



**SIMULADOR DE LA MECÁNICA DE FLUIDOS PARA OPERACIONES DE  
LIMPIEZA Y SEPARACIÓN DE FASES EN POZOS VERTICALES Y  
HORIZONTALES CON TECNOLOGIA COILED TUBING**

**Proyecto de Grado, modalidad Tesis de Grado  
Presentado por:**

**JENNY CATALINA GONZALEZ PEÑA  
Cód. 2009179044**

**MAGDA ALEXANDRA TRUJILLO JIMÉNEZ  
Cód. 2009283378**

**PROGRAMA DE INGENIERÍA ELECTRÓNICA  
UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
NEIVA, HUILA  
2014**



**SIMULADOR DE LA MECÁNICA DE FLUIDOS PARA OPERACIONES DE  
LIMPIEZA Y SEPARACIÓN DE FASES EN POZOS VERTICALES Y  
HORIZONTALES CON TECNOLOGIA COILED TUBING**

**Proyecto de Grado presentado para optar el título de:  
INGENIERO ELECTRÓNICO**

**JENNY CATALINA GONZALEZ PEÑA  
Cód. 2009179044**

**MAGDA ALEXANDRA TRUJILLO JIMÉNEZ  
Cód. 2009283378**

**Director  
ING. GERMAN MARTINEZ B.  
Docente Programa Ingeniería Electrónica**

**PROGRAMA DE INGENIERÍA ELECTRÓNICA  
UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
NEIVA, HUILA  
2014**



Nota de aceptación:

---

---

---

---

---

---

---

---

---

Firma del Director del Proyecto

---

Firma del Primer Jurado

---

Firma del Segundo Jurado

Neiva, 22 de Septiembre 2014

## DEDICATORIA

A Dios y la Santísima Virgen por bendecirme en cada momento de mi vida y colmarme de sabiduría y entendimiento ante cada circunstancia presentada;

A mis padres Edna Yamile y Jacob, y mi hermana Goretty por brindarme su amor y creer siempre en mí, porque ellos cada día me fortalecen con sus consejos y alientos haciendo de mí una gran persona para enfrentar el diario vivir;

A mi maravillosa familia en general, pero en especial a mi abuela Albina, tío Gilberto, tía Dennis y tía Damaris quienes me guiaron y ayudaron para el logro de mis objetivos y a quienes les debo toda su dedicación y amor para conmigo;

A mi novio Daniel Gilberto Barreiro quien siempre me apoya de manera incondicional en cada uno de mis momentos pero que sobre todo siente gran admiración por mí.

A todas aquellas personas que me han brindado su ayuda sin esperar nada a cambio y quienes depositan su confianza en mí.

**JENNY CATALINA GONZALEZ PEÑA**

A Dios primeramente, por darme la vida y permitirme disfrutarla cerca de mi familia, por poner en mí la chispa del conocimiento y creatividad con la que hoy puedo contar y sigo cultivando.

A las personas que me han dado su infinito amor y apoyo incondicional, mi familia, quienes son mi fuerza e inspiración. Pilar, mi madre, ejemplo de fe, fuerza y perseverancia; Carlos, mi padre, quien ha instruido en mi responsabilidad, esfuerzo y disciplina. Mi querido hermano Mario, a quien admiro por su nobleza y gran carisma. A Luis, mi novio, por su amor y compañía.

**MAGDA ALEXANDRA TRUJILLO JIMÉNEZ**

## AGRADECIMIENTOS

*Los autores expresamos en primer lugar un enorme agradecimiento a Dios y La Virgen Santísima porque sin ayuda de ellos nada sería posible; y a nuestras familias por brindarnos su apoyo incondicional y constante, pero sobre todo por creer en cada momento en las capacidades nuestras como personas para llegar a ser grandes profesionales.*

*Igualmente agradecemos a la Empresa Guacamaya Oil Services S.A.S y su personal en general, pero de manera especial al Ingeniero Juan David Tarache Serrano, Gerente de Operaciones, quien confió éste excelente proyecto en nosotras brindándonos así su apoyo y guía en cada etapa del mismo.*

*A su vez manifestamos nuestros más sinceros agradecimientos al Ingeniero German Martínez B. docente de nuestro Programa, quien fue maestro y guía constante como Director en el presente proyecto de Grado y a todos nuestros docentes en el transcurso de la Carrera de Ingeniería Electrónica ya que gracias a sus conocimientos y enseñanzas han aportado para nuestro fortalecimiento como profesionales.*

*A todas aquellas personas que de una u otra manera nos colaboraron para culminar nuestros estudios universitarios de Pre-grado.*

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	14
1. INFORMACIÓN DEL PROYECTO.....	16
1.1 Formulación del Problema .....	16
1.2 Descripción del Proyecto.....	16
1.3 Justificación.....	16
1.4 OBJETIVOS .....	17
1.4.1 Objetivo General.....	17
1.4.2 Objetivos Específicos .....	17
2. MARCO TEORICO .....	19
2.1 COILED TUBING .....	19
2.1.1 Ventajas del Coiled Tubing Operativas.....	19
2.1.2 Limpieza de Pozos .....	20
2.2 EJECUCIÓN OPERACIONES (LIMPIEZA DE ARENAS – SEPARACIÓN DE FASES) EN POZOS PETROLEROS UTILIZANDO COILED TUBING .....	20
2.2.1 Pasos de proceso de Limpieza de Arenas .....	23
3. GENERALIDADES.....	26
3.1 REVISIÓN BIBLIOGRÁFICA.....	26
3.1.1 Solid Works.....	26
3.1.2 CONTROL ON/OFF.....	29
3.1.2.1 Sistema Operativo Android.....	29
3.1.2.2 Arduino Leonardo .....	32
3.1.2.3 Actuadores.....	40
3.1.3 SHARP DEVELOP .....	41
3.1.3.1 ZedGraph.....	42
3.1.4 Medición de Presión .....	43
3.1.5 Medición de Caudal a través de Flujo.....	44
4. FUNCIONAMIENTO GENERAL .....	46

