion de autenticacion	✓			google		
client_id	varchar(254)	Id del cliente de la API	✓			n/a		
secret_key	varchar(254)	Clave secreta de la API	✓			n/a		
password	varchar(20)	Contraseña de acceso a la API				password12345		

Tabla 58. Estructura diccionario de datos socialapp.

TABLA SOCIALAPPSITES								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, identificador de todos los elementos	✓		✓	3		
socialapp_id	integer	Referencia al elemento de API creado				6	✓	one-to-many
sites_id	integer	Referencia al elemento del sitio				8	✓	one-to-many

Tabla 59. Estructura diccionario de datos socialappsites.

TABLA SOCIALACCOUNT								
NOMBRE CAMPO	TIPO DATO	DESCRIPCIÓN	UNIQUE	NULL	PK	EJEMPLO	FK	RELACIÓN
id	integer	Llave primaria, identificador de todos los elementos de cuentas de red social	✓		✓	3		
socialapp_id	integer	Referencia a la API o aplicacion de autenticacion				4	✓	one-to-many
usuario_id	integer	Referencia al usuario al que se le asociara la cuenta de red social				5	✓	one-to-many
uid	uuid	Identificador unico, traído por la API externa				n/a		
last_login	date	Ultimo login con la cuenta de red social				01/8/2022/04:41		
date_joined	date	Fecha de vinculacion de la cuenta				01/10/2022/09:41		
extra_data	json	Datos externos traídos por la API, contiene los nombres, correos y demas informacion del usuario				n/a		

Tabla 60. Estructura diccionario de datos socialaccount.

6.4.2.5 Arquitectura del sistema

Una vez planteada la estructura de la base de datos se pudo definir la arquitectura y flujo del sistema. Puesto que la aplicación será construida utilizando Django, se decidió aplicar el patrón de arquitectura MVT, este permite la modularidad de las diferentes funcionalidades y fácil acceso a la base de datos a través del ORM.

En la **Figura 70** se puede observar el proceso de interacción entre el cliente, el servidor y las APIs externas que ayudan al funcionamiento de **Needer**. En primera instancia el cliente hace

una petición HTTP al servidor. El cual se encontrará alojado en un servidor con partición Linux específicamente Ubuntu Server gracias a los servicios de **Digital Ocean**.

El servidor busca el endpoint solicitado a través del archivo de configuraciones de URLs. Posteriormente, si encuentra la vista puede comunicarse con archivos externos traídos desde los servidores de Google ya sea en el CDN que contendrá todos los videos, imágenes y audios que suban los Usuarios a la plataforma, o, así como los datos externos contenidos en las APIs Google. Hay que tener en cuenta que para la entrega del prototipo las APIs se aplicarán en versión de prueba y se usará un CDN diferente al planteado en la inversión inicial, mientras se comprueba la funcionalidad de la aplicación.

Por otro lado si el usuario solicita contenido de la base de datos esta query se hará a través de los Modelos, haciendo uso del ORM para comunicarse con la base de datos. La cual estará subida en un servidor de Heroku y se montará en el gestor de base de datos PostgreSQL. Una vez el servidor completa la petición solicitada busca los archivos estáticos vinculados con la vista requerida y la renderiza en el navegador del cliente.

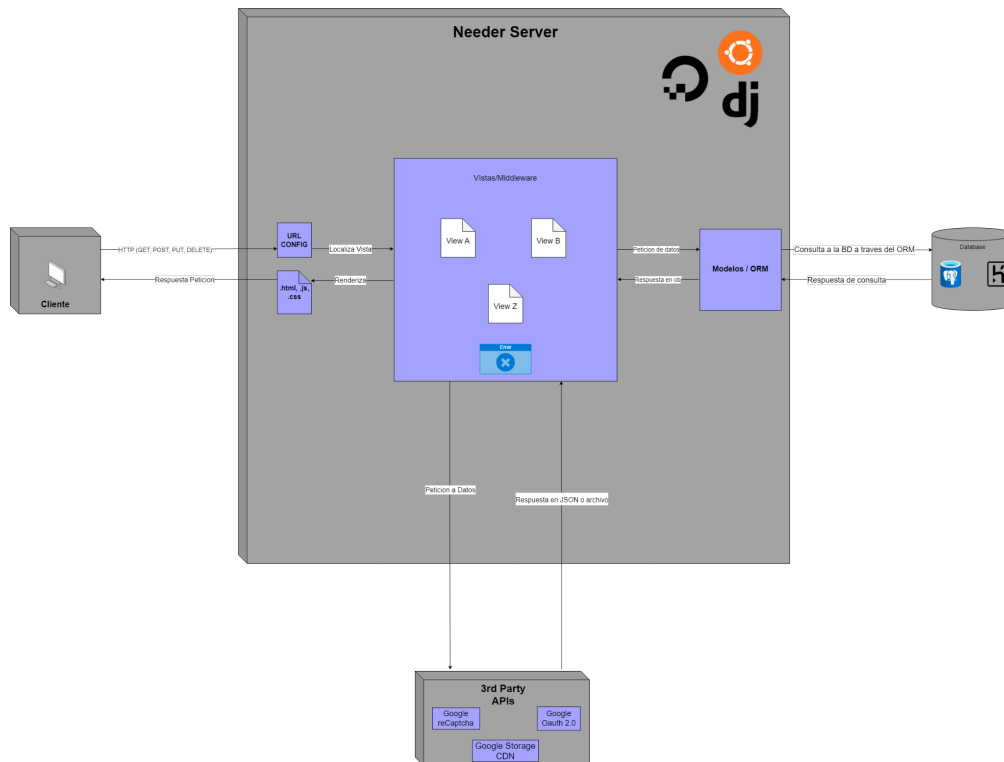


Figura 70. Arquitectura del sistema Needer.

6.4.2.6 Diagrama de despliegue

En cuanto a la estructura física del aplicativo, se presenta una estructura similar que la arquitectura pero con los componentes lógicos y tangibles del flujo. Como se observa en la siguiente (**Figura 71**), el inicio del ciclo se dará por la petición HTTP realizada por el cliente desde un navegador web. Esta petición será recibida por el servidor, y la comunicación se permitirá gracias al servidor NGINX, y a la interfaz WSGI que funcionará de enlace entre el servidor y la aplicación. El aplicativo estará soportado bajo el framework Django en su versión 4.1.7, y el proyecto estará escrito en Python 3.9. Finalmente, el servidor en el cual estará alojado el aplicativo correrá una versión de Ubuntu Server 22.04 de 64bits.

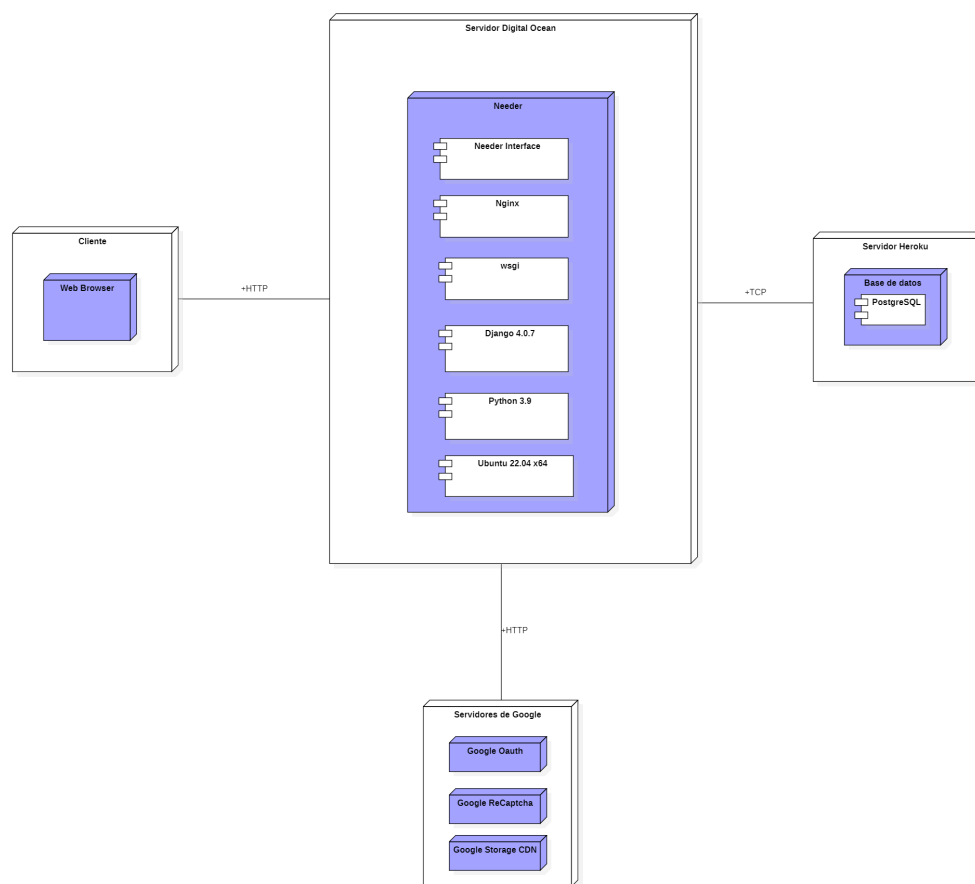


Figura 71. Diagrama de despliegue Needer.

Finalmente, en cuanto a la comunicación con los datos de la base de datos y los servidores de google. Los datos serán traídos gracias al protocolo TCP desde el servidor de heroku, el cual soportará a la base de datos con el SGBD (Sistema Gestor de Base de datos) PostgreSQL. Por otro lado, los datos externos serán solicitados a los servidores de google haciendo uso del protocolo HTTP. El CDN que hace parte de Google, será utilizado para almacenar todos los

archivos multimedia subidos por los usuarios, en este caso se usa uno diferente al planteado en la inversión, dado que se aplicará solo para el primer estado del prototipo.

6.4.2.7 Diseño de Historias de Usuario

Por último, se encuentran los diseños de las historias de usuario. Esta fase fue planteada y ejecutada por Etapas, explicando los diseños correspondientes que se realizaron para el desarrollo de frontend. Cabe resaltar que, se realizaron diseños de backend, pero se dejaron como documentos independientes y no se incluyeron en la explicación de las Etapas. Se consideraron las funcionalidades de cada historia de usuario y se concluyó que se debía tener un diseño exclusivo para cada una, organizando los componentes y facilitando su desarrollo.

Para el desarrollo backend, se utilizó la metodología orientada a objetos, al igual que en los diagramas generales, en donde se basa cada componente en una clase. Por otro lado, para el diseño frontend, se utilizaron los métodos definidos en la metodología de diseño UX/UI, los cuales explican la organización de la información presentada al usuario, la estructuración de la interfaz del usuario y los patrones de diseño que tendrán cada uno de los elementos de la interfaz.

Para el diseño de los Mockups, se utilizó la herramienta Figma, permitiendo desarrollar diseños que corresponden a las necesidades de los usuarios, planteadas en las entrevistas y las encuestas realizadas en el análisis de la demanda. Además, para el diseño de los Mockups, se tomaron en cuenta los resultados analizados en el estudio de comercialización, permitiendo obtener un diseño estandarizado para el producto:

Botones

Los botones tienen los siguientes valores estándares: Alto → 45
Ancho → 180

Color : #5C28FF

Boton

Default

Color : #3131B3

Boton

Hover

Color : #D1D1FF

Boton

Disabled

Tipografías

La tipografía estará definida por las fuentes Inter y Source Sans Pro

Categoría de escala	Fuente	Tipo de fuente	Tamaño(px)
H1	Source Sans Pro	Bold	55
H2	Source Sans Pro	SemiBold	45
H3	Source Sans Pro	Regular	37
H4	Source Sans Pro	Regular	30
H5	Source Sans Pro	Regular	26
H6	Source Sans Pro	Light	22
BUTTON	Inter	ExtraBold	21
Links	Inter	Medium	18
Parrafo	Inter	Regular	15

Figura 72. Diseño estándar botones y tipografías para el producto.

Formulario

Los formularios van a tener los siguientes detalles: Altura → 45 Ancho → 300

Estilos de los cuadros de texto

Cuadro de texto

Default



Focused



James Rodríguez

Filled



Texto desactivado

Disabled



Estilos de los selectores

Seleccione una opción

Default



Opción seleccionada

Filled



Desactivado

Disabled



Estilo de los buscadores


Buscar

Icon Input



James Rodríguez

Icon Input Filled



Buscar

Icon Input Focused



Colores



#5C28FF



#3131B3



#D1D1FF



#A3A3FF

Figura 73. Diseño estándar formulario y colores para el producto.

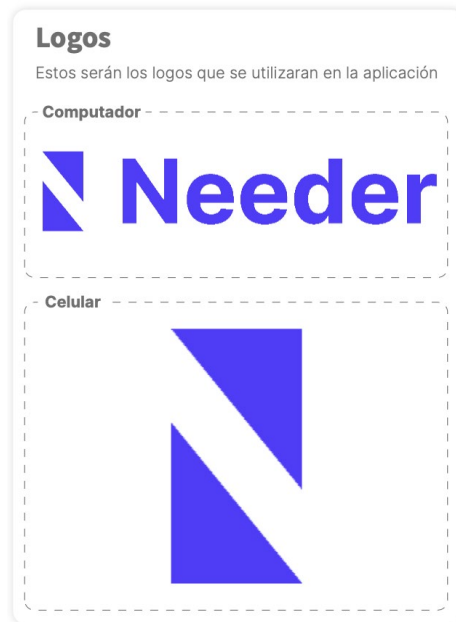


Figura 74. Diseño estándar de logos en diferentes dispositivos para el producto.

Etapa 1

Como se mencionó anteriormente, la etapa 1 está conformada por los Sprints 1, 2 y 3. Los cuales contienen las funciones de la red social, contando con el nivel de mayor prioridad entre las historias de usuario. Teniendo en cuenta esto, se da inicio a la explicación del proceso que el usuario debe llevar a cabo, para hacer uso de dichas funcionalidades. Empezando con realizar el registro a la plataforma, el usuario deberá ingresar sus datos personales, los cuales serán utilizados posteriormente para su autenticación y reconocimiento dentro de la plataforma.

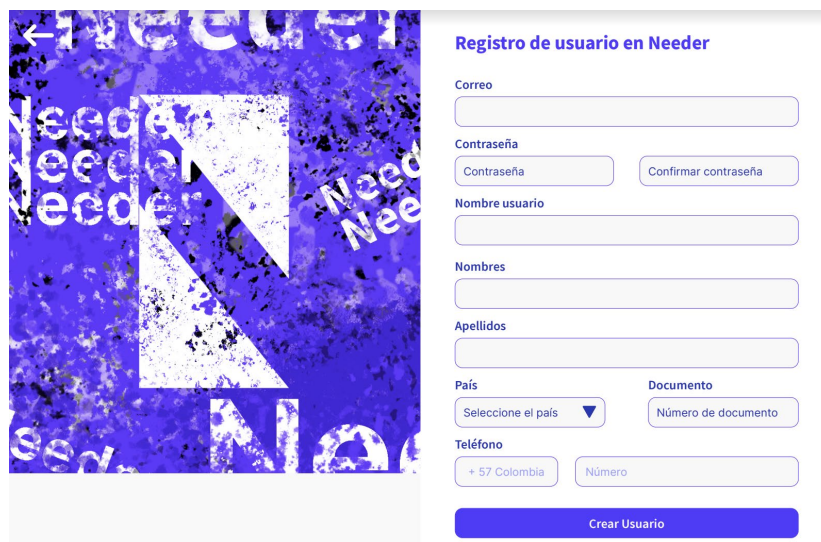


Figura 75. Mockup U-1.

Luego de realizar el registro en la plataforma, el usuario deberá de autenticar sus datos para poder acceder a las funcionalidades de la aplicación. Dirigiéndose al apartado de Login, el usuario deberá ingresar su correo y su contraseña. Otro punto importante acá, es que también se podrá realizar este proceso mediante la API de Google, permitiendo la vinculación directa de su cuenta Google con su perfil de usuario.



The image shows a login form titled "Iniciar sesión en Needer". It features a blue header with a logo and the title. Below the header, there are two input fields: "Correo" and "Contraseña". A blue button labeled "Iniciar Sesión" is positioned below the password field. Underneath the button, there is a link that says "¿Tienes problemas con iniciar sesión?" and a "Regístrate" link. At the bottom, there is a "Inicia sesión con Google" button with the Google logo.

Figura 76. Mockup U-2.

Una vez dentro de la plataforma, el usuario será redirigido inicialmente a los ajustes de su perfil de Needer. Esto con el fin de realizar la respectiva actualización de datos, puesto que su perfil solamente cuenta con los datos ingresados en el formulario de registro. En este apartado, el usuario podrá ingresar una foto de perfil, una foto de banner, una biografía, un apodo a su gusto, un link de alguna red social ajena a él, entre otros datos. Al final de esta sección, se podrá visualizar un pequeño apartado, el cual permitirá al usuario actualizar su correo electrónico o administrar las cuentas registradas en ese perfil.

Needer


Buscar...

Home
Mi perfil
Publicaciones
Marketplace
Mensajes
Panel de Creador
Ajustes

Donación a la pagina
Información
Cerrar Sesión

Ajustes

Perfil Método de pago Actualizar Contraseña Privacidad Avanzados



Datos personales

Nombre	Biografía
<input type="text"/>	<input type="text"/>
Apellidos	
<input type="text"/>	
Nombre Usuario	Pais
<input type="text"/>	<input type="text"/>
Apodo	Link
<input type="text"/>	<input type="text"/>
Fecha de nacimiento	Genero
<input type="text"/>	<input type="text"/>

Tipo celebridad (Seleccione Máx. 5)

<input type="checkbox"/> Artista	<input type="checkbox"/> Cantante	<input type="checkbox"/> Cosplayer	<input type="checkbox"/> Creador de Videos
<input type="checkbox"/> Deportista	<input type="checkbox"/> Modelo	<input type="checkbox"/> Streamer	<input type="checkbox"/> Webcamer


Foto

Banner

Datos de acceso

Email Principal

Redes asociadas



Google

La cuenta se encuentra vinculada con Google
cuenta123@gmail.com

Figura 77. Mockup U-3.

Todos los cambios realizados por el usuario en los ajustes del perfil, podrán ser visualizados en el apartado de perfil. Este apartado cuenta con una pequeña sección, la cual permite la creación de publicaciones. Una vez se haga clic en esta sección, el usuario será enviado al panel de creación de las publicaciones.

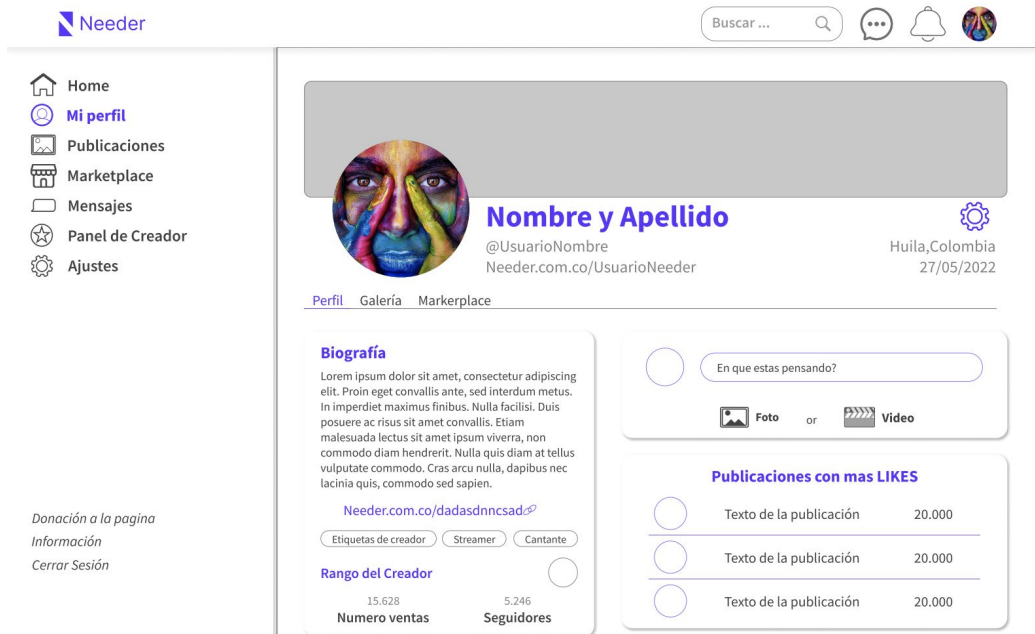


Figura 78. Mockup U-4.

En este panel, se podrán observar, además del formulario para crear la publicación, las reglas que se deben tener al crear dicha publicación y un enlace que permitirá al usuario ser redirigido a las políticas de contenido de Needer. Este formulario permite ingresar una descripción y un archivo, ya sea una imagen, un vídeo o un audio.



Figura 79. Mockup U-6.

Una vez creada la publicación, se podrá ingresar al detalle de dicha publicación, el cual mostrará la información más relevante, como su autor, su descripción, su archivo y su fecha

de publicación. Otro punto importante acá, es la interacción con dicha publicación, puesto que los usuarios podrán dar like a la publicación o crear sus comentarios en esta.

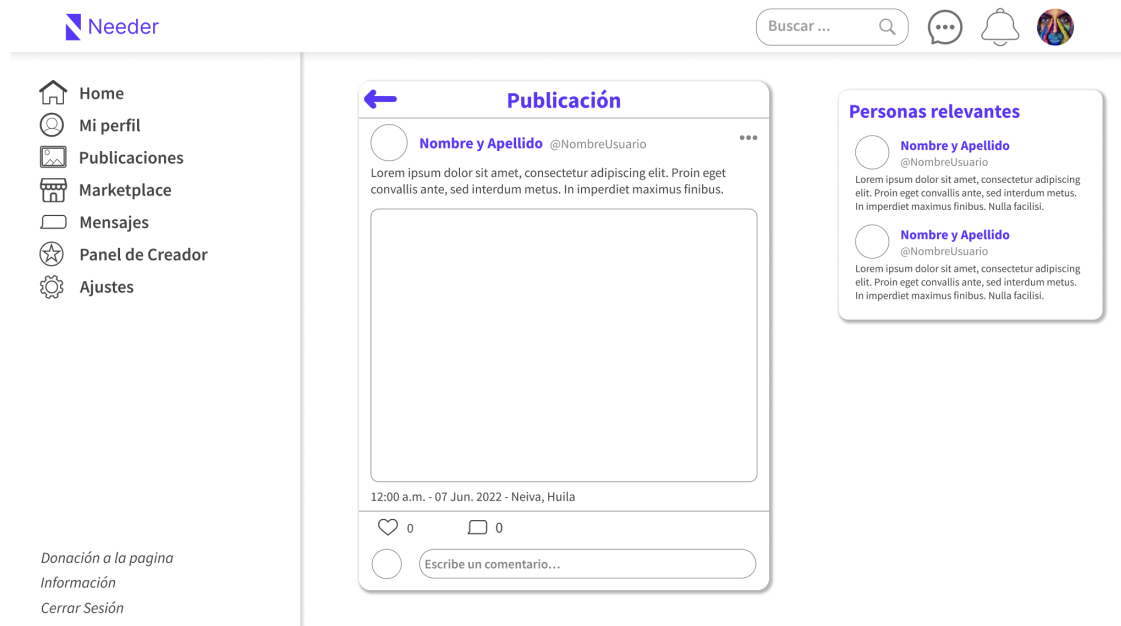


Figura 80. Mockup U-8.

Continuando con las funcionalidades, el usuario podrá ingresar a los perfiles de otros usuarios y poder observar la información que estos tengan registrados en sus perfiles. También, se podrán ver e interactuar con las publicaciones que ellos hayan realizado dentro de Needer.

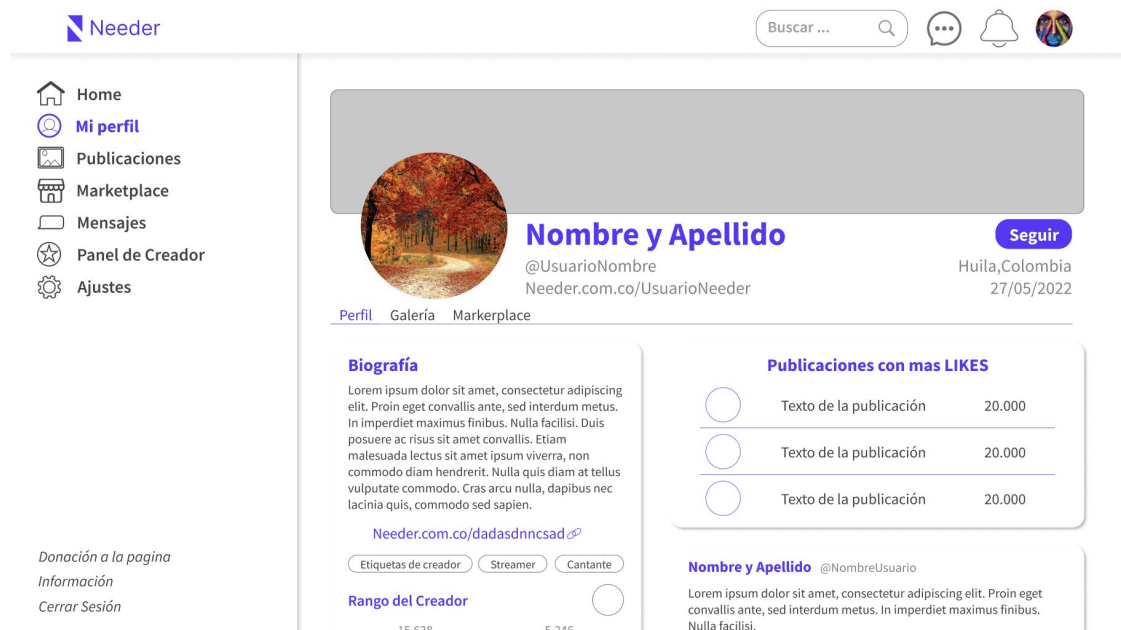


Figura 81. Mockup U-5.

El usuario podrá observar el feed de la red social, este estará ubicado en el apartado de Home y contará con un diseño intuitivo. Los usuarios podrán crear publicaciones gracias a la sección ubicada en la parte superior del feed, además de poder ver las publicaciones realizadas por todos los usuarios registrados dentro de la plataforma. Contará con unas pequeñas secciones a la derecha, las cuales mostrarán a las personas más relevantes de la aplicación y a las personas que sigues.

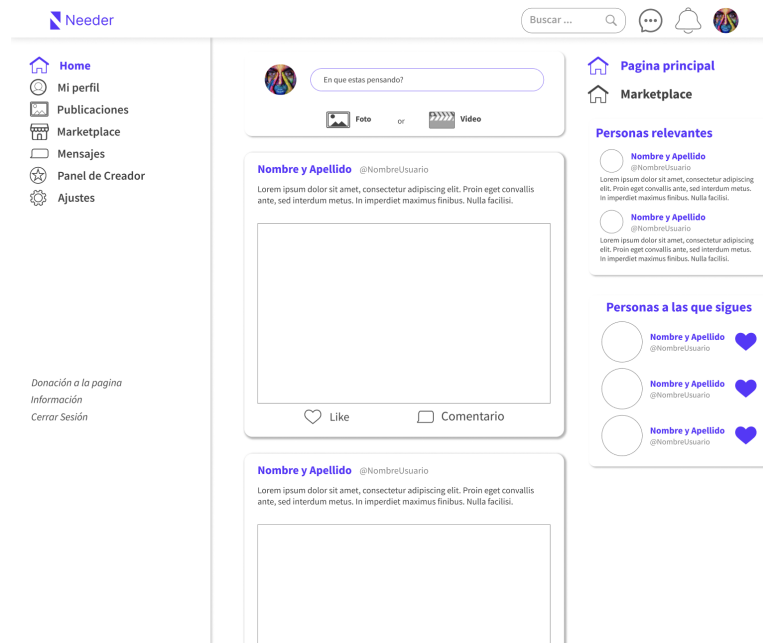


Figura 82. Diseño Mockup U-21.

Una de las funcionalidades más importantes y relevantes que ofrece Needer como red social, es el de permitir a los usuarios la comunicación entre ellos mediante mensajes. Para esta función, ubicada en el apartado de mensajes, se realizaron 2 Mockups. El primer Mockup muestra una lista de los mensajes recibidos, además de contar con una barra de búsqueda, en la parte superior derecha y un botón que le permitirá al usuario crear un chat en caso tal de que quiera hacerlo.

El segundo diseño, muestra los mensajes enviados por parte de los 2 usuarios, cuenta con una casilla, en la parte inferior, que permite al usuario escribir y enviar los mensajes. También, cuenta con un botón en la parte superior izquierda, que le permitirá al usuario salir de esa vista.



Figura 83. Mockup U-36.

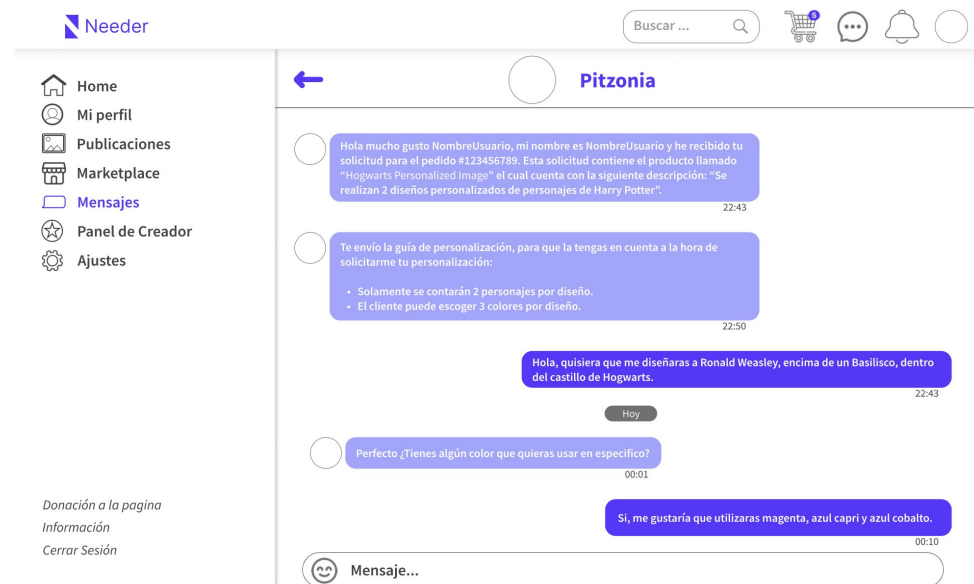


Figura 84. Mockup U-36.

Etapa 2

Una vez expuestas las funcionalidades más importantes de la red social, se continúa el proceso con la presentación de las funcionalidades del marketplace. En la etapa 2, conformada por los sprints 4 y 5, se encuentran las funcionalidades con prioridad media entre las historias de usuario. Estas funcionalidades están relacionadas con el creador de contenido, publicación de contenido multimedia y compra y venta de este contenido dentro de la plataforma.

Comenzando con la verificación del usuario como creador de contenido, ubicado en el apartado del panel de creador, permite al usuario poder autenticar su identidad para ser reconocido como creador de contenido. Está compuesto por una simple descripción, la cual explica su funcionamiento y el proceso a realizar por el usuario.

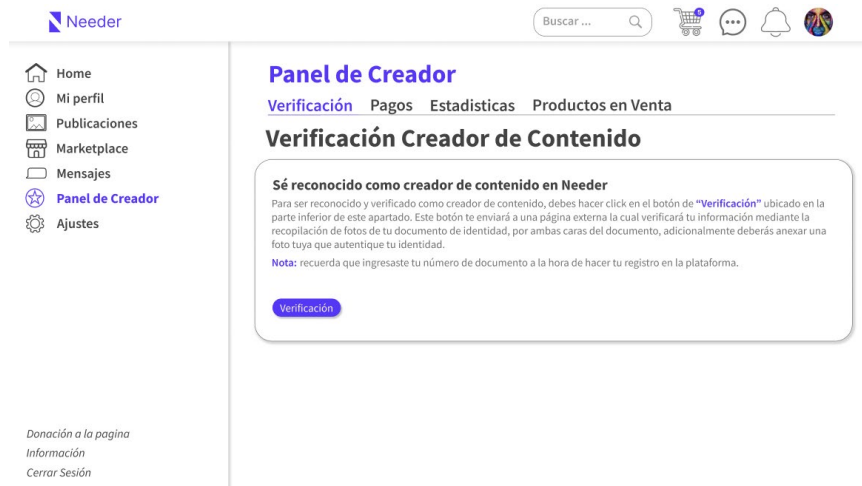


Figura 85. Mockup CD-1.

Una vez se haya confirmado la identidad del usuario y se le reconozca como creador de contenido, deberá de registrar una cuenta bancaria para poder recibir los pagos de sus productos publicados en venta. Para eso, el usuario deberá entrar en la sección de Pagos, ubicado en el menú superior del diseño. Al ingresar, se podrán observar los métodos de pagos que se pueden vincular en la cuenta del usuario, siendo estas Paypal y Stripe.

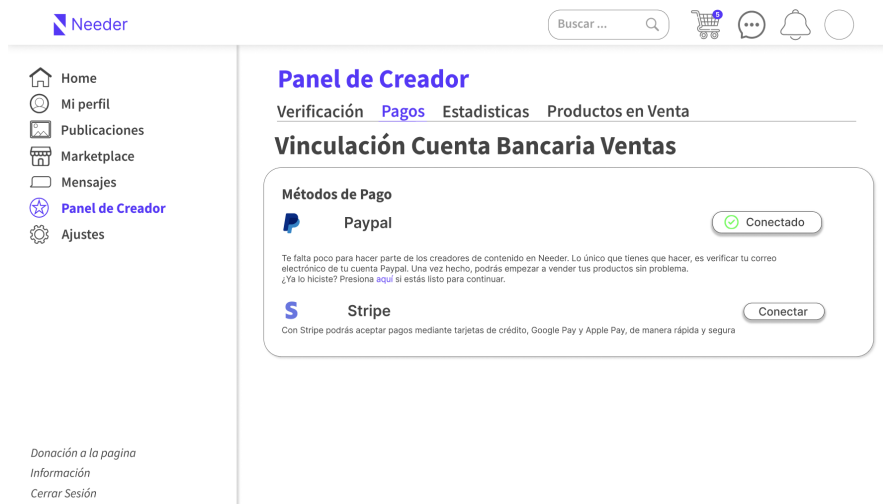


Figura 86. Mockup CD-2.

Siendo reconocido como creador de contenido y teniendo registrada una cuenta bancaria para recibir los pagos por la venta de su contenido multimedia. Se podrá dar inicio a la venta de contenido multimedia dentro de Needer, para eso, el usuario deberá dirigirse al apartado del perfil, sección de marketplace y comenzar a crear contenido multimedia.

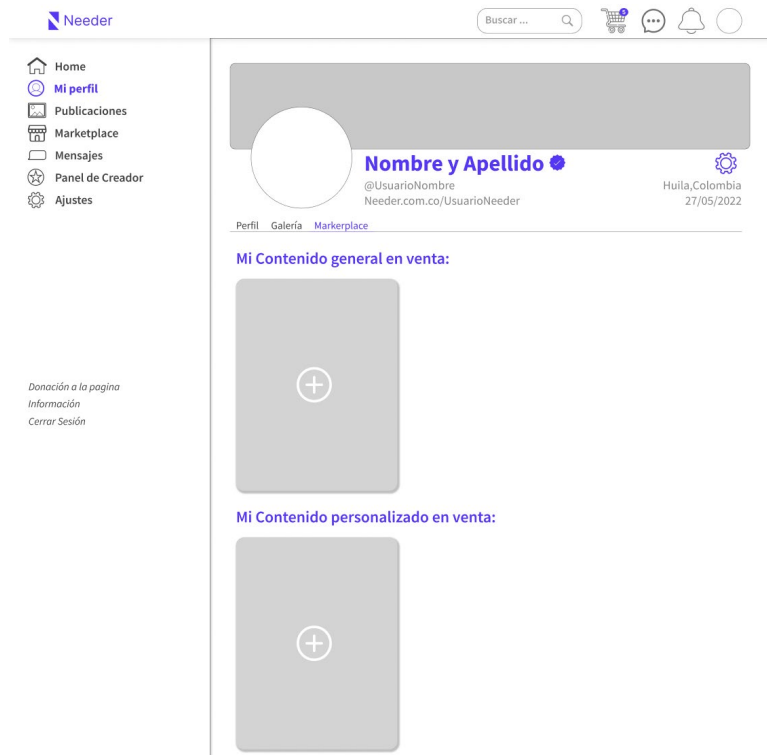


Figura 87. Mockup CD-10.

Se pueden observar 2 secciones, mi contenido general en venta y mi contenido personalizado en venta. Estas opciones cuentan con un botón que redirigirá al usuario al formulario para crear contenido multimedia. Este formulario varía según el tipo de contenido multimedia que se quiera crear, por ejemplo, para el caso del contenido personalizado, se agrega una sección para detallar la guía de personalización del producto, pero no se solicita un archivo multimedia.

Needer

🛒
💬
🔔
👤

- 🏠 Home
- 👤 Mi perfil
- 📄 Publicaciones
- 🏪 Marketplace
- ✉ Mensajes
- 👤 Panel de Creador
- ⚙ Ajustes

[Donación a la página](#)
[Información](#)
[Cerrar Sesión](#)

← Agregar Contenido

Contenido genérico
Contenido personalizado

Crea un nuevo producto

Título

Descripción

Recuerda que tus productos deben tener una descripción, la cual ayudará a tus compradores a la hora de adquirirlos, debe contar con un máximo 2200 caracteres.


Se realizan diseños de logos para todo tipo de productos, marcas, trabajos, entre otros. Se realizarán por pedido entre 5 u 8 diseños, dependiendo de la solicitud enviada por el comprador.

Miniatura del producto (opcional)


Esta es una imagen (no obligatoria) de ejemplo de cómo el producto puede ser ofrecido dentro del marketplace de Needer. No es el producto que se desea vender. Por ejemplo:

- Teaser de la foto que deseas vender.
- Miniatura del video que estas ofreciendo.
- Caratula del audio.

En caso de dejar vacío el campo se te asignara una miniatura ya sea a partir de una imagen por defecto, o en caso de ser video de una fraccion del mismo.



Agregar =



El formato del archivo debe ser de imagen (.jpg, .png, .jpeg) y no superar las 300mb por archivo, maximo 3 imagenes

Deseas que la miniatura del producto tenga una capa de sombreado para no mostrar el producto del todo?

Precio

Recuerda que los precios estan dados en dolares (USD). ej: \$30.00

Descuento

El porcentaje de descuento puede estar entre 0-100%. Ten en cuenta que si lo dejas vacío contara como 0 o si Ingresas 100% el producto sera **gratis**

Guía de Personalizacion

Es tiempo de poner tu producto en venta. Para eso necesitamos que coloques una pequeña descripción sobre la personalización que el cliente podrá solicitar sobre el producto. A continuación se exponen los puntos a tener en cuenta de las solicitudes del contenido personalizado:

- Al momento de que el cliente solicite un producto personalizado, al pedido sera asignado un moderador.
- La solicitud será finalizada solamente cuando el moderador y el cliente den el visto bueno del producto.
- Cualquier compra realizada bajo esta modalidad no tendra posibilidad a a devoluciones.

Cantidad de Productos

Este campo permite conocer la cantidad de productos multimedia que ofrecerás. En caso de querer exclusividad en tus productos puedes poner un limite de ejemplares. Si dejas el campo vacío por defecto se asignara a 300 unidades.

Previsualizacion Externa

Si deseas que tus compradores conozcan algo del producto antes de adquirirlo puedes agregar un enlace a un video, audio o imagen.

Cancelar
Guardar y Publicar

Figura 88. Mockup CD-06.

Por otro lado, para el formulario de contenido general, no se solicitará una guía de personalización pero si un archivo multimedia. Estos formularios comparten los campos de

título, descripción, miniatura del producto, precio, descuento, archivo del producto, cantidad de productos.

Needer

Buscar...

Home
Mi perfil
Publicaciones
Marketplace
Mensajes
Panel de Creador
Ajustes

Donación a la pagina
Información
Cerrar Sesión

← Agregar Contenido

Contenido genérico Contenido personalizado

Creando un nuevo producto

Título
Blue People


Descripción
Recuerda que tus productos deben tener una descripción, la cual ayudará a tus compradores a la hora de consumir, debe contar con un máximo 2200 caracteres.

Imagen diseñada a partir de una idea personal, está basada en la película Avatar 2 la tribu de agua, se utilizaron colores propios de las tribus incas.

Miniatura del producto (opcional)
Esta es una imagen (no obligatoria) de ejemplo de cómo el producto puede ser ofrecido dentro del marketplace de Needer. No es el producto que se deseas vender. Por ejemplo:

- Teaser de la foto que deseas vender.
- Miniatura del video que estas ofreciendo.
- Caratula del audio.

En caso de dejar vacío el campo se te asignara una miniatura ya sea a partir de una imagen por defecto, o en caso de ser video de una fracción del mismo.



Deseas que la miniatura del producto tenga una capa de sombreado para no mostrar el producto del todo?

Precio
\$ 20
Recuerda que los precios estan dados en dolares (USD), ej: \$30.00

Descuento
% 50
El porcentaje de descuento puede estar entre 0-100%. Ten en cuenta que si lo dejas vacío contara como 0 o si ingresas 100% el producto sera gratis

Producto a Ofrecer
Es tiempo de poner tu producto en venta. Para eso necesitamos que subas el archivo que deseas vender, siguiendo los siguientes parametros:

- El formato del archivo debe ser de imagen, audio o video.
- No debe pesar mas de 950Mb
- Debe cumplir las normas de la comunidad

Sube un archivo

Ejemplo de formatos: .jpg, .mp3, .mp4, .avi

Cantidad de Productos
Este campo permite conocer la cantidad de productos multimedia que ofrecerás. En caso de querer exclusividad en tus productos puedes poner un límite de ejemplares. Si dejas el campo vacío por defecto se asignara a 300 unidades

15

Previsualización Externa
Si deseas que tus compradores conozcan algo del producto antes de adquirirlo puedes agregar un enlace a un video, audio o imagen.

<https://www.youtube.com/watch?v=dQw4w9WgXcQ>

Cancelar Guardar y Publicar

Figura 89. Mockup CD-03.

Una vez creado y publicado el contenido multimedia, se podrá visualizar en la sección de marketplace ubicado en el perfil del usuario. En esta sección, el usuario podrá ingresar a su contenido publicado para realizar ediciones en caso tal de que el producto lo necesite.

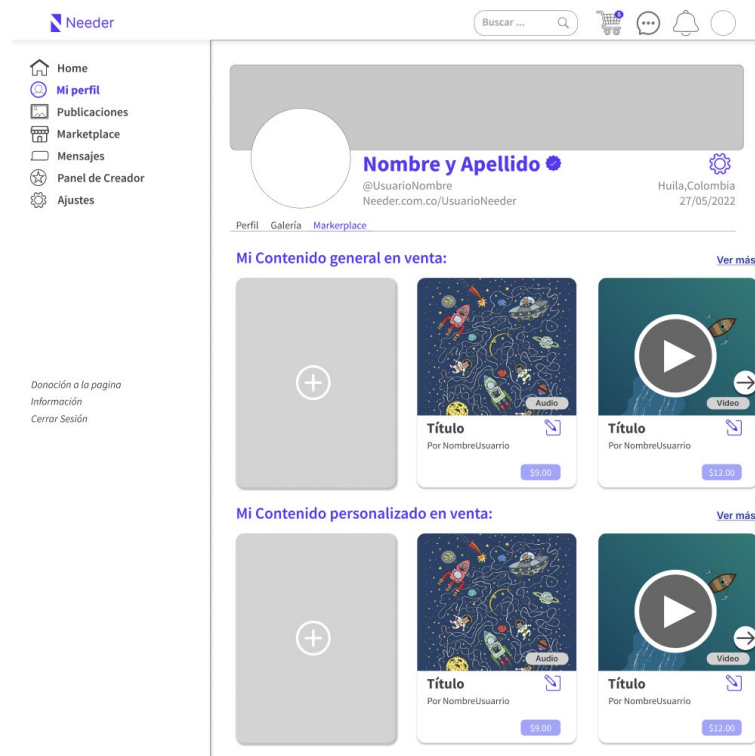


Figura 90. Diseño Mockup CD-10.

Ahora bien, existen 2 tipos de marketplace que permiten al usuario adquirir contenido multimedia dentro de la plataforma. El primero se encuentra ubicado dentro del perfil del creador de contenido y el segundo es el marketplace general de la aplicación, ubicado en el apartado de marketplace. El marketplace del creador de contenido, cuenta con un diseño muy simple pero intuitivo, permitiendo al usuario poder filtrar el contenido mediante la categoría del producto, el precio y el tipo de contenido.

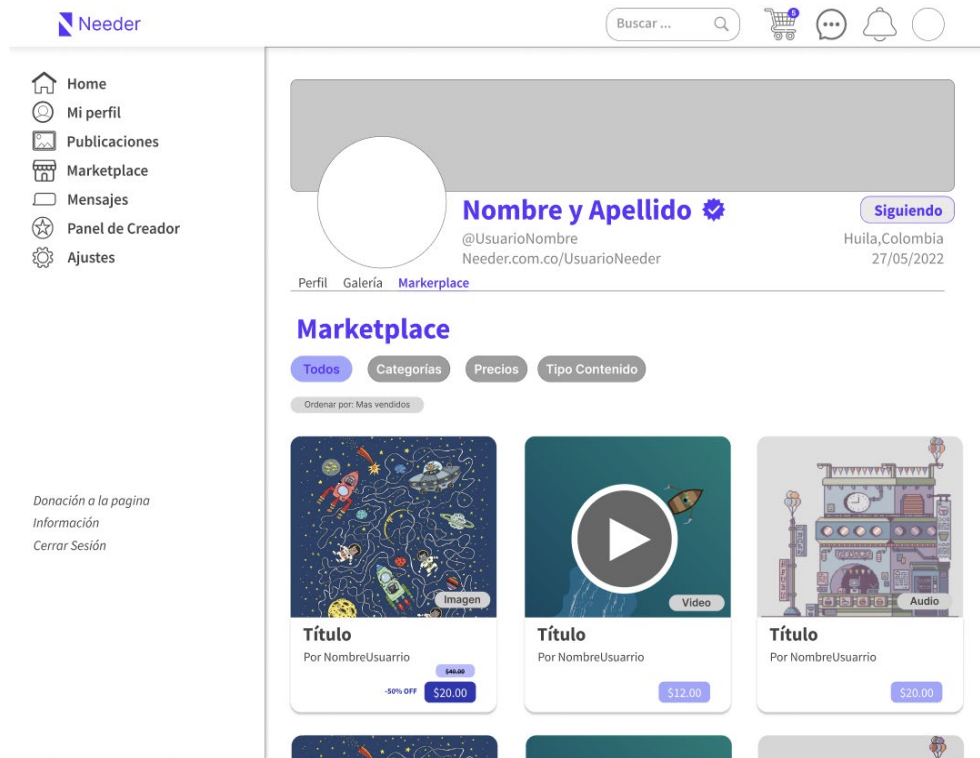


Figura 91. Mockup U-44.

En cuanto al marketplace general de la plataforma, se encuentra ubicado en el apartado de marketplace, reflejado en la barra de navegación lateral. Este diseño es más sofisticado, puesto que cuenta con una barra de búsqueda, la cual ayuda al usuario a encontrar contenido de interés de manera más rápida. También, se pueden filtrar los contenidos publicados mediante las categorías de creador de contenido, ubicadas debajo de la barra de búsqueda. Cabe resaltar que el contenido multimedia reflejado en este marketplace, se encuentra dividido por las secciones del contenido más vendido y del contenido local, además de mostrar a los creadores de contenido más famosos dentro de la plataforma.

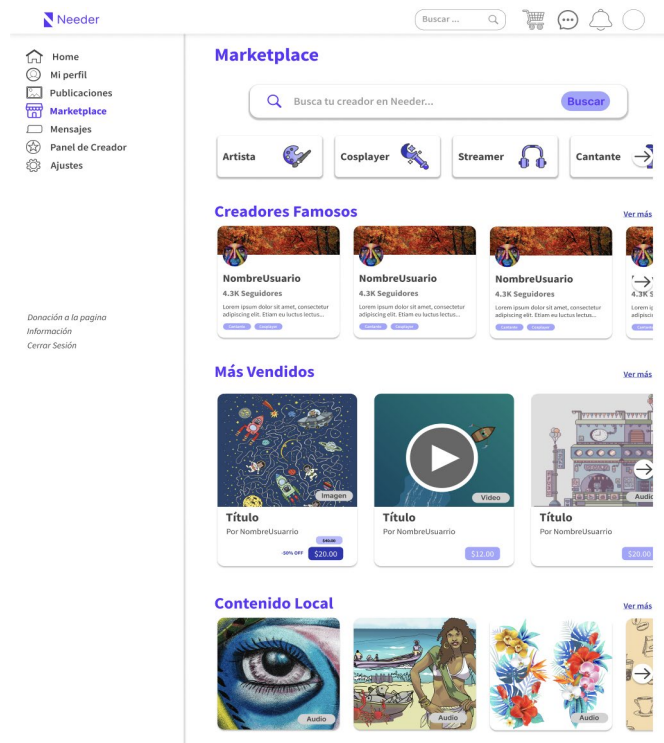


Figura 92. Mockup CD-22.

Para adquirir algún producto dentro de la plataforma, se tendrá que tener agregado algún método de pago en la plataforma. Este apartado se encuentra ubicado en los ajustes de la cuenta, sección método de pago. Aquí el usuario podrá agregar los métodos de pago que mejor le convengan, ya sean tarjetas de crédito (MasterCard, Visa o American Express) o cuentas electrónicas (Paypal o Venom).



Figura 93. Mockup U-24.

Luego de registrar un método de pago en la plataforma, el usuario podrá realizar la compra de cualquier producto multimedia ofrecido en el marketplace. Cuando se realiza el proceso de compra, el usuario deberá escoger el método de pago que se ajuste a su compra. Se mostrarán las tarjetas de crédito o débito agregadas por el usuario o sus cuentas electrónicas vinculadas. Como dato importante, se mostrará al costado derecho, la cantidad de productos a comprar, los impuestos (en caso tal de que sea contenido local) y la comisión que Needer cobrará por la transacción, más el total a pagar.

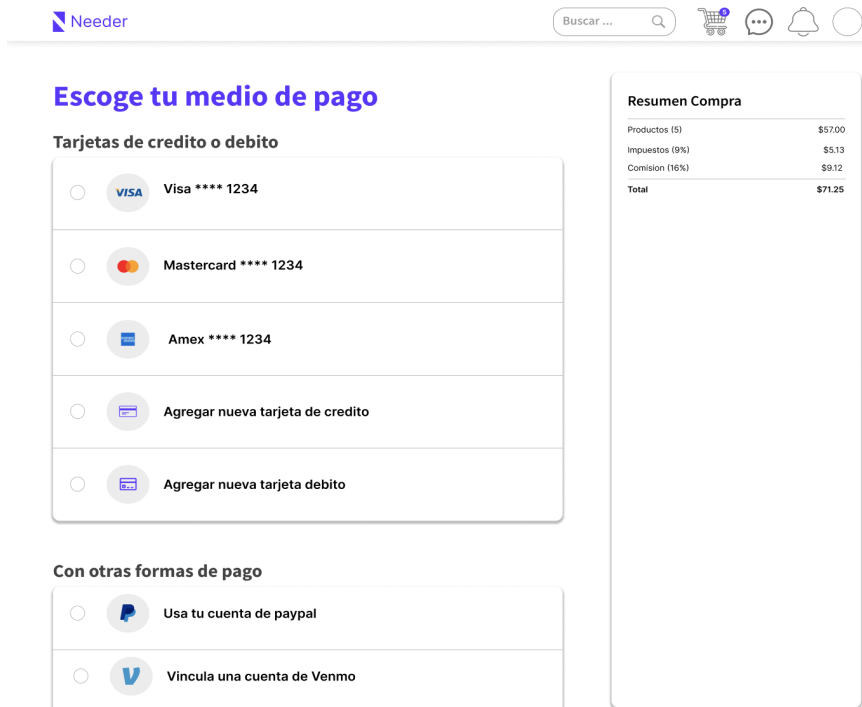


Figura 94. Mockup U-26.

Etapa 3

Para finalizar, se presenta la tercera y última etapa, la cual contiene las funcionalidades bajas de la red social y del marketplace. Esta etapa está conformada por 4 Sprints, siendo el 6to Sprint el primero y el 9no Sprint el último. Las funcionalidades presentadas en esta etapa, están relacionadas con el creador de contenido, el usuario general y el moderador.

Llegando ya al final de las funcionalidades presentes para el creador de contenido, se encuentra la visualización de las ganancias obtenidas por las ventas en la plataforma. Estas ganancias se encuentran en el apartado del panel del creador de contenido, en la sección de estadísticas. Refleja las ganancias mensuales del creador de contenido, el total de productos vendidos en el mes y la cantidad de seguidores con los que cuenta su perfil. También, se pueden encontrar gráficas de ganancias mensuales, los productos más vendidos y los listados de todos los productos vendidos hasta el momento. Finalmente se encuentran las notificaciones de usuario, recibidas por publicaciones, likes, compras o ventas.

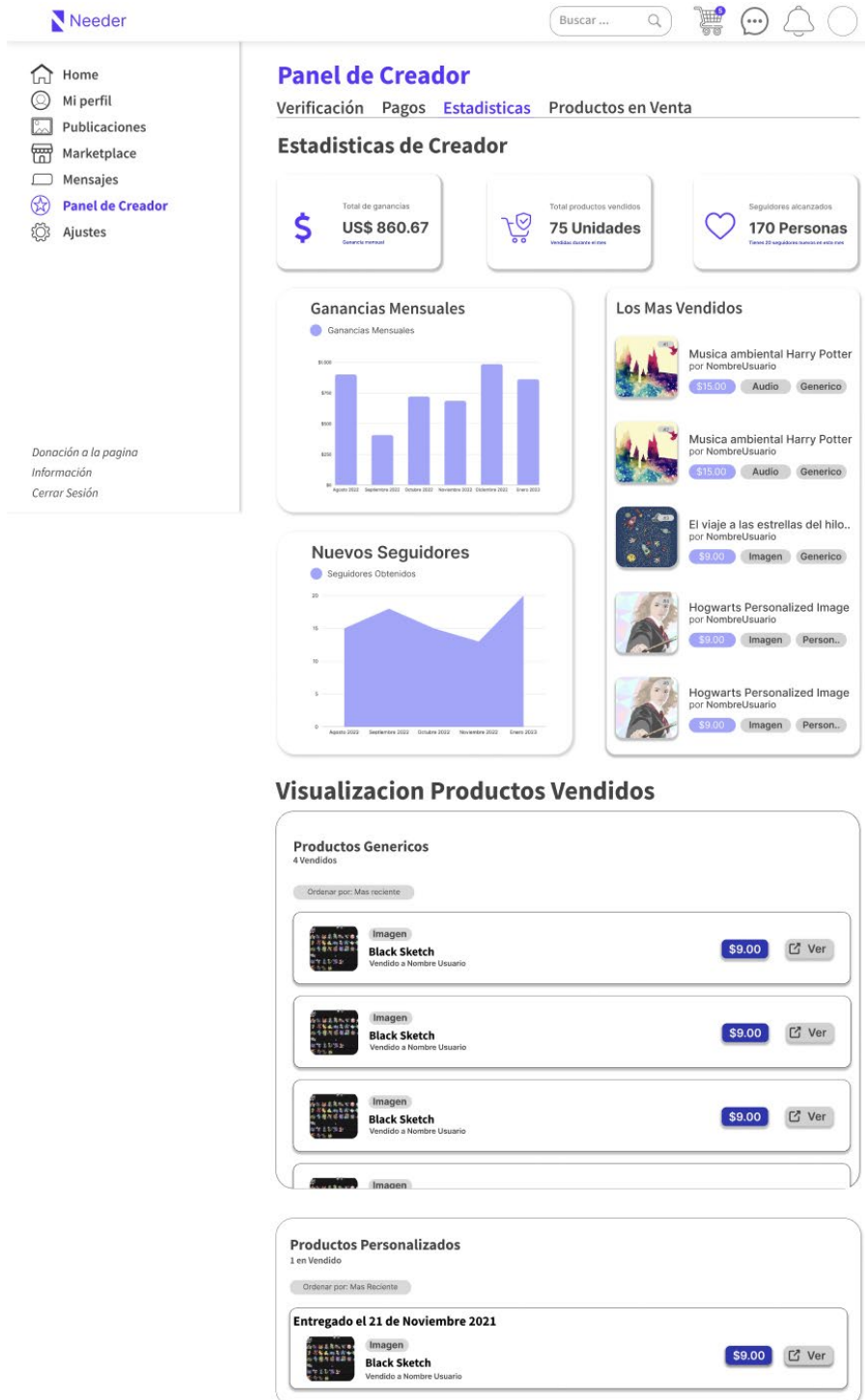


Figura 95. Mockup CD-11.

Por otra parte, en el caso del usuario, Needer todavía ofrece funcionalidades que son prácticas para su uso. Para los productos personalizados adquiridos por el usuarios, se debe solicitar el producto, puesto que este tipo de contenido cuenta con unas características totalmente diferentes al contenido genérico.

Para solicitar el producto personalizado, el usuario deberá ingresar al apartado de perfil, la sección de inventario e ingresar al producto personalizado que haya adquirido. Este diseño muestra la factura de la compra, el producto comprado y unas casillas con información, ubicadas al costado derecho. Una vez allí, sólo deberá hacer clic en el botón de solicitar producto.

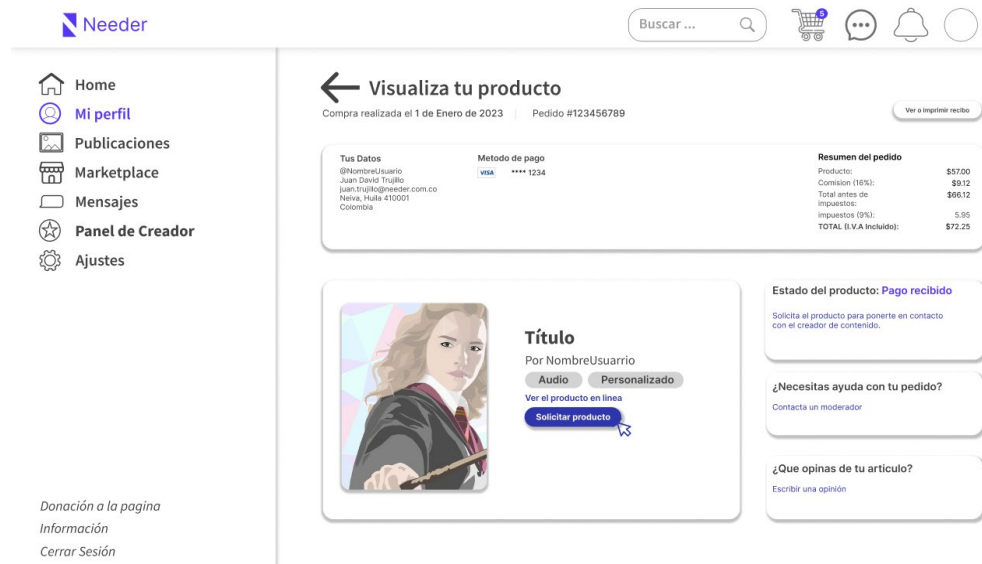


Figura 96. Mockup U-32.

Al momento de ser solicitado el producto, se creará un grupo de chat en el cual estarán involucrados el usuario, el creador de contenido y un moderador. El moderador será el encargado de supervisar la solicitud del producto. El creador de contenido y el usuario, deberán llegar a un acuerdo en base a la guía de personalización.

Esta sala de chat cuenta con un diseño totalmente independiente de las salas de chat general. Cuenta con el nombre del producto comprado, el número de la solicitud y cierta información ofrecida por la plataforma, como el número del moderador asignado a la solicitud y una tarjeta que contiene los datos necesarios del producto.

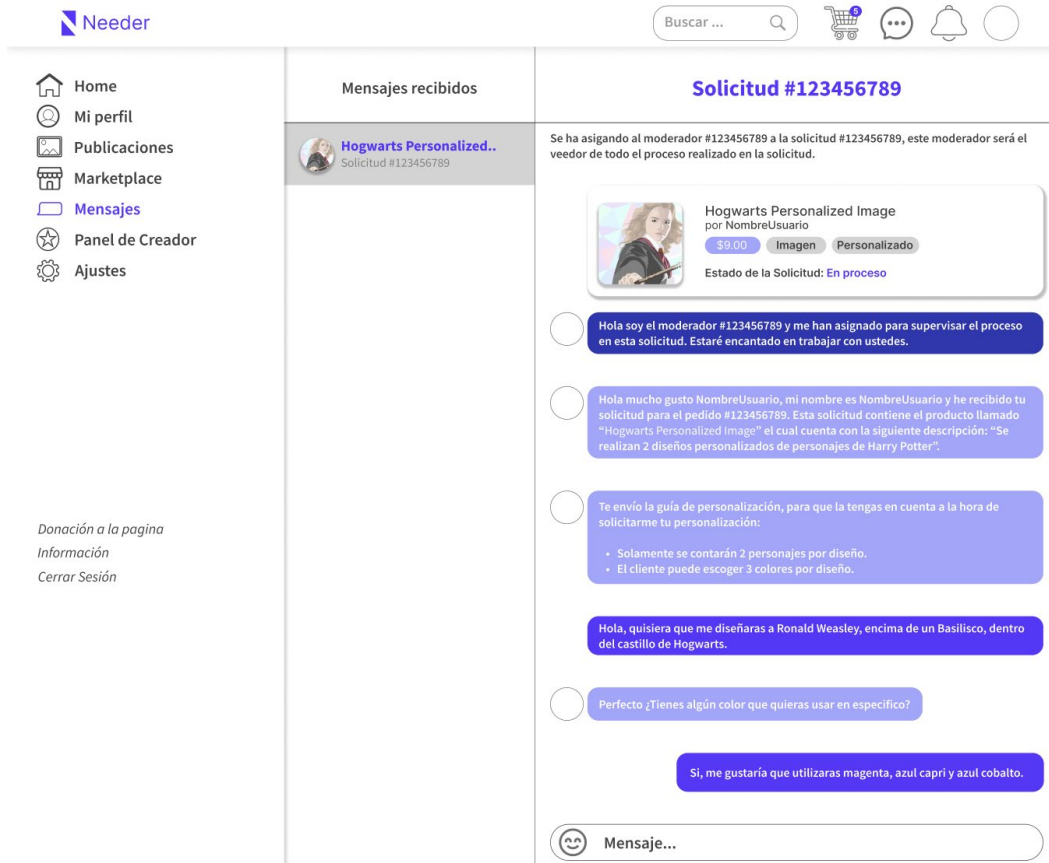


Figura 97. Mockup U-35.

Ahora bien, centrándose un poco en la red social, los usuarios podrán generar reportes a otros usuarios por actitudes indebidas dentro de la plataforma. También, podrán generar reportes a los productos ofrecidos en el marketplace. Para generar el reporte del usuario, se deberá realizar desde el perfil del usuario, escogiendo la opción de reportar.

El usuario será redirigido a un formulario de reporte, el cual contendrá el Id del Usuario a reportar, el tipo de reporte, el asunto del reporte, la descripción del reporte y una casilla para agregar algún tipo de evidencia que soporte el reporte. El reporte del usuario y del producto cuentan con el mismo diseño, cambiando solamente el tipo de reporte y el Id a reportar (para el caso del producto, es el Id del producto).

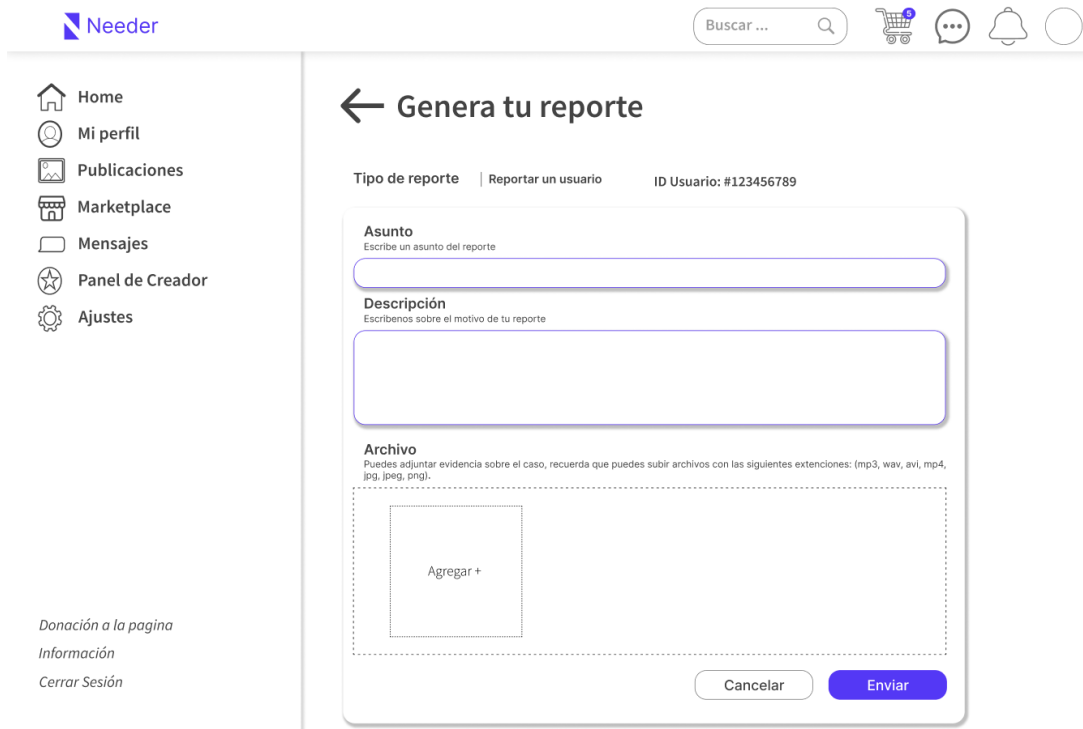


Figura 98. Mockup U-30.

Los reportes generados por el usuario, podrán ser visualizados en el apartado de ajustes, sección de avanzados. Los reportes se mostrarán en forma de lista, divididos por días y contarán con una herramienta de filtrado, la cual se puede realizar por medio de fechas. Estos reportes contarán con cierta información suministrada por el usuario, la hora en la que se generó el reporte y el estado del reporte.



Figura 99. Mockup U-31.

Los diseños del moderador cuentan con un estilo totalmente diferente a los mostrados inicialmente, teniendo un fondo oscuro y cambiando el color de la letra por colores más claros. Uno de los diseños que más sobresale, es la aprobación o negación de contenido personalizado. Este diseño contiene los datos del producto de la solicitud a la cual fue asignada al moderador, teniendo como punto extra, la cantidad de archivos entregados por el creador de contenido al comprador. El moderador deberá de reconocer si se cumplió o no lo solicitado por el comprador al creador de contenido.

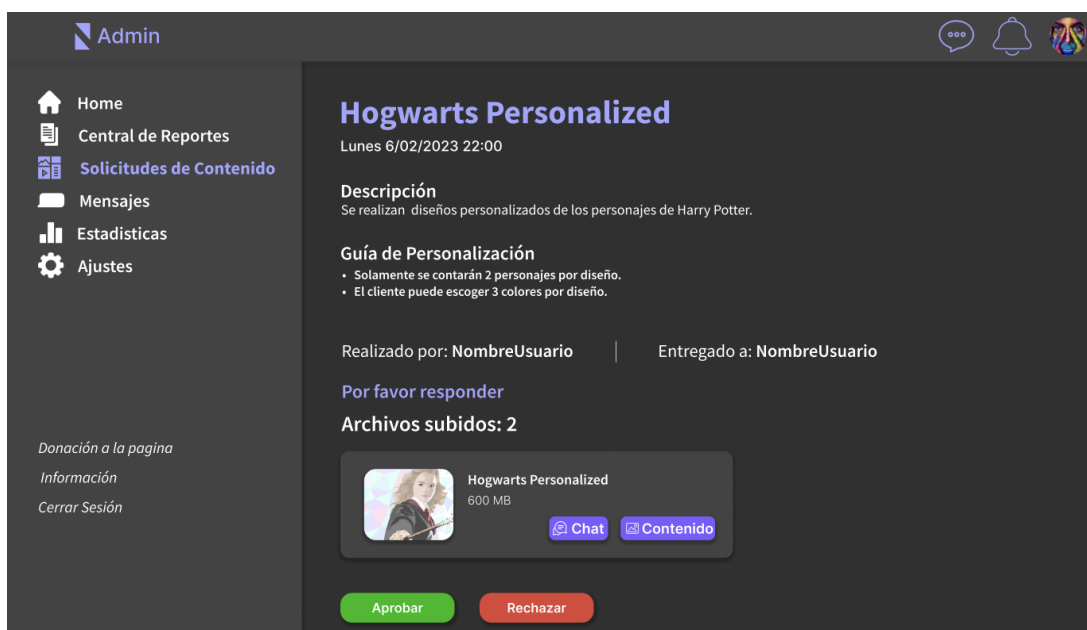


Figura 100. Mockup M-7.

Ahora bien, continuando con las funcionalidades importantes del moderador, se tiene la visualización de las ganancias en la plataforma. Este diseño es similar al diseño de ganancias por parte del creador de contenido, pero refleja otro tipo de información sobre la plataforma. Se pueden observar el total de ganancias mensuales a nivel global, la cantidad de productos vendidos mensualmente a nivel global y la cantidad de usuarios nuevos registrados en el mes actual. También cuenta con un apartado de los productos más vendidos dentro de la plataforma y un listado de todos los productos vendidos en la plataforma.

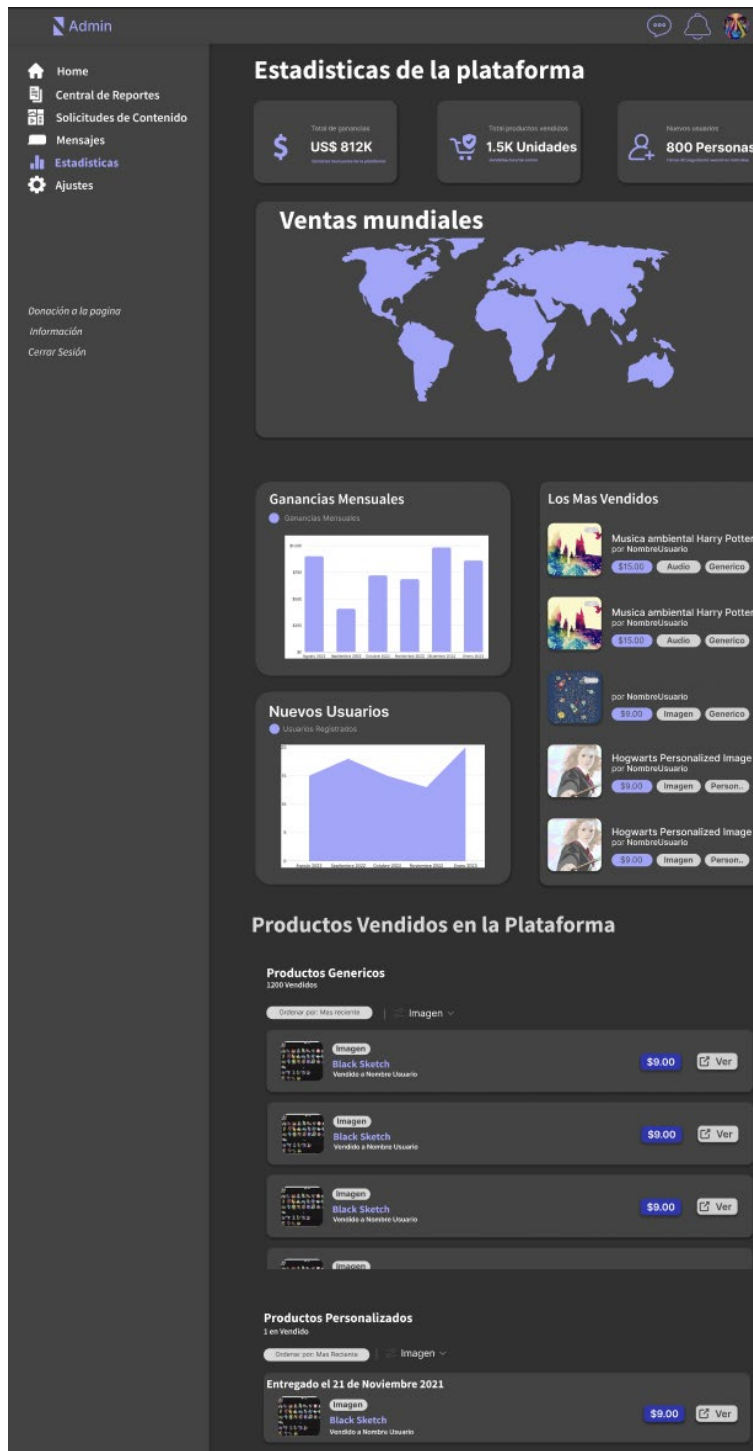


Figura 101. Mockup M-10.