4.1	DISEÑO DEL PROTOTIPO SIMULADOR.....	46
4.1.1	Componente Ingeniería de Diseño.....	50
4.1.2	DISEÑOS E IMPRESIÓN EN SOLID WORKS (3D).....	50
4.1.2.1	Diseños HSEQ.....	50
4.1.2.2	Iluminación Locación.....	51
4.1.2.3	Unidad de Coiled Tubing.....	51
4.1.2.4	Arme sobre Cabezal de Pozo.....	52
4.1.2.5	Unidad de Bombeo.....	52
4.1.2.6	Locación.....	53
4.2	DISEÑO DEL CONTROL DE LA BOMBA SUMERGIBLE Y LA ILUMINACIÓN SIMULADOR.....	55
4.2.1	APLICACIÓN MOVIL.....	57
4.2.1.1	Componentes de una Aplicación Android en App Inventor 2.....	57
4.2.1.2	Programación Diagramas de Bloques App Móvil 'SIMULADOR_GOS'.....	58
4.2.1.3	Vistas Pantallas de App Móvil.....	63
4.2.1.4	Configuración del Archivo Android.....	65
4.3	DISEÑO DE LA INTERFAZ GRÁFICA.....	66
4.3.1	Diseño de la Ventana (Formulario).....	66
4.3.2	DESCRIPCIÓN DEL CÓDIGO FUENTE.....	67
4.3.2.1	Instrucciones (Uso de objetos IDisposable).....	67
4.3.2.2	Componentes de inicialización.....	68
4.3.2.3	Configuración de los botones 'Iniciar' y 'Detener'.....	69
4.3.2.4	SetSize.....	70
4.3.2.5	Temporizador (Timer).....	71
4.3.2.6	Función 'Leer Datos' 70.....	71
4.3.2.7	Créditos.....	72
4.4	CÓDIGO ARDUINO.....	73
4.4.1	Declara Variables e Inicializa.....	74
4.4.2	Función de Configuración o Setup.....	74
4.4.3	Función de Ciclo o Loop.....	75

4.4.4	Ejecución del Control.....	76
4.5	DISEÑO DEL CIRCUITO .....	78
4.5.1	Simulación .....	78
4.5.2	Diseño del Circuito Impreso PCB .....	79
	CONCLUSIONES.....	80
	RECOMENDACIONES .....	81
	BIBLIOGRAFIA .....	82
	ANEXOS .....	84



## LISTA DE TABLAS

**Tabla 1.** Características y especificaciones actuales del sistema operativo Android

**Tabla 2.** Especificaciones Arduino Leonardo.

**Tabla 3.** Diseños en Solid Works piezas HSEQ en arme de locación (Caneca de Residuos, cono, extintor Multipropósito)

**Tabla 4.** Diseños en Solid Works piezas Iluminación Locación en arme de locación (Luz Posta, Planta Estadio, Reflector)

**Tabla 5.** Diseños en Solid Works piezas Unidad de Coiled Tubing (Rollo, caja de herramientas, manifold)

**Tabla 6.** Diseños en Solid Works piezas sobre arme de Cabezal de Pozo (BOP, Inyector de Unidad de Coiled Tubing).

**Tabla 7.** Diseños en Solid Works piezas Unidad de Bombeo (motor, control de mandos, bomba, reja, tanque de almacenamiento de fluidos, escalera).

**Tabla 8.** Diseños en Solid Works piezas de Locación (caseta, stop parking de entrada a locación).

## LISTA DE FIGURAS

- Figura 1.** Unidad de Coiled Tubing realizando operación en Pozo petrolero y SIMULADOR\_GOS.
- Figura 2.** Se baja la herramienta de limpieza con Coiled Tubing hasta el tope del relleno
- Figura 3.** Penetración de la herramienta en el relleno durante la circulación, lava y moviliza los sólidos y toma un bocado
- Figura 4.** Se alcanza una longitud de bocado previamente planificada y se sube la herramienta de limpieza por chorro hacia el extremo superior de la tubería de revestimiento corta (Liner), para dar comienzo al proceso de barrido
- Figura 5.** El relleno está siendo barrido a través de una porción de la sección angular crítica (40 a 65 grados) del pozo.
- Figura 6.** Columna Estratigráfica – Cuenca Valle Superior del Magdalena.
- Figura 7.** Diagrama de Bloques Control ON/OFF
- Figura 8.** Tarjeta Arduino Leonardo
- Figura 9.** Esquemático ATMEGA32U4 U1.
- Figura 10.** Esquemático ATMEGA32U4 U2.
- Figura 11.** USB – Arduino Leonardo
- Figura 12.** LMV358DGKR – Arduino Leonardo
- Figura 13.** +5V Auto Selector – Arduino Leonardo
- Figura 14.** Bomba Sumergible SP-9000 utilizada en el Prototipo SIMULADOR\_GOS
- Figura 15.** Entorno Sharp Develop
- Figura 16.** Sensor de Presión MPX 5010.
- Figura 17.** Sensor de Flujo (Caudal) SEN\_0394
- Figura 18.** Diseño del Simulador de la Mecánica de Fluidos para operaciones de limpieza y Separación de Fases en pozos verticales y horizontales con tecnología Coiled Tubing.
- Figura 19.** Prototipo a escala de tuberías de pozo horizontal y vertical de petróleo implementado en el Simulador.
- Figura 20.** Diagrama en bloques Ingeniería de Diseño.
- Figura 21.** Resultados Diseño e implementación de piezas 3D en Prototipo Simulador
- Figura 22.** Diagrama de Bloques Control ON/OFF

**Figura 23.** Representación gráfica del Funcionamiento de la Aplicación Móvil

**Figura 24.** Diagrama de Bloques Screen1 de la app móvil *SIMULADOR\_GOS*

**Figura 25.** Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*

**Figura 26.** Declaraciones variables A, B, C y D Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*

**Figura 27.** Declaraciones variables G y H Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*

**Figura 28.** Diagrama de Bloques “Desconectar Bluetooth” y “Salir” de la pantalla de CONTROL de la app móvil *SIMULADOR\_GOS*.