Así mismo, se cuenta con la funcionalidad de poder visualizar a los usuarios que cuentan con una falta o que estén baneados. Este diseño está ubicado en el apartado de usuarios del panel del moderador. Al ingresar, se podrá observar una pequeña descripción explicativa para el moderador, un filtrado para buscar a los usuarios existentes dentro de la plataforma y el listado de todos los usuarios con su respectiva información y reflejando el estado actual de la cuenta.

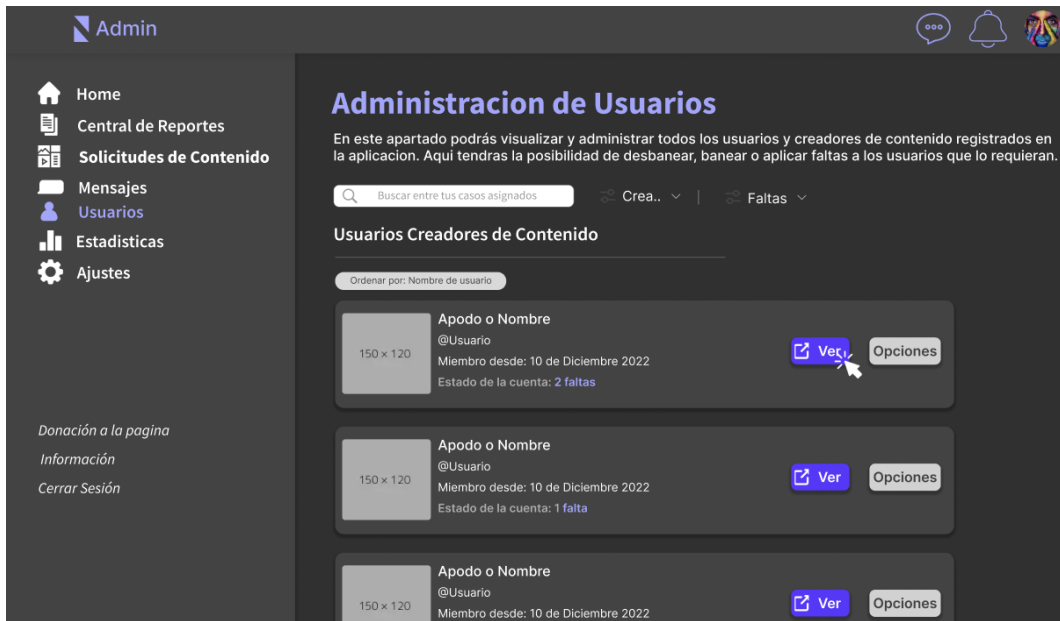


Figura 102. Mockup M-9.

Por último, se realizó el diseño de las notificaciones para los usuarios, con la finalidad de mostrar la cantidad de notificaciones y el tipo de notificaciones que han recibido. Al recibir una notificación, el usuario será avisado por un número ubicado en el ícono de la campana de la barra de navegación. Al hacer clic en este ícono, se desplegará un listado de todas las notificaciones recibidas por el usuario, organizadas por fecha. El usuario podrá acceder a estas notificaciones, mediante un clic.

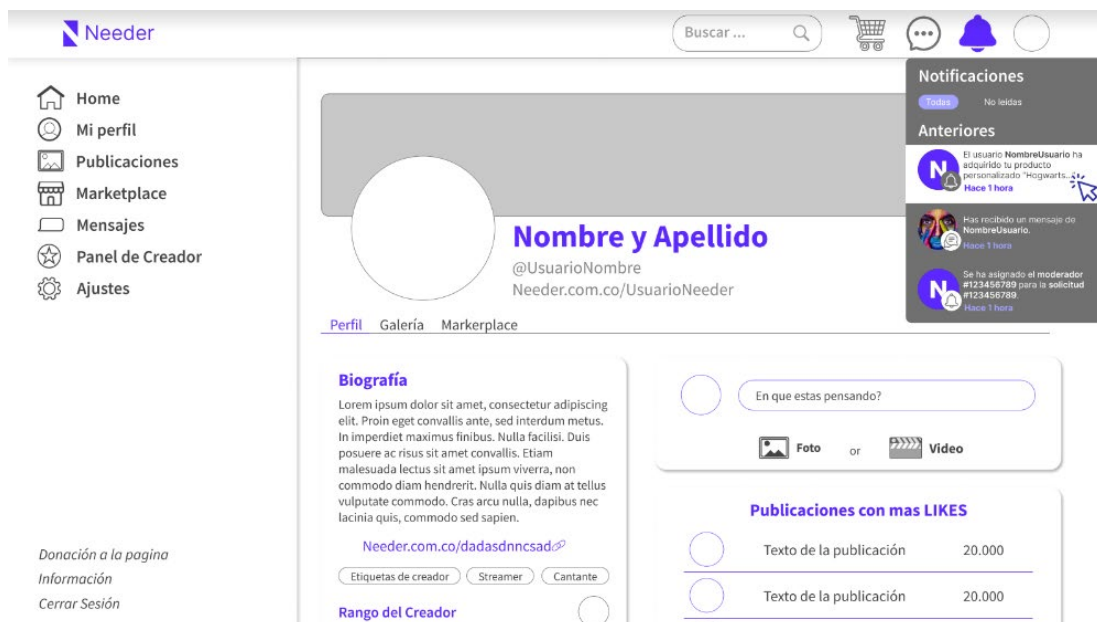


Figura 103. Mockup U-45.

6.4.3 Desarrollo

Una vez obtenidos los diseños de la plataforma, se da inicio al desarrollo del prototipo web. Este proceso de desarrollo se divide en 2 etapas principales. Primero, la preparación del entorno de desarrollo y posteriormente la segunda etapa se divide en el desarrollo de los Sprints definidos en el alcance del proyecto. El proceso de codificación para cada Sprint se dividió en la creación de los Modelos, las vistas para manejar las peticiones y lógica, la administración de las URLs y finalmente la maquetación de las plantillas

6.4.3.1 Preparación del entorno de trabajo

Con el fin de poder iniciar con el desarrollo del proyecto de manera ordenada, se crea primero la Carpeta denominada Proyecto, el cual se almacenará todo el código de la aplicación, así como el entorno virtual de Python. Para esto, se utiliza el comando presentado en la siguiente figura:

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado>mkdir Proyecto
```

Figura 104. Creación de la carpeta Proyecto.

Una vez creada la carpeta proyecto, se ingresa a la misma, con el fin de crear el entorno virtual del proyecto. Para esto, se utiliza el comando “*python3 -m venv.*”, de esta forma se da inicio al entorno virtual, a través del módulo “*Virtual Environment*”(venv) de Python.

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto>python3 -m venv .
```

Figura 105. Creación del entorno virtual.

Posteriormente, se hace la instalación del módulo “Django”. Esta instalación, estuvo acompañada de descargas de múltiples subpaquetes, los cuales fueron utilizados para el correcto funcionamiento del framework, un ejemplo de esto es “*sqlparse*”, módulo el cual se encarga de formatear todas las solicitudes SQL realizadas a través del ORM.

```
(Proyecto) C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto>pip install django==4.1.7
```

Figura 106. Instalación de Django versión 4.1.7.

Luego, se inicia el proyecto de Django, al cual se le denominó “needer”. Justo después de la creación del proyecto, Django se encarga de crear los paquetes y módulos para su funcionamiento.

```
(Proyecto) C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto>django-admin startproject needer
```

Figura 107. Creación del proyecto needer.

Para poder empezar la interacción con la base de datos, se ejecutaron las migraciones necesarias desde la carpeta que contiene el *manage.py* (en este caso needer). Estas migraciones crearon las tablas iniciales del proyecto (**Figura 108**).

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto>cd needer  
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto\needer>python manage.py runserver
```

Figura 108. Migraciones realizadas a la base de datos local.

Inmediatamente después, fue creado el super usuario para acceder directamente al panel de administrador, luego se inicializó el servidor local, haciendo uso del comando “*python manage.py runserver*” y se accedió a la url “*127.0.0.1/admin*” para acceder al panel, aquí se inició sesión con los datos diligenciados a la hora de crear el super usuario. La **Figura 109**, muestra las tablas iniciales del proyecto, las cuales fueron Groups y Users, tablas de autenticación.

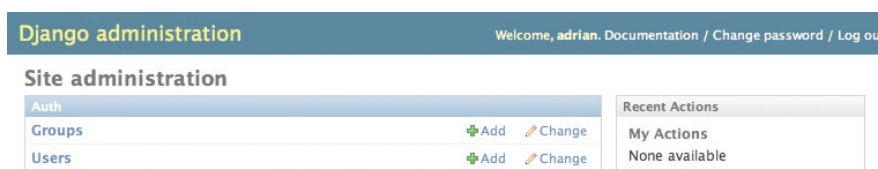


Figura 109. Interfaz gráfica administrador Django.

Después, se ingresó al módulo settings.py, ubicado en el paquete de needer y se realizaron los ajustes necesarios para definir el directorio donde se almacenarán las plantillas (**Figura 109**) y los archivos estáticos, es decir, el CSS, JS y las imágenes estáticas (**Figura 110**).

```
settings.py 4
needer > needer > settings.py > ...
52 'django.middleware.csrf.CsrfViewMiddleware',
53 'django.contrib.auth.middleware.AuthenticationMiddleware',
54 'django.contrib.messages.middleware.MessageMiddleware',
55 'django.middleware.clickjacking.XFrameOptionsMiddleware',
56 ]
57
58 AUTHENTICATION_BACKENDS = [
59     'django.contrib.auth.backends.ModelBackend',
60 ]
61
62 ]
63
64
65
66 ROOT_URLCONF = 'needer.urls'
67
68 TEMPLATES = [
69     {
70         'BACKEND': 'django.template.backends.django.DjangoTemplates',
71         'DIRS': [os.path.join(BASE_DIR, 'templates')],
72         'APP_DIRS': True,
73         'OPTIONS': {
74             'context_processors': [
75                 'django.template.context_processors.debug',
76                 'django.template.context_processors.request',
77                 'django.contrib.auth.context_processors.auth',
78                 'django.contrib.messages.context_processors.messages',
79             ],
80         },
81     },
82 ]
```

Figura 110. Configuración de las plantillas.

```
STATIC_URL = 'static/'
```

Figura 111. Configuración de la URL de los archivos estáticos.

Finalizado ya el proceso de configuración inicial en el módulo de settings.py, se continuó el proceso creando el repositorio local con Git. Después, se creó el archivo .gitignore, utilizado para los archivos sensibles que no se quieran enviar al repositorio remoto, como por ejemplo, el archivo que almacena las variables de entorno y el archivo **“README.md”**, el cual contiene una breve presentación del proyecto.

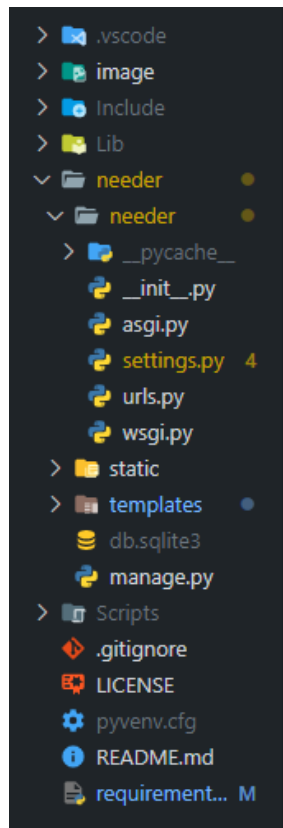


Figura 112. Árbol de directorios virtual de needer.

Una vez finalizado el proceso de creación del entorno, la estructura de las carpetas quedó organizada como se puede observar en la figura anterior. Además, se realizó el primer Commit al repositorio remoto, con el fin gestionar el control de versiones y empezar a trabajar en equipo.

6.4.3.1 Sprint 1

Durante el primer Sprint, se desarrollaron todas las historias de usuario relacionadas con las funcionalidades esenciales para los usuarios, como el registro y autenticación en la plataforma, la actualización de información personal, la vinculación de cuentas de Google mediante la API, entre otros. Para lograr esto, inicialmente se creó la aplicación “users”, la cual contiene todo lo relacionado con los usuarios y la autenticación.

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto\needer>python manage.py startapp users
```

Figura 113. Creación de la aplicación users.

Posteriormente, se agregó la aplicación al módulo `settings.py` del paquete `needer`. Es importante destacar que, al igual que la aplicación de usuarios, se instaló una aplicación denominada `Main`, que incluye las plantillas y vistas iniciales de la aplicación, como la página de inicio, las políticas de privacidad, los términos y condiciones y la información sobre la empresa.

Ambas aplicaciones fueron añadidas al listado de aplicaciones instaladas en el proyecto desde el archivo de ajustes, es importante recalcar que, por defecto Django tiene aplicaciones instaladas como la de autenticación ("*django.contrib.auth*"). Adicionalmente, se instalaron módulos externos desde el entorno virtual como "*allauth*" para facilitar la gestión de cuentas de google y captcha para brindar seguridad a los formularios.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.sites',  
    'captcha',  
    'allauth',  
    'allauth.account',  
    'allauth.socialaccount',  
    'allauth.socialaccount.providers.google',  
    'needer',  
    'users',  
    'main',  
]
```

Figura 114. Aplicaciones instaladas en el módulo `settings.py`.

Una vez instaladas las aplicaciones en el módulo de *settings.py*, se crearon los modelos de las clases, con el fin de almacenar información en la base de datos. Estos modelos se encuentran dentro del paquete de "*users*" y cuentan con una preparación realizada mediante "*imports*", los cuales permiten hacer el llamado a las clases, funciones o modelos de las librerías de Django. Para interactuar directamente con las tablas que conforman la base de datos, Django cuenta con un ORM, el cual se representa como una implementación de conceptos de mapeo relacional, permitiendo la inserción de valores a las tablas, como lo haría con instrucciones de SQL.


```

from django.db import models
from django.contrib.auth.models import AbstractUser, Group
from django_resized import ResizedImageField
from django.urls import reverse
from .utils import *
from django.core.exceptions import ValidationError

# Create your models here.

Juan David Trujillo Suarez, 9 months ago | 1 author (Juan David Trujillo Suarez)
class Pais(models.Model):
    """
    Pais

    id -> int
    nombre -> string

    """

    nombre = models.CharField(max_length=100, verbose_name="Nombre", null=False, blank=False)

Juan David Trujillo Suarez, 9 months ago | 1 author (Juan David Trujillo Suarez)
class Meta:
    ordering = ["nombre"]
    verbose_name = "Pais"
    verbose_name_plural = "Paises"

    def __str__(self) -> str:
        return self.nombre

```

Figura 115. Creación del modelo Pais.

En la **Figura (166)**, se puede observar que, para hacer uso del ORM de Django, se hizo el llamado del módulo ubicado en la librería “*db*” llamado “*models*”. Este módulo contiene las clases de la clase Models que permite la creación de tablas en la base de datos, esta se implementó directa e indirectamente (a través de modelos de librerías como oauth) para la creación de los modelos.

El primer modelo que se creó, es el modelo de “*Pais*”, puesto que se usa como llave foránea por el modelo “*User*”. Está compuesto por el campo “*nombre*”, el cual es utilizado para crear una columna en la base de datos y utiliza el modelo “*CharField*” para definir el tipo de dato que se almacena en ella. Este campo cuenta con atributos que permiten establecer su longitud, identificador dentro de las funciones y validadores. Además, estos atributos también aseguran que no se puedan ingresar valores nulos o vacíos en este campo.

Una vez creado el modelo, se realizó una clase llamada “*Meta*”, la cual se encarga de modificar el comportamiento de los modelos. Como se pudo observar en la figura anterior, la clase “*Meta*” contiene atributos, que permiten organizar la información que contiene el modelo “*Pais*” alfabéticamente. También, se modificaron los nombres en singular y en plural, con el fin de ser visualizados de manera ordenada desde el administrador de Django.

Finalmente, se realizó la función STR, con el fin de recibir un String a la hora de imprimir las instancias de los modelos.

Se creó un segundo modelo llamado "*TipoCelebridad*", el cual se utilizó como llave foránea por el modelo "*Users*". Este modelo está conformado por un campo llamado "*nombre*" que tuvo definidos parámetros de longitud, identificador y validadores. Además, este modelo contiene una clase "*Meta*", encargada de organizar la información visualizada en el administrador de Django

```
class TipoCelebridad(models.Model):
    """
    TipoCelebridad

    id -> int
    nombre -> string

    """
    nombre = models.CharField(max_length=24, verbose_name="Nombre", null=False, blank=False)

    class Meta:
        ordering = ['nombre']
        verbose_name = "Tipo Celebridad"
        verbose_name_plural = "Tipos de Celebridad"

    def __str__(self):
        return self.nombre
```

Figura 116. Creación del modelo Celebridad.

Finalmente, se creó el modelo "*User*", con el fin de administrar los usuarios registrados en el prototipo web. Este modelo se creó como hijo del modelo "*AbstracUser*", esta clase se utiliza en Django para definir un modelo de usuario personalizado. Este modelo cuenta con los siguientes atributos:

- **id:** Definido como IntegerField, almacena el identificador generado por defecto por Django. Este identificador es un número entero.
- **username:** El username se estableció como un campo de tipo CharField. Se encarga de almacenar el nombre de usuario que tendrá el usuario dentro del prototipo web. Este atributo es de tipo String.

- **first_name:** Se identificó el atributo first_name como CharField, encargado de guardar el nombre del usuario, ingresado inicialmente en el formulario de registro, este atributo es almacenado como String.
- **last_name:** El last_name, es definido como un atributo de tipo CharField, el cual almacena el apellido del usuario, dato ingresado en el formulario de registro e identificado como String.
- **password:** Es un atributo de tipo CharField, el cual almacena la contraseña del usuario para realizar su validación en el prototipo web. Este atributo es de tipo String y está encriptado.
- **email:** Hace referencia al email ingresado por el usuario en el formulario de registro. Este atributo es almacenado como String y se identifica como un atributo de tipo EmailField.
- **num_documento:** Identificado como CharField. Este atributo contiene el número de identificación del usuario y es almacenado como String. Cuenta con una longitud máxima de 10 caracteres, es único y no puede estar vacío ni ser nulo.
- **pais:** Este atributo se identifica como un ForeignKey, el cual proviene del modelo Pais y se almacena como un entero. Permite ser un atributo nulo y estar vacío.
- **slug:** Guarda una sección de la url del perfil del usuario, con el fin de identificar las vistas en el entorno de desarrollo. Se almacena como String y se identifica como un atributo de tipo SlugField, siendo un atributo único, permitiendo estar vacío y ser nulo.
- **biografía:** El atributo biografía, almacena una descripción que el usuario ingresa en su perfil. Se guarda como una variable de tipo String, identificada como un atributo de tipo TextField. Cuenta con una longitud máxima de 200 caracteres, permitiendo encontrarse vacía.
- **genero:** Este atributo, se identificó como CharField, contando con una longitud máxima de 7 caracteres y contiene una lista con las opciones a escoger. Se encarga de

almacenar el tipo de género con el que se identifica el usuario dentro del prototipo web.

- **fecha_nacimiento:** Siendo identificado como un atributo de tipo DateField, se encarga de guardar la fecha de nacimiento del usuario. Permite encontrarse vacío y ser nulo.
- **tipo_celebridad:** Este atributo se define como ManyToManyField, puesto que permite relacionar varios registros con un solo usuario. Se encarga de almacenar el tipo de celebridad con el que se identifica el usuario. Puede estar vacío y se almacena como un entero
- **foto:** Almacena la foto subida por el usuario en su perfil. Se identifica como un atributo de tipo ResizedImageField, puesto que permite redimensionar el tamaño de las imágenes subidas a una longitud de 500 píxeles por 500 píxeles. Este atributo cuenta con una función upload_to que indica el lugar en dónde se deben almacenar las imágenes. Permite encontrarse vacío.
- **banner:** Este atributo se identifica como ResizedImageField, permitiendo redimensionar imágenes a una longitud de 1280 píxeles por 720 píxeles. Este atributo almacena las imágenes subidas por el usuario para el banner de su perfil y son almacenadas en el directorio denominado user_directory_path_profile. Permite encontrar vacío
- **cartera:** La cartera es identificada como un atributo de tipo DecimalField, teniendo una longitud de 19 caracteres enteros, 2 caracteres decimales, un valor por defecto, puede ser nulo y estar vacío. Este atributo almacena la cartera que el usuario ingresará para gestionar su dinero dentro del prototipo web.
- **link:** El atributo link, se identifica como un atributo de tipo Charfield, contando con una longitud de 80 caracteres, permitiendo ser nulo y encontrarse vacío. Este atributo, almacena un enlace subido por el usuario en su perfil, en la sección de redes sociales.

- **verificado:** Identificado como un atributo de tipo BooleanField. El atributo verificado almacena el estado en el que se encuentra el usuario dentro del prototipo web, siendo false por defecto.

Teniendo identificados y definidos todos los atributos que hacen parte del modelo User, se continúa el proceso mediante la creación de métodos, los cuales se encargaron de realizar una función en específico dentro del modelo. A continuación se presentarán los métodos utilizados en el modelo User:

- **get_absolute_url:** Este método se encarga de obtener la url de un usuario para una instancia específica del modelo. La url se genera utilizando la función reverse() de Django, la cual toma como parámetro el nombre de la vista “detalle_creador” y un diccionario de palabras claves, el cual contiene como clave “slug”, con un valor de self.slug.
- **delete:** El método delete se encarga de eliminar la foto de la instancia a la cual se encuentra asociada, utilizando el método “delete” del “storage” imagen. Este método cuenta con los parámetros de “using”, para identificar la base de datos a la que pertenece la instancia y “keep_parents”, que indica si se deben borrar los objetos asociados a la instancia.
- **clean:** Este método prohíbe el uso de las palabras home o marketplace como nombre de usuario o como slug. Mediante un condicional if self.username.lower() y self.slug.lower(). Al encontrarse estos valores dentro de los campos username o slug, el método presenta un ValidationError con un mensaje.

```

class User(AbstractUser):
    """
    User Model

    attrs from AbstractUser:

    id -> int
    username -> string
    first_name -> string
    last_name -> string
    password -> string(encrypted)
    email -> string
    num_documento -> string
    pais -> comes from Pais Model
    slug -> string
    biografia -> string
    genero -> string
    fecha_nacimiento -> date
    tipo_celebridad -> comes from TipoCelebridad
    foto -> file
    cartera int
    link link
    methods:

    save(self, *args, **kwargs) -> saves instance

    """

    # Datos personales de facturación.
    num_documento = models.CharField(verbose_name='Numero de documento', max_length=10, null = True, blank = True, unique=True)
    pais = models.ForeignKey(Pais, verbose_name="Pais", null=True, blank=True, on_delete=models.SET_NULL)
    # TODO NUMERO DE CELULAR

```

Figura 117. Creación del modelo User.

Una vez finalizado el proceso de creación del modelo de usuario y definidos todos los métodos a utilizar. El siguiente paso que se realizó fue la migración de dichos modelos, utilizando el comando `python manage.py makemigration` y `python manage.py migrate`.

Ya con la migración de los modelos realizada, se modificó el archivo llamado `admin.py` dentro de la carpeta de Usuario, el cual se encargó de registrar los modelos en el administrador de Django. Este proceso se realizó importando los atributos de la clase padre `ModelAdmin` dentro del módulo de administración de Django llamado `admin`, del paquete `django.contrib` y los modelos desde el archivo `models.py`.

```

from django.contrib import admin
from .models import TipoCelebridad, User, Pais

# Register your models here.

class UserAdmin(admin.ModelAdmin):
    pass

admin.site.register(Pais)
admin.site.register(User, UserAdmin)
admin.site.register(TipoCelebridad)

```

Figura 118. Gestión del documento admin.py.

Luego, se realizó una clase llamada UserAdmin, la cual tiene como parámetro los atributos de la clase Padre ModelAdmin del módulo de administración de Django. Esta clase contiene los modelos importados, utilizando una extensión admin.site.register, la cual permite registrar los modelos en la interfaz gráfica del administrador, para poder visualizar mejor los datos de la base de datos.

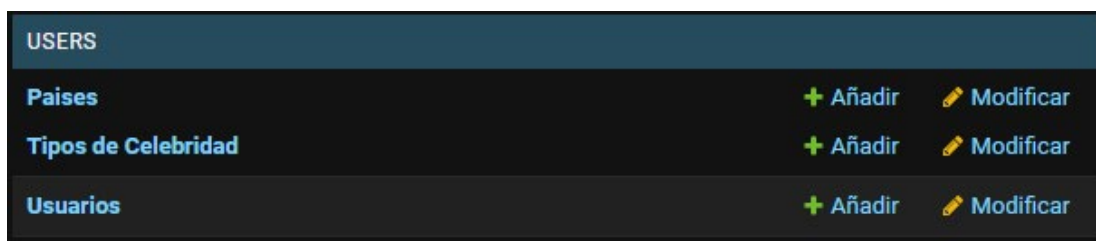


Figura 119. Interfaz gráfica actualizada del administrador de Django.

Con la interfaz gráfica del administrador actualizada con los modelos creados y con sus respectivos atributos y métodos. Se inició el proceso de creación de las vistas, las cuales contienen la mayoría de las funcionalidades que se ejecutaron en el proyecto.

Al momento de crear la aplicación Users, dentro de la carpeta de dicha aplicación, se encontró un archivo llamado view.py, el cual fue modificado para realizar las vistas que reciben las peticiones del proyecto. Cabe resaltar que, en este archivo se desarrollaron todas las vistas relacionadas con el registro de usuario, actualización de contraseña, actualización de correo

electrónico y autenticación de usuario. Para este caso, se explicarán las vistas que se pueden observar en la **Figura (120)**:

La primera vista creada fue la vista `SocialUserSignupView`, la cual tiene como parámetro la vista `SocialSignupView`, vista creada por defecto gracias a la librería de `allauth`. Esta vista se creó con la finalidad de que los usuarios puedan registrarse en la plataforma utilizando la API de Google. Para esto, se creó una variable llamada `form_class`, la cual contiene el formulario `SocialCustomForm` y una función denominada `get_context_data`, la cual envía un contexto al `template`, con el fin de visualizar la información.

La segunda vista creada, fue la vista `UserSignupView`, la cual contiene como parámetro la vista `SignupView`, vista creada por defecto gracias a la librería de `allauth`. Esta vista contiene la función de registro en la plataforma de manera normal. Contiene una variable llamada `form_class`, la cual contiene el formulario `SignupCustomForm`. Además, dentro de la clase, se puede encontrar la función `get_context_data`, esta función se encarga de enviar un contexto al `template`, con la finalidad de poder visualizar la información.

```
# User CREATION VIEWS
class SocialUserSignupView(SocialSignupView):
    """
    SocialSignupView

    Vista para registrarse a traves de API.

    Arguments: Requests, *args, **kwargs

    Retorna: Instancia del usuario creado. / Redirect al login
    """
    form_class = SocialCustomForm

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context["pais"] = Pais.objects.all()
        return context

class UserSignupView(SignupView):
    """
    SignupView

    Vista para registrarse.

    Arguments: Requests, *args, **kwargs

    Retorna: Instancia del usuario creado. / Redirect al login
    """
    # Segun el tipo de usuario se actualizaran los campos
    form_class = SignupCustomForm

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context["pais"] = Pais.objects.all()
        return context
```

Figura 120. Vistas registro de usuario.

Finalizado la creación de las vistas, se realizaron las urls de las vistas. Para esto, se creó un archivo dentro de la carpeta Users nombrado como urls.py. Dentro de este archivo, se realizaron los imports de los módulos path e include, los cuales ayudaron a definir las rutas de URL del prototipo web. También se importaron vistas como page_not_found, la cual se utilizó para manejar los errores 404 de página no encontrada; la vista PasswordResetView, para procesar las solicitudes de restablecimiento de contraseña en el marco de autenticación Allauth; todas las vistas realizadas en el archivo views.py de la aplicación Users y la vista LogoutView, utilizada para procesar la solicitud de cierre de sesión del usuario. Por otra parte, se importó el formulario CustomResetPasswordForm, el cual es utilizado para la vista de restablecimiento de contraseña.

Una vez se realizaron todos los imports, se creó un listado de todas las urls que se utilizaron en el archivo urls.py en el paquete Users. Estas urls son rutas de acceso, las cuales implementan las funcionalidades de cada una de las vistas importadas. A continuación se presentarán todas las vistas creadas para el Sprint 1 del prototipo web:

```
"""
users URLs
"""
from django.urls import path, include
from django.views.defaults import page_not_found
from allauth.account.views import PasswordResetView
from .views import *
from django.contrib.auth.views import LogoutView
from .forms import CustomResetPasswordForm

urlpatterns = [
    path('social/signup/', SocialUserSignupView.as_view(), name="socialaccount_signup"),
    path('social/connections/', CustomConnectionsView.as_view(), name="socialaccount_connections"),
    path('signup/', UserSignupView.as_view(), name="account_signup"),
    path('password/set', page_not_found, {'exception': Exception('Not Found')}, name="account_set_password"),
    path('logout/', LogoutView.as_view(), name="account_logout"),
    path('email/', CustomEmailChangeView.as_view(), name="account_email"),
    path('password/reset/', PasswordResetView.as_view(form_class = CustomResetPasswordForm), name="account_reset_password"),
    path('password/change/', CustomPasswordChangeView.as_view(), name="account_change_password"),
    path('inactive/', CustomInactiveView.as_view(), name="account_inactive"),
    path('profile/', updateAccountViewResolver, name='account_profile'),
    path('', include('allauth.urls')),
]
```

Figura 121. URLS realizadas en el archivo urls.py del paquete Users.

Finalmente, se realizaron todas las plantillas que se encuentran asociadas a las vistas creadas en esta aplicación. Estas plantillas cuentan con unos diseños únicos, asociados a las funcionalidades de las vistas que hacen uso de ellas. Están conformadas por una estructura realizada en HTML, estilos implementados con CSS y la librería de Bootstrap, la cual facilitó la implementación del “responsive” en todos los diseños. Estos diseños se encuentran dentro de la carpeta llamada template.

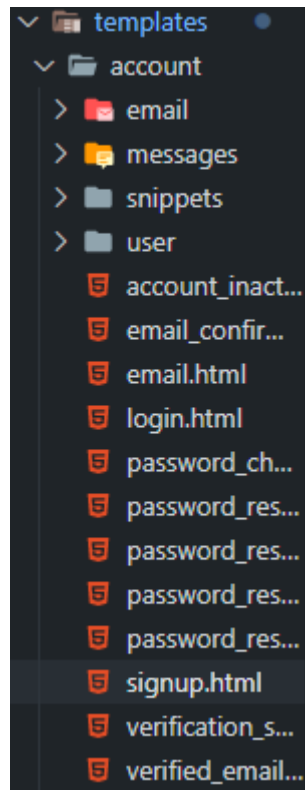


Figura 122. Templates diseñados en el paquete Users.

A continuación se podrán observar algunos de los resultados obtenidos, vistos desde la perspectiva del usuario:

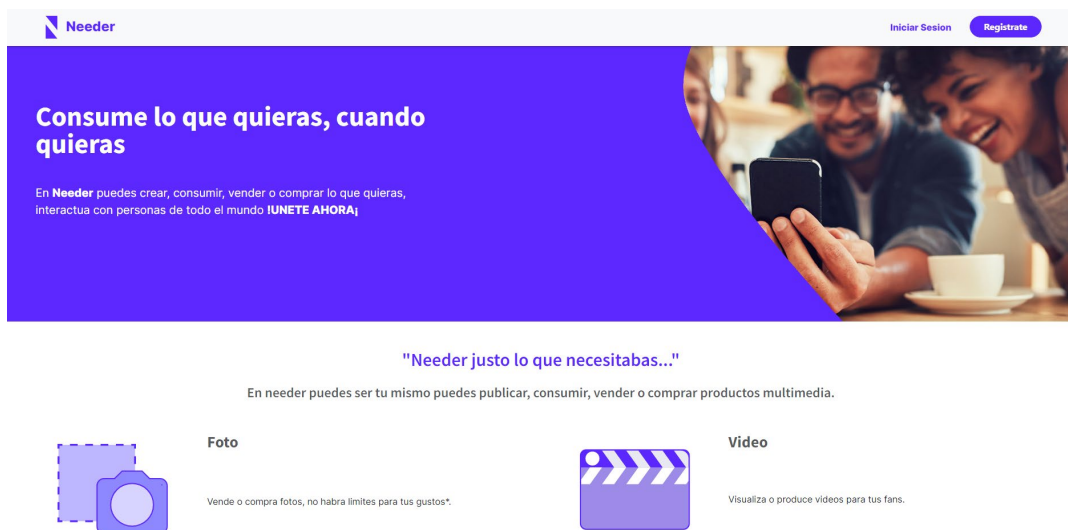


Figura 123. Landing-Page de Needer.

Bienvenido a Needer

¿Ya tienes una cuenta? Por favor [Iniciar Sesión](#).

Email:

Nunca compartemos tu email con otras empresas.

Contraseña: **Repite tu Contraseña:**

Usuario:

Nombres: **Apellidos:**

País: **Numero de documento:**
Tu número de identificación solo será consultado por nosotros para validar tu información.

[Regístrate](#)

Figura 124. Formulario de Registro de Needer.

Iniciar sesión en Needer

Correo electrónico:

Contraseña:

Recordarme

[Iniciar sesión](#)

¿Tienes problemas con iniciar sesión? [Únete](#)

[Inicia sesión con google](#)

Figura 125. Formulario de autenticación de Needer.

Ajustes

[Perfil](#) [Método de pago](#) [Actualizar Contraseña](#)

django It's Web Framework for the masses.

Datos personales

First name

Last name

Biografía

Dona a Needer
 Información
 Cerrar sesión

Needer © - 2022

Figura 126. Ajustes del perfil del usuario.

6.4.3.1 Sprint 2

Para el segundo sprint de la primera etapa, se realizaron todas las funcionalidades básicas esenciales de la red social. Es decir, crear, actualizar, listar y eliminar publicaciones, comentarios, dar likes a publicaciones y ver perfiles de otros usuarios. Para esto, en primera medida fue necesario crear la aplicación social, donde se encontrará todo lo relacionado a la red social (**Figura 127**).

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto\needer>python manage.py startapp social
```

Figura 127. Creación de la app social.

Una vez creada la aplicación, se debió ingresar al listado de aplicaciones instaladas al igual que como se realizó en el primer Sprint con la aplicación users. Una vez hecho esto, se generó el paquete social dentro de la ruta del proyecto y se pudo dar inicio a la creación de las vistas, modelos y plantillas. Para eso, primero se importaron los módulos y clases externas necesarias para la creación de los modelos, en este caso la más importante es la clase User, modelo creado en el Sprint anterior y que fue utilizado para asociar a cada publicación con un autor (usuario).

```
class Publicacion(models.Model):
    class Meta:
        verbose_name_plural = "Publicaciones"

    user = models.ForeignKey(User, on_delete= models.CASCADE, verbose_name='Autor')

    descripcion = HTMLField(verbose_name='Descripcion', max_length=2200, null=True, blank=True)
    archivo = models.FileField(upload_to = postDirectory, blank=True, null=True, validators=[valid_file_extention, valid_file_size])
    fecha_creacion = models.DateTimeField(verbose_name='Fecha Publicacion', auto_now_add=True, auto_now=False)
    fecha_actualizacion = models.DateTimeField(verbose_name='Fecha Actualizacion', auto_now=True, auto_now_add=False, blank=True, null=True)
    nsfw = models.BooleanField(verbose_name='NSFW', default=False, null=False, blank=False)

    def get_absolute_url(self):
        return reverse('detalle-publicacion', kwargs={'pk': self.pk, 'user_slug': self.user.slug})
```

Figura 128. Asociación modelo user.

Para la creación del modelo primero se utilizó el atributo **“verbose_name_plural”** dentro de la Clase Meta para definir cómo se debe nombrar la clase Publicación cuando se encuentre en plural. Posteriormente, se crearon los atributos de la publicación, los campos son los siguientes:

- **user:** Utiliza el campo de llave foránea, referenciando a la clase Usuario. A partir del parámetro “on_delete” se definió que sucede cuando el usuario se elimina, en este caso si se borra el usuario, se eliminan todas las publicaciones.
- **descripción:** El campo descripción almacena una descripción de la publicación, en este caso se utiliza un campo del módulo tinymce, el cual permite crear formularios con formateo. Es decir, estilos en títulos, subtítulos, permite ingresar emojis, etc.
- **archivo:** El archivo almacena un string con el directorio donde se encuentra subido un archivo que el usuario desea compartir. En este caso se utiliza validadores para comprobar el tamaño y extensión del archivo.
- **fecha_creacion:** La fecha creación almacena la fecha y hora de creación de la actualización, con el fin de ordenar las publicaciones por orden de creación.
- **fecha_actualizacion:** Por otro lado, la fecha de actualización guarda la fecha y hora en la que se actualiza la publicación. Por lo que inicialmente se almacena vacía.
- **nsfw:** Finalmente, en el atributo nsfw (“Not Safe For Work”) se almacena un booleano el cual definirá si la publicación es apta para todo público.

Una vez creados los atributos básicos, se sobrescriben algunos métodos con el fin de manipular la publicación a la hora de guardarse o para obtener la ruta del detalle de publicación accediendo directamente desde el objeto como por ejemplo en el método “*get_absolute_url*”. Este último, retorna la ruta del detalle de la publicación a partir del id de la misma.

Posteriormente, se crearon los modelos de *LikedPublicacion*, el cual referencia la tabla enlace para la relación *ManyToMany* y permite los likes en las publicaciones, el modelo Comentarios, para almacenar los comentarios y por último el modelo *LikeComentarios*, para poder dar likes a los comentarios.

```

class LikedPublicacion(models.Model):
    id_publicacion = models.ForeignKey(Publicacion, verbose_name= ("Publicacion"), on_delete=models.CASCADE)
    user = models.ForeignKey(User, verbose_name= ("Usuario"), on_delete=models.CASCADE)
    fecha = models.DateTimeField(verbose_name = 'Fecha de Like', auto_now_add=True, auto_now=False)

class Comentarios(models.Model):
    id_publicacion = models.ForeignKey(Publicacion, verbose_name= ("Publicacion"), on_delete=models.CASCADE)
    user = models.ForeignKey(User, verbose_name= ("Autor"), on_delete=models.CASCADE)
    comentario = models.TextField(max_length=520, verbose_name= ("Comentario"), blank=False)
    fecha_creacion = models.DateTimeField(verbose_name = 'Fecha de Like', auto_now_add=True, auto_now=False)

```

Figura 129. Modelos likepublicacion y comentario.

Para los likes se almacena el id de la publicación al que se le da like, el usuario que dio like a la misma y la fecha en que se dio like (esto con el fin de más adelante administrar las notificaciones y registros de actividad). Por otro lado los comentarios almacenan de igual forma el id de la publicación para referenciar a través de la llave Foránea al modelo **“Publicacion”**, el usuario, un comentario como campo de texto y la fecha y hora en la que se creó el comentario.

Finalmente se crea el modelo **LikeComentario**, el cual permite almacenar los likes de los comentarios. Este almacena los mismos campos que el **LikePublicacion**, con la diferencia que en vez de guardar el id de la publicación, guarda el id del comentario **“id_comentario”**.

Una vez creados los modelos se realizan las migraciones de todos los modelos y se actualiza el módulo **“admin.py”** para visualizar las tablas dentro del panel de administración. Luego, como se puede observar en la siguiente figura, se crearon los métodos de las clases como lo es para agregar un nuevo like, listar los likes, quitar el link, obtener la url del comentario para eliminarlo, etc.

```

def likePublicacion(self): return reverse('addlike', kwargs={'pk':self.id})
def disPublicacion(self): return reverse('removelike', kwargs={'pk':self.id})

def cantidadLikes(self):
    # Metodo para contar Likes
    return LikedPublicacion.objects.filter(id_publicacion=self).count()

def cantidadComentarios(self):
    # Metodo para contar comentarios
    return Comentarios.objects.filter(id_publicacion=self).count()

```

Figura 130. Métodos creados para las clases likepublicacion y comentario.

Continuando con el proceso del Sprint 2, se crearon las vistas necesarias para suplir las necesidades de las Historias de Usuario. Como ejemplo se toma la vista **“CrearPublicacionView”**, en este caso se usó la clase llamada **CreateView** para agilizar el

proceso de la creación de la vista (**Figura 131**). Para todas las demás vistas, según la necesidad se utilizaron las vistas genéricas como lo son: “*CreateView*”, “*UpdateView*”, “*DeleteView*”, “*ListView*” y “*DetailView*”.

```
class CrearPublicacionView(DispatchAuthenticatedUserMixin, LoginRequiredMixin,
                          SuccessMessageMixin, CreateView):
    model = Publicacion
    template_name = 'social/user/crear-publi.html'
    form_class = CrearPublicacionForm
    success_message = 'La publicacion fue creada satisfactoriamente!'

    # Se sobrescribe el form_valid para guardar el autor.
    def form_valid(self, form):
        self.object = form.save(commit=False)
        self.object.user = self.request.user
        self.object.save()

        return redirect(self.get_success_url())

    def get_success_url(self) → str:
        return reverse('detalle-publicacion', kwargs={'pk': self.object.pk, 'user_slug': self.object.user.slug})
```

Figura 131. Vista crearpublicacon.

Para la creación de este y los demás objetos, en algunos casos fue necesario crear un formulario, con el fin de poder manipular la verificación de los campos. Dado que por defecto si se usa la vista *CreateView*, se crea un formulario a partir de los atributos del modelo, pero no se validan según la necesidad que haya. Por ejemplo, para el caso de las Publicaciones como se observa en las siguientes figuras (**Figura 132** y **Figura 133**), fue necesario validar si la publicación iba con un archivo, esta puede ir sin la descripción o en caso contrario si no hay archivo que haya descripción.

```
class CrearPublicacionForm(forms.ModelForm):
    descripcion = forms.CharField(required=False, widget=tinymce_widgets.AdminTinyMCE(attrs={'placeholder': 'Cuentale a tu publico de
    class Meta:
        model = Publicacion
        fields = ('descripcion', 'archivo', 'nsfw')

    def __init__(self, *args, **kwargs):
        super(CrearPublicacionForm, self).__init__(*args, **kwargs)

        for visible in self.visible_fields():
            visible.field.widget.attrs['class'] = 'form-control'

        self.fields['descripcion'].widget.attrs['class'] = 'form-control'
        self.fields['archivo'].widget.attrs['class'] = 'form-control'
        self.fields['nsfw'].widget.attrs['class'] = 'form-check-input d-none'
```

Figura 132. Formulario crearpublicacon.

```

def clean(self):
    cleaned_data = super().clean()

    if not 'descripcion' in cleaned_data:
        cleaned_data['descripcion'] = ''
    else:
        if len(cleaned_data['descripcion']) > 2200:
            raise forms.ValidationError('La descripción debe tener menos de 280 caracteres')
        # Si ambos campos van vacíos tirara error.
        if len(cleaned_data['descripcion']) == 0 and not cleaned_data["archivo"]:

            raise forms.ValidationError('Debes agregar una descripción o un archivo a la publicación.')

    return cleaned_data

```

Figura 133. Función dentro de formulario crearpublicacon.

Posteriormente, una vez creadas las vistas y formularios se estableció el archivo de urls para administrar las rutas de la aplicación social. En este archivo se puede observar todas las vistas que se realizaron para el segundo Sprint.

```

settings.py  admin.py 2  urls.py 1
needer > social > urls.py > ...
1
2
3
4
5
6 urlpatterns = [
7
8     # lista red social
9     path('home/', HomeSocialView.as_view(), name='social-home'),
10
11     # CREADOR DE CONTENIDO DETAIL
12
13     path('<str:slug>/', DetailCreador.as_view(), name='detalle-creador'),
14
15     # CRUD PUBLICACION:
16     path('<str:user_slug>/publicacion/<int:pk>/', DetallePublicacionView.as_view(), name='detalle-publicacion'),
17     path('publicacion/create/', CrearPublicacionView.as_view(), name='crear-publicacion'),
18     path('<str:user_slug>/publicacion/<int:pk>/update/', UpdatePublicacionView.as_view(), name='update-publicacion'),
19     path('<str:user_slug>/publicacion/<int:pk>/delete/', DeletePublicacionView.as_view(), name='delete-publicacion'),
20
21     #likes
22     path('publicacion/<int:pk>/like/', AddLikesPublicacion.as_view(), name='addlike'),
23     path('publicacion/<int:pk>/dislike/', RemoveLikesPublicacion.as_view(), name='removelike'),
24
25     #Comentarios
26     path('comentario/<int:pk>/', CrearComentarioView.as_view(), name='comentario'),
27     path('comentario/<int:pk>/delete/<str:user_slug>/', DeleteComentarioView.as_view(), name='delete-comentario'),
28     path('comentario/<int:pk>/like/', AddLikesComentarios.as_view(), name='comentario-addlike'),
29     path('comentario/<int:pk>/dislike/', RemoveLikesComentarios.as_view(), name='comentario-removelike'),
30
31
32     #Galeria Red Social
33     path('<str:slug>/galeria/', GaleriaSocial.as_view(), name='galeria-social'),
34
35 ]

```

Figura 134. URLS aplicación social.

Como último paso, se crearon las plantillas que están asociadas a cada vista. Si se observa en la figura de la vista de crear publicación, hay un atributo llamado **“template_name”** en el cual se guarda el string con el enlace de las plantillas. En este caso todos los templates, fueron almacenados en un paquete llamado social, con un subpaquete llamado user, dentro del directorio de templates.

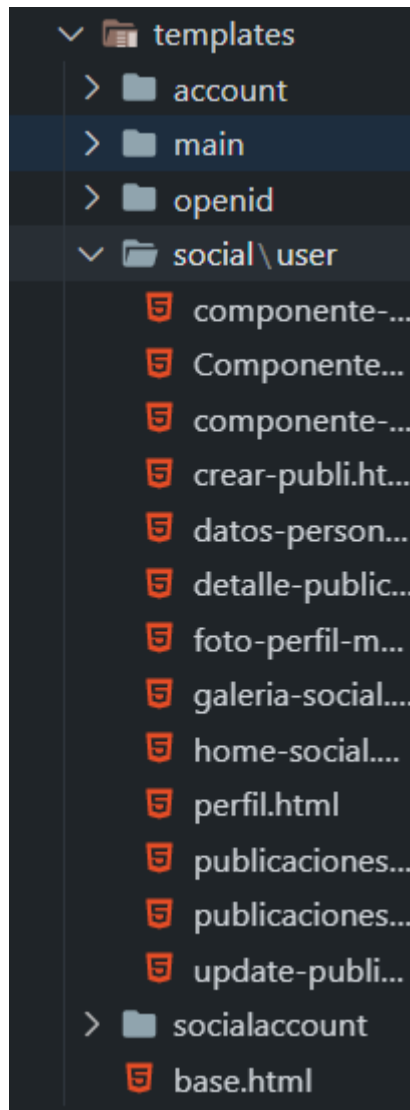


Figura 135. Templates aplicación social.

Finalmente, se puede observar en las siguientes figuras el resultado del desarrollo del segundo sprint:

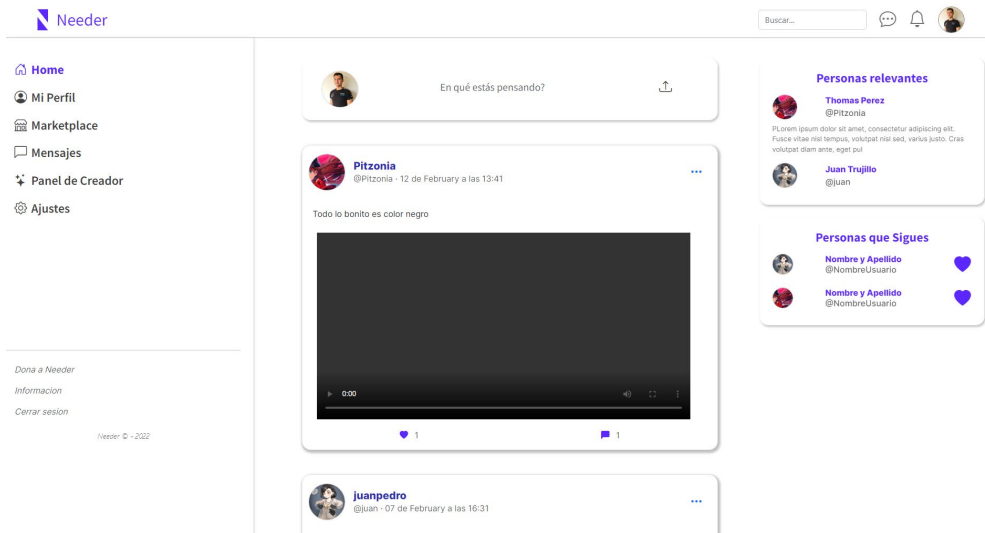


Figura 136. Vista Home social.

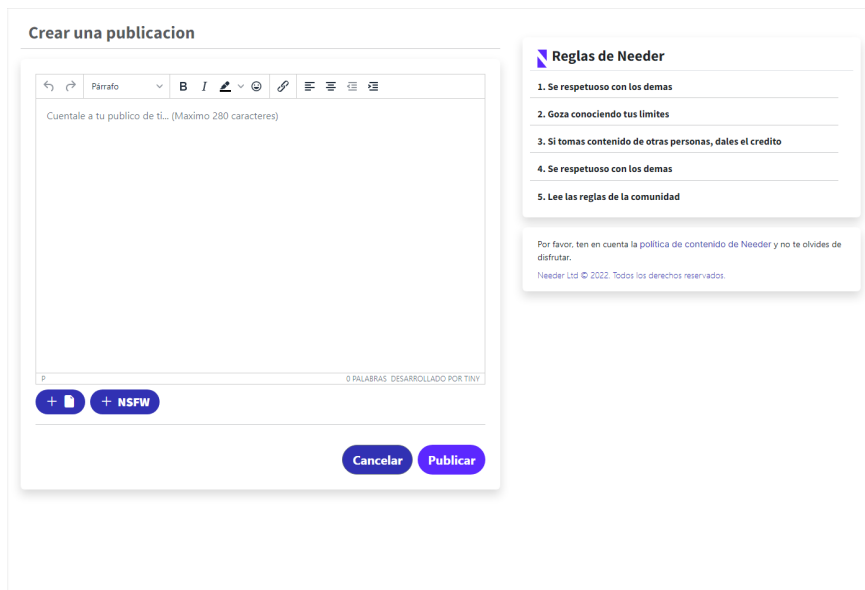


Figura 137. Vista Crear Publicación.

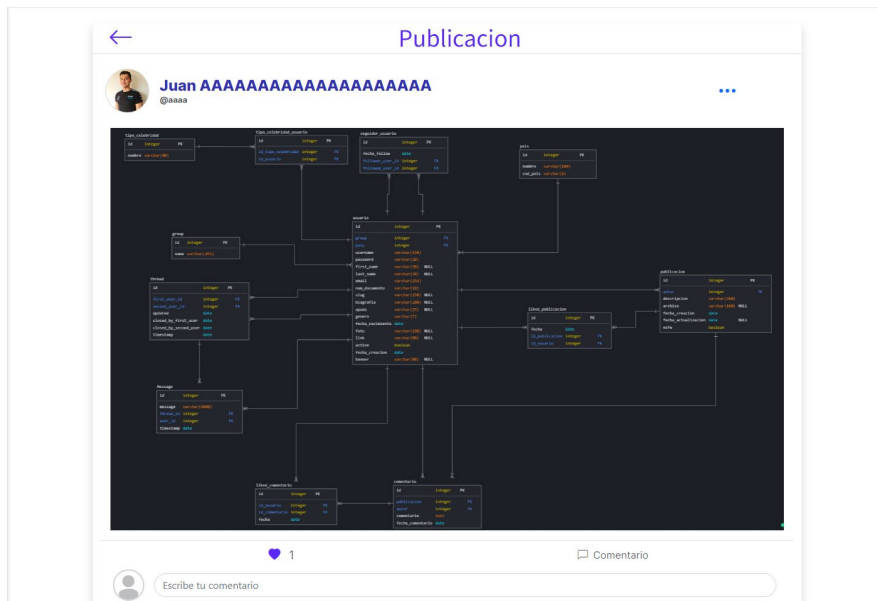


Figura 138. Vista Publicación Creada.

6.4.3.1 Sprint 3

Para el tercer y último sprint desarrollado se realizaron las historias de usuario de búsqueda de contenido, manejo de seguidores y seguidos, mensajes entre usuarios y eliminar la cuenta de usuario. En este caso puesto que ya existen las aplicaciones social y users, en ellas se realizaron las funcionalidades de la Galería, seguidores y búsqueda de contenido, por esa razón, no fue necesario volver a crearlas. Sin embargo, para la aplicación de chat se decidió crear una aplicación aparte para manejar los websockets (Canales de comunicación) y las rutas del chat aparte.

```
C:\Users\sarsu\Desktop\Home\Universidad\Modalidad de grado\Proyecto\needer>python manage.py startapp chat
```

Figura 139. Creación aplicación chat.

Al igual que en la creación de las aplicaciones anteriormente mencionadas, se debe registrar la aplicación en el módulo de ajustes y una vez hecho esto se cargan los paquetes necesarios para la creación del chat. Continuando con el proceso, se crearon los modelos necesarios en el modulo **“models.py”** dentro del paquete chat y users, donde se presentan las Clases necesarias que almacenan los seguidores (en los modelos de users), y los hilos y mensajes dentro de los modelos de la aplicación chat.

```

class SeguidorUsuario(models.Model):
    fecha_follow = models.DateTimeField(auto_now_add=True)
    follower_user_id = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Seguidor', related_name='user_follower_id')
    followed_user_id = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Seguido', related_name='user_followed_id')

    def clean(self):
        objeto = SeguidorUsuario.objects.filter(follower_user_id=self.follower_user_id, followed_user_id=self.followed_user_id).order_by('id')
        if objeto:
            raise ValidationError('El usuario ya sigue a este otro usuario')

```

Figura 140. Modelo seguidorusuario.

Primero se presenta la creación del modelo *SeguidorUsuario* presente en la aplicación de users, este modelo es utilizado como enlace para la relación *ManyToMany* de usuario a usuario, para llevar el contenido de los seguidores. Almacenando dos instancias de usuario, y la fecha del follow, a través de los siguientes atributos:

- **fecha_follow:** Fecha y hora del follow, es utilizada para organizar los followers dentro del listado. Es almacenada como DateTimeField y se añade automáticamente al crearse el objeto, gracias al parámetro “auto_now_add = True”.
- **follower_user_id:** Almacena la instancia del objeto User y su relación es OneToMany a través del campo ForeignKey. Este almacena el objeto del usuario que decidió seguir a otro.
- **followed_user_id:** Al igual que el campo anterior, se almacena la instancia del objeto a través del ForeignKey. En este caso es el usuario que fue seguido, por el usuario seguidor.

Por otro lado, el método clean, dentro de la clase SeguidorUsuario, asegura que al crearse una nueva instancia de la clase, no exista ya una coincidencia de dos usuarios que se siguen. Es decir, que no se repita el seguidor y el seguido.

Después se crearon las vistas para seguir a otro usuario, dejar de seguirlo y listarlos. Estas vistas se realizaron con la ayuda de javascript y peticiones AJAX, puesto que se requería que el usuario no recargará la página para realizar la petición de seguir o dejar de seguir, y en cambio se hacen a través de elementos del DOM. Un ejemplo de esto es la vista de seguir un usuario, como se observa en la **Figura 141**.

```

"""Vista de Seguir"""
#TODO: PROBAR FUNCIONAMIENTO
class SeguirUsuario(PreventGetMethodMixin, LoginRequiredMixin, View):
    model = SeguidorUsuario

    def get_queryset(self):
        self.queryset = self.model.objects.filter(followed_user_id = self.usuario)
        return self.queryset

    def post(self, request, slug, *args, **kwargs):
        self.usuario = get_object_or_404(User, slug=slug)

        if self.request.user == self.usuario:
            raise Http404()

        query = self.get_queryset()
        usuario_follower = get_object_or_404(User, id = request.user.id)

        cantidadFollows = len(self.usuario.get_user_followers())
        result = {}
        listado = []
        result['pk'] = self.usuario.id
        result['url'] = self.usuario.dejarseguirUsuario()

        """ Si el usuario ya dio like devuelva la cantidad de follows que tiene la publicacion """
        if query.filter(follower_user_id = self.request.user).exists():
            result['followers'] = str(cantidadFollows)
            listado.append(result)
            listado = json.dumps(listado)
            return JsonResponse({'result': listado} )
        """ Si no ha dado like se crea el objeto y se suma + 1 """
        query.create(followed_user_id = self.usuario, follower_user_id =usuario_follower)
        cantidadFollows += 1
        result['followers'] = str(cantidadFollows)
        listado.append(result)
        listado = json.dumps(listado)

        return JsonResponse({'result': listado} )

    def get_success_url(self) → str:
        return reverse('detalle-creador', kwargs={'slug':self.usuario.slug})

```

Figura 141. Vista seguirusuario y sus funciones.

Una vez creado el modelo y las vistas fue posible además, crear métodos para el usuario que facilitan el trabajo de traer el listado de los seguidores, seguir a otros usuarios y dejar de seguirlos como se observa a continuación:

```

def get_user_followers(self):
    seguidores = [object_list.follower_user_id for object_list in SeguidorUsuario.objects.filter(followed_user_id=self).order_by(
    return seguidores

def get_user_followed(self):
    seguidos = [object_list.followed_user_id for object_list in SeguidorUsuario.objects.filter(follower_user_id=self).order_by('
    return seguidos

def seguirUsuario(self):
    return reverse('follow-usuario', kwargs={'slug':self.slug})

def dejarseguirUsuario(self):
    return reverse('unfollow-usuario', kwargs={'slug':self.slug})

```

Figura 142. Métodos seguidores.

En este caso para seguir o dejar de seguir, se utilizó la función reverse para que automáticamente devuelva la URL que haría el llamado a la vista del AJAX dependiendo de la funcionalidad que se requiera, de esta forma el código se mantiene con los patrones de programación DRY (No te repitas en inglés) y se hace el llamado de la función dentro de los

templates. Estos últimos fueron creados dentro del paquete de templates, en el subdirectorio de “social/users”, manteniendo todas los templates de la red social en un solo lugar.

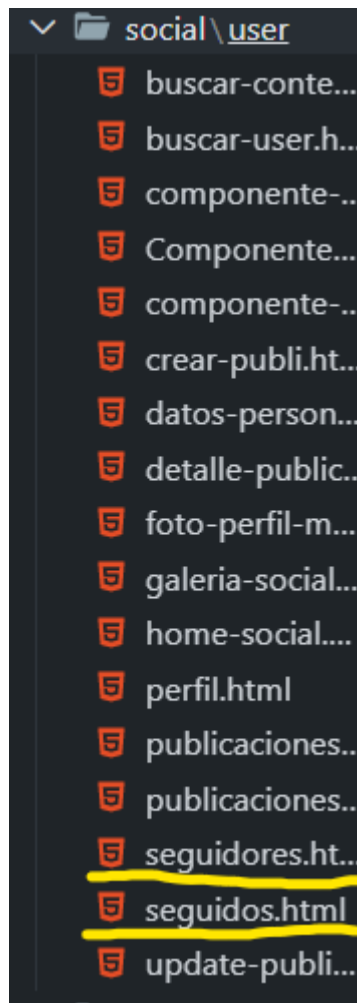


Figura 143. Templates seguidores y seguidos.

Después de lo anterior, se ingresan las vistas realizadas al módulo de urls, dentro del paquete social para empezar a utilizar las vistas (**Figura 144**).

```
path('<str:slug>/unfollow', DejarSeguirUsuario.as_view(), name='unfollow-usuario'),  
path('<str:slug>/followers', SeguidoresUsuario.as_view(), name='followers-usuario'),  
path('<str:slug>/followed', SeguidosUsuario.as_view(), name='followed-usuario'),
```

Figura 144. URLS aplicación chat.

Una vez realizadas las funcionalidades de seguidores se pudo dar paso a la codificación de los canales de mensajería o chats. Puesto que ya la aplicación fue creada, el primer proceso

fue crear los modelos necesarios para almacenar los hilos y los mensajes. Estos se realizaron dentro del paquete “*models.py*” en el directorio de chat.

```
from django.db import models
from django.contrib.auth import get_user_model
from django.db.models import Q
from django.urls import reverse
from django.core.exceptions import ValidationError
from django.utils import timezone

User = get_user_model()

# Create your models here.

class ThreadManager(models.Manager):
    def by_user(self, **kwargs):
        user = kwargs.get('user')
        lookup = Q(first_person=user) | Q(second_person=user)
        qs = self.get_queryset().filter(lookup).distinct()
        return qs
```

Figura 145. Modelo threadmanager.

Inicialmente en los modelos se importan las librerías necesarias para crear cada clase. Posteriormente se crea la clase inicial llamada *ThreadManager*, la cual se usara para realizar peticiones personalizadas desde el ORM, dentro de los Hilos (Threads). Para esto se hereda de la clase *models.Manager* y se crea el metodo “*by_user*”, el cual filtra el queryset por los usuarios vinculados al Hilo.

```
class Thread(models.Model):
    first_person = models.ForeignKey(User, on_delete=models.CASCADE, null=False, blank=False, related_name='thread_first_person')
    second_person = models.ForeignKey(User, on_delete=models.CASCADE, null=False, blank=False, related_name='thread_second_person')
    updated = models.DateTimeField(auto_now=False)
    closed_by_first_user = models.DateTimeField(auto_now=False, auto_now_add=False)
    closed_by_second_user = models.DateTimeField(auto_now=False, auto_now_add=False)
    timestamp = models.DateTimeField(auto_now_add=True)

    objects = ThreadManager()
    class Meta:
        unique_together = ['first_person', 'second_person']

    def get_absolute_url(self):
        return reverse('thread', kwargs={'pk': self.pk})

    def save(self, *args, **kwargs):
        if not self.pk:
            thread = Thread.objects.filter((Q(first_person=self.first_person) & Q(second_person=self.second_person)) | (Q(first_per
            if thread:
                raise ValidationError('Ya existe un chat entre estos dos usuarios')
            return super(Thread, self).save(*args, **kwargs)
```

Figura 146. Modelo thread.

Posteriormente, se crea la clase “Thread” o hilo en la cual se creará la tabla necesaria para crear los canales de chat. La clase contiene los siguientes atributos:

- **first_person:** Identificado como un ForeignKey, define el primer integrante en hacer parte del hilo. Proviene del modelo User, cuenta con on_delete, el cual especifica si el

usuario es eliminado, el registro también se elimina. No puede ser nulo ni quedar en blanco.

- **second_person:** Se identifica como un ForeignKey, define al segundo integrante en hacer parte del hilo. Proveniente del modelo User, cuenta con on_delete, el cual especifica si el usuario es eliminado, el registro también se elimina. No puede ser nulo ni quedar en blanco.
- **updated:** Identificado como un DateTimeField y contiene la variable auto_now en false, puesto que no se requiere que se actualice.
- **closed_by_first_user:** Se establece como un DateTimeField el cual contiene las variables de auto_now en false y auto_now_add en false. Se encarga de cerrar el hilo por parte del primer integrante del hilo y no agrega a otro integrante en caso de contar con 1.
- **closed_by_second_user:** Establecido como un DateTimeField el cual contiene las variables de auto_now en false y auto_now_add en false. Se encarga de cerrar el hilo por parte del segundo integrante del hilo y no agrega a otro integrante en caso de contar con 1.
- **timestamp:** Timestamp se definió como un DateTimeField, con la variable de auto_now_add en True. Utilizado para marcar el momento exacto de creación del hilo.
- **objects:** se define como un ThreadManager y se encarga de hacer una consulta personalizada entre los 2 usuarios del hilo.

Además se crearon los siguientes métodos dentro la clase:

- **save:** Evita la creación de hilos duplicados entre los mismos dos usuarios en la aplicación web.
- **get_absolute_url:** Se encarga de almacenar la URL del hilo creado.


```

class ChatMessage(models.Model):
    thread = models.ForeignKey(Thread, null=True, blank=True, on_delete=models.CASCADE, related_name='chatmessage_thread')
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    message = models.TextField(max_length=650)
    timestamp = models.DateTimeField(auto_now_add=True)
    class Meta:
        ordering = ["timestamp"]

    def save(self, *args, **kwargs):
        if not self.message:
            raise ValidationError('Mensaje vacío')
        elif self.message.strip() == '':
            raise ValidationError('Mensaje vacío')
        else:
            if self.user == self.thread.first_person or self.user == self.thread.second_person:
                return super(ChatMessage, self).save(*args, **kwargs)
            else:
                raise ValidationError('El usuario no pertenece al thread')

```

Figura 147. Modelo chatmessage.

Luego se creó el modelo de *ChatMessage*, el cual almacena todos los mensajes vinculados a un hilo. Este contiene los siguiente atributos:

- **thread:** Es el medio por el cual será enviado el mensaje. Está establecido como un ForeignKey, puede ser nulo y estar vacío, además al ser eliminado, se eliminará todo registro gracias al `on_delete` con `models.CASCADE`.
- **user:** Identificado como un ForeignKey. Hace referencia al usuario que envía el mensaje.
- **message:** Establecido como un TextField, cuenta con una longitud máxima de 650 caracteres y representa el mensaje que el usuario va enviar.
- **timestamp:** Identificado como un DateTimeField, con la variable de `auto_add_now` en true, Se encarga de definir el momento exacto de creación del mensaje.

Además, se sobrescribió la función `save` que viene de la clase `Model` con el fin de validar si el mensaje viene vacío, o si el usuario pertenece al canal.

Luego como se puede observar en la siguiente figura (**Figura 147**), se crean las vistas que suplen las peticiones del chat, en este caso se pone como ejemplo la vista de crear un nuevo hilo. Sin embargo, dentro del módulo se presentan vistas para listar los chats, eliminarlos, entre otros.

```

class ThreadCreateView(ExtendsInnerContentMixin, LoginRequiredMixin, CreateView):
    model = Thread
    form_class = ThreadForm
    template_name = 'chat/thread-crear.html'

    def form_valid(self, form):
        cleaned_data = form.cleaned_data
        user = cleaned_data['second_person']
        thread = Thread.objects.filter((Q(first_person = user) & (Q(second_person = self.request.user))) | (Q(first_person = self
        if thread:
            # AGREGAR LINK PARA IR AL DETALLE
            link_filtrar = reverse('chat')
            messages.add_message(self.request, 40, mark_safe(f'Error, ya hay un chat con esta persona <a href="{thread[0].get_abs
            return super().form_invalid(form)

        self.object = form.save(commit = False)
        self.object.first_person = self.request.user
        self.object.save()
        return super().form_valid(form)

    def get_success_url(self, **kwargs) -> str:
        return reverse('thread', kwargs={'pk':self.object.pk})

```

Figura 148. Vista threadcreatview.

La vista hereda de la clase `CreateView` de las vistas genéricas de Django y además utiliza unos mixins, que validan si el usuario está autenticado y sobrescriben algunos métodos internos de la clase `CreateView`. A la vista, se le agrego un formulario personalizado llamado ***ThreadForm*** el cual ayuda a crear nuevos hilos, por otro lado dado que se creó el método ***“form_valid”*** se valida el formulario desde la vista, luego de que se realiza la petición POST. Una vez el formulario se haya validado, si el resultado es satisfactorio, el método ***“get_success_url”*** retorna una vista de redireccionamiento.

```

import json
from channels.consumer import AsyncConsumer
from channels.db import database_sync_to_async
from django.contrib.auth import get_user_model
from django.db.models import Q
from datetime import datetime

from chat.models import Thread, ChatMessage

User = get_user_model()

```

Figura 149. Configuración models consumer.

Para enviar mensajes dentro de los canales fue necesario crear canales asíncronos, esto a través de la librería `channels` de Django, de la cual se importa la clase `AsyncConsumer`. Esta última, permite utilizar los métodos para crear conexiones entre usuarios.

```

class ChatConsumer(AsyncConsumer):

    async def websocket_connect(self, event):
        user = self.scope['user']
        thread_id = self.scope['url_route']['kwargs']['pk']
        chat_room = f'user_chatroom_{user.id}'
        self.chat_room = chat_room

        thread = await self.user_in_thread(user.id, thread_id)
        if thread:
            await self.channel_layer.group_add(
                chat_room,
                self.channel_name
            )
            await self.send({
                'type': 'websocket.accept'
            })
        else:
            await self.websocket_disconnect(event)

```

Figura 150. Clase chat y métodos.

Dentro de la clase *ChatConsumer* se crea el canal asíncrono al cual se le definieron métodos para conectarse al websocket, enviar los mensajes, recibirlos, validar los usuarios, entre otros.

```

from django.urls import path
from . import consumers

websocket_urlpatterns = [
    path('chat/<int:pk>/', consumers.ChatConsumer.as_asgi()),
]

```

Figura 151. URLs consumer.

Luego, se agrega una url al módulo routing que se encargará de manejar la ruta de los mensajes asíncronos.

```

from django.urls import path
from .views import *

urlpatterns = [
    path('chat/', ThreadListView.as_view(), name='chat'),
    path('chat/create', ThreadCreateView.as_view(), name='chat-create'),
    path('chat/<int:pk>/', ThreadDetailView.as_view(), name='thread'),
    path('chat/<int:pk>/update', ThreadDeleteView.as_view(), name='thread-update'),
    path('chat/filter/', ThreadFilterView.as_view(), name='thread-filter')
]

```

Figura 152. URLs thread.

Las urls dentro del paquete de chats, se configuraron como se muestra en la **Figura 152**. De esta forma, ya se puede acceder a todas las funcionalidades del chat. Sin embargo, para

visualizarlo es necesario crear los templates dentro de cada vista. Una vez creados los templates en el paquete de templates de chat, se observa el árbol de directorios en la siguiente figura:

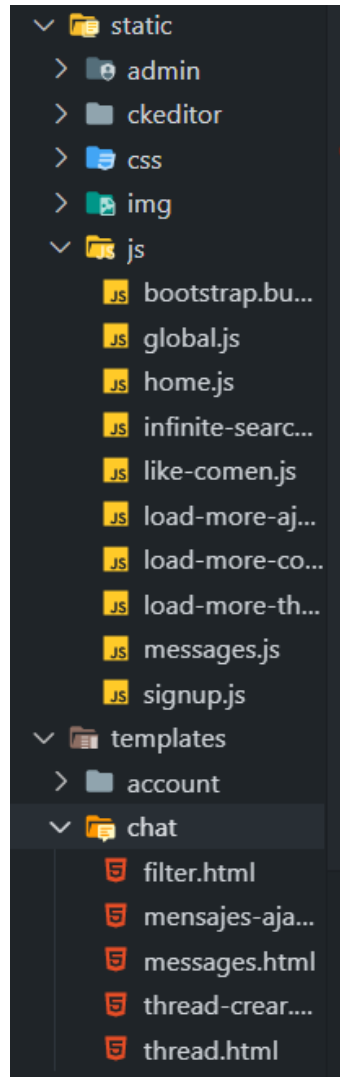


Figura 153. Organización de templates y archivos javascript.

Una vez finalizado el tercer sprint se muestra el resultado del desarrollo en las siguientes imágenes.

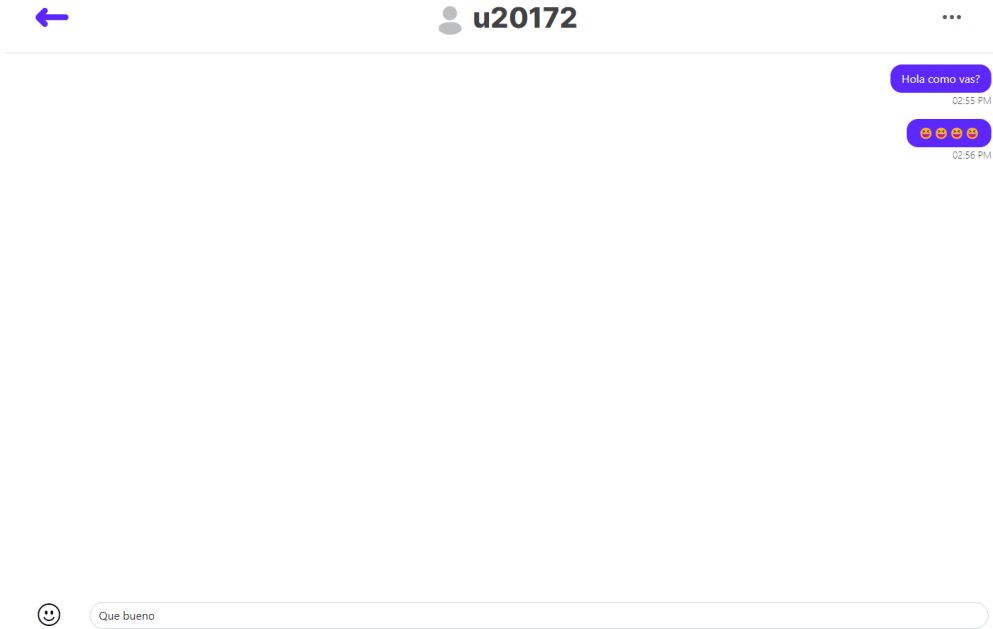


Figura 154. Vista chat.