**Figura 29.** Créditos de la app móvil *SIMULADOR\_GOS*.

**Figura 30.** Vista “Screen1” *SIMULADOR\_GOS*.

**Figura 31.** Vista “CONTROL” *SIMULADOR\_GOS*

**Figura 32.** Vista “CONTROL ON/OFF” Screen *SIMULADOR\_GOS*

**Figura 33.** Vista “Créditos” *SIMULADOR\_GOS*

**Figura 34.** Vista del QR de la App móvil “*SIMULADOR\_GOS*”

**Figura 35.** Respuesta Data Sensores Electrónicos (Interfaz gráfica)

**Figura 36.** MessageBox Créditos

**Figura 37.** Simulación

**Figura 38.** Circuito impreso

## RESUMEN

### TITULO:

SIMULADOR DE LA MECÁNICA DE FLUIDOS PARA OPERACIONES DE LIMPIEZA Y SEPARACIÓN DE FASES EN POZOS VERTICALES Y HORIZONTALES CON TECNOLOGIA COILED TUBING

### AUTORES:

JENNY CATALINA GONZALEZ PEÑA

MAGDA ALEXANDRA TRUJILLO JIMÉNEZ

### PALABRAS CLAVE:

Prototipo, Simulador, Mecánica de Fluidos, Coiled Tubing, Bomba sumergible, Control ON-OFF, Aplicación móvil, Android, Arduino, Sensor, Sharp Develop, Zed Graphic, Solid Works, 3D.

### DESCRIPCIÓN:

Éste proyecto busca la simulación de los procesos que se llevan a cabo en los pozos de petróleo que manejan la técnica Coiled Tubing, pozos tanto verticales como horizontales, de manera que sus operarios, personal en general de la empresa y cualquiera que quiera observar estos procesos, pueda evidenciar de manera clara en qué consisten sus métodos de limpieza y separación de fases en dichos pozos, y a su vez, de qué manera se controlan las sustancias que entran y salen del pozo, permitiendo así que una persona, sin conocimientos previos del tema, logre entender fácilmente cuál es su objetivo de la inyección de fluidos mediante el C.T. en pozos de petróleo.

Cabe mencionar, que este proyecto de ingeniería interdisciplinaria, financiado por Guacamaya Oil Services S.A.S, contribuiría no solo a la capacitación de manera didáctica y novedosa del personal, utilizando herramientas hardware y software al igual que la elaboración de cada una de las piezas ubicadas en superficie con tecnología 3D para hacerlo atractivo y de fácil acceso, sino que también, al tener la característica de ser a escala, consigue así ser lo más semejante a pozos de petróleo en los cuales dicha compañía presta sus servicios y por sus características de simulador y la medición en tiempo real de variables físicas directamente sobre el fluido que se está inyectando, ser material para futuras experimentaciones con nuevas técnicas y sustancias, prestándose como prototipo para numerosas aplicaciones.

## **ABSTRACT**

### **TITLE:**

MECHANIC SIMULATOR FLUIDS FOR CLEAN-UP OPERATIONS AND PHASES SEPARATION IN VERTICAL AND HORIZONTAL WELLS WITH COILED TUBING TECHNOLOGY

### **AUTHORS:**

JENNY CATALINA GONZALEZ PEÑA

MAGDA ALEXANDRA TRUJILLO J.

### **KEYWORDS:**

Prototype, Simulator, Fluid Mechanics, Coiled Tubing, submersible pump, ON-OFF Control, Mobile application, Android, Arduino, Sensor, Sharp Develop, Zed Graphic, Solid Works, 3D.

### **DESCRIPTION:**

This Project look for the processes simulation that take place in the oil Wells that operate the Coiled Tubing technique both vertical and horizontal Wells, so its workers, Company personal and anyone that wants to look these processes, can prove that consist in their cleaning methods and phase separation in these Wells, and at the same time, in what way the substances that in and out of well are controlled, and allow that a person, without previous knowledge about the topic, may understand easily what is it injection fluid purpose by means of the Coiled Tubing in oil Wells.

Note that this interdisciplinary engineering Project, funded by Guacamaya Oil Services S.A.S is going to contribute not just the training and novel way of didactic staff, using hardware and software tools, like the making of each pieces located on the Surface with 3D Technology to make it attractive and easy access, but also to have the characteristics of being scaled get well be more like oil Wells in which such Company provides services and their simulator characteristics and measurement real time of physical changeable directly above the fluid that is injected, be material for future experimentation with new techniques and substances, paying as a Prototype for many applications.

## INTRODUCCIÓN

Dentro de los procesos clave para la explotación eficiente de los yacimientos de petróleo está la óptima selección y el desempeño de los fluidos de control utilizados en las operaciones de perforación, mantenimiento, reacondicionamiento y estimulación de pozos en las que desde hace algunos años la tecnología de Coiled Tubing se ha venido posicionando como una alternativa eficiente y versátil para los retos que establece la industria del petróleo en la actualidad; dados los innumerables procedimientos que se pueden realizar con esta tecnología en campo dentro de los cuales se destacan la limpieza de pozos, las operaciones de pesca, la estimulación matricial acida, el fracturamiento hidráulico, entre otras, las cuales requieren de la circulación de fluidos en el pozo sin necesidad de matar o controlar el pozo.