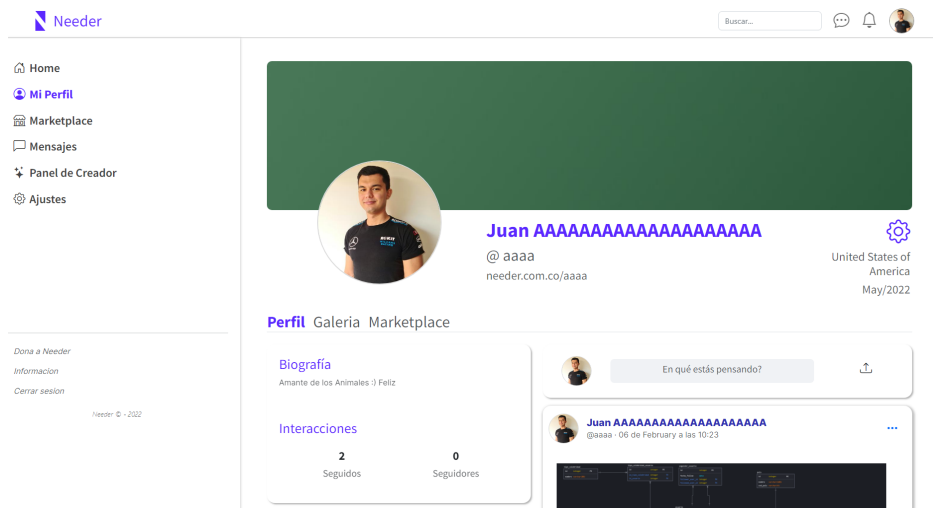


Figura 155. Vista perfil seguidos.

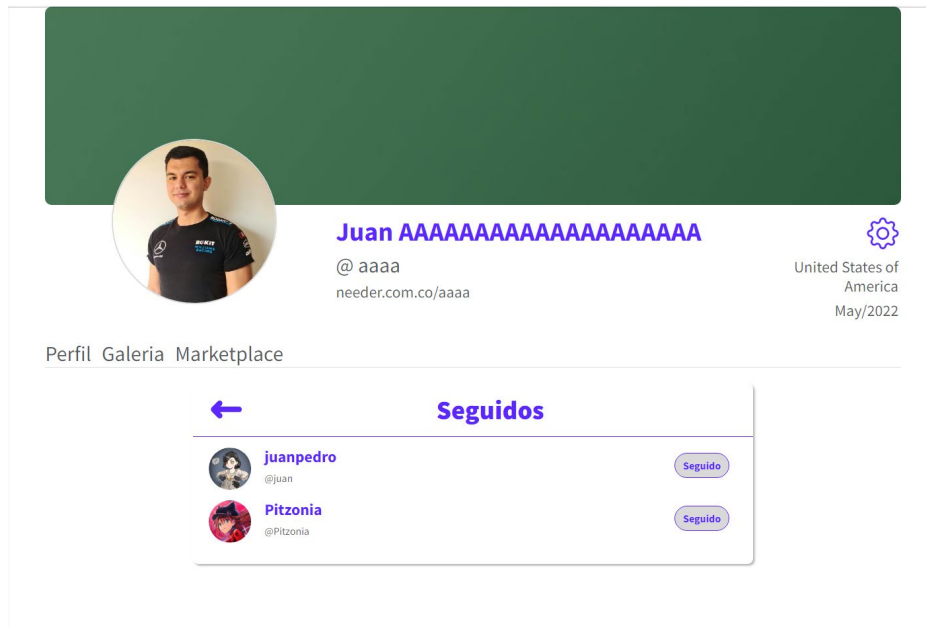


Figura 156. Vista usuarios seguidos.

6.4.4 Testing

Culminada la fase del sprint, se continúa el proceso con la etapa de testing. Esta fase fue aplicada a las historias de usuario más importantes de cada Sprint, estas historias de usuario son:

- U-1 Registro de Usuario (Sprint 1)
- U-2 Autenticación de Usuario (Sprint 1)
- U-3 Actualización de información Personal (Sprint 1)
- U-6 Creación de Publicación (Sprint 1)
- U-14 Comentar una publicación (Sprint 2)
- U-36 Mensaje entre usuarios (Sprint 3)

6.4.4.1 Plan de Pruebas

Inicialmente para desarrollar las pruebas, el primer paso fue realizar los respectivos planes de pruebas para cada una de las historias de usuario. Este plan de pruebas está realizado bajo una estructura que se encarga de definir todo el procedimiento a realizar dentro del escenario. Este plan de pruebas contiene los elementos observados en la siguiente tabla:

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	COND
R-1	1	Registro en la plataforma realizado de manera correcta	Se realiza el registro de usuario en la plataforma, ingresando todos los datos de manera correcta	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais, Documento)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario será registrado en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-2	2	Registro en la plataforma - Problemas con el Documento	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Documento o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais), el dato Documento se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Documento)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-3	3	Registro en la plataforma - Problemas con el Email	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Email o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Password, Username, First_name, Last_name, Pais), el dato Email se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Email)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-4	4	Registro en la plataforma - Problemas con la Password	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea las Password o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Username, First_name, Last_name, Pais), el dato Password se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Password)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-5	5	Registro en la plataforma - Problemas con el Username	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Username o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , First_name, Last_name, Pais), el dato Username se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Username)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-6	6	Registro en la plataforma - Problemas con el First_name	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el First_name o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , Last_name, Pais), el dato First_name se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato First_name)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-7	7	Registro en la plataforma - Problemas con el Last_name	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Last_name o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , First_name, Pais), el dato Last_name se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Last_name)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha

R-8	8	Registro en la plataforma - Problemas con el Pais	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Pais o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , First_name, Last_name), el dato Pais se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Pais)	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-9	9	Registro en la plataforma - Duplicación de datos	Se realiza el registro de usuario en la plataforma, ingresando los mismos datos utilizados en el registro de otra cuenta	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais, Documento)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha
R-10	10	Registro en la plataforma - Formulario vacío	Se realiza el registro de usuario en la plataforma, sin ingresar ningún dato en el formulario	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	No se ingresan datos	Se enviará el formulario completamente vacío	El usuario no se pudo registrar en la plataforma	No aplica	En espera de procedimiento	No realizado hasta la fecha

Tabla 61. Plan de Pruebas HU U-1.

La tabla anterior muestra el plan de pruebas de la historia de usuario U-1 y está conformada por los siguientes elementos:

- **ID:** El ID es un código creado para poder identificar y cuantificar la cantidad de tests a realizar en un plan de pruebas, está conformado por la inicial de la funcionalidad de la historia de usuario y un número, dependiendo del orden de creación. En este caso, la funcionalidad se llama Registro de Usuario, dando como resultado la R del ID.
- **Número de escenarios:** Se encargan de enumerar la cantidad de tests a realizar en un plan de pruebas. En este caso, se presentan 10 escenarios, dando a entender que se deben realizar 10 tests en este plan de pruebas.
- **C.D.P o Caso de Prueba:** Es un título descriptivo que se encarga de nombrar el escenario.
- **Descripción:** Se encarga de describir el Caso de Prueba, indicando sus elementos, sus funcionalidades y las acciones que se van a realizar en la prueba.
- **Fecha:** Fecha de creación de escenario.

- **A.F o Área funcional:** Describe el área funcional al cual está asociado el escenario.
- **Funcionalidad:** Describe la funcionalidad que se está probando en el plan de pruebas. En este ejemplo, la funcionalidad a testear es el registro de usuario.
- **D.I o Datos de Ingreso:** En este campo se especifican todos los datos de entradas necesarios para validar la prueba del escenario. Un ejemplo claro es el escenario 10, el cual especifica el envío del formulario completamente vacío, no cuenta con datos de entrada.
- **A.R o Acciones a Realizar:** Se encarga de especificar las acciones que se deben realizar para poder ejecutar de manera correcta la prueba.
- **R.E o Resultados Esperados:** Como su nombre lo indica, son los resultados que se esperan obtener al ejecutar todos los pasos de la prueba de manera correcta.
- **REQ.E o Requerimientos Especiales:** Especifica cualquier condición de hardware o software necesaria para ejecutar la prueba.
- **R.O o Resultados Obtenidos:** Son los resultados obtenidos al ejecutar la prueba. En este caso, al no estar todavía implementado, cuenta con una condición de *“En espera de procedimiento”*.
- **Condición:** Menciona la condición en la que se encuentra el escenario. En este ejemplo, al no estar todavía ejecutado, se encuentra en estado de *“No realizado hasta la fecha”*.

Para una explicación más detallada sobre el proceso realizado en estas historias de usuario, se decidió incluir el plan de pruebas de las historias de usuario U-2 y U-6. A continuación, se mostrarán los planes de pruebas de las historias de usuario U-2 y U-6:

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	COND
----	---	-------	------	------	-----	-----	-----	-----	-----	-------	-----	------

A-1	1	Autenticación de usuario confirmada de manera correcta	Se realiza la autenticación de datos, permitiendo el ingreso al usuario a la plataforma	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email, Password)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario será autenticado en la plataforma, accediendo a las funcionalidades de esta.	No aplica	En espera de procedimiento	No realizado hasta la fecha
A-2	2	Autenticación de usuario - Sin cuenta activa	Se realiza la autenticación de datos, sin contar con una cuenta activa en la plataforma	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email, Password)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	En espera de procedimiento	No realizado hasta la fecha
A-3	3	Autenticación de usuario - Problemas con la Password	Se realiza la autenticación de datos, presentando problemas con la Password (Dato ingresado de manera errónea o vacío)	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email) el dato Password se ingresa de manera errónea o no se ingresa	Se enviará el formulario con los datos solicitados de manera correcta (menos el dato Password)	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	En espera de procedimiento	No realizado hasta la fecha
A-4	4	Autenticación de usuario - Problemas con el Email	Se realiza la autenticación de datos, presentando problemas con el Email (Dato ingresado de manera errónea o vacío)	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Password) el dato Email se ingresa de manera errónea o no se ingresa	Se enviará el formulario con los datos solicitados de manera correcta (menos el dato Email)	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	En espera de procedimiento	No realizado hasta la fecha
A-5	5	Autenticación de usuario - Formulario vacío	Se realiza la autenticación del formulario sin datos ingresados	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	No se ingresan datos	Se enviará el formulario completamente vacío	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	En espera de procedimiento	No realizado hasta la fecha

Tabla 62. Plan de Pruebas HU U-2.

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	CO
CRP-1	1	Crear una Publicación - realizado de manera correcta	Se realiza la creación de la publicación de manera correcta	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, Archivo, NSFW)	Se enviará el formulario con todos los datos solicitados de manera correcta	La publicación ha sido creada de manera exitosa	No aplica	En espera de procedimiento	No realizado hasta la fecha
CRP-2	2	Crear una Publicación - Sin un Archivo	Se realiza la creación de la publicación sin contener un Archivo	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, NSFW)	Se enviará el formulario con una descripción pero sin un archivo	La publicación ha sido creada de manera exitosa	No aplica	En espera de procedimiento	No realizado hasta la fecha
CRP-3	3	Crear una Publicación - Sin una Descripción	Se realiza la creación de la publicación sin contener una Descripción	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Archivo, NSFW)	Se enviará el formulario con un archivo pero sin una descripción	La publicación ha sido creada de manera exitosa	No aplica	En espera de procedimiento	No realizado hasta la fecha
CRP-4	4	Crear una Publicación - Problemas con el Archivo	Se realiza la creación de la publicación, pero ingresando de manera errónea el Archivo	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, Archivo, NSFW)	Se enviará el formulario con una descripción pero con archivo no permitido	La publicación no ha sido creada	No aplica	En espera de procedimiento	No realizado hasta la fecha

CRP-5	5	Crear una Publicación - Formulario vacío	Se realiza la creación de la publicación, sin datos ingresados	11/02/2023	Crear una Publicación	Creación de una publicación	No se ingresan datos	Se enviará el formulario totalmente vacío	La publicación no ha sido creada	No aplica	En espera de procedimiento	No realizado hasta la fecha
-------	---	--	--	------------	-----------------------	-----------------------------	----------------------	---	----------------------------------	-----------	----------------------------	-----------------------------

Tabla 63. Plan de Pruebas HU U-6.

6.4.4.2 Pruebas Unitarias

Una vez desarrollado el plan de pruebas, se da inicio a las pruebas unitarias. Para este caso, se explicarán las pruebas unitarias de la historias de usuario U-1, U-2 y U-6 como se había mencionado anteriormente. Para el desarrollo de las pruebas unitarias, se debe tener en cuenta la cantidad de escenarios presentados en los planes de pruebas de las historias de usuario. Estos escenarios son indispensables, puesto que mencionan el proceso que se debe realizar en la prueba unitaria, para que su validación sea exitosa.

6.4.4.2.1 Pruebas Unitarias U-1

Se da inicio al desarrollo de las pruebas unitarias con la historia de usuario U-1, la cual cuenta con la funcionalidad llamada **“Registro de usuario”** y 10 escenarios, pero se explicarán las pruebas unitarias de los escenarios 1, 2, 9 y 10. Para el caso de las pruebas unitarias de los escenarios restantes, se podrán ver reflejadas dentro del archivo `test.py` en el paquete de **Users**. Para poder realizar las pruebas unitarias, se creó un archivo llamado **test.py** en la carpeta de **Users**, la cual contiene todo lo relacionado con el usuario.

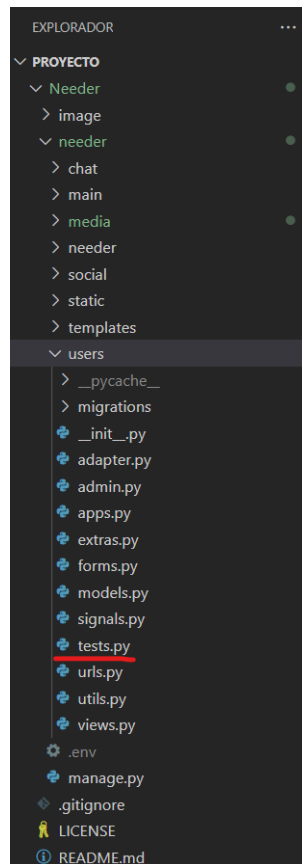


Figura 157. Organización de las carpetas del proyecto.

Dentro de este módulo, se importa la clase de la librería unittest de Python, la cual es reutilizada por Django para realizar test llamada *TestCase*. Luego se importa del módulo *forms.py* del paquete *Users* el formulario de registro llamado *SignupCustomForm*. Después, se importa del módulo *models.py* las clases de *Pais* y *User*. Posterior a esto, se importa la función *patch* de la librería de *Mock*, esta librería se utiliza con el fin de poder imitar objetos reales en el código. La función *patch* es utilizada para reemplazar objetos reales del código con instancias de *Mock*. Por último, se importa la clase *RecaptchaResponse* del módulo *captcha.client*. Esta clase se utiliza para representar la respuesta a un *reCAPTCHA*.

```
Needer > needer > users > tests.py > ...
1  from django.test import TestCase
2  from .forms import SignupCustomForm
3  from .models import Pais, User
4  from unittest.mock import patch
5  from captcha.client import RecaptchaResponse
```

Figura 158. Imports prueba unitaria HU U-1.

Una vez se tengan los elementos necesarios para el desarrollo de las pruebas unitarias, se crea la clase *TestUserForm*, la cual hereda la clase *TestCase* de Django y se inicia el proceso con el escenario 1.

Escenario 1

Para este escenario, la prueba debe enviar el formulario de registro con todos los datos correctos. Sabiendo esto, la prueba unitaria utiliza la función decoradora *patch* aplicada a la función del escenario *test_registro_plataforma* para simular el comportamiento de la función *submit* del módulo *captcha.fields.client*. Esta simulación es almacenada en la variable *mocked_submit* la cual es pasada como segundo parámetro al método *test_registro_plataforma*.

Este método comprueba que el registro de un usuario se realiza correctamente en la plataforma. Para esto, se crea un objeto *Pais* con el nombre *Colombia*. Luego, se establece el valor *is_valid=True* para la variable *mocked_submit*. Esto garantiza que la prueba no falle, debido a un error en la validación del *reCAPTCHA*. Posterior a esto, se crea un objeto llamado *SignupCustomForm*, el cual recibe los datos de prueba. Finalmente, se utiliza el método *assertTrue()* de la clase *TestCase*, con el fin de comprobar que el formulario es válido.

```
12 class TestUserForm(TestCase):
13     @patch("captcha.fields.client.submit")
14     def test_registro_plataforma(self, mocked_submit):
15         """ R-1 Registro en la plataforma de forma correcta """
16         Pais.objects.create(nombre='Colombia')
17         mocked_submit.return_value = RecaptchaResponse(is_valid=True)
18         form = SignupCustomForm(data={
19             'username': 'Juan',
20             'email': 'puentesjs903@gmail.com',
21             'first_name': 'Juan',
22             'last_name': 'Puentes',
23             'password1': 'sebas_111@',
24             'password2': 'sebas_111@',
25             'pais': 'Colombia',
26             'num_documento': '1075316291',
27             'captcha': 'PASSED'
28         })
29         self.assertTrue(form.is_valid())
```

Figura 159. Prueba unitaria U-1 escenario 1.

Escenario 2

Para este escenario, la prueba debe enviar el formulario de registro con todos los datos correctos, menos el número de identificación. Teniendo en cuenta esto, se realiza el mismo procedimiento inicial de la prueba unitaria anterior. El método *test_registro_plataforma_error_documento* comprueba que se produce un error en el formulario cuando el número de documento del usuario no es un valor numérico, entre 6 y 10 dígitos. Luego, se crea el objeto *Pais*, el cual contiene un valor igual a *Colombia*.

Este objeto es utilizado para validar el dato de *Pais* que se encuentra dentro del formulario de la clase *SignCustomForm*. Posterior a esto, se establece el valor *is_valid=True* para la variable *mocked_submit*. Esto garantiza que la prueba no falle, debido a un error en la validación del *reCAPTCHA*.

Por último, se crea un objeto llamado *SignCustomForm*, el cual recibe los datos de prueba que incluyen un número de documento no válido (*aaaaa9810*). Por último, se utiliza el método *assertTrue()* de la clase *TestCase* para comprobar si se produce un error en el campo *num_documento* del formulario.

```
32 @patch("captcha.fields.client.submit")
33 def test_registro_plataforma_error_documento(self, mocked_submit):
34     """ R-2 Registro en la plataforma con error en el documento vacío """
35     Pais.objects.create(nombre='Colombia')
36     mocked_submit.return_value = RecaptchaResponse(is_valid=True)
37     form = SignupCustomForm(data={
38         'username': 'Juan',
39         'email': 'JuanDavid123@gmail.com',
40         'first_name': 'Juan',
41         'last_name': 'Trujillo',
42         'password1': 'Juan_111@',
43         'password2': 'Juan_111@',
44         'pais': 'Colombia',
45         'num_documento': 'aaaaa9810',
46         'captcha': 'PASSED'
47     })
48
49     self.assertEqual(
50         form.errors['num_documento'], ['Error en el numero de documento debe contener solo numeros (De 6 - 10 numeros).']
51     )
```

Figura 160. Prueba unitaria U-1 escenario 2.

Escenario 9

Para este escenario, la prueba debe enviar el formulario de registro con todos los datos correctos, pero presentando el inconveniente de contar con datos duplicados, es decir, se repiten los datos del formulario del registro de usuario con los de una cuenta previamente creada. Teniendo en cuenta esto, se realiza el mismo procedimiento inicial de la prueba unitaria anterior. Repitiendo el mismo procedimiento también con el objeto *Pais*.

Una vez realizado esto, se crea un objeto llamado *User*, utilizando el modelo *User* de Django. Este objeto cuenta con datos específicos, los cuales serán utilizados para realizar una comparación con los datos del objeto *SignupCustomForm*. Este objeto contiene los mismos datos específicos del objeto *User*, los datos a comparar son *email*, *username* y *documento*. Por último, se prueba que el formulario no sea válido, mediante el método *assertFalse* de la clase *TestCase*. Como resultado, se debe obtener un *False* por parte del *form.is_valid()* para que la prueba pase de manera correcta.

```
173 @patch("captcha.fields.client.submit")
174 def test_registro_plataforma_datos_duplicados(self, mocked_submit):
175     """ R-9 Datos duplicados """
176     pais = Pais.objects.create(nombre='Colombia')
177     mocked_submit.return_value = RecaptchaResponse(is_valid=True)
178     User.objects.create(
179         username='thomas',
180         email='thomas123@gmail.com',
181         first_name='Thomas',
182         last_name='Perez',
183         password='thomas_@1998',
184         pais = pais,
185         num_documento='1075319499'
186     )
187     form = SignupCustomForm(data={
188         'username': 'thomas',
189         'email': 'thomas123@gmail.com',
190         'first_name': 'Thomas',
191         'last_name': 'Perez',
192         'password1': 'thomas_@1998',
193         'password2': 'thomas_@1998',
194         'pais': 'Colombia',
195         'num_documento': '1075319499',
196         'captcha': 'PASSED'
197     })
198     self.assertFalse(form.is_valid())
```

Figura 161. Prueba unitaria U-1 escenario 9.

Escenario 10

Para el escenario 10, la prueba debe enviar el formulario de registro completamente vacío. Teniendo en cuenta esto, la prueba unitaria crea el método *test_registro_plataforma_sin_datos*, el cual contiene un formulario llamado *SignupCustomForm* completamente sin parámetros, es decir, no tiene información dentro del formulario. Una vez creado el formulario vacío, se procede a utilizar el método *assertFalse*, de la clase *TestCase* de Django, con el fin de que al retornar un valor falso, el formulario sea válido y la prueba pase de manera exitosa.

```

200 def test_registro_plataforma_sin_datos(self):
201     """ R-10 Se envia un formulario vacio """
202
203     form = SignupCustomForm(data={
204         'username': '',
205         'email': '',
206         'first_name': '',
207         'last_name': '',
208         'password1': '',
209         'password2': '',
210         'pais': '',
211         'num_documento': ''
212     })
213     self.assertFalse(
214         form.is_valid()
215     )

```

Figura 162. Prueba unitaria U-1 escenario 10.

6.4.4.2.2 Pruebas Unitarias U-2

Se da inicio al desarrollo de las pruebas unitarias con la historia de usuario U-2, la cual cuenta con la funcionalidad llamada “*Autenticación de usuario*” y 5 escenarios, pero se explicarán las pruebas unitarias de los escenarios 1, 2 y 3. Para el caso de las pruebas unitarias de los escenarios restantes, se podrán ver reflejadas dentro del archivo *test.py* en el paquete de *Users*. Para poder realizar las pruebas unitarias, se utilizó el módulo llamado *test.py* del paquete *Users*, utilizado para realizar las pruebas unitarias de la U-1.

Dentro de este módulo, se importa la clase de la librería unittest de Python, la cual es reutilizada por Django para realizar test llamada *TestCase*. También se importa la clase Client, la cual proporciona una interfaz para hacer solicitudes HTTP y obtener una respuesta en los casos de prueba. Luego se importa el método *reverse*, el cual se encuentra en el módulo Django.urls.

Este método trae el enlace almacenado de la *url* que se encuentra ubicada en el archivo *urls*. Siguiendo con el proceso, se importan los *settings* con el fin de hacer uso de las configuraciones que tiene este archivo en el proyecto. Por último, se importa, de la librería *django-allauth*, el modelo *EmailAddress*, utilizado para almacenar emails asociados a los usuarios en la aplicación.


```

1  from django.test import TestCase, Client
2  from django.urls import reverse
3  from django.conf import settings
4  from allauth.account.models import EmailAddress
5

```

Figura 163. Imports prueba unitaria HU U-2.

Antes de empezar con el desarrollo de los escenarios, primero se debe preparar el ambiente necesario para las pruebas unitarias. Para esto, se crea la clase *TestUserForm*, la cual hereda la clase *TestCase* de Django. Luego se hace uso del método *setUp*, el cual contiene un diccionario de credenciales, denominado *self.credenciales*, este diccionario almacena el correo electrónico y la contraseña asociados a la cuenta.

Se inicializa la variable *self.client*, con el fin de imitar las acciones de un usuario. Luego, se crea una variable llamada *self.user*, la cual almacena los datos de los campos *email*, *password* y *username*. Después, se crea una variable llamada *self.email*, la cual contiene el objeto *user* y el *email* y se utiliza para la verificación del correo.

```

219 class TestUserLogin(TestCase):
220     def setUp(self):
221         self.credenciales = {
222             'login': 'Juandt903@gmail.com',
223             'password': 'david_1998',
224         }
225         self.client = Client()
226         self.user = User.objects.create_user(email='Juandt903@gmail.com', password='david_1998', username='juan')
227         self.email = EmailAddress.objects.create(user=self.user, email='Juandt903@gmail.com', verified=True, primary=True)
228         return super().setUp()

```

Figura 164. SetUp pruebas unitarias HU U-2.

Escenario 1

Para el escenario 1, se realiza la autenticación del usuario con todos los datos ingresados de manera correcta. Teniendo en cuenta esto, la prueba unitaria crea un método llamada *test_autenticacion_usuario*. En la primera línea, se utiliza el *self.client.post* para realizar una solicitud *HTTP POST* a la *URL* de inicio de sesión de la aplicación, utilizando el nombre de la vista *account_login* y las credenciales almacenadas en el atributo *credenciales*. Es importante resaltar que, la *URL reverse("account_login")* se construye a partir del nombre de la vista, en lugar de escribir la *URL* completa.

Luego, se utiliza el `self.assertRedirects` para verificar que la respuesta a la solicitud *HTTP* de inicio de sesión contiene una redirección y que la *URL* de redirección, es la *URL* definida en la variable `settings.LOGIN_REDIRECT_URL`. La opción `fetch_redirect_response=False`, se utiliza para evitar que se solicite la *URL* de redirección real.

```
229
230 def test_autenticacion_usuario(self):
231     """ A- 1 Ingreso a la plataforma """
232     response = self.client.post(reverse('account_login'),self.credenciales)
233     self.assertRedirects(response, settings.LOGIN_REDIRECT_URL, fetch_redirect_response=False)
```

Figura 165. Prueba unitaria U-2 escenario 1.

Escenario 2

Para el escenario 2, se realiza la autenticación del usuario con todos los datos ingresados de manera correcta, sin contar con una cuenta activa en la plataforma. En base a esto, la prueba unitaria crea un método llamado `test_autenticacion_usuario_sin_cuenta_activa` y se vuelve hacer uso del `self.client.post`, para realizar una solicitud *HTTP POST* a la *URL* de inicio de sesión de la aplicación, utilizando el nombre de la vista `account_login`. Para este escenario, se usan credenciales inválidas para simular un intento fallido de inicio de sesión.

Después de generar la solicitud, se utiliza la variable `validador_error` para almacenar un mensaje de error esperado, el cual indica que las credenciales especificadas no son correctas. Por último, se hace uso de la función `assert` para verificar que este mensaje de error está presente en el contenido de la respuesta *HTTP* de la solicitud. Solicitud obtenida, mediante el atributo `content`, este atributo es decodificado de bytes a una cadena de texto, utilizando el `decode("utf-8")`, con el fin de reconocer los caracteres especiales.

```
235 def test_autenticacion_usuario_sin_cuenta_activa(self):
236     """ A- 2 Autenticacion del usuario sin tener una cuenta activa """
237     response = self.client.post(reverse('account_login'),{"login":"no_estoy_registrado@gmail.com","password":"wrong"})
238     validador_error = 'El correo electrónico y/o la contraseña que se especificaron no son correctos.'
239     assert validador_error in response.content.decode('utf-8')
```

Figura 166. Prueba unitaria U-2 escenario 2.

Escenario 3

Para el escenario 3, se realiza la autenticación del usuario ingresando el dato de *email* de manera correcta, pero el dato de *password* de manera incorrecta. En base a esto, la prueba unitaria crea un método llamado `test_autenticacion_usuario_error_password`, después, se vuelve hacer uso del `self.client.post` para las peticiones *HTTP POST* a la *URL* de la aplicación,

haciendo uso del nombre de la variable `account_login`. Para esta prueba, se hace uso de una dirección de correo electrónico correcta y una contraseña incorrecta, simulando un intento fallido de inicio de sesión.

Después de realizar la solicitud, se vuelve a hacer uso de la variable `validador_error`, para el almacenamiento de un mensaje de error esperado. Posteriormente, se utiliza la función `assert`, para verificar que este mensaje de error está presente en el contenido de la respuesta `HTTP` de la solicitud realizada. Esta solicitud se obtiene mediante el atributo `content` y se decodifica de bytes a una cadena de texto, utilizando `decode("utf-8")`.

```
241 def test_autenticacion_usuario_error_password(self):
242     """ A-3 Error en la contraseña del usuario """
243     response = self.client.post(reverse('account_login'), {'login': 'Juandt903@gmail.com', 'password': 'a'})
244     validador_error = 'El correo electrónico y/o la contraseña que se especificaron no son correctos.'
245     assert validador_error in response.content.decode('utf-8')
```

Figura 167. Prueba unitaria U-2 escenario 3.

6.4.4.2.3 Pruebas Unitarias U-6

Se da inicio al desarrollo de las pruebas unitarias con la historia de usuario U-6, la cual cuenta con la funcionalidad llamada **“Crear Publicación”** y 5 escenarios, pero se explicarán las pruebas unitarias de los escenarios 1, 2 y 5. Para el caso de las pruebas unitarias de los escenarios restantes, se podrán ver reflejadas dentro del archivo `test.py` en el paquete de `Social`. Para poder realizar las pruebas unitarias, se creó un módulo llamado `test.py` en el paquete `Social`.

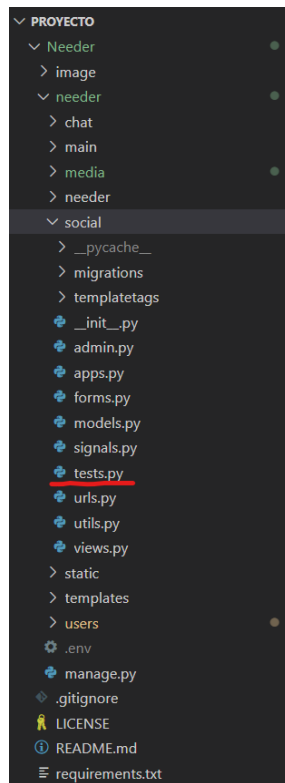


Figura 168. Organización de las carpetas del proyecto.

Dentro de este módulo, se importa la clase de la librería unittest de Python, la cual es reutilizada por Django para realizar test llamada *TestCase*. Luego se importa del módulo *forms.py* del paquete *Social*, el formulario llamado *CrearPublicacionForm*. Después, se importa de Django el módulo *SimpleUploadedFile*, el permite crear objetos que simulan un archivo subido a través de un formulario *HTML* en un servidor web.

```
1 from django.test import TestCase
2 from .forms import CrearPublicacionForm, CrearComentarios
3 from django.core.files.uploadedfile import SimpleUploadedFile
```

Figura 169. Imports prueba unitaria HU U-6.

Una vez se tengan los elementos necesarios para el desarrollo de las pruebas unitarias, se crea la clase *TestUserForm*, la cual hereda la clase *TestCase* de Django y se inicia el proceso con el escenario 1.

Escenario 1

Para el escenario 1, se crea la publicación, ingresando todos los datos de manera correcta. Teniendo en cuenta esto, la prueba unitaria crea un método llamado *test_crear_publicacion*,

luego, se crea el el objeto llamado archivo, el cual almacena un archivo llamado *default-user.png* y lo abre en modo binario, utilizando la función *open()*. Luego, se utiliza la clase *SimpleUploadedFile*, para crear un archivo cargado simulado con el contenido de *default-user.png*. Este objeto se guarda en la variable *file*.

Después, se crea una instancia del formulario *CrearPublicacionForm* y se le pasan dos diccionarios como argumentos. El primer diccionario, contiene la descripción de la publicación y un valor booleano, el cual indica si la publicación tiene contenido *NSFW (Not Safe For Work)*. El segundo diccionario, contiene el archivo simulado, creado anteriormente, y es utilizado para simular la carga del archivo desde un formulario web.

Finalmente, se llama al método *is_valid()* en el formulario, para validar la información suministrada. Para verificar que el formulario es válido, se utiliza el método *assertTrue()*, lo que indica que se espera un resultado positivo de la prueba.

```
7 class TestPublicacionForm(TestCase):
8     def test_crear_publicacion(self):
9         """ Crear una publicacion de manera correcta """
10        archivo = open('media/default-user.png', 'rb')
11        file = SimpleUploadedFile('default-user.png', content=archivo.read(), content_type='image/jpeg')
12        form = CrearPublicacionForm({'descripcion': 'Descripcion', 'nsfw': True}, {'archivo': file})
13
14        self.assertTrue(form.is_valid())
```

Figura 170. Prueba unitaria U-6 escenario 1.

Escenario 2

Para el escenario 2, se crea la publicación, ingresando una descripción pero no se ingresa un archivo. Teniendo en cuenta esto, la prueba unitaria crea un método llamado *test_crear_publicacion_sin_archivo*. Se crea una instancia del formulario *CrearPublicacionForm* sin un archivo de imagen adjunto, contando únicamente con los datos de *descripcion* y *nsfw*. Luego, se llama al método *is_valid()* del formulario para validar la información proporcionada. Para verificar que el formulario es válido, se utiliza el método *assertTrue()*, indicando que el resultado esperado de la prueba es verdadero.

```
17 def test_crear_publicacion_sin_archivo(self):
18     """ Crear una publicacion sin archivo """
19     form = CrearPublicacionForm(data={
20         'descripcion': 'Pruebas de testeo',
21         'NSFW': False})
22     self.assertTrue(form.is_valid())
```

Figura 171. Prueba unitaria U-6 escenario 2.

Escenario 5

Para el escenario 5, se envía el formulario para crear la publicación completamente vacío. Gracias a esto, la prueba unitaria crea un método llamado *test_crear_publicacion_sin_datos*. Para este caso, se crea una instancia del formulario *CrearPublicacionForm* con un diccionario vacío, lo que significa que no se proporciona ningún tipo de información a esta instancia. Se hace el llamado al método *is_valid()* del formulario, con el fin de verificar si los datos ingresados son aceptados. Para verificar si el formulario es válido o no, se utiliza el método *assertFalse()*, indicando que se espera que el resultado de la prueba sea falso.

```
30 def test_crear_publicacion_sin_datos(self):
31     """ Crear una publicacion de manera correcta """
32     form = CrearPublicacionForm(data={})
33     self.assertFalse(form.is_valid())
```

Figura 172. Prueba unitaria U-6 escenario 5.

6.4.4.3 Resultados

Para la obtención de los resultados, se realizaron las pruebas unitarias a todos los escenarios de las historias de usuario seleccionadas para esta etapa. Como se mencionó anteriormente, se explicó el procedimiento de algunos escenarios de las historias U-1, U-2 y U-6, pero el proceso abarcó todos sus escenarios. Los resultados obtenidos fueron los siguientes:

Resultados U-1

Para la historia de usuario U-1, se realizaron las pruebas unitarias para todos sus escenarios, es decir, se aplicaron las pruebas unitarias para los 10 escenarios que conforman su plan de pruebas. Como se puede observar en la siguiente imagen, a la hora de ejecutar las pruebas, la consola indica que se encontraron 10 tests a evaluar. Luego, procede a crear una base de datos de prueba, llamada default. El sistema identifica que no se encuentran problemas en el código y procede a ejecutar las pruebas. Al realizar las pruebas unitarias, se obtuvo como resultado que los 10 test fueron ejecutados en menos de un segundo y que no presentaron error alguno, entregando un OK. Esto quiere decir que todos los tests pasaron la prueba:

```

PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> python manage.py test users.tests.TestUserForm
Found 10 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 10 tests in 0.034s

OK
Destroying test database for alias 'default'...
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> 

```

Figura 173. Resultados de las pruebas unitarias HU U-1.

Teniendo como resultado que, los 10 escenarios pasaron las pruebas unitarias, se procede a actualizar el estado de estos escenarios en el plan de pruebas:

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	CO
R-1	1	Registro en la plataforma realizado de manera correcta	Se realiza el registro de usuario en la plataforma, ingresando todos los datos de manera correcta	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais, Documento)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario será registrado en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-2	2	Registro en la plataforma - Problemas con el Documento	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Documento o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais), el dato Documento se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Documento)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-3	3	Registro en la plataforma - Problemas con el Email	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Email o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Password, Username, First_name, Last_name, Pais), el dato Email se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Email)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-4	4	Registro en la plataforma - Problemas con la Password	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea las Password o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Username, First_name, Last_name, Pais), el dato Password se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Password)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-5	5	Registro en la plataforma - Problemas con el Username	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Username o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , First_name, Last_name, Pais), el dato Username se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Username)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa

R-6	6	Registro en la plataforma - Problemas con el First_name	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el First_name o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , Last_name, Pais), el dato First_name se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato First_name)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-7	7	Registro en la plataforma - Problemas con el Last_name	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Last_name o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , First_name, Pais), el dato Last_name se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Last_name)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-8	8	Registro en la plataforma - Problemas con el Pais	Se realiza el registro de usuario en la plataforma, ingresando de manera errónea el Pais o dejando vacía la casilla	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Documento, Email, Password , Username , First_name, Last_name), el dato Pais se ingresa de manera errónea o no se ingresa	Se enviará el formulario con todos los datos solicitados de manera correcta (menos el dato Pais)	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-9	9	Registro en la plataforma - Duplicación de datos	Se realiza el registro de usuario en la plataforma, ingresando los mismos datos utilizados en el registro de otra cuenta	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	Se ingresan los siguientes datos: (Email, Password, Username, First_name, Last_name, Pais, Documento)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
R-10	10	Registro en la plataforma - Formulario vacío	Se realiza el registro de usuario en la plataforma, sin ingresar ningún dato en el formulario	11/02/2023	Registro de usuario	Registro en la plataforma de Needer	No se ingresan datos	Se enviará el formulario completamente vacío	El usuario no se pudo registrar en la plataforma	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa

Tabla 64. Plan de pruebas actualizado HU U-1.

Resultados U-2

Se realizaron pruebas unitarias para todos los escenarios de la historia de usuario U-2, los cuales fueron 5. Al iniciar con el proceso de ejecución de las pruebas unitarias, la consola indica que se encontraron 5 pruebas a evaluar, como se puede observar en la imagen adjunta. Seguidamente, se crea una base de datos de prueba denominada default. Luego, el sistema identifica que el código no presenta problemas y procede a ejecutar las pruebas. Después de ejecutar las pruebas unitarias, se obtuvo como resultado que los 5 tests se ejecutaron en un segundo sin presentar errores y se entregó un OK. Esto significa que todas las pruebas se aprobaron correctamente:


```

PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> python manage.py test users.tests.TestUserLogin
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 1.035s

OK
Destroying test database for alias 'default'...
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> 

```

Figura 174. Resultados de las pruebas unitarias HU U-2.

Una vez se ha confirmado que los 5 escenarios han superado las pruebas unitarias, se procede a actualizar el estado de dichos escenarios en el plan de pruebas:

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	CO
A-1	1	Autenticación de usuario confirmada de manera correcta	Se realiza la autenticación de datos, permitiendo el ingreso al usuario a la plataforma	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email, Password)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario será autenticado en la plataforma, accediendo a las funcionalidades de esta.	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
A-2	2	Autenticación de usuario - Sin cuenta activa	Se realiza la autenticación de datos, sin contar con una cuenta activa en la plataforma	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email, Password)	Se enviará el formulario con todos los datos solicitados de manera correcta	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
A-3	3	Autenticación de usuario - Problemas con la Password	Se realiza la autenticación de datos, presentando problemas con la Password (Dato ingresado de manera errónea o vacío)	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Email) el dato Password se ingresa de manera errónea o no se ingresa	Se enviará el formulario con los datos solicitados de manera correcta (menos el dato Password)	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
A-4	4	Autenticación de usuario - Problemas con el Email	Se realiza la autenticación de datos, presentando problemas con el Email (Dato ingresado de manera errónea o vacío)	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	Se ingresan los siguientes datos: (Password) el dato Email se ingresa de manera errónea o no se ingresa	Se enviará el formulario con los datos solicitados de manera correcta (menos el dato Email)	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
A-5	5	Autenticación de usuario - Formulario vacío	Se realiza la autenticación del formulario sin datos ingresados	11/02/2023	Autenticación de Usuario	Ingreso en la plataforma	No se ingresan datos	Se enviará el formulario completamente vacío	El usuario no será autenticado, impidiendo su acceso a la plataforma.	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa

Tabla 65. Plan de pruebas actualizado HU U-2.

Resultados U-6

En la historia de usuario U-6 se llevaron a cabo pruebas unitarias para sus 5 escenarios. En la siguiente imagen, se puede observar cómo una vez iniciado el proceso de ejecución de las pruebas, la consola indica la presencia de 5 pruebas a evaluar. Seguido de esto, se crea una base de datos de prueba llamada default. Luego de verificar que no existen problemas en el código, el sistema procede a ejecutar las pruebas, entregando como resultado que los 5 test se ejecutaron en un segundo sin presentar ningún problema y se recibe una confirmación de OK. Dando a entender los 5 escenarios pasaron las pruebas de manera satisfactoria.

```
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> python manage.py test social.tests.TestPublicacionForm
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 0.004s

OK
Destroying test database for alias 'default'...
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> █
```

Figura 175. Resultados de las pruebas unitarias HU U-6.

Contando con los respectivos resultados de las pruebas unitarias y teniendo conocimiento previo de su aplicación en todos los escenarios, se procede a actualizar el estado de los escenarios en el plan de pruebas:

ID	#	C.D.P	DESC	FECH	A.F	FUN	D.I	A.R	R.E	REQ.E	R.O	CO
CRP-1	1	Crear una Publicación - realizado de manera correcta	Se realiza la creación de la publicación de manera correcta	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, Archivo, NSFW)	Se enviará el formulario con todos los datos solicitados de manera correcta	La publicación ha sido creada de manera exitosa	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
CRP-2	2	Crear una Publicación - Sin un Archivo	Se realiza la creación de la publicación sin contener un Archivo	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, NSFW)	Se enviará el formulario con una descripción pero sin un archivo	La publicación ha sido creada de manera exitosa	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
CRP-3	3	Crear una Publicación - Sin una Descripción	Se realiza la creación de la publicación sin contener una Descripción	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Archivo, NSFW)	Se enviará el formulario con un archivo pero sin una descripción	La publicación ha sido creada de manera exitosa	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
CRP-4	4	Crear una Publicación - Problemas con el Archivo	Se realiza la creación de la publicación, pero ingresando de manera errónea el Archivo	11/02/2023	Crear una Publicación	Creación de una publicación	Se ingresan los siguientes datos: (Descripción, Archivo, NSFW)	Se enviará el formulario con una descripción pero con archivo no permitido	La publicación no ha sido creada	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa

CRP-5	5	Crear una Publicación - Formulario vacío	Se realiza la creación de la publicación, sin datos ingresados	11/02/2023	Crear una Publicación	Creación de una publicación	No se ingresan datos	Se enviará el formulario totalmente vacío	La publicación no ha sido creada	No aplica	Se pasó la prueba de manera exitosa	Prueba realizada de manera exitosa
-------	---	--	--	------------	-----------------------	-----------------------------	----------------------	---	----------------------------------	-----------	-------------------------------------	------------------------------------

Tabla 66. Plan de pruebas actualizado HU U-6.

Las historias de usuario U-3, U-14 y U-36 también se han sometido al proceso de las pruebas unitarias y se obtuvieron resultados satisfactorios. Para la U-3, se llevaron a cabo pruebas unitarias en sus 6 escenarios y se confirmó que todos ellos han pasado las pruebas de manera satisfactoria. En el caso de la U-14, se realizaron pruebas unitarias en sus 2 escenarios y se obtuvo la confirmación de que superaron las pruebas. Por último, para la historia de usuario U-36, se llevaron a cabo pruebas unitarias en sus 5 escenarios, obteniendo como resultado que todos pasaron las pruebas. Estos resultados se verán reflejados en el documento de Plan de Pruebas y se anexará al documento final.

6.4.5 Despliegue

Una vez que las fases de análisis, diseño, desarrollo y testing del ciclo de vida del software han sido finalizadas, con un total de no más de 480 horas de trabajo, se procede a dar paso a la fase de despliegue. Esta etapa incluye una serie de procesos para llevar a cabo de manera efectiva el lanzamiento del proyecto.

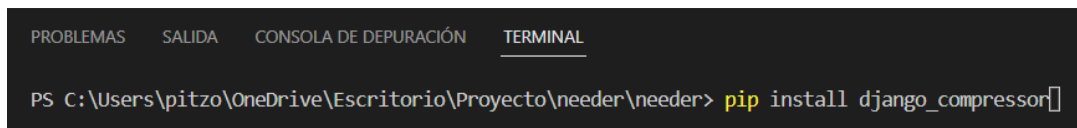
En primer lugar, se realizó la preparación del proyecto para el despliegue. En esta fase, se tomaron medidas para asegurar que el proyecto estuviera listo para ser subido al servidor y que todo el software necesario estuviera disponible y actualizado. Se realizaron pruebas adicionales y se corrigieron todos los errores encontrados. Posteriormente, se llevó a cabo la ejecución del despliegue, donde se procede a subir el proyecto al servidor. En este paso se verificó el cumplimiento de todas las especificaciones del proyecto y se aseguró que todo funcione correctamente en el servidor.

Luego de realizar la etapa anterior, se realizó la configuración del DNS y la aplicación del certificado SSL. Este paso fue esencial para asegurarse de que la aplicación estuviera disponible y accesible para los usuarios a través de internet. Se establecieron las configuraciones necesarias para que el DNS y el certificado estuvieran en funcionamiento y el proyecto estuviera listo para su uso.

6.4.5.1 Preparación del proyecto para el despliegue

6.4.5.1.1 Purge CSS

Se tomó la decisión de utilizar Purge CSS para mejorar la eficiencia del código CSS generado por Bootstrap y para reducir los tiempos de carga del prototipo web, mediante la eliminación de las clases innecesarias que no se utilizan en el HTML. Para llevar a cabo esta tarea, lo primero que se realizó fue la instalación de la aplicación Compressor. La instalación de Compressor se logró mediante la inclusión de la siguiente línea de petición:



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> pip install django_compressor
```

Figura 175. Instalación de la aplicación Compressor.

La aplicación Compressor se encargó de comprimir todos los archivos de CSS y Javascript, permitiendo reducir la cantidad de solicitudes que se realizan al servidor para cargar el prototipo web. También eliminó los comentarios y espacios en blanco innecesarios de los archivos, lo que redujo su tamaño y aceleró la carga de las vistas del proyecto.

Instalada la librería Compressor, el siguiente paso que se realizó fue el de la modificación del archivo `setting.py`. Primero se agregó en el listado de apps la aplicación Compressor. Luego de este proceso, se tuvo que agregar el buscador de archivos de Django Compressor dentro del listado de buscadores `STATICFILES_FINDERS`, con el fin de encontrar los archivos comprimidos generados por la aplicación Compressor.

```

21  INSTALLED_APPS = [
22      'django.contrib.admin',
23      'django.contrib.auth',
24      'django.contrib.contenttypes',
25      'django.contrib.sessions',
26      'django.contrib.messages',
27      'django.contrib.staticfiles',
28      'django.contrib.sites',
29      'captcha',
30      'allauth',
31      'allauth.account',
32      'allauth.socialaccount',
33      'allauth.socialaccount.providers.google',
34      'tinymce',
35      'needer',
36      'users',
37      'main',
38      'social',
39      'chat',
40      'channels',
41      'compressor',
42  ]
43

```

Figura 176. Adición de la aplicación Compressor al listado de apps de Needer.

```

242  STATICFILES_FINDERS = (
243      'django.contrib.staticfiles.finders.FileSystemFinder',
244      'django.contrib.staticfiles.finders.AppDirectoriesFinder',
245      'compressor.finders.CompressorFinder',
246  )

```

Figura 177. Adición del buscador Compressor al listado de búsqueda de Needer.

Una vez finalizada la configuración del archivo settings.py, se ejecutó la aplicación Compressor para realizar el Purge CSS mediante la siguiente línea de petición y se obtuvieron los CSS comprimidos.

```

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL
PS C:\Users\pitzo\OneDrive\Escritorio\Proyecto\needer\needer> python manage.py compress --force[]

```

Figura 178. Ejecución del Compressor en el proyecto.

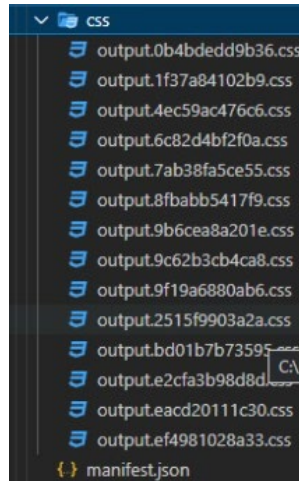


Figura 179. Obtención de los CSS Compuestos.

6.4.5.1.2 Configuración de la base de datos y variables de entorno

El siguiente proceso que se realizó fue la configuración de la base de datos y las variables de entorno, con el fin de seguir utilizando las APIs externas dentro del proyecto desplegado.

Como se puede observar en la **Figura (180)**, las variables de entorno son:

- **SECRET_KEY:** Esta es una variable de entorno usada por Django, como llave secreta, para proteger los proyectos. Hace el llamado a la llave secreta de Django `DJANGO_SECRET_KEY`, en caso de no encontrar dicha llave, esta variable generará una llave aleatoria.
- **DEBUG:** Durante la fase de desarrollo de un proyecto, resulta fundamental contar con la capacidad de observar los errores en tiempo real a través de la terminal. Es por ello que, en el entorno de desarrollo, se establece un valor de "True" que permite la visualización de los errores, mientras que se utiliza "False" para el caso de despliegue, en donde no se podrán visualizar los errores. De esta forma, se facilita la detección y corrección de cualquier problema que pueda surgir en el proceso de desarrollo del proyecto.
- **DEVELOPMENT_MODE:** Variable de entorno creada, con el fin de permitir intercambios, en el entorno de desarrollo, entre las base de datos de prueba y las base de datos de producción.

- **ALLOWED_HOSTS:** Se define como una lista de cadenas que representan los nombres de host/dominio que utiliza Django para visualizar el prototipo web. Hace el llamado a la variable de entorno DJANGO_ALLOWED_HOSTS, solicitada por DigitalOcean para poder realizar el despliegue del proyecto, adicionalmente se adjunta la dirección del localhost utilizada en el proyecto.

```
load_dotenv()
BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = os.getenv("DJANGO_SECRET_KEY", get_random_secret_key())

DEBUG = os.getenv("DEBUG", "False") == "True"

DEVELOPMENT_MODE = os.getenv("DEVELOPMENT_MODE", "False") == "True"

ALLOWED_HOSTS = os.getenv("DJANGO_ALLOWED_HOSTS", "127.0.0.1,localhost").split(",")
```

Figura 180. Variables de entorno.

Una vez se definieron las variables de entorno, se configuró la base de datos para ejecutar el proceso de despliegue. Se creó el condicional mostrado en la **Figura (181)** gracias a las recomendaciones de DigitalOcean para el proceso de despliegue. Este condicional muestra que, al encontrarse en un entorno de desarrollo, se hace uso de la base de datos alojada en localmente por SQLite. En caso tal de no encontrarse en un entorno de desarrollo y presentarse en un entorno de producción, se hará uso de la base de datos alojada en Heroku, gracias a las credenciales guardadas en la variable **DATABASE_URL**.

```
90 if DEVELOPMENT_MODE is True:
91     DATABASES = {
92         "default": {
93             "ENGINE": "django.db.backends.sqlite3",
94             "NAME": os.path.join(BASE_DIR, "db.sqlite3"),
95         }
96     }
97 elif len(sys.argv) > 0 and sys.argv[1] != 'collectstatic':
98     if os.getenv("DATABASE_URL", None) is None:
99         raise Exception("DATABASE_URL environment variable not defined")
100     DATABASES = {
101         "default": dj_database_url.parse(os.environ.get("DATABASE_URL")),
102     }
```

Figura 181. Base de datos.

6.4.5.1.3 Configuración del CDN

Continuando con el proceso de preparación del entorno, se configuró el CDN para poder ejecutar el despliegue. El CDN es un almacenamiento en la nube, encargado de almacenar todo el contenido multimedia de la aplicación. Como se muestra en la **Figura (182)**, se realizaron los siguientes pasos para configurar el CDN:

Lo primero que se realizó fue un condicional con la variable de entorno `DEBUG`, en donde si se encuentra en un entorno de desarrollo su valor es `True` y el directorio de `STATICFILES_DIR` sería igual a la ruta local “static”. En caso de encontrarse en un entorno de desarrollo, se procede a ejecutar la segunda parte del condicional. La carpeta `STATIC_ROOT` establece la ubicación de la carpeta raíz de los archivos estáticos de la aplicación web, lo que permite que se pueda almacenar de manera ordenada y accesible para su uso en el despliegue.

Luego, se ejecuta la variable de entorno `GS_CREDENTIALS`, la cual crea las credenciales de Google Cloud Storage necesarias para acceder y manipular los archivos alojados en Google Cloud Storage. Esta variable se configura a partir de un archivo de credenciales de JSON, ubicado en la carpeta `BASE_DIR` del proyecto.

Después, se ejecuta la variable de entorno `GOOGLE_APPLICATION_CREDENTIALS`, la cual se encarga de establecer la ubicación del archivo de credenciales de cuenta de los servicios de Google Cloud. Permitiendo que la aplicación se autentique de manera automática en los servicios de Google Cloud.

Continuando con la ejecución del condicional, el siguiente proceso que se ejecuta, es el del listado de buscadores, encargados de buscar los archivos estáticos de la aplicación para su uso. Por último, se encuentran las variables de entorno `DEFAULT_FILE_STORAGE`, `GS_BUCKET_NAME` y `STATICFILES_FILES`, las cuales permiten que los archivos cargados por los usuarios y los archivos estáticos de la aplicación web, sean almacenadas en Google Cloud Storage y no de manera local.


```

229 """ Cuando se realice el despliegue se usara el CDN """
230 if DEBUG is True:
231     STATICFILES_DIRS = [
232         BASE_DIR / "static",
233     ]
234 else:
235     STATIC_ROOT = BASE_DIR / "static"
236
237     GS_CREDENTIALS = service_account.Credentials.from_service_account_file(os.path.join(BASE_DIR,"credentials.json"))
238     GOOGLE_APPLICATION_CREDENTIALS = '/path/service-account.json'
239     ## STATIC FILES ##
240
241     STATICFILES_FINDERS = (
242         'django.contrib.staticfiles.finders.FileSystemFinder',
243         'django.contrib.staticfiles.finders.AppDirectoriesFinder',
244         'compressor.finders.CompressorFinder',
245     )
246
247
248
249     DEFAULT_FILE_STORAGE = 'storages.backends.gcloud.GoogleCloudStorage'
250     GS_BUCKET_NAME = 'needer'
251
252     STATICFILES_STORAGE = 'storages.backends.gcloud.GoogleCloudStorage'

```

Figura 182. Configuración del CDN.

6.4.5.2 Ejecución del despliegue

Finalizado el proceso de configuración del CDN, se realizó el despliegue utilizando los servicios de DigitalOcean para el manejo del servidor y Heroku para el manejo de la base de datos. Cabe resaltar que, se utilizó un plan estudiantil, brindado por DigitalOcean, el cual regala a los estudiantes US \$200, los cuales se utilizaron únicamente para el despliegue del proyecto y se hizo uso de Hostinger, con el fin de comprar el dominio.

Lo primero que se realizó, fue hacer la configuración del DNS en Hostinger, estableciendo los nameservers brindados por DigitalOcean, con el fin de realizar una conexión entre el dominio y el servidor, cómo se puede apreciar en la **Figura (183)** y **Figura (184)**:

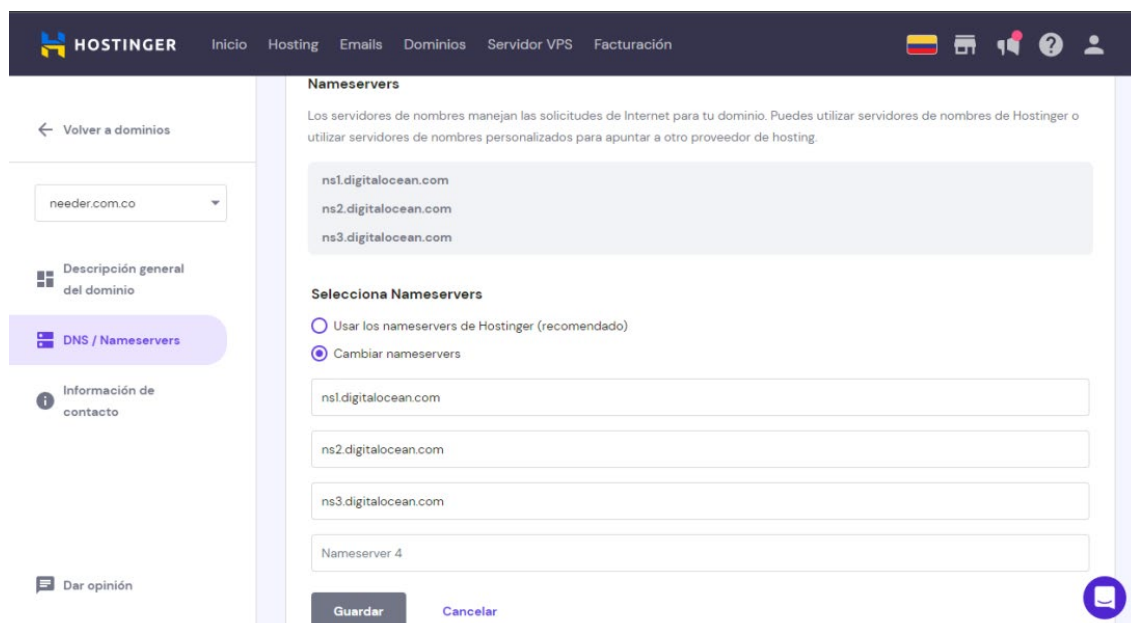


Figura 183. Actualización nameservers en Hostinger.

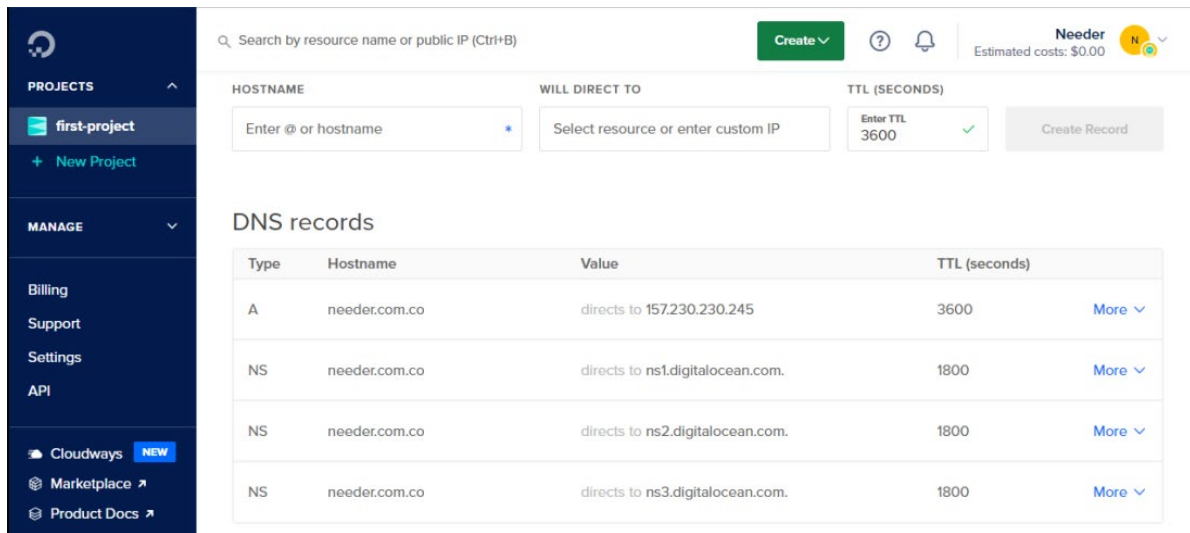


Figura 184. DNS enlazados con el Hostname DigitalOcean.

Una vez finalizada la configuración entre el dominio y el servidor, se creó el droplet, entorno virtual el cual permite establecer interacciones con el servidor. Para la creación del droplet, se escogió la región de Estados Unidos, más específicamente New York. Esta elección define el datacenter en donde se alojará el servidor. Luego, se escogió el ISO a implementar en el droplet. Cómo se especificó anteriormente, se utilizó el Ubuntu server versión 22.04 (LTS) x64 y después, se escogió el tamaño que iba a tener el droplet. Para este caso, como el proyecto inicialmente es pequeño y se sigue encontrando en estado de prueba, se decidió hacer uso de una configuración regular, con un tamaño de 512MB, 10 GB SSD Disk y 500GB de transferencia. Esta configuración es la más baja y tiene un costo mensual de US \$4. Sin embargo, como se previó en las proyecciones de costos, estos valores se espera incrementen en la medida que el proyecto crezca.

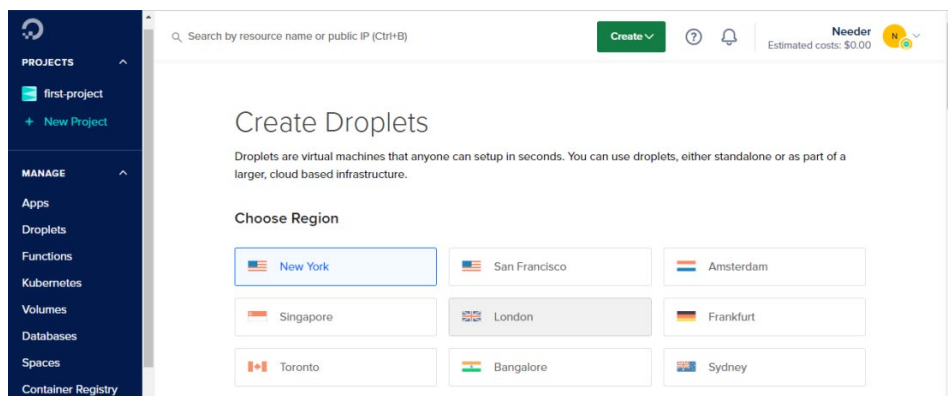


Figura 185. Selección de la región para el alojamiento del servidor.

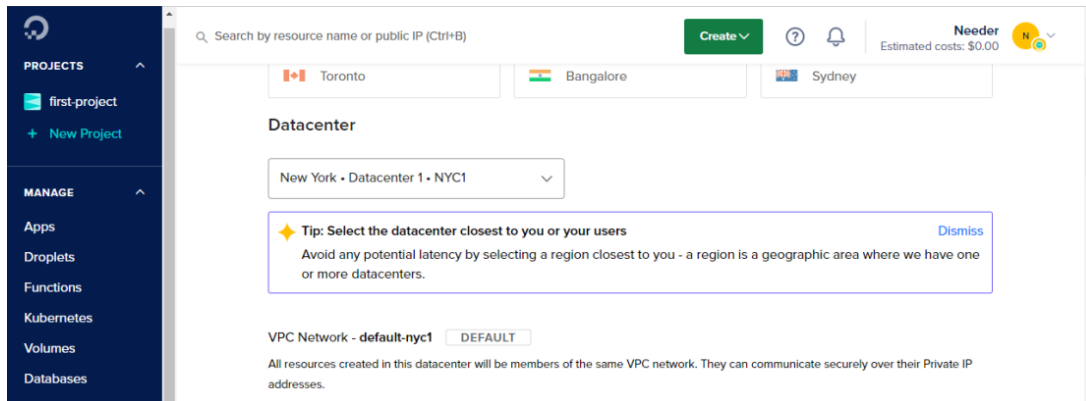


Figura 186. Datacenter situado en New York.

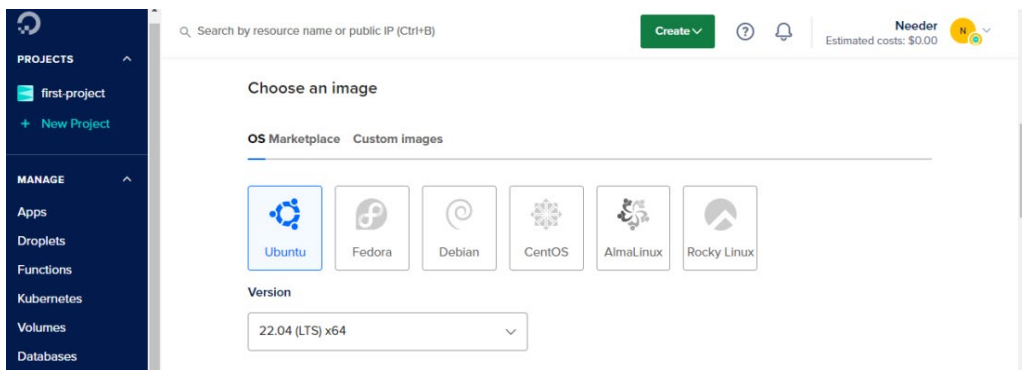


Figura 187. Selección del Ubuntu Server versión 22.04.

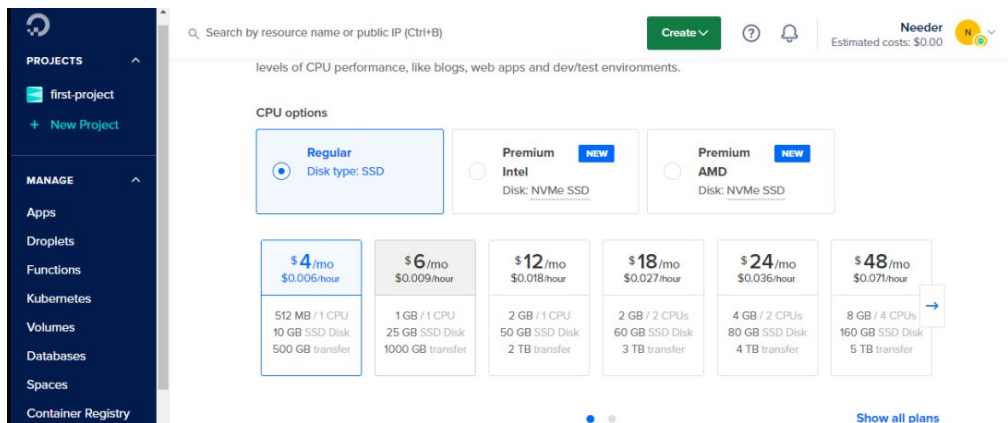


Figura 188. Selección del tamaño del entorno virtual.

Finalizado ya el proceso de creación del droplet, el siguiente paso fue ingresar a dicho droplet, mediante SSH. Como se puede observar en la **Figura 189**:

```
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-50-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon Feb 20 15:05:32 UTC 2023

System load: 0.3203125      Users logged in:      0
Usage of /:  32.0% of 9.51GB IPv4 address for eth0: 159.203.181.125
Memory usage: 87%          IPv4 address for eth0: 10.10.0.5
Swap usage:  0%           IPv4 address for eth1: 10.116.0.2
Processes:   99

106 updates can be applied immediately.
59 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Feb 20 14:21:44 2023 from 162.243.190.66
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~# █
```

Figura 189. Droplet en ejecución.

Con el fin de poder ejecutar el prototipo web dentro del servidor, se creó una directorio, el cual va a contener todo el proyecto. Dentro de este directorio se creó un entorno virtual, el cual va a contener todas las instalaciones requeridas para la ejecución del proyecto. Posteriormente, se realizó la conexión al repositorio en github de manera remota, con el fin de hacer pull a la rama creada para este proceso, llamada dist, sin tener la necesidad de manejar directamente la rama master del proyecto.

Una vez se contó con el proyecto dentro del directorio, se procedió a inicializar el entorno virtual mediante el comando `source bin/activate` y posterior a esto, se realizó la instalación de los paquetes necesarios que se encuentran dentro del archivo `requirements.txt` utilizando el comando `pip install -r requirements.txt`.

```
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~# mkdir project█
```

Figura 190. Creación del directorio project.

```
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~# cd project
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~/project# ls
'=.8'  README.md  image  lib  needer  requirements.txt
LICENSE  bin  include  lib64  pyenv.cfg
```

Figura 191. Configuración en el directorio del project.

Luego de haber hecho las respectivas instalaciones de los paquetes dentro del entorno virtual. Se ingresó a la aplicación de needer, la cual contiene los archivos de configuración del proyecto, ubicada dentro del directorio de needer y se creó un archivo llamado `credentials`

utilizando el comando `nano credential.json`. Este archivo contiene las variables que permiten la conexión con el CDN. También, se configuraron las variables de entorno creando el archivo llamado `.env` utilizando el comando `nano .env`.

```
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~/project/needer/needer# ls -a
.      .env      __pycache__  credentials.json  urls.py
..     __init__.py  asgi.py      settings.py      wsgi.py
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~/project/needer/needer#
```

Figura 192. Creación de los archivos `credentials.json` y `.env`.

Ya con las variables de entorno configuradas y con las configuraciones que permiten el enlace con el CDN, se procede a editar el archivo `gunicorn.socket`, el cual permitirá definir y configurar el socket para la comunicación entre el Gunicorn y Nginx. Dentro de este archivo, se crearon 3 secciones, la sección Unit para describir el socket, una sección de Socket, encargada de definir la ubicación del socket y la sección Install, la cual se asegura de que el socket es creado en el momento adecuado.

```
[Unit]
Description=gunicorn socket
[Socket]
ListenStream=/run/gunicorn.sock
[Install]
WantedBy=sockets.target
```

Figura 193. Configuración del `gunicorn.socket`.

Configurado ya el `gunicorn.socket`, luego se editó el archivo `gunicorn.service`, el cual permite el funcionamiento del proyecto. Este archivo será creado con el comando `sudo nano /etc/systemd/system/gunicorn.service` y dentro de este archivo, se realizarán los ajustes mostrados en la **Figura 193**.

```

[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=sammy
Group=www-data
WorkingDirectory=/root/project/needer
ExecStart=/root/project/bin/gunicorn \
    --access-logfile - \
    -k uvicorn.workers.UvicornWorker \
    --workers 3 \
    --bind unix:/run/gunicorn.sock \
    needer[.]asgi:application

[Install]
WantedBy=multi-user.target

```

Figura 193. Configuración del archivo `gunicorn.service`.

Luego, se debe verificar la creación del archivo `gunicorn.socket` utilizando el comando `sudo journalctl -u gunicorn.socket`. Después, se inicia el gunicorn utilizando el comando `sudo systemctl start gunicorn.socket` y se habilita con el comando `sudo systemctl enable gunicorn.socket`. Una vez habilitado el socket, se revisa su status con el comando `sudo systemctl status gunicorn.socket` y se verifica que el gunicorn está ejecutando correctamente como se puede observar en la Figura 194.

```

root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needer# sudo nano /etc/systemd/
system/gunicorn.socket
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needer# sudo systemctl start gu
nicorn.socket
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needer# sudo systemctl enable g
unicorn.socket
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needer# sudo systemctl statut g
unicorn.socket
Unknown command verb statut.
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needer# sudo systemctl status g
unicorn.socket
● gunicorn.socket - gunicorn socket
   Loaded: loaded (/etc/systemd/system/gunicorn.socket; enabled; vendor prese
   Active: active (running) since Sun 2023-02-19 07:07:25 UTC; 1 day 10h ago
   Triggers: ● gunicorn.service
   Listen: /run/gunicorn.sock (Stream)
   CGroup: /system.slice/gunicorn.socket

Feb 19 07:07:25 ubuntu-s-1vcpu-512mb-10gb-nycl-01 systemd[1]: Listening on guni
lines 1-8/8 (END) ...skipping...
● gunicorn.socket - gunicorn socket
   Loaded: loaded (/etc/systemd/system/gunicorn.socket; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-02-19 07:07:25 UTC; 1 day 10h ago
   Triggers: ● gunicorn.service
   Listen: /run/gunicorn.sock (Stream)
   CGroup: /system.slice/gunicorn.socket

```

Figura 194. Ejecución del `gunicorn.service`.

Estando configurado el gunicorn, el siguiente paso que se realizó es configurar el Nginx para poder pasar tráfico al proceso. Para esto, se utilizó el comando `sudo nano /etc/nginx/sites-available/needier` y se configuró como se muestra en la **Figura 195**.

```
server {
    listen 80;
    server_name needier.com.co;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root root/project/needier;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix://run/gunicorn.sock;
    }
}
```

Figura 195. Configuración Nginx.

Esta configuración básicamente crea un bloque *location* `/` para que todas las solicitudes coincidan. Dentro de este *location*, estará situado el archivo `proxy_params`, el cual es incluido con la instalación de Nginx. Luego, se pasará todo el tráfico de información directamente al `gunicorn.socket`. Esta configuración será ejecutada con el comando `sudo ln -s /etc/nginx/sites-available/needier /etc/nginx/sites-enabled/`.

Una vez ejecutado el proceso, el siguiente paso que se realizó fue el reinicio del Nginx, utilizando el comando `sudo systemctl restart nginx`, con el fin de que los cambios tomen lugar y se apliquen. Después, se testeó la configuración del nginx y se verificó que todo estuviera correcto.

```
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~/project/needier# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-s-1vcpu-512mb-10gb-nyc1-01:~/project/needier#
```

Figura 196. Test realizado al nginx.

Por último, se abrió el firewall para el tráfico normal en el puerto 80. Además, como ya no hay necesidad de acceder al servidor local de desarrollo, se eliminó el rule para abrir el puerto 8000.