En vista a lo anterior surge una necesidad apremiante de desarrollar un Simulador de la mecánica de fluidos para las diversas operaciones de limpieza y Separación de fases que se pueden llevar a cabo en pozos de petróleo verticales y horizontales mediante la tecnología de Coiled Tubing, que permita de forma didáctica, real y novedosa la capacitación del personal de la Empresa Guacamaya Oil Services S.A.S y a su vez que se pueda utilizar como material de apoyo para futuras experimentaciones con nuevas técnicas y sustancias, prestándose como prototipo para numerosas aplicaciones.

Éste simulador cuenta con un sistema de control a través de una bomba electro sumergible de inyección de fluido y a su vez medidas de presión y caudal para los fluidos que se desplazan en los pozos de petróleo en las operaciones descritas.

Igualmente la realización de una interfaz gráfica informática que permite evidenciar el comportamiento de las variables de operación presión de inyección de fluido, presión de anular Tubing – Coiled Tubing y caudal de inyección de fluido para su posterior análisis, permitiéndonos evidenciar su función correspondiente en la operación establecida; el diseño de una aplicación móvil que por medio de Bluetooth establece comunicación entre el Simulador y el usuario, lo cual permite realizar control On/Off sobre el mismo, siendo de esta manera más atractivo y funcional a lo que la ingeniería respecta para el usuario.

Ahora bien, éste Prototipo cuenta en superficie con los diferentes cabezales requeridos, implementos HSEQ, herramientas y equipos donde van las unidades en quienes se lleva la tubería flexible en este caso para realizar las operaciones y al mismo tiempo la iluminación correspondiente para la estética y más real representación posible siendo así un diseño novedoso y práctico de ingeniería debido al exitoso manejo de software y hardware para creación y ensamble de piezas en 3D, en las cuales son elaboradas éstas piezas por los autores del presente proyecto.

## **1) INFORMACIÓN DEL PROYECTO**

A continuación se presentan los aspectos más relevantes que describen y justifican la elaboración de este proyecto.

### **1.1) Formulación del Problema**

Diseñar, construir e implementar un modelo a escala que permita visualizar los procesos que se llevan a cabo en la inyección de fluidos en pozos verticales y horizontales con tecnología Coiled Tubing utilizados por la empresa Guacamaya Oil Services S.A.S en operaciones de limpieza de arenas donde se evidencia la separación de fases.

### **1.2) Descripción del Proyecto**

El presente proyecto consistió en diseñar y realizar un prototipo simulador de la mecánica de fluidos para operaciones de Limpieza de arenas y Separación de fases en pozos petroleros a través de tecnología Coiled Tubing con el fin de capacitar al personal de Guacamaya Oil Services S.A.S y brindar la posibilidad de uso en investigaciones futuras, utilizando para éste diseño de ingeniería, el cual se refleja en el uso de software para modelamiento de piezas (Solid Works) y hardware para su posterior impresión; además una aplicación móvil con la cual se realiza el control ON-OFF del mismo y a su vez una interfaz gráfica informática que posibilita el monitoreo y representación de las respectivas mediciones de presión de inyección de fluido, presión de anular Tubing – Coiled Tubing y caudal de inyección de fluido.

### **1.3) Justificación**

Guacamaya Oil Services, una empresa huilense que suministra servicios para la industria petrolera en la línea de estimulación de pozos mediante la técnica de Coiled Tubing, Flushing, bombeos de químicos e inyección de fluidos y nitrógeno utilizando los recursos más eficientes, pese a la necesidad de capacitar su personal e invertir en investigación e innovación, busca dentro del programa de ingeniería electrónica de la Universidad Surcolombiana, estudiantes interesados en el fortalecimiento y progreso de la industria, la oportunidad de generar proyectos de ingeniería que contribuyan al crecimiento del departamento del Huila



y de nuestro país en áreas afines a sus tareas, y en particular en las novedosas técnicas Coiled Tubing, pues en Colombia ninguna institución educativa o empresa cuenta con un prototipo simulador del control de la mecánica de fluidos para operaciones en pozos petroleros.

Es por esta razón que éste proyecto busca la simulación de los procesos que se llevan a cabo en los pozos de petróleos que manejan la técnica Coiled Tubing, pozos tanto verticales como horizontales, de manera que sus operarios, personal en general de la empresa y cualquiera que desee observar estos procesos, pueda evidenciar de manera clara en qué consisten sus métodos de limpieza y separación de fases en dichos pozos, y a su vez, de qué manera se controlan las sustancias que entran y salen del pozo, permitiendo así que una persona sin conocimientos previos del tema, logre entender fácilmente cuál es su objetivo de la inyección de fluidos mediante el C.T. en pozos de petróleo.

## **1.4) Objetivos**

### **1.4.1) Objetivo General**

Diseñar e implementar un Simulador del modelo a escala de pozos petroleros verticales y horizontales, que permita visualizar la mecánica de los fluidos utilizados para las operaciones de limpieza y separación de Fases mediante tecnología Coiled Tubing.

### **1.4.2) Objetivos Específicos**

- Diseñar e implementar un sistema que controle la bomba de inyección de los fluidos utilizados en operaciones de limpieza y separación de fases, utilizando software para el desarrollo de una interfaz de dispositivos móviles, que permita por comunicación Bluetooth que el usuario encienda el simulador y a su vez active o desactive la misma desde su teléfono celular.
- Realizar el diseño e implementación de una interfaz gráfica informática que permita evidenciar el comportamiento de las variables de operación; presión de inyección de fluido, presión de anular Tubing – Coiled Tubing y caudal de inyección de fluido para su posterior análisis.