```
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needers#  
sudo ufw delete allow 8000  
Could not delete non-existent rule  
Could not delete non-existent rule (v6)  
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needers# sudo ufw allow 'Nginx Full'  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
root@ubuntu-s-1vcpu-512mb-10gb-nycl-01:~/project/needers#
```

Figura 197. Paso de tráfico puerto 80 y configuración del puerto 8000.

Una vez desplegada la aplicación se puede ingresar a ella a través de un navegador, mediante la URL <https://needer.com.co>.

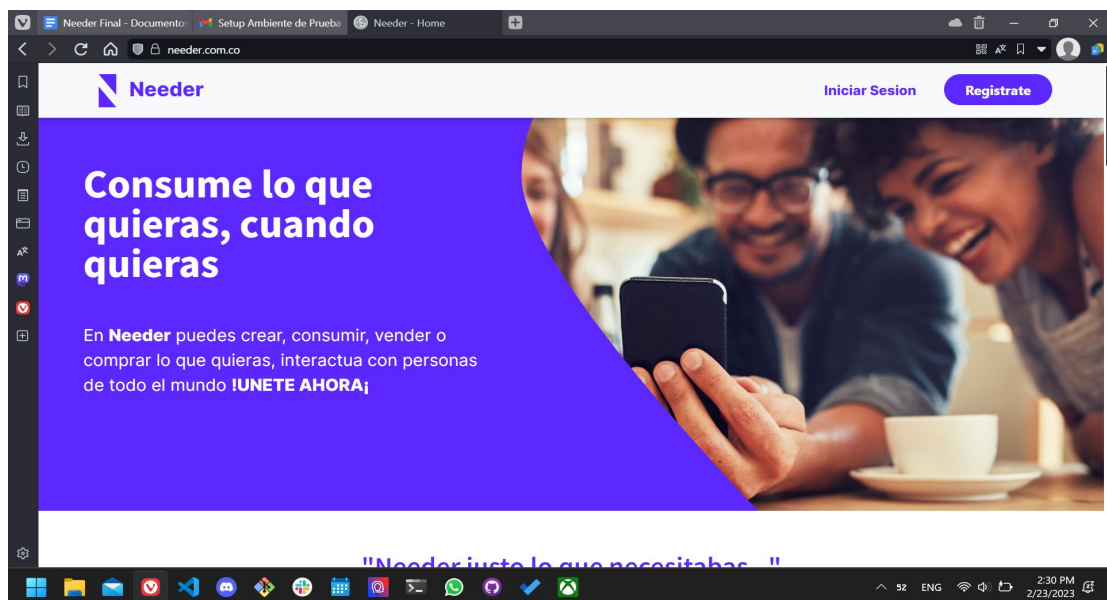


Figura 198. Proyecto desplegado

6.5 Proyección de ingresos, costos y gastos

La estimación de ingresos, costos y gastos se realizó con acompañamiento del centro de emprendimiento. Este proceso es de suma importancia, dado que en conjunto con el análisis de demanda, se logrará determinar la viabilidad del proyecto. Se ejecutó en 6 etapas o procesos, presentados a continuación.

6.5.1 Análisis de Ingresos

Este primer paso ayudó a encontrar las posibles ganancias mensuales y anuales de la compañía. Se utilizó la misma información de ingresos planteada en el modelo de negocio y se expuso en las siguientes tablas.

Needer sas					
PRESUPUESTO DE COMPRAS Y VENTAS					
En Años					
EN MILES(\$000)					
VENTAS TOTALES / AÑO	1	2	3	4	5
venta de contenido	5.858	15.966	39.916	71.849	129.328
	0	0	0	0	0
	0	0	0	0	0
TOTAL	5.858	15.966	39.916	71.849	129.328

Tabla 67. Proyección de ingresos anuales

Se espera que los ingresos sean percibidos gracias a las ventas del marketplace, por lo que en el apartado de red social no se ofrecera publicidad y no representará ingresos para la empresa. Estos ingresos se basan en el porcentaje de comisión a partir de las ventas estimadas, por lo que cada ingreso bruto representará el **13%** de las posibles ventas que se realizarán en el año. Estas incrementarán año a año, presentando un aumento **173%** para el segundo año, posterior a esto los ingresos aumentarán **150%** al 3 año, y para los últimos 2 años proyectados se estima crezcan solo un **80%**. En base a lo anterior para el 5to año se proyecta que ingrese a la compañía, **\$129 millones de pesos**. Por otro lado, los ingresos mensuales se estimaron dividiendo la ganancia anual entre los meses del año (**Tabla n**).

GANANCIAS MENSUALES AÑO A AÑO					
EN MILES (\$000)					
POLÍTICA DE VENTAS MENSUAL	1	2	3	4	5
CRÉDITO	0	0	0	0	0
CONTADO	488	1.331	3.326	5.987	10.777
TOTAL VTAS MES	488	1.331	3.326	5.987	10.777
TOTAL CRÉDITO AÑO	0	0	0	0	0
TOTAL CONTADO AÑO	5.858	15.966	39.916	71.849	129.328
TOTAL VTAS AÑO	5.858	15.966	39.916	71.849	129.328

Tabla 68. Proyección de ingresos mensuales

6.5.2 Proyección de gastos

En el apartado de gastos se estimaron los gastos de personal, la depreciación de los bienes fijos y los gastos generales de la empresa.

6.5.2.1 Gastos de Personal

Inicialmente, para los gastos de personal puesto que no se pagarán sueldos inicialmente, y que a los dos miembros del equipo se les atribuirá según las ganancias de la plataforma, se desestiman estos cálculos. Sin embargo, en la inversión inicial se tendrá en cuenta el salario para los 3 primeros meses, representando el periodo de desarrollo del prototipo.

6.5.2.2 Gastos Depreciación de Bienes Fijos y Amortización de Inversiones Diferidas

Los gastos de depreciación definen el porcentaje de devaluación que tendrán tanto los bienes fijos como las inversiones diferidas. Los equipos de cómputo representan la inversión fija inicial, a los cuales se estimó que pierdan anualmente un 20% de su valor, disminuyendo 2 millones anualmente, hasta perder totalmente su valor para el 5to año. Por otro lado, los precios de las inversiones diferidas, las cuales representarán los costos de servidores, almacenamiento, dominio y demás, suman un valor de **COP \$19.550.000**. Estas últimas, no se representaron como pérdidas para el primer año en la empresa, si no, un valor de inversión que será retornado conforme se obtengan ganancias.

Needer sas							
RELACIÓN DE INVERSIONES FIJAS Y DIFERIDAS - GASTOS DE DEPRECIACIÓN Y AMORTIZACIÓN							
En Años							
EN MILES (\$000)							
ACTIVO	Vida útil	INVER INICIAL	1	2	3	4	5
INVERSIONES FIJAS							
	0		0	0	0	0	0
Equipos de computo	5	10.000	2.000	2.000	2.000	2.000	2.000
TOTAL		10.000	2.000	2.000	2.000	2.000	2.000
INVERSIONES							

DIFERIDAS							
			0	0	0	0	0
Servidores, Hosting, etc.	1	19.550	19.550	0	0	0	0
TOTAL		19.550	19.550	0	0	0	0

Tabla 69. Gastos de depreciación y amortización

6.5.2.3 Gastos Generales

Por último, los gastos generales percibidos únicamente para el primer año fueron los servicios públicos (concretamente la energía eléctrica) y el internet. Estos valores sumados a los gastos de personal representarán la inversión en capital de trabajo. En total, los gastos de funcionamiento para el primer año se proyectaron por **COP \$2.400.000**, esto dado que los gastos operacionales se proyectan a partir del 2do año. Representado en los servidores, dominios, base de datos, etc. Por este motivo a partir de este año, los gastos aumentan a **COP \$23.926.000**.

Los valores anuales de ambos servicios, incrementarán un 9% cada año, por la inflación.

Needer sas					
RELACIÓN DE GASTOS DE FUNCIONAMIENTO					
En Años					
EN MILES (\$000)					
Gastos Administrativos	1	2	3	4	5
Servicios Públicos	720	785	855	932	1.016
Internet	1.680	1.831	1.996	2.176	2.371
Costo Operacional	0	21.310	23.227	25.318	27.596
TOTAL GASTOS	2.400	23.926	26.079	28.426	30.984
% Comisiones sob/ vtas contado		crédito			
Incremento % gastos	9,00%	9,00%	9,00%	9,00%	

Tabla 70. Gastos Generales

6.5.3 Inversiones Iniciales

Ahora bien, las inversiones iniciales se tomaron en proyección para el año 0, estas serán utilizadas para cubrir el sueldo inicial durante la construcción del prototipo, los gastos generales iniciales, el servidor, dominio entre otros.

Inicialmente se estimó la inversión de capital de trabajo, representada en los gastos generales de la compañía. A la cual se le suman los gastos de sueldos por los 3 primeros meses de desarrollo. Dando como total la magnitud de inversión inicial de capital de trabajo, es decir **COP 19.660.000**.

INVERSIÓN INICIAL	
En Años	
EN MILES (\$000)	
CÁLCULO CAPITAL DE TRABAJO	
Gastos de Nómina	0
Gastos Generales	2.400
TOTAL	2.400
TOTAL GASTOS Y COSTOS REQUERIDOS	2.400
TOTAL GASTOS Y COSTOS DIARIOS	7
TOTAL DÍAS PARA CAPITAL DE TRABAJO	249
TOTAL CAPITAL DE TRABAJO REQUERIDO	1.660
DÍAS PARA CAPITAL DE TRABAJO CARTERA	
VENTAS A CRÉDITO DIARIAS	0

Tabla 71. Inversión Inicial Capital de Trabajo

Por otro lado, las inversiones fijas y diferidas, planteadas anteriormente en el capítulo **Modelo de Negocio** se estimaron nuevamente para la definición de las inversiones iniciales. En total la inversión inicial requerida es de COP 49.210.000, destinada para el año 0. El financiamiento de esta inversión se realizará accediendo a un crédito o a convocatorias de financiamiento. Por esta razón, el 10% de las inversiones iniciales será recaudada por recursos propios y el restante con ayuda externa.

VALOR INVERSIÓN INICIAL		RECUPERACIÓN INVERSIÓN	
		%	VALOR
INVERSIONES FIJAS	10.000	20%	2.000
INVERSIONES DIFERIDAS	19.550	0%	0
CAP. DE TRAB. COS. Y GAS.	19.660	100%	19.660
CAP. DE TRAB. CARTERA	0	0%	0
TOTAL INVERSIÓN INICIAL	49.210		21.660

Tabla 72. Inversiones Iniciales

FUENTES DE FINANCIACIÓN	% PARTIC.	VALOR INVERSIÓN
CAPITAL PROPIO	10%	4.921
PRÉSTAMO BANCARIO 1	90%	44.289
PRÉSTAMO BANCARIO 2		0
PRÉSTAMO BANCARIO 3		0
TOTAL	100%	49.210

Tabla 73. *Financiamiento de Inversiones*

6.5.4 Flujo de caja

El proceso de encontrar el flujo de caja, garantiza el conocimiento del resultado que hay de la diferencia entre los ingresos y egresos. Este proceso ayuda a reconocer si la empresa estará en una situación económica sana. Es importante recalcar que parte de los ingresos iniciales percibidos son producto de la inversión inicial, por lo que no se iniciará con déficit de efectivo en el primer año. Sin embargo, dado que los ingresos producto de las ganancias al principio serán bajos, no se obtendrá un flujo positivo hasta el 4to año, donde a partir de este, la empresa empezará a ser solvente.

Needer sas FLUJO DE CAJA PROYECTADO En Años EN MILES (\$000)						
CONCEPTO	0	1	2	3	4	5
INGRESOS						
Ingresos por Ventas de Contado en el Periodo	0	5.858	15.966	39.916	71.849	129.328
Caja inicial	0	0	-16.092	-24.086	-10.345	28.780
Préstamo	44.289	0	0	0	0	0
Aporte de capital	4.921	0	0	0	0	0
TOTAL INGRESOS	49.210	5.858	-125	15.830	61.504	158.108
EGRESOS						
Gastos de funcionamiento	0	2.400	23.926	26.079	28.426	30.984
Gastos de personal	19.660	0	0	0	0	0
Amortización préstamo		0	0	0	0	0
Compra de activos fijos	10.000	0	0	0	0	0
Gastos de iniciación y montaje	19.550	19.550	0	0	0	0
Pagos de impuestos	0	0	35	96	4.299	14.778
TOTAL EGRESOS	49.210	21.950	23.961	26.175	32.725	45.762

SALDO FINAL EN CAJA	0	-16.092	-24.086	-10.345	28.780	112.345
----------------------------	----------	----------------	----------------	----------------	---------------	----------------

Tabla 74. Flujo de caja de la empresa

6.5.5 Estado de Resultado

Una vez encontrado el saldo final en caja a este análisis se estiman los gastos e ingresos planteados en el análisis anterior, pero, se agregan elementos como el pago de impuestos, pago de iva, etc, esto con el fin de encontrar las ganancias netas de la empresa. Este proceso llamado estado de resultado ayuda a conocer si la empresa en el periodo de 5 años logrará resultados positivos a partir de los gastos que se realizarán.

Needer sas ESTADO DE RESULTADOS PROYECTADO En Años EN MILES (\$000)					
Concepto	1	2	3	4	5
Ventas estimadas					
Ingresos por ventas	5.858	15.966	39.916	71.849	129.328
TOTAL VENTAS	5.858	15.966	39.916	71.849	129.328
Costo de ventas	0	0	0	0	0
UTILIDAD BRUTA	5.858	15.966	39.916	71.849	129.328
Gastos administrativos					
Gastos de funcionamiento	2.400	23.926	26.079	28.426	30.984
Impuesto de ICO	35	96	239	431	776
Depreciación	2.000	2.000	2.000	2.000	2.000
Amortización	19.550	0	0	0	0
Gasto de intereses	0	0	0	0	0
Total gastos administrativos	23.985	26.021	28.318	30.857	33.760
Gastos de ventas					
Gastos de ventas	0	0	0	0	0
Total gastos de ventas	0	0	0	0	0
Total gastos	23.985	26.021	28.318	30.857	33.760

	409%	163%	71%	43%	26%
UTILIDAD OPERACIONAL	-18.127	-10.055	11.598	40.992	95.568
UTILIDAD OPERACIONAL	-18.127	-10.055	11.598	40.992	95.568
Impuesto de Renta	0	0	4.059	14.347	33.449
Impuesto Cree y Otros	0	0	0	0	0
Utilidad después de impuestos	-18.127	-10.055	7.539	26.645	62.119
UTILIDAD NETA	-18.127	-10.055	7.539	26.645	62.119

Tabla 75. Estado de resultados

IMPUESTO RENTA	35%
IMPUESTO CREE Y OTROS	0%
RESERVA LEGAL	0%
OTRAS RESERVAS	0%

Tabla 76. Porcentaje pago de impuestos y otros

Como se observó en las tablas anteriores a partir del análisis realizado se concluye que la empresa obtendrá utilidades a partir del 3er y 4to año.

6.5.6 Balance

Ahora bien, determinar el balance permitió determinar el patrimonio neto de la empresa durante los años. Es decir, cuantificar que es de la empresa luego de gastos, impuestos, pago de pasivos y demás. Ahora bien, puesto que como se mencionó anteriormente se espera obtener financiamiento para la inversión inicial, la empresa contará con un pasivo con la magnitud de la inversión, el cual se irá pagando conforme se vaya teniendo recursos en la empresa. Por esta razón, las utilidades acumuladas irán en negativo hasta el 5to año. Por otro lado, la empresa empezará a contar con patrimonio en el 4to año, donde éste será de **COP \$10.992.000**, representado por elementos como el flujo de caja, inventario de planta, depreciación y demás, y a partir de ahí obtendrá ganancias monetarias.

Needer sas

BALANCE GENERAL PROYECTADO

En Años

EN MILES (\$000)

CUENTA CONTABLE	0	1	2	3	4	5
ACTIVO						
Corriente						
Caja y Bancos	0	0	0	0	28.780	112.345
Inventario	0	0	0	0	0	0
Clientes		0	0	0	0	0
Total Activo Corriente	0	0	0	0	28.780	112.345
Propiedad Planta y equipo	10.000	10.000	10.000	10.000	10.000	10.000
Depreciación Acumulada		2.000	4.000	6.000	8.000	10.000
Total Propiedad, Planta y Equipo	10.000	8.000	6.000	4.000	2.000	0
Otros Activos						
Diferidos ajustados	19.550	0	0	0	0	0
Amortización acumulada		0	0	0	0	0
Total Otros Activos	19.550	0	0	0	0	0
TOTAL ACTIVO	29.550	8.000	6.000	4.000	30.780	112.345
PASIVOS						
Cuentas por Pagar por Flujo caja		16.092	24.086	10.345	0	0
Obligaciones Financieras	44.289	44.289	44.289	44.289	44.289	44.289
Impuestos por Pagar		35	96	4.299	14.778	34.225
Cuentas por Pagar proveedores		0	0	0	0	0
TOTAL PASIVO	44.289	60.416	68.471	58.932	59.067	78.514
PATRIMONIO						
Aportes de Capital	4.921	4.921	4.921	4.921	4.921	4.921
Reserva Legal		0	0	0	0	0
Utilidades del periodo		-18.127	-10.055	7.539	26.645	62.119
Utilidades Acumuladas		0	-18.127	-28.182	-20.643	6.001
TOTAL PATRIMONIO	4.921	-13.206	-23.261	-15.722	10.922	73.041
TOTAL PASIVO Y	49.210	47.210	45.210	43.210	69.990	151.555

PATRIMONIO						
------------	--	--	--	--	--	--

Tabla 77. Balance de la empresa

6.6 Evaluación financiera, social, ambiental

Una vez realizada la proyección de ingresos, costos y gastos, fue posible realizar el análisis financiero, social y ambiental. El primer análisis definirá la rentabilidad y viabilidad del proyecto, comprendiendo aspectos como la tasa interna de retorno, la tasa interna de oportunidad, la relación costo/beneficio y la tasa verdadera de rentabilidad.

6.6.1 Análisis Financiero de Rentabilidad

EN MILES

ANÁLISIS FINANCIERO		RESULTADO
Tasa Interna de Retorno (TIR)		28%
Valor Presente Neto (VPN)		38.221
Tasa Interna de Oportunidad (TIO)		13%
Relación Beneficio / Costo (B/C)		1,69
Tasa verdadera de Rentabilidad (TVR)		25%

Tabla 78. Análisis financiero

La tasa interna de oportunidad es el primer valor que se estimó, dado que esta determina a nivel teórico la rentabilidad del proyecto. Es decir, describe los valores presentes netos esperados en base al flujo de caja del proyecto. En este caso la Tasa Interna de Oportunidad (TIO) esperada fue del 13%, con esto, si se invirtieron \$49.210.000 COP inicialmente, a valor futuro ese monto se transformara en, \$55.617.300 COP.

Por otro lado, la tasa interna de retorno dicta el rendimiento real del proyecto. Según lo anterior, puesto que la TIR (Tasa Interna de Retorno) fue del 28%, la ganancia en base a la primera inversión será de \$13.778.800 COP en base a los posibles ingresos de la plataforma. Sin embargo, este valor presenta errores en su cálculo, por lo que fue necesario plantear la Tasa Verdadera de Rentabilidad (TVR), siendo esta del 25% que en otras palabras describiría que la empresa para el 5to año tendrá un valor de \$61.512.500 COP.

Finalmente, la relación costo beneficio garantiza que por cada peso invertido en el proyecto se obtendrá una rentabilidad de 0.69 pesos, por lo que se concluye que el proyecto es rentable en el tiempo.

6.6.2 Impacto Social

En cuanto a los beneficios sociales que se espera garantizar por medio de la empresa y la plataforma son:

1. Brindar espacios donde artistas y personas puedan crear comunidades, ganar dinero y obtener visibilidad.
2. Por otro lado, se desea generar primeros empleos a desarrolladores y trabajadores jóvenes, promoviendo el trabajo inicialmente en la ciudad de Neiva.
3. Así mismo, se espera garantizar un ambiente donde se respete la dignidad de las personas y sobre todo de la mujer, donde ellas puedan sentirse cómodas e identificadas.
4. Por último, Needer pretende crear organizaciones que ayuden a comunidades vulnerables alrededor de Colombia. Donde se apoye a las personas en estados de pobreza, niños sin estudio y los afectados por el conflicto armado.

6.6.3 Impacto Ambiental

Como aspecto final pero no menos importante, se definió el impacto ambiental que generará la plataforma en el mundo. Inicialmente dado que en la empresa las facturas y documentación se llevarán de forma digital, se estima no gastar papel y por lo tanto reducir en la deforestación de bosques y tala de árboles. Además, se planea que aparte de apoyar causas sociales, Needer pueda crear causas ambientales, que apoyen la batalla contra el calentamiento global, la conservación de hábitats naturales y la defensa de animales.

VII. CONCLUSIONES Y FUTUROS TRABAJOS

En conclusión, se ha determinado que la guía metodológica del centro de emprendimiento de la Universidad Surcolombiana ha sido fundamental para estructurar el modelo de negocio de la empresa. A través de los estudios de mercado realizados, se ha logrado definir los productos y servicios que la empresa espera ofrecer una vez que esté conformada. Además, esta guía ha permitido establecer los componentes organizacionales y legales de la startup. Así mismo, gracias a las asesorías, se permitieron llevar a cabo el estudio financiero que dio como resultado los índices que se evaluaron en el análisis financiero. Por otra parte, estas asesorías permitieron realizar la evaluación ambiental y económica de la empresa. Todos estos procesos, dieron como resultado que la empresa es sostenible y que su realización es rentable. En consecuencia, se puede afirmar que el producto generado a partir del modelo de negocio es satisfactorio.

Además, se utilizó la metodología Scrum para la planificación y el desarrollo ágil y seguro de aplicaciones en un proyecto que se completó en tan solo tres meses. Este enfoque demostró ser efectivo en la entrega de resultados rápidos y confiables. También, gracias al uso de Bootstrap y Purge CSS, se pudo agilizar el proceso de desarrollo frontend y gracias a las metodologías de diseño, se pudo realizar un diseño acorde a los estándares de calidad. Por otro lado, Django se empleó para el desarrollo ágil y rápido de aplicaciones web, permitiendo construir aplicaciones complejas de manera eficiente, lo que aumentó la productividad del equipo y redujo los tiempos de entrega. En conclusión, la implementación de estas metodologías y tecnologías resultó fundamental para el éxito de la startup.

Como trabajo futuro, se espera que los miembros del equipo u otras personas puedan seguir avanzando en el proyecto, con el fin de desarrollar los demás productos establecidos en el modelo de negocio y los requerimientos planteados de las demás etapas. Por otro lado, se propone modificar la arquitectura de la aplicación, puesto que esta va a seguir en crecimiento a medida que se agreguen más productos, por lo tanto, se espera que se transicione a una arquitectura de microservicios

Referencias bibliográficas

- [1, p.43] *Abbate, Janet. A Social History of the Internet, Cambridge, Massachusetts: MIT Press, 1999.*
- [2, p.25] *R. Cañedo Andalia, "Aproximaciones para una historia de Internet", Acimed, vol. 12, n.º 1, 2004.*
- [3] *"Personas que usan Internet (% de la población)". World Bank Open Data | Data. https://datos.bancomundial.org/indicador/IT.NET.USER.ZS?most_recent_value_desc=true (accedido el 28 de abril de 2022).*
- [4] *"Historia de las Redes Sociales". Autor: Julia Máxima Uriarte. Para: Caracteristicas.co. Última edición: 15 de mayo de 2020. Disponible en: <https://www.caracteristicas.co/historia-de-las-redes-sociales/>. Consultado: 28 de julio de 2022.*
- [5] *Centro de Emprendimiento, Universidad Surcolombiana, "Guía Metodología para Argumentar la Idea Primera Aproximacion de la Idea al Modelo de Empresa", PDF, Version 2.0.*
- [6] *F. Rocca Jordán. "¿Qué es la propuesta de valor? - Clase Ejecutiva UC". Clase Ejecutiva UC. <https://www.claseejecutiva.com.co/blog/articulos/que-dice-la-propuesta-de-valor/> (accedido el 28 de julio de 2022).*
- [7] *"Definitions of Marketing". American Marketing Association. <https://www.ama.org/the-definition-of-marketing-what-is-marketing/#:~:text=Product,of%20tangible%20and%20intangible%20forms>. (accedido el 27 de julio de 2022).*
- [8, p.12] *Urueña, A., Ferrari, A., Blanco, D. y Valdecasa, E. (diciembre de 2011). Las Redes Sociales en Internet. [redes_sociales-documento_0.pdf](#), España. https://www.ontsi.es/sites/ontsi/files/redes_sociales-documento_0.pdf*
- [9, p.24] *S. S. Valera Martínez, "GUÍA DE ANÁLISIS PARA UN MARKETPLACE PARA PYMES MEXICANAS", Trabajo de grado, Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, 2005.*
- [10] *NTCHosting. "NTC Encyclopedia - Multimedia Content". NTCHosting. <https://www.ntchosting.com/encyclopedia/multimedia/multimedia-content/> (accedido el 27 de julio de 2022).*
- [11] *Universidad UNADE. "Tipos de comercio electrónico: desde B2B a G2C". Universidad Americana de Europa. <https://unade.edu.mx/tipos-de-comercio-electronico/> (accedido el 28 de julio de 2022).*

[12] D. da Silva. "¿Qué es el mercado objetivo? 5 pasos para definir el tuyo". Zendesk. <https://www.zendesk.com.mx/blog/mercado-objetivo/> (accedido el 28 de julio de 2022).

[13] R. Muñoz González. "5. Análisis competitivo". Marketing-XXI. <https://www.marketing-xxi.com/analisis-competitivo-17.htm> (accedido el 28 de julio de 2022).

[14] G. J. Giner. "Canales de Distribución, ¿Cuál es el adecuado para tu Negocio?" Escuela de Negocios y Dirección. <https://www.escueladenegociosydireccion.com/revista/business/emprendedores/canales-de-distribucion-cual-es-el-adecuado-para-tu-negocio/> (accedido el 28 de julio de 2022).

[15] R. Peiró. "Programa de fidelización - Economipedia". Economipedia. <https://economipedia.com/definiciones/programa-de-fidelizacion.html#:~:text=Un%20programa%20de%20fidelización%20es,y%20duradera%20con%20la%20marca> (accedido el 28 de julio de 2022).

[16] L. Agüero Cobo, "Estrategia de Fidelización de Clientes", Trabajo de Grado, Universidad de Cantabria, Cantabria, 2014. Accedido el 28 de julio de 2022. [En línea]. Disponible: <http://hdl.handle.net/10902/4474>

[17] J. F. López. "Inversión". Economipedia. <https://economipedia.com/definiciones/inversion.html> (accedido el 29 de julio de 2022).

[18] J. Betancourt Milan, "Proyecto de Inversión para Instalar una Planta Embotelladora de Refrescos en Torreón, Coahuila", Trabajo de Grado, Universidad Nacional Autónoma de México, Ciudad de México, 2001. [En línea]. Disponible: <https://repositorio.unam.mx/contenidos/429095>

[19, p.10] J. Lizcano Álvarez y E. Castelló Taliani, *Rentabilidad Empresarial Propuesta Práctica de Análisis y Evaluación*. Madrid: Cámaras de Comercio. Servicios de Estudios., 2004.

[20] R. Muñoz González. "1. Concepto de investigación de mercados". Marketing-XXI. <https://www.marketing-xxi.com/concepto-de-investigacion-de-mercados-23.htm#:~:text=Se%20puede%20definir%20como%20la,del%20marketing%20estratégico%20y%20operativo>. (accedido el 28 de julio de 2022).

[21, pp.66-67] E. Nunez Jimenez, *Guía para la preparación de Proyectos de servicios públicos municipales*. Cuajimalpa: INAP, 1997.

[22, p.83] A. E. Miguel, *Proyectos de inversión. Formulación y Evaluación para Micro y Pequeñas Empresas*, 4a ed. Oaxaca: Instituto Tecnológico de Oaxaca., 2001.

[23, p.10] L. L. Garber, R. R. Burke y J. Morgan Jones, *The role of package color in consumer purchase consideration and choice*. Cambridge, MA: Marketing Science Institute, 2000.

[24, pp.2-3] "Taller de producción de mensajes". https://perio.unlp.edu.ar.https://unblogdemarketing.files.wordpress.com/2014/03/mision_vision-claras.pdf.

[25, pp.5-7] "Taller de producción de mensajes". https://perio.unlp.edu.ar.https://unblogdemarketing.files.wordpress.com/2014/03/mision_vision-claras.pdf.

[26] H. Rodríguez. "¿Cómo hacer un organigrama para tu empresa? ¡Estás a 6 pasos de lograrlo!" Crehana. <https://www.crehana.com/blog/negocios/como-hacer-un-organigrama/>.

[27] F. A. CÁCERES FUENTES y D. R. ROMERO VARGAS, *MANUAL DE FUNCIONES Y PROCEDIMIENTOS DE LA EMPRESA SARMIENTO Y FARIETA AGENTES INMOBILIARIOS S.A.S*, 2016, *2016_manual_funciones_procedimientos.pdf*, Villavicencio, Colombia.

[28] *Manual para determinar y verificar los requisitos habilitantes en los Procesos de Contratación*, M-DVRHPC-05, Departamento Nacional de Planeación Colombiano, Bogotá D.C.

[29] 15 ejemplos de valores corporativos y 5 consejos para definir los tuyos • Asana. (18 de junio de 2022). Asana. <https://asana.com/es/resources/company-values-examples>.

[30] "¿Qué son las Cámaras de Comercio y sus principales funciones?" DataCrédito. <https://www.datacreditoempresas.com.co/blog-datacredito-empresas/que-son-las-camaras-de-comercio-y-sus-principales-funciones/> (accedido el 25 de octubre de 2022).

[31] Cámara de Comercio de Ipiales. "Funciones De Las Cámaras De Comercio". <http://ccipiales.org.co>.

[32] "¿Cuándo se considera que hay homonimia? - Cámara de Comercio de Bogotá". Cámara de Comercio de Bogotá. <https://www.ccb.org.co/Preguntas-frecuentes/Registros-Publicos/Cuando-se-considera-que-hay-homonimia#:~:text=La%20homonimia%20implica%20la%20existencia,el%20uso%20del%20público%20en>

[33] M. Murillo Uribe-Echeverría, "Sociedades por Acciones Simplificadas (S.A.S)", ABCES Jurídico.

[34] Valentinast. (14 de septiembre de 2020). ¿Qué impuestos debe pagar una empresa SAS en Colombia? Rankia. <https://www.rankia.co/blog/dian/4715040-que-impuestos-debe-pagar-empresa-sas-colombia>.

[35] "¿Cuál es la ley del comercio electrónico en Colombia? | Pymas". Pymas. <https://www.pymas.com.co/ideas-para-crecer/ecommerce/ley-comercio-electronico-colombia> (accedido el 26 de octubre de 2022).

[36] Colombia, Congreso de la República. Ley n.º 1581, Ley 1581 de 2012. [En línea]. Disponible: <https://www.minambiente.gov.co/politica-de-proteccion-de-datos-personales/#:~:text=Ley%20de%20proteccion%20de%20Datos,de%20naturaleza%20pública%20o%20privada.>

[37] "¿Por qué es importante leer los términos y condiciones de una página en internet o aplicación?" Argentina.gob.ar. <https://www.argentina.gob.ar/justicia/convosenlaweb/situaciones/porque-es-importante-leer-los-terminos-y-condiciones-de-una-pagina-en-internet-o-aplicacion#:~:text=¿Qué%20son%20los%20términos%20y,web%20o%20de%20una%20aplicación.> (accedido el 26 de octubre de 2022).

[38] G. Byrum y D. Nucera, *How Does the Internet Work?* 2018. [En línea]. Disponible: https://uploads-ssl.webflow.com/5f96dbf70b28bb61abdf6b62/606d1087b3515d8df3a46d60_Publication_howtheinternetworks_v18.pdf

[39, p.47] C. P. FLÓREZ RAMÍREZ, H. MÉNDEZ COLLO, "ESTUDIO DE INGENIERÍA SOCIAL EN EL USO DE LAS REDES SOCIALES", Trabajo de grado, UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA, Bogotá D.C., 2017. [En línea]. Disponible: <https://repository.unad.edu.co/bitstream/handle/10596/14505/17659358.pdf?sequence=1&isAllowed=y>

[40] Fu, K., Feamster, N., Sit, E. y Smith, K. (s. f.). *Dos and Don'ts of Client Authentication on the Web*. 16. <https://pdos.csail.mit.edu/papers/webauth:sec10.pdf>

[41] MLSDev. "How to Build a Social Network Website from Scratch". Medium. <https://mlsdev.medium.com/how-to-build-a-social-network-website-from-scratch-66034be120e2> (accedido el 11 de noviembre de 2022).

[42] D. Boyd y N. Ellison, "Social network sites: definition, history, and scholarship", *IEEE Engineering Management Review*, vol. 38, n.º 3, p. 3, 2010. Accedido el 11 de noviembre de 2022. [En línea]. Disponible: <https://doi.org/10.1109/emr.2010.5559139>

[43] Vitality y Alina. "How to Make a Social Media Website like Facebook From Scratch: The Success Story | Aimprosoft". Aimprosoft. <https://www.aimprosoft.com/blog/how-to-build-a-social-media-website/>.

[44] A. Richter y M. Koch, "Functions of Social Networking Services", *Contribution136*, Neubiberg - Munich, Alemania. [En línea]. Disponible: <https://dl.eusset.eu/bitstream/20.500.12015/2802/1/00467.pdf>

[45] D. Martín Critikián y M. Medina Núñez, "REDES SOCIALES Y LA ADICCIÓN AL LIKE DE LA GENERACIÓN Z", *Revista de Comunicación y Salud*, vol. 11, n.º 55-76, p. 22, 2021.

[46] J. M. CORONADO ZÚÑIGA, U. HERNANDEZ PINO "MODELO DE CONECTIVIDAD PARA REDES HUMANAS", Trabajo de grado, UNIVERSIDAD DEL CAUCA, Popayán, 2004. [En línea]. Disponible: <http://www.iered.org/joiner/docfinal.html>

[47] D. Solís Fonseca, W. Roque Pérez y M. L. Morilla Faurés, PASARELA DE PAGOS PARA LA SEGURIDAD DE TRANSACCIONES BANCARIAS EN LINEA, 29 de agosto de 2013, *pasarela-de-pagos.pdf*. [En línea]. Disponible: <https://www.3ciencias.com/wp-content/uploads/2013/08/pasarela-de-pagos.pdf>

Mostrar menos

[48, p.10] A. Alcides Castillo, "SISTEMA DE SISTEMA DE INVENTARIO WEB PARA INVENTARIO WEB PARA LA GESTIÓN Y EL LA GESTIÓN Y EL CONTROL DE ACTIVOS CONTROL DE ACTIVOS FIJOS DE LA FCEYT FIJOS DE LA FCEYT", Trabajo de grado, UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO, Santiago del Estero, 2018. [En línea]. Disponible: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjkyvuO9LD7AhXAczABHZtCCNEQFnoECBsQAQ&url=http://bcdigi.unse.edu.ar:8080/js/pui/bitstream/123456789/218/1/Trabajo%20de%20Graduacion%20Siweb.pdf&usq=AOvVaw2t09NyqxZW_xhtm27-F4XR

[49] "Ciclo de vida del software: todo lo que necesitas saber". Intelequia. <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>.

[50] H. Verma, "Software Development Life Cycle: Factors affecting Software Quality", *JETIR*, vol. 6, n.º 3, marzo de 2019, art. n.º JETIR1903A29. [En línea]. Disponible: <https://www.jetir.org/papers/JETIR1903A29.pdf>

[51, p.4] P. D. Lavappa, "Guidelines for Planning and Development of Software for Buildings and Building Systems", junio de 2008, *Guidelines for Planning and Development of Software for Buildings and Building Systems*, Gaithersburg, Estados Unidos. [En línea]. Disponible: <https://www.govinfo.gov/content/pkg/GOVPUB-C13-5ee54872c409fdf078c794476b4c062a/pdf/GOVPUB-C13-5ee54872c409fdf078c794476b4c062a.pdf>

[52] H. Verma, "Software Development Life Cycle: Factors affecting Software Quality", *JETIR*, vol. 6, n.º 3, marzo de 2019, art. n.º JETIR1903A29. [En línea]. Disponible: <https://www.jetir.org/papers/JETIR1903A29.pdf>

[53] Valpadasu Hema et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 981 022060

[54] K. Schwaber y J. Sutherland, "La Guía Scrum, La Guía Definitiva de Scrum: Las Reglas del Juego", noviembre de 2020, 2020-Scrum-Guide-Spanish-European.pdf. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

[55, p.4] K. Schwaber y J. Sutherland, "La Guía Scrum, La Guía Definitiva de Scrum: Las Reglas del Juego", noviembre de 2020, 2020-Scrum-Guide-Spanish-European.pdf. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