- Analizar y evaluar el comportamiento de la presión de inyección, presión de anular Tubing – Coiled Tubing y caudal de inyección en el bombeo de fluidos en pozos petroleros a través de Tecnología Coiled Tubing y observar su correspondiente función en la operación establecida.
- Facilitar y brindar a la empresa Guacamaya Oil Services S.A.S la posibilidad de capacitar al personal de trabajo de manera didáctica, novedosa y lo más cercano a la realidad posible para día a día ofrecer un mejor servicio, con la posibilidad abierta a futuras investigaciones de la misma en ésta área; lo anterior haciendo énfasis en HSEQ. (Salud, Ambiente, Seguridad, Calidad).

## 2) MARCO TEÓRICO

### 2.1 Coiled Tubing

Esta nueva tecnología que cuenta con tubería flexible de diámetro pequeño que se permite introducir en pozos petroleros en operaciones Workover, permite la inyección de fluidos simulado en este proyecto, cuenta con características similares a las tuberías convencionales pero con la gran ventaja de que no es necesario conectarla tramo por tramo para bajarla o retirarla del pozo, ya que esta se enrolla como una manguera en un carrete adecuado, que es accionado mecánicamente.

**Figura 1.** Unidad de Coiled Tubing realizando operación en Pozo petrolero y SIMULADOR\_GOS.



**Fuente:** Guacamaya Oil Services S.A.S - Autores.

#### 2.1.1 Ventajas del Coiled Tubing Operativas:

- Permite realizar operaciones sin necesidad de matar el pozo, con permanente control de surgencia.
- Es bastante versátil para una amplia gama de trabajos.
- Posibilita la realización de soluciones globales (Servicios integrados).
- Ofrece rapidez operativa y de movilización.

- Bajo costo de Locación y bajo impacto sobre el terreno en el que realiza trabajos.
- Disminución del impacto audio-visual.
- Posibilidad de comando a distancia, lo cual contribuye a la Seguridad personal.

### 2.1.2 Limpieza de Pozos

Durante la producción de pozos petroleros y las diferentes operaciones de fractura y perforación existe la problemática de acumulación de arena o sólidos provenientes de dichos procesos en éstos.

En la actualidad gracias al versátil y rápido avance de la tecnología los ingenieros de campo pueden acceder a métodos más eficaces y seguros para la eliminación de dichos escombros y arenas del pozo; entre dichos procesos la técnica más eficiente y común hoy en día para la limpieza utiliza una herramienta por chorro que se baja en el pozo con Tubería Flexible (CT), ya que ésta a diferencia de otras técnicas posee innumerables ventajas tales como el bombeo continuo durante este tipo de operaciones puesto que en tuberías con trabajos standard éste debe detenerse en varias ocasiones para realizar las correspondientes conexiones de tubería. Una vez se inicia el bombeo del fluido de limpieza escogido de acuerdo a las diferentes características del pozo (estado mecánico, tamaño de las partículas) a través de la tubería de producción, ésta herramienta se baja y allí empieza el trabajo de lavado de la arena u otros escombros.

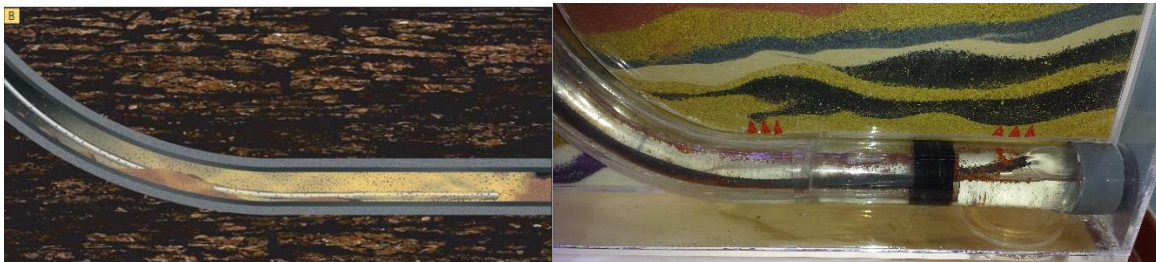
### 2.2 Ejecución Operaciones (Limpieza de Arenas - Separación de Fases) en pozos petroleros utilizando Coiled Tubing

**Figura 2.** Se baja la herramienta de limpieza con Coiled Tubing hasta el tope del relleno



**Fuente:** Oilfield Review. Sistemas Integrados de Limpieza de Pozos: Mejoramiento de la eficiencia y Reducción del riesgo. file:///C:/Users/ACER/Desktop/01\_integrated\_wellbore\_cleanout.pdf - AUTORES

**Figura 3.** Penetración de la herramienta en el relleno durante la circulación, lava y moviliza los sólidos y toma un bocado



**Fuente:** Oilfield Review. Sistemas Integrados de Limpieza de Pozos: Mejoramiento de la eficiencia y Reducción del riesgo. file:///C:/Users/ACER/Desktop/01\_integrated\_wellbore\_cleanout.pdf - AUTORES

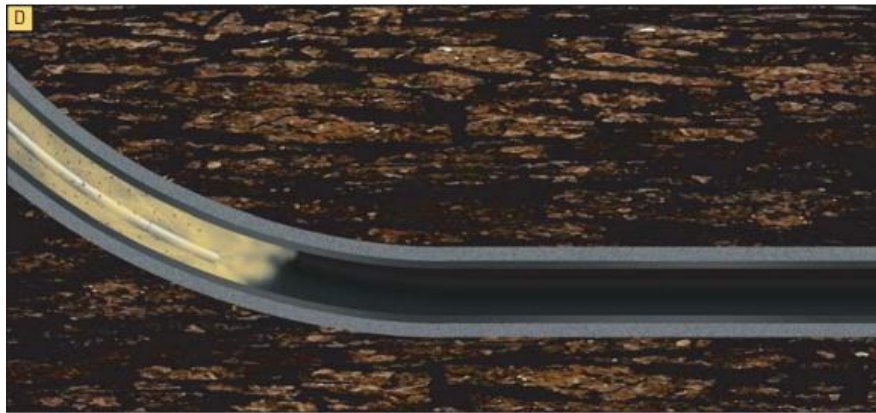
**Figura 4.** Se alcanza una longitud de bocado previamente planificada y se sube la herramienta de limpieza por chorro hacia el extremo superior de la tubería de revestimiento corta (liner), para dar comienzo al proceso de barrido





**Fuente:** Oilfield Review. Sistemas Integrados de Limpieza de Pozos: Mejoramiento de la eficiencia y Reducción del riesgo. file:///C:/Users/ACER/Desktop/01\_integrated\_wellbore\_cleanout.pdf

**Figura 5.** El relleno está siendo barrido a través de una porción de la sección angular crítica (40 a 65 grados) del pozo.



**Fuente:** Oilfield Review. Sistemas Integrados de Limpieza de Pozos: Mejoramiento de la eficiencia y Reducción del riesgo. file:///C:/Users/ACER/Desktop/01\_integrated\_wellbore\_cleanout.pdf

En general, una vez que los sólidos son barridos hacia el extremo superior de la tubería de revestimiento corta, la boquilla vuelve a fondo, se toma el siguiente bocado y el proceso se repite hasta remover todos los sólidos del pozo.

Las operaciones de limpieza de pozos constituyeron unas de las primeras aplicaciones para los servicios de Coiled Tubing. Las estimaciones globales indican que casi un 50% de las operaciones con Coiled Tubing se llevan a cabo para remover y sólidos, tales como arena producida o restos de apuntalante provenientes de los tratamientos de fracturamiento hidráulico.<sup>1</sup> En general, los desarrollos continuos registrados en los sistemas de operaciones con Coiled Tubing han permitido que lo operadores avancen al mismo tiempo que se

---

<sup>1</sup> Rolovic R, Weng X, Hill S, Robinson G, Zmlak k y Najafov J: "An Integrated System Approach to Wellbore Cleanouts with Coiled Tubing", artículo de la SPE 89333, presentado en la conferencia y Exhibición sobre Tubería Flexible de las SPE/ICoTA, The Woodlands, Texas, EUA, 23 al 24 de marzo de 2004.

incrementan las profundidades y tortuosidades de los pozos y aumentan las complejidades de las condiciones ambientales de fondo de pozo.<sup>2</sup>

### 2.2.1 Pasos de Proceso de Limpieza de Arenas

A continuación se describe procedimiento a seguir para un servicio de limpieza de arenas según experiencia en campo de los autores del presente proyecto y capacitaciones y normatividad a seguir por parte de la empresa Guacamaya Oil Services S.A.S:

- ✓ Se realiza la ubicación de equipo Coiled Tubing, Grúa Hidráulica, Unidad de bombeo, unidad de Nitrógeno choke manifold, tanque de retornos y Frack Tank (en el caso que la operación lo requiera), dejando suficiente espacio entre y alrededor de los equipos, que permita el paso del personal, tomando en cuenta la dirección del viento, y 60 ft como mínimo del cabezal al punto caliente más cercano.
- ✓ Arme de equipos de Coiled Tubing. Configuración del equipo de superficie adicionando línea de control de pozo hacia la BOP, conectando la Flow tee con el Kill port de la BOP.
- ✓ Realización de prueba de funcionalidad de BOP (Blow Out Preventer) cerrando y abriendo todos los rams, Revisión del correcto Funcionamiento de elastómeros de los Stripper.
- ✓ Instalación del conector de Coiled Tubing, se realiza Pull test con 15.000 lbf registrando con el sistema de adquisición de datos del Coiled Tubing DART.
- ✓ Arme BHA (Botton Hole Assembly) compuesto por conector de CT (1.5"OD), doble flapper, desconector hidráulico y boquilla de limpieza.
- ✓ Identificación del estado mecánico actual del pozo y las profundidades del intervalo a ser limpiado.
- ✓ Se conecta a la válvula máster en la cabeza del pozo y T de flujo.

---

<sup>2</sup> Para más información sobre Coiled Tubing, consulte: Afghoul AC, Amaravadi S, Boumali A, Calmeto JCN, Lima J, Lovell J, Tinkham S, Zemlak K y Staal T: "Tubería flexible: La próxima generación," Oilfield Review 16, no. 1 (Primavera de 2004): 38-57.