[56] Scrum: A Framework To Reduce Risk And Deliver Value Sooner. The Liberators, 2021. [En línea]. Disponible: <https://scrumorg-website-prod.s3.amazonaws.com/drupal/2021-03/Scrum-%20A%20Framework%20to%20Reduce%20Risk%20and%20Deliver%20Value%20Sooner.pdf>

[57, p.1] R. A. Amézquita Mejía, "ANÁLISIS DE LA APLICACIÓN DE LA METODOLOGÍA SCRUM COMO COMPLEMENTO DE LAS METODOLOGÍAS DEL PMI PARA EL CONTROL DE PROYECTOS DE DESARROLLO DE SOFTWARE.", p. 12.

[58, p.4] R. A. Amézquita Mejía, "ANÁLISIS DE LA APLICACIÓN DE LA METODOLOGÍA SCRUM COMO COMPLEMENTO DE LAS METODOLOGÍAS DEL PMI PARA EL CONTROL DE PROYECTOS DE DESARROLLO DE SOFTWARE.", p. 12.

[59, p.4] R. A. Amézquita Mejía, "ANÁLISIS DE LA APLICACIÓN DE LA METODOLOGÍA SCRUM COMO COMPLEMENTO DE LAS METODOLOGÍAS DEL PMI PARA EL CONTROL DE PROYECTOS DE DESARROLLO DE SOFTWARE.", p. 12.

[60, p.5] K. Schwaber y J. Sutherland, "The Scrum Guide™", p. 16, julio de 2013. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

[61] Grimaldi Pablo David Salazar Joaquín Ismael, "El desafío de Scrum distribuido en diferentes locaciones", Trabajo de grado, Universidad Nacional de la Plata, 2022. [En línea]. Disponible: <http://sedici.unlp.edu.ar/bitstream/handle/10915/133263/Tesis.pdf-PDFA.pdf?sequence=3&isAllowed=y>

[62, p.12] K. Schwaber y J. Sutherland, "The Scrum Guide™", p. 16, julio de 2013. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

[63, p.12-15] K. Schwaber y J. Sutherland, "The Scrum Guide™", p. 16, julio de 2013. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

[64, p.7-12] K. Schwaber y J. Sutherland, "The Scrum Guide™", p. 16, julio de 2013. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

[65] O. J. Gascón Busio, "Todo Scrum 2021 Un resumen para ser un Scrum Master", 2019, [guia-scrum-agilpm_2019.pdf](#).

[66] "The Official Kanban Guide To The Kanban Method", Kanban University, vol. 6, p. 15, febrero de 2021. [En línea]. Disponible: https://kanban.university/wp-content/uploads/2021/03/The-Official-Kanban-Guide_A4.pdf

[67] B. K. Swee Khoo, "The Software Design Methodology", p. 5. [En línea]. Disponible: <https://userpages.umbc.edu/~khoo/survey1.html>

[68] "Plantilla de Diagrama de Clases UML | Moqups". Online Mockup, Wireframe & UI Prototyping Tool · Moqups. <https://moqups.com/es/templates/diagrams-flowcharts/uml-diagrams/class-diagram/>.

[69] "Tutorial de diagrama de secuencia UML". Lucidchart. <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>.

[70] H. Braumann. "¿Qué es UX y UI? Descubre todo lo que hay detrás en el diseño de un sitio o una app". Crehana. <https://www.crehana.com/blog/transformacion-digital/que-es-ux-y-como-te-ayudara-ser-profesional-completo/>.

[71] "Organization Schemes | Usability.gov". Home | Usability.gov. <https://www.usability.gov/how-to-and-tools/methods/organization-schemes.html>.

[72] "Organization Structures | Usability.gov". Home | Usability.gov. <https://www.usability.gov/how-to-and-tools/methods/organization-structures.html>.

[73, p.6] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea]. Disponible: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[74, p.32] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea]. Disponible: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[75, p.34] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[76, p.37] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[77, pp.43-44] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[78, p.62] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[79, p.76] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[80, p.92] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[81, p.95] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[82, p.98] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[83, p.104] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[84, p.108] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea].
Disponibile: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[85, p.119] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea]. Disponible: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[86, p.128] K. Zieba, J. Cao, B. Gremillion y M. Ellis, *Web UI Design Patterns 2016*. 2016. [En línea]. Disponible: <http://www.bionicrab.com/temp/ux-books/Web%20UI%20Design%20Patterns%202016%20Volume%201.pdf>

[87, p.1] Diagrama de despliegue, 8 de diciembre de 2012, *Diagrama de despliegue*, Bolivia. [En línea]. Disponible: <http://virtual.usalesiana.edu.bo/web/conte/practica/22012/2132.pdf>

[88] Bass, L. (2003). *Software architecture in practice (2a ed.)*. Addison-Wesley.

[89] K. McArthur, *Pro PHP, Patterns, Frameworks, Testing and more*. Apress, 2008. [En línea]. Disponible: <https://doi.org/10.1007/978-1-4302-0279-0>

[90] "¿Qué es el desarrollo web? | Euroinnova". Euroinnova Business School. <https://www.euroinnova.co/blog/que-es-el-desarrollo-web>.

[91] "HTML: Lenguaje de etiquetas de hipertexto | MDN". MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTML>.

[92] J. Eguíluz Pérez, *Introducción a CSS*. [En línea]. Disponible: https://www.jesusda.com/docs/ebooks/introduccion_css.pdf

[93] D. P. Pérez Miranda, "TECNÓLOGA EN INFORMÁTICA MENCIÓN ANÁLISIS DE SISTEMAS", INSTITUTO SUPERIOR TECNOLÓGICO "SAN GABRIEL", Riobamba, 2021. [En línea]. Disponible: <https://sangabrielriobamba.edu.ec/tesis/sistemas/tesis30.pdf>

[94] "Introduction | PurgeCSS". PurgeCSS - Remove unused CSS | PurgeCSS. <https://purgecss.com/introduction.html>.

[95] "Lenguajes de programación", octubre de 2008, *Lenguajes de programación*, España. [En línea]. Disponible: https://guimi.net/descargas/Monograficos/G-Lenguajes_de_programacion.pdf

[96] J. Eguíluz Pérez, *Introducción a JavaScript*. 2008. [En línea]. Disponible: https://www.jesusda.com/docs/ebooks/introduccion_javascript.pdf

[97] Tutorialspoint, *Jquery Tutorial*. Tutorialspoint, 2015. [En línea]. Disponible: https://www.tutorialspoint.com/jquery/jquery_tutorial.pdf

[98] *Ada, Introducción a la programación backend, Introduccion-a-Backend-v20201110_1900.pptx.pdf*, Buenos Aires, Argentina. [En línea]. Disponible: https://adaitw.org/wp-content/uploads/2020/11/Introduccion-a-Backend-v20201110_1900.pptx.pdf

[99] *L. G. Vazquez, Introducción a Python, 012_Introduccion a Python.pdf*, Argentina. [En línea]. Disponible: http://cdr.ing.unlp.edu.ar/files/presentaciones/012_Introduccion%20a%20Python.pdf

[100] *K. Reitz y T. Schlusser, The Hitchhiker's Guide to Python. Sebastopol: O Reilly, 2016.* [En línea]. Disponible:

<https://books.google.es/books?hl=es&lr=&id=nHDtDAAAQBAJ&oi=fnd&pg=PR2&dq=pep8+python&ots=kz5Nli9->

[Ps&sig=WvLNsyYOAESP2bvNSwSjfyCXYgg#v=onepage&q=pep8%20python&f=false](https://books.google.es/books?hl=es&lr=&id=nHDtDAAAQBAJ&oi=fnd&pg=PR2&dq=pep8+python&ots=kz5Nli9-Ps&sig=WvLNsyYOAESP2bvNSwSjfyCXYgg#v=onepage&q=pep8%20python&f=false)

[101] *S. Mukherjee, A. Almanza y C. Rubio-González, "Fixing Dependency Errors for Python Build Reproducibility", SIGSOFT, n.º 978, p. 13, julio de 2021.* [En línea]. Disponible: <https://doi.org/10.1145/3460319.3464797>

[102] *J. J. Gutiérrez, ¿Qué es un framework web?, Microsoft Word - Framework.doc.* [En línea]. Disponible: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

[103] *"Introducción a Django - Aprende sobre desarrollo web | MDN". MDN Web Docs.* <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>.

[104] *"4.1. Introducción a WSGI — Materiales del entrenamiento de frameworks de Desarrollo Web en Python - Nivel básico". Frameworks de Desarrollo Web en Python — Materiales del entrenamiento de frameworks de Desarrollo Web en Python - Nivel básico.* https://entrenamiento-frameworks-web-python.readthedocs.io/es/latest/leccion4/introduccion_wsgi.html.

[105] *A. Corrales Rubiano y C. Y. López Herrera, IDENTIFICADORES DIGITALES: UNA HERRAMIENTA QUE APOYA LA RECUPERACIÓN DE INFORMACIÓN, indentificadores.pdf.* [En línea]. Disponible: <http://eprints.rclis.org/10599/1/indentificadores.pdf>

[106] *"Tutorial de Django Parte 8: Autenticación y permisos de Usuario - Aprende sobre desarrollo web | MDN". MDN Web Docs.* <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Authentication>.

[107] *M. A. Cruz-Chávez, "Bases de Datos, Conceptos y sus Características", 10 de septiembre de 2006, Capítulo 1, Cuernavaca, México.* [En línea]. Disponible: <http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/bd1.pdf>

[108] L. A. SALTOS PEREZ, "ESTUDIO COMPARATIVO ENTRE BASES DE DATOS RELACIONAL Y NO RELACIONAL", Trabajo de grado, UNIVERSIDAD TÉCNICA DE BABAHOYO, BABAHOYO. [En línea]. Disponible: <http://dspace.utb.edu.ec/bitstream/handle/49000/11669/E-UTB-FAFI-SIST-000326.pdf?sequence=1&isAllowed=y>

[109] E. M. Suárez, "¿Que es una base de datos relacional?", agosto de 2008, Base de datos relacional.pdf. [En línea]. Disponible: <http://www.uprh.edu/adem/Base%20de%20datos%20relacional.pdf>

[110] "QuÃ© es un diagrama entidad-relaciÃ³n". Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>.

[111, p.13] de Guevara, L. V. (6 de julio de 2021). Gestión de Bases de Datos Versión 1.0. Gestión de Bases de Datos. <https://readthedocs.org/projects/gestionbasesdatos/downloads/pdf/latest/>

[112] "Fork - pgPedia - a PostgreSQL Encyclopedia". pgPedia - a PostgreSQL Encyclopedia. <https://pgpedia.info/f/fork.html>.

[113] P. Denzer, "PostgreSQL", 23 de octubre de 2002, informe.pdf. [En línea]. Disponible: <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>

[114] "¿Qué es una API?" Red Hat - We make open source technologies for the enterprise. <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>.

[115] "Software Testing: Conocé El Ciclo De Vida De Las Pruebas De Software". Trans-Ti. <https://trans-ti.com/2021/01/11/software-testing-conoce-el-ciclo-de-vida-de-las-pruebas-de-software/>.

[116, p.6] V. KHORIKOV, Unit Testing: Principles, Practices, and Patterns. MANNING. [En línea]. Disponible: [https://sd.blackball.lv/library/unit_testing_\(2020\).pdf](https://sd.blackball.lv/library/unit_testing_(2020).pdf)

[117] P. Kua, "UNIT TESTING", Microsoft Word - AppsUnitTesting-1.doc. [En línea]. Disponible: <https://www.thekua.com/publications/AppsUnitTesting.pdf>

[118] "Writing and running tests | Django documentation | Django". Django documentation | Django documentation | Django. <https://docs.djangoproject.com/en/4.1/topics/testing/overview/>.

[119] "Despliegue de Software - EcuRed". EcuRed. https://www.ecured.cu/Despliegue_de_Software.

[120, p.84] Á. Mompó Camarasa, "Estudio de sistema de control de versiones y uso de los mismos en proyectos de integración continua", Trabajo de grado, Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València. [En línea]. Disponible:

<https://m.riunet.upv.es/bitstream/handle/10251/87104/MOMPÓ%20-%20Estudio%20de%20sistema%20de%20control%20de%20versiones%20y%20uso%20de%20los%20mismos%20en%20proyectos%20de%20integración....pdf?sequence=1&isAllowed=y>

[121, p.5-6] Itaú, "Gestión Financiera", 2019, *Manual-gestion-financiera-vfinal-2.pdf*, Sao Pablo, Brasil. [En línea]. Disponible:

[122, p.30] O. E. Casteblanco, *Cómo definir los costos de tu empresa*. Bogotá D.C: Editorial Kimpres Ltda., 2009.

[123, p.31] O. E. Casteblanco, *Cómo definir los costos de tu empresa*. Bogotá D.C: Editorial Kimpres Ltda., 2009.

[124, p.6] J. Vásquez, B. Aguilar, B. Robledo y B. Montecinos, "Material de profundización AMORTIZACIONES, DEPRECIACIONES Y PROVISIONES", febrero de 2017, *Amortizaciones, depreciaciones y provisiones*, Santiago de Chile, Chile, Versión 1.0. [En línea]. Disponible: http://www.inacap.cl/web/material-apoyo-cedem/alumno/Contabilidad/Contabilidad-Basica%20CCCB01/CCPR03_U5_MP2.pdf

[125, p.82] B. H. S. DILIA M, "DISEÑO DE UN SISTEMA DE COSTOS PARA OPTIMIZAR EL PROCESO DE TRAMITACIÓN DE EQUIVALENCIAS. CASO: URBE", Trabajo de grado, UNIVERSIDAD DR. RAFAEL BELLOSO CHACÍN, Maracaibo, 2000.

[126] H. Sy Corvo. "Inversión inicial en una empresa: qué es, estructura, ejemplo". Lifered. <https://www.lifered.com/inversion-inicial/>.

[127, p.4] A. L. Cordoba, "EL FLUJO DE CAJA COMO HERRAMIENTA GERENCIAL PARA EL CONTROL FINANCIERO", *Ensayo de Postgrado*, UNIVERSIDAD MILITAR NUEVA GRANADA, Bogotá D.C, 2017.

[128, p.13] H. E. González de Paz, I. C. Quintanilla Rivas, V. L. Sanchés Raymundo, "NIIF PYMES PARA LA ELABORACIÓN, PRESENTACIÓN Y REVELACIÓN DE ESTADOS FINANCIEROS DE LAS EMPRESAS COMERCIALIZADORAS DE PRODUCTOS FARMACÉUTICOS, EN EL MUNICIPIO DE SAN SALVADOR", Trabajo de grado, UNIVERSIDAD DE EL SALVADOR, San Salvador, 2012.

[129, p.4] J. Vásquez, B. Aguilar, B. Robledo y B. Montecinos, "ESTADOS FINANCIEROS BÁSICOS " febrero de 2017, *Estados financieros básicos*, Santiago de Chile, Chile, Versión 1.0

[130] Santiago. "¿Qué son las empresas de TRIPLE IMPACTO? Las mejores del FUTURO". Red Kopora. <https://redkopora.com/desarollo-sostenible/empresas-de-triple-impacto/>.

[131, p.17] K. S. LUCUMI GAMBOA, "ANÁLISIS DE VIABILIDAD FINANCIERA PARA UNA ENTIDAD SIN ÁNIMO DE LUCRO (ESAL) DEL DISTRITO ESPECIAL DE BUENAVENTURA: PROYECTO ASOCIACIÓN MAREA VIVA DEL PACÍFICO 2019", *Master en Administración Financiera, UNIVERSIDAD EAFIT, CALI, 2019.*

[132, p.15] K. S. LUCUMI GAMBOA, "ANÁLISIS DE VIABILIDAD FINANCIERA PARA UNA ENTIDAD SIN ÁNIMO DE LUCRO (ESAL) DEL DISTRITO ESPECIAL DE BUENAVENTURA: PROYECTO ASOCIACIÓN MAREA VIVA DEL PACÍFICO 2019", *Master en Administración Financiera, UNIVERSIDAD EAFIT, CALI, 2019.*

[133, p.5] J. Arias Ramírez, "ANEXO 3. PRESUPUESTO", ANEXO;jsessionid=1D072BA55B62DBE7D235B293D9B03218, Colombia.

[134, p.64] C. M. MARTÍNEZ STONE, ""Evaluación Económica e Inversión sobre un Condominio Horizontal en la Delegación Álvaro Obregón"", Tesis, Universidad Nacional Autónoma de México, Ciudad de México, 2002. [En línea]. Disponible: <http://www.economia.unam.mx/secss/docs/tesisfe/MartinezSCM/tesis.html>

[135, p.18] K. S. LUCUMI GAMBOA, "ANÁLISIS DE VIABILIDAD FINANCIERA PARA UNA ENTIDAD SIN ÁNIMO DE LUCRO (ESAL) DEL DISTRITO ESPECIAL DE BUENAVENTURA: PROYECTO ASOCIACIÓN MAREA VIVA DEL PACÍFICO 2019", *Master en Administración Financiera, UNIVERSIDAD EAFIT, CALI, 2019.*

[136] "Impacto social - ES|IMPACT". ES|IMPACT. <https://www.esimpact.org/impacto-social/>.

[137] "IMPACTO AMBIENTAL IMPACTOS MEDIO AMBIENTALES | GRN". CONSULTORIA AMBIENTAL. <https://www.grn.cl/impacto-ambiental.html>.

[138] A. PASTRÁN, "Colombia superó los 8,2 millones de accesos fijos a internet en segundo trimestre", *La República*, p. 1, 12 de noviembre de 2021. [En línea]. Disponible: <https://www.larepublica.co/economia/colombia-supero-los-82-millones-de-accesos-fijos-a-internet-en-el-segundo-trimestre-3261097>

[139] B. X. BECERRA, "Colombia, entre los países que más tiempo pasan en las redes sociales", *La República*, p. 1, 15 de mayo de 2022. [En línea]. Disponible: <https://www.larepublica.co/globoeconomia/colombia-es-uno-de-los-paises-en-los-que-mas-tiempo-se-gasta-en-las-redes-sociales-3363907#:~:text=El%20estudio%20publicado%20por%20DataReportal,alta%20que%20el%20promedio%20mundial.>

3363907#:~:text=El%20estudio%20publicado%20por%20DataReportal,alta%20que%20el%20promedio%20mundial.

[140] K. García. "El consumo de social media en Latinoamérica - #LatamDigital 2021". Interlat. <https://interlat.co/el-consumo-de-social-media-en-latinoamerica-2021/>.

[141] J. TORO, "Acceso a redes sociales en América Latina aumentó a 82% en el año de la pandemia", *La República*, p. 1, 24 de abril de 2021. [En línea]. Disponible: <https://www.larepublica.co/globoeconomia/acceso-a-redes-sociales-en-america-latina-aumento-a-82-en-primer-ano-de-pandemia-3158549>

[142] M. Mohsin. "10 Video Marketing Statistics You should Know for 2022 [Infographic]". Oberlo | Where Self Made is Made. <https://www.oberlo.com/blog/video-marketing-statistics>.

[143] S. QUEIRUGA. "Qué es y cómo funciona Twitch, la plataforma de streaming de referencia". *Marketing 4 Ecommerce - Tu revista de marketing online para e-commerce*. [https://marketing4ecommerce.net/que-es-y-como-funciona-twitch/#:~:text=Dos%20tercios%20de%20los%20usuarios,%20y%20millennials%20\(32%\)](https://marketing4ecommerce.net/que-es-y-como-funciona-twitch/#:~:text=Dos%20tercios%20de%20los%20usuarios,%20y%20millennials%20(32%)).

[144] International Telecommunication Union, *Measuring digital development Facts and figures 2021*. Geneva: ITU, 2021. [En línea]. Disponible: <https://img.lalr.co/cms/2021/12/10163813/Facts-and-figures-2021.pdf>

[145] "Así perciben las redes sociales los jóvenes de Neiva". <https://www.alcaldianeiva.gov.co/NuestraAlcaldia/SalaDePrensa/Paginas/Asi-perciben-las-redes-sociales-los-jovenes-de-Neiva.aspx>.

[146] K. Molenaar. "20+ Patreon Statistics You Need to Know". *Influencer Marketing Hub*. <https://influencermarketinghub.com/patreon-stats-revenue-users/>.

[147] B. Dean. "Fiverr Usage and Growth Statistics: How Many People Use Fiverr in 2022?" *Backlinko*. <https://backlinko.com/fiverr-users>.

[148] J. M. LÓPEZ BEJARANO, "Las millonarias cifras que está recaudando Onlyfans en Colombia con creadores locales", *La República*, p. 1, 26 de abril de 2021. [En línea]. Disponible: <https://www.larepublica.co/ocio/las-millonarias-cifras-que-esta-recaudando-onlyfans-en-colombia-con-creadores-locales-3158917>

[149] L. H. Wei. "10 OnlyFans Statistics You Need to Know in 2022". *Followchain*. <https://www.followchain.org/onlyfans-statistics/#1>.

[150] "OnlyFans". *OnlyFans*. <https://onlyfans.com/help/3/16/59>.

[151] "OnlyFans". OnlyFans. <https://onlyfans.com/help/2/12/134>.

[152] A. K. "Go Live On OnlyFans - OnlyFans Blog". OnlyFans Blog. <https://blog.onlyfans.com/go-live-on-onlyfans/>.

[153] B. Dean. "Patreon: Subscriber and Creator Statistics for 2022". Backlinko. <https://backlinko.com/patreon-users>.

[154] "¿Qué tarifas tendré que pagar como creador?" Patreon Help Center. <https://support.patreon.com/hc/es-es/articles/360027674431--Qué-tarifas-tendré-que-pagar-como-creador->.

[155] "Pricing | Patreon". Creativity powered by membership | Patreon. <https://www.patreon.com/pricing>.

[156] "Ko-fi Shop | Sell Digital and Physical Products With Just a Link. No Listing Fees." Ko-fi | More. <https://more.ko-fi.com/shop>.

[157] "Using Ko-fi for Commissions and Services". Ko-fi Help Center. <https://help.ko-fi.com/hc/en-us/articles/360016170433-Using-Ko-fi-for-Commissions-and-Services#:~:text=Ko-fi%20Commissions%20is%20a,advance%20directly%20into%20your%20PayPal>

[158] "MODELO DE CONSTITUCIÓN DE UNA SOCIEDAD POR ACCIONES SIMPLIFICADAS – S.A.S.", desc_56a07bd36fab096885ddf53495f15f65.pdf, Cúcuta, Colombia.

Figuras

Fig 1. Organigrama de “ Bizneo“ .[3]

Fig 2. Manual de funciones de “Alteco consultores Desarrollo y Gestión” .[3]

Fig 3. Requisitos de contratación de “Calameo By kenneth Samayoa Estrada”. [1]

Fig 4. Objetivos estratégicos de “Hubspot” .[5]

Fig 5. Diagrama de secuencias de “It SourceCode” .[1]

Fig 6. Ciclo de vida del software desde la metodología ágil de desarrollo de “Okeke” .[1]

Fig 7. Ciclo del Sprint en la metodología Scrum de “pm-partners” .[1]

Fig 8. Plantilla de los componentes de las historias de usuario de " oficinaproyectosinformatica ". [1]

Fig 9. Tablero Kanban para proyectos de Software de “Trello” .[2]

Fig 10. Diagrama de clases de un e-commerce de “Moqups” .[1]

Fig 11. Diagrama de secuencias de “El Lenguaje Unificado de Modelado By Grady Booch, Jim Rumbaugh e Ivar Jacobson” .[4]

Fig 12. Diagrama de casos de uso de “Wondershare” .[5]

Fig 13. Estructura Jerárquica de “Palbin” .[3]

Fig 14. Inputs Hints de “Interactions Design Foundation” .[4]

Fig 15. Forgiving Formats de “Web UI Design Patterns 2016 By Jerry Cao, Ben Gremillion, Kamil Zieba y Matt Ellis” .[31]

Fig 16. Undo de “Franbravo” .[14]

Fig 17. Inline Validation de “FalconMasters” .[1]

Fig 18. Social Login "Zendeska". [4]

Fig 19. Flagging/Reporting de "Sergio Ramirez" . [1]

Fig 20. WYSIWYG Text Editor de "iQuery AZ" . [1]

Fig 21. Searches de "Trucos" .[1]

Fig 22. Notifications de "This Intersts Me".[1]

Fig 23. Sticky (Fixed) " Contesntsquare " Navigation [4]

Fig 24. Vertical Navigation "Patternfly" [1]

Fig 25. Modals "Appcues" [7]

Fig 26. Recommendations "Barilliance" [3]

Fig 27. Diagrama de despliegue de "DiagramasUML" .[19]

Fig 28. Interacción entre el cliente y el servidor en el patrón Modelo Vista Controlador "Jose Aguilar" [1]

Fig 29. Arquitectura MVT "Espifreelancer" [2]

Fig 30. Conceptos del modelo entidad relación
"<https://cursos.aiu.edu/Base%20de%20Datos/pdf/Tema%203.pdf>" [1]

Fig 31. Diagrama entidad relación "wikiwand" [2]

Fig 32. Testing Unitario "Django Documentation" [1]

Fig 33. Uso de internet en EEUU "relevantaudience" [2]

Fig 34. Consumo de OnlyFans "La República" [1]

Fig 35. Diagrama No 1. Causa y efecto de "Needer By Juan Trujillo y Thomas Pérez".

Fig 36. Diagrama No 2. Causa y efecto de "Needer By Juan Trujillo y Thomas Pérez".

Fig 37. Diagrama No 3. Causa y efecto de "Needer By Juan Trujillo y Thomas Pérez".

Fig 38. Diagrama No 4. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 39. Diagrama No 5. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 40. Diagrama No 6. Causa y efecto de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 41. Gráfico de edad de los encuestados de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 42. Gráfico de género de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 43. Gráfico de visitantes de las aplicaciones web de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 44. Gráfico de Dinero gastado mensualmente en el consumo de contenido multimedia por parte de los encuestados de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 45. Gráfica de horas gastadas mensualmente en redes sociales de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 46. Gráfico de interés de uso de prototipo web “Needer By Juan Trujillo y Thomas Pérez”.

Fig 47. Gráfico de interés de uso de prototipo web con productos multimedia de “Needer By Juan Trujillo y Thomas Pérez”.

Fig 48. Patreon y OnlyFans de "Sofía Gray" [6]

Fig 49. Colores de la marca “Needer By Juan Trujillo y Thomas Pérez”.

Fig 50. Organigrama de Needer “Needer By Juan Trujillo y Thomas Pérez”.

Fig 51. Búsqueda de Homónimos de "Leegales" [1]

Fig 52. Resultados de Homónimos con respecto a Needer de "Cámara de Comercio de Bogotá"

Fig 53. Historias de usuario organizadas en el Backlog de Trello.

Fig 54. Planning Poker.

Fig 55. Sprint 1 Burndown

Fig 56. Sprint 2 Burndown

Fig 57. Sprint 3 Burndown

Fig 58. Sprint 4 Burndown

Fig 59. Sprint 5 Burndown

Fig 60. Sprint 6 Burndown

Fig 61. Sprint 7 Burndown

Fig 62. Sprint 8 Burndown

Fig 63. Sprint 9 Burndown

Fig 64. Sprint Burndown General

Fig 65. Checklist de una historia de usuario.

Fig 66. Diagrama de casos de uso Needer.

Fig 67. Diagrama de secuencias Needer.

Fig 68. Diagrama de clases Needer.

Fig 69. Diseño de base de datos Needer.

Fig 70. Arquitectura del sistema Needer.

Fig 71. Diagrama de despliegue Needer.

Fig 72. Diseño estándar botones y tipografías para el producto.

Fig 73. Diseño estándar formulario y colores para el producto.

Fig 74. Diseño estándar de logos en diferentes dispositivos para el producto.

Fig 75. Mockup U-1.

Fig 76. Mockup U-2.

Fig 77. Mockup U-3.

Fig 78. Mockup U-4.

Fig 79. Mockup U-6.

Fig 80. Mockup U-8.

Fig 81. Mockup U-5.

Fig 82. Diseño Mockup U-21.

Fig 83. Mockup U-36.

Fig 84. Mockup U-36.

Fig 85. Mockup CD-1.

Fig 86. Mockup CD-2.

Fig 87. Mockup CD-10.

Fig 88. Mockup CD-06.

Fig 89. Mockup CD-03.

Fig 90. Mockup CD-10.

Fig 91. Mockup U-44.

Fig 92. Mockup CD-22.

Fig 93. Mockup U-24.

Fig 94. Mockup U-26.

Fig 95. Mockup CD-11.

Fig 96. Mockup U-32.

Fig 97. Mockup U-35.

Fig 98. Mockup U-30.

Fig 99. Mockup U-31.

Fig 100. Mockup M-7.

Fig 101. Mockup M-10.

Fig 102. Mockup M-9.

Fig 103. Mockup U-45.

Fig 104. Creación de la carpeta Proyecto.

Fig 105. Creación del entorno virtual.

Fig 106. Instalación de Django versión 4.1.7.

Fig 107. Creación del proyecto needer.

Fig 108. Migraciones realizadas a la base de datos local.

Fig 109. Interfaz gráfica administrador Django.

Fig 110. Configuración de las plantillas.

Fig 111. Configuración de la URL de los archivos estáticos.

Fig 112. Árbol de directorios virtual de needer.

Fig 113. Creación de la aplicación users.

Fig 114. Aplicaciones instaladas en el módulo settings.py.

Fig 115. Creación del modelo Pais.

Fig 116. Creación del modelo Celebridad.

Fig 117. Creación del modelo User.

Fig 118. Gestión del documento admin.py.

Fig 119. Interfaz gráfica actualizada del administrador de Django.

Fig 120. Vistas registro de usuario.

Fig 121. URLS realizadas en el archivo urls.py del paquete Users.

Fig 122. Templates diseñados en el paquete Users.

Fig 123. Landing-Page de Needer.

Fig 124. Formulario de Registro de Needer.

Fig 125. Formulario de autenticación de Needer.

Fig 126. Ajustes del perfil del usuario.

Fig 127. Creación de la app social.

Fig 128. Asociación modelo user.

Fig 129. Modelos likepublicacion y comentario.

Fig 130. Métodos creados para las clases likepublicacion y comentario.

Fig 131. Vista crearpublicacon.

Fig 132. Formulario crearpublicacon.

Fig 133. Función dentro de formulario crearpublicacon.

Fig 134. URLS aplicación social.

Fig 135. Templates aplicación social.

Fig 136. Vista Home social.

Fig 137. Vista Crear Publicación.

Fig 138. Vista Publicación Creada.

Fig 139. Creación aplicación chat.

Fig 140. Modelo seguidorusuario.

Fig 141. Vista seguirusuario y sus funciones.

Fig 142. Métodos seguidores.

Fig 143. Templates seguidores y seguidos.

Fig 144. URLS aplicación chat.

Fig 145. Modelo threadmanager.

Fig 146. Modelo thread.

Fig 147. Modelo chatmessage.

Fig 148. Vista threadcreatview.

Fig 149. Configuración models consumer.

Fig 150. Clase chat y métodos.

Fig 151. URLS consumer.

Fig 152. URLS thread.

Fig 153. Organización de templates y archivos javascript.

Fig 154. Vista chat.

Fig 155. Vista perfil seguidos.

Fig 156. Vista usuarios seguidos.

Fig 157. Organización de las carpetas del proyecto.

Fig 158. Imports prueba unitaria HU U-1.

Fig 159. Prueba unitaria U-1 escenario 1.

Fig 160. Prueba unitaria U-1 escenario 2.

Fig 161. Prueba unitaria U-1 escenario 9.

Fig 162. Prueba unitaria U-1 escenario 10.

Fig 163. Imports prueba unitaria HU U-2.

Fig 164. SetUp pruebas unitarias HU U-2.

Fig 165. Prueba unitaria U-2 escenario 1.

Fig 166. Prueba unitaria U-2 escenario 2.

Fig 167. Prueba unitaria U-2 escenario 3.

Fig 168. Organización de las carpetas del proyecto.

Fig 169. Imports prueba unitaria HU U-6.

Fig 170. Prueba unitaria U-6 escenario 1.

Fig 171. Prueba unitaria U-6 escenario 2.

Fig 172. Prueba unitaria U-6 escenario 5.

Fig 173. Resultados de las pruebas unitarias HU U-1.

Fig 174. Resultados de las pruebas unitarias HU U-2.

Fig 175. Instalación de la aplicación Compressor.

Fig 176. Adición de la aplicación Compressor al listado de apps de Needer.

Fig 177. Adición del buscador Compressor al listado de búsqueda de Needer.

Fig 178. Ejecución del Compressor en el proyecto.

Fig 179. Obtención de los CSS Compuestos.

Fig 180. Variables de entorno.

Fig 181. Base de datos.

Fig 182. Configuración del CDN.

Figura 183. Actualización nameservers en Hostinger.

Figura 184. DNS enlazados con el Hostname DigitalOcean.

Figura 185. Selección de la región para el alojamiento del servidor.

Figura 186. Datacenter situado en New York.

Figura 187. Selección del Ubuntu Server versión 22.04.

Figura 188. Selección del tamaño del entorno virtual.

Figura 189. Droplet en ejecución.

Figura 190. Creación del directorio project.

Figura 191. Configuración en el directorio del project.

Figura 192. Creación de los archivos credentials.json y .env.

Figura 193. Configuración del gunicorn.socket.

Figura 194. Ejecución del gunicorn.service.

Figura 195. Configuración Nginx.

Figura 196. Test realizado al nginx.

Figura 197. Paso de tráfico puerto 80 y configuración del puerto 8000.

Figura 198. Proyecto Desplegado.

Tablas

1. Rango de precios promedio de contenido multimedia.

2. Inversiones diferidas

3. Capital de trabajo

4. Inversión Fija

5. Total inversión inicial

6. Ganancias Brutas

7. Porcentaje de ganancias

8. Problemas encontrados por parte de los usuarios

9. Cuadro comparativo entre la competencia.

10. Manual de Funciones CEO.

11. Requisitos de contratación.

12. Capital suscrito por accionistas

13. Características de las acciones

14. Epicas

15. HU U-2

16. Niveles de priorización de las historias de usuario.

17. Sprint 1

18. Sprint 2

19. Sprint 3

20. Sprint 4

21. Sprint 5

22. Sprint 6

23. Sprint 7

24. Sprint 8

25. Sprint 9

26. Storypoints Sprint 1

27. Storypoints Sprint 2

28. Storypoints Sprint 3

29. Storypoints Sprint 4

30. Storypoints Sprint 5

31. Storypoints Sprint 6

32. Storypoints Sprint 7

33. Storypoints Sprint 8

34. Storypoints Sprint 9

35. Sprint Burndown 1

36. Sprint Burndown 2

37. Sprint Burndown 3

38. Sprint Burndown 4

39. Sprint Burndown 5

40. *Sprint Burndown 6*
41. *Sprint Burndown 7*
42. *Sprint Burndown 8*
43. *Sprint Burndown 9*
44. *Diccionario de datos tabla usuario.*
45. *Diccionario datos tabla país.*
46. *Diccionario de datos tabla group.*
47. *Diccionario de datos tabla tipo_celebridad.*
48. *Diccionario de datos tabla tipo_celebridad_usuario.*
49. *Diccionario de datos seguidores_usuario.*
50. *Estructura diccionario de datos tabla publicacion.*
51. *Diccionario de datos tabla liked_publicacion.*
52. *Estructura diccionario de datos tabla comentario.*
53. *Estructura diccionario de datos liked_comentario.*
54. *Estructura diccionario de datos thread.*
55. *Estructura diccionario de datos message.*
56. *Estructura diccionario de datos emailaddress.*
57. *Estructura diccionario de datos sites.*
58. *Estructura diccionario de datos socialapp.*
59. *Estructura diccionario de datos socialappsites.*

60. Estructura diccionario de datos socialaccount.

61. Plan de Pruebas HU U-1.

62. Plan de Pruebas HU U-2.

63. Plan de Pruebas HU U-6.

64. Plan de pruebas actualizado HU U-1.

65. Plan de pruebas actualizado HU U-2.

66. Plan de pruebas actualizado HU U-6.

67. Proyección de ingresos anuales

68. Proyección de ingresos mensuales

69. Gastos de depreciacion y amortizacion

70. Gastos Generales

71. Inversión Inicial Capital de Trabajo

72. Inversiones Iniciales

73. Financiamiento de Inversiones

74. Flujo de caja de la empresa

75. Estado de resultados

76. Porcentaje pago de impuestos y otros

77. Balance de la empresa

78. Análisis financiero