- ✓ Izaje y arme conjunto inyector, stripper, BOP, líneas de bombeo y válvulas en la cabeza de pozo habiendo calculado previamente el plan de izaje.
- ✓ Se verifica el correcto arme de línea de retorno rígida hasta línea al choke manifold y alineación de válvulas.
- ✓ Inicio del sistema de adquisición de datos y se calibra lecturas de presión, caudal y volumen a cero.
- ✓ Realización prueba hidrostática con baja y alta presión. Para la baja presión se usa 500 psi, y se mantendrá por 5 min. Para la alta presión se usa 4500 psi, y se mantendrá por 10 min. Se procede a verificar líneas de bombeo y equipos por posibles fugas. En caso de que haya pérdida de presión, relajar, reparar fugas y probar nuevamente.
- ✓ Se verifica correcta alineación de válvulas y disposición de tanques para fluidos a la bomba y Choke.
- ✓ Se calcula profundidad cero (0) de Coiled Tubing teniendo en cuenta la altura del cero (0) de Coiled Tubing respecto a la altura de la mesa rotaria de perforación y así ingresar la diferencia en el sistema de adquisición de datos electrónico de Coiled Tubing.
- ✓ Se pone en (0) sensores electrónicos de presión, peso y volumen en el sistema del equipo de Coiled Tubing, al igual que los contadores de profundidad.
- ✓ Se verifica alineación correcta de válvulas, BOP del pozo, presiones y se informa a todo el personal involucrado el inicio de descenso de Coiled Tubing a pozo con la aprobación del representante del cliente en pozo.
- ✓ Inicia descenso del Coiled Tubing a una velocidad entre 10 - 20 ft/min incrementándose hasta un máximo de 60 ft/min. A continuación se Baja bombeando a tasa mínima 0,5 bpm en caso de ser necesario para mantener lleno el Coiled Tubing. En caso de observar una pérdida de peso, evaluar las condiciones del pozo para definir el procedimiento a seguir.
- ✓ Cada 1.500 ft se realiza prueba de tensión o según el criterio del operador de Coiled Tubing (CT), teniendo en cuenta gráfica de fatiga de la tubería utilizada.



- ✓ Se baja el Coiled Tubing chequeando en todo momento el peso, monitoreando los retornos y la presión de cabeza.
- ✓ Se desciende hasta verificar tope de arena cuando se encuentre tope, se marca la tubería en superficie y se realiza correlación de la profundidad del cuentapiés mecánico y de la adquisición de data con el dato que suministre el cliente.
- ✓ Inicio de bombeo de gel lineal con N<sub>2</sub> (Unidad de Nitrógeno) a un caudal líquido de 0.7 bpm. Una vez estabilizado este caudal (aproximadamente 10 barriles), se empieza bombeo de N<sub>2</sub> a 300 scfm, esperando así a que haya retorno en superficie para proceder a verificar que la circulación sea continua y posteriormente proceder a bajar CT para limpieza de arena.  
**NOTA: Si la limpieza se realiza sin N<sub>2</sub> se inicia bombeo únicamente del fluido de limpieza en las mismas condiciones expuestas**
- ✓ Se procede a levantar 50 ft por encima de la profundidad a la que se perdió peso y se continúa con bombeo de gel lineal con N<sub>2</sub> a un caudal líquido de 0.7 bpm y caudal de 350 scfm de N<sub>2</sub>. **NOTA: Si no se establece circulación en el pozo con las condiciones de caudal anteriormente descritas, se debe incrementar el caudal de nitrógeno hasta obtener condiciones de circulación constantes y si no se está empleado nitrógeno, incrementar caudal de líquido.**
- ✓ A la profundidad máxima alcanzada, se bombea circulando fondo arriba 4 veces la capacidad del pozo de gel lineal o fluido de limpieza a un caudal líquido de 0.7 bpm y caudal de N<sub>2</sub> de 350 scfm, verificando retorno de fluido 100 % limpio en superficie.
- ✓ Con Coiled Tubing en superficie se asegura presiones en 0 psi y se procede a realizar Rig Down de equipos
- ✓ Finalmente se realiza entrega de pozo al representante en pozo del cliente en condiciones adecuadas de orden y aseo.

### 3) GENERALIDADES

#### 3.1 REVISIÓN BIBLIOGRÁFICA

El Prototipo Simulador se implementa con el fin de poder brindar capacitación al personal de la empresa huilense Guacamaya Oil Services S.A.S acerca de dos de las diferentes operaciones a los cuales ellos se ven enfrentados a diario en el campo de trabajo durante sus procedimientos operativos de Coiled Tubing, lo cual les permitirá a éstos tener un visión más clara de las diferentes estructuras y problemáticas aparte de sus conocimientos adquiridos previamente a través de sus estudios o experiencias laborales, permitiéndoles así en el momento que se requiera tener una visión más amplia y brindar posibles soluciones de forma rápida y certera al cliente; lo anterior debido al diseño empleado y funcionalidad del mismo, puesto que cuenta con una infraestructura en superficie de última tecnología a lo que la Ingeniería de Diseño respecta con el uso de software y hardware para elaboración y construcción de piezas en 3D (Solid Works); la más real representación estratigráfica de algunos de los diferentes pozos a los que esta compañía presta sus servicios, la cual se encuentra en el Prototipo Simulador dentro de una Cajetilla de Formaciones distribuida en diferentes capas de arenas de colores recubriendo éstos y teniendo en cuenta la 'Cuenca Valle Superior del Magdalena'.

Una app móvil la cual controla de forma ON-OFF el funcionamiento de éste haciéndolo más atractivo y práctico a quien lo manipule; igualmente con una interfaz gráfica (Data) la cual muestra y monitorea en tiempo real las mediciones de Presión de Inyección, Caudal de inyección, y Presión del anular Tubing- Coiled Tubing durante la simulación de una limpieza de arenas donde se evidencia Separación de Fases en pozos petroleros a escala lo cual lo hace aún más semejante a la realidad.

##### 3.1.1 Solid Works

Es un programa desarrollado para el sistema operativo Microsoft Windows con el objetivo de permitir diseñar piezas en planos 2D y 3D con características propias de material de producción entre otras que permite la simulación de las mismas. Este software además permite guardar los

archivos de las piezas en extensión .STL que es la que utilizan las impresoras 3D.

El software permite modelar no solo piezas, también conjuntos que permiten modelar ensambles y extraer de ellos tanto planos técnicos como otra información necesaria para la producción de estas, funcionando con base en las nuevas técnicas de modelado con sistemas CAD de manera bastante automatizada con herramientas de diseño de dibujo, pieza, ensamble, análisis y simulación.

### 3.1.2 Estratigrafía

El simulador tomo como referencia estratigráfica la cuenca del Valle Superior del Magdalena, <sup>3</sup> que es una cuenca ubicada en la parte alta del Rio Magdalena, que limita al Norte con el Cinturón Plegado de Girardot, al Sureste parcialmente con el sistema de fallas de Algeciras – Garzón, al Noreste con el sistema de fallas de Bituima – La Salina y al Oeste con la Cordillera Central. Un alto de basamento llamado alto de Natagaima – El Pata divide esta cuenca en las sub-cuencas de Girardot y Neiva.

Dentro de las características principales de esta estratigrafía encontramos que cuenta con un área de  $26.200 \text{ km}^2$ , su ubicación se extiende desde la población Pitalito en el Sur, hasta Honda en el Norte.

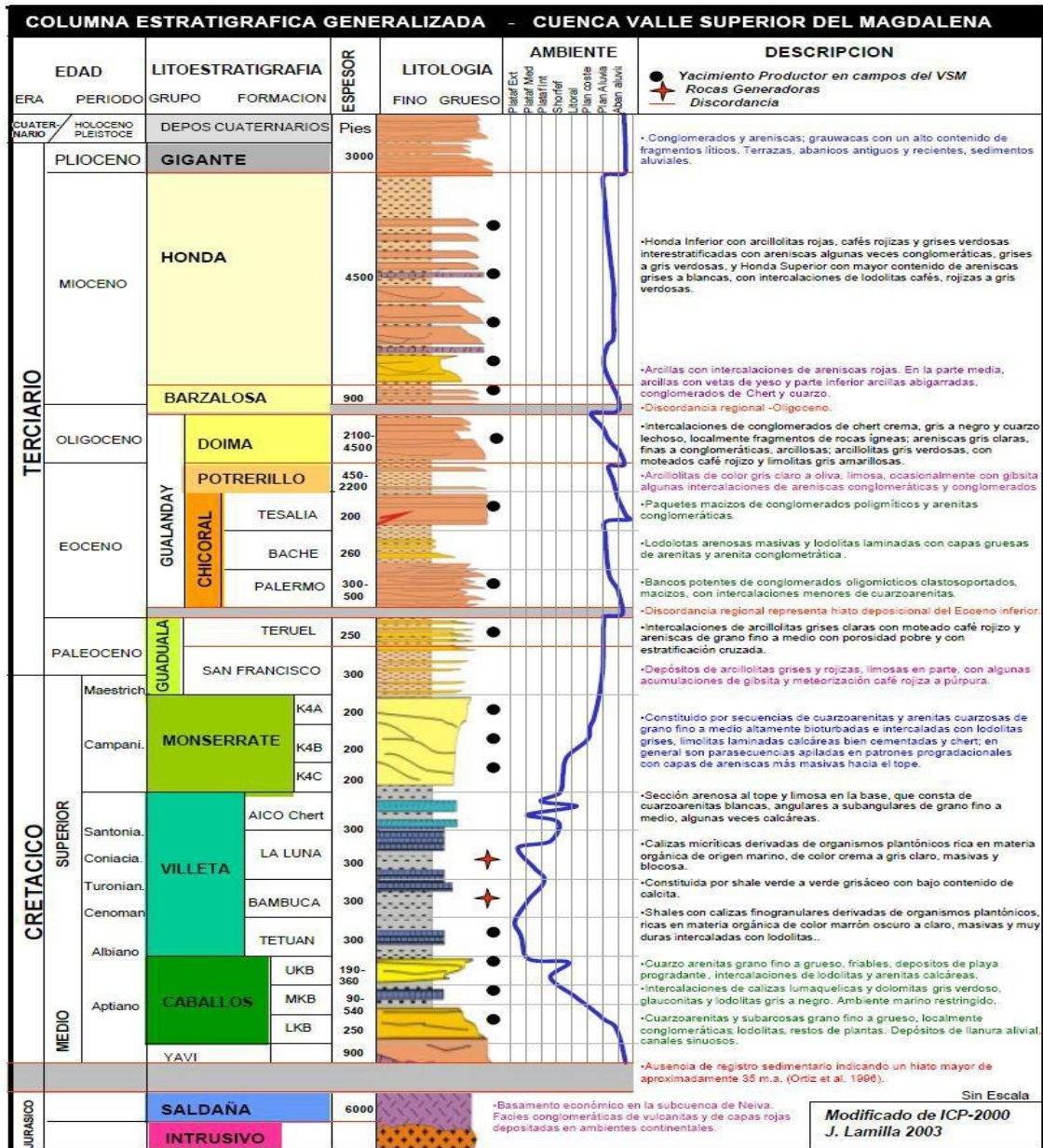
Ésta cuenca produce diariamente en promedio cerca de 88.000 bpd, de los cuales 26.000 son crudos pesados, 53.000 son crudos medianos y 7.800 son crudos livianos; en la actualidad produce 28 millones de Barriles de Petróleo por año a partir de 28 campos. Es alta producción debido a la Deposición de Lutitas y Calizas con alto contenido de materia Orgánica en las formaciones Tetuán, Bambuca y La Luna.

EL principal transportador de hidrocarburos son las areniscas de la Formación Caballos y en menor escala las areniscas de las Formación Monserrate.

---

<sup>3</sup> Cuenas Sedimentarias de Colombia. <http://es.slideshare.net/macorca123/cuenas-sedimentarias-de-colombiadoc-1>

**Figura 6. Columna Estratigráfica – Cuenca Valle Superior del Magdalena.**



**Fuente.** Cuencas Sedimentarias de Colombia.

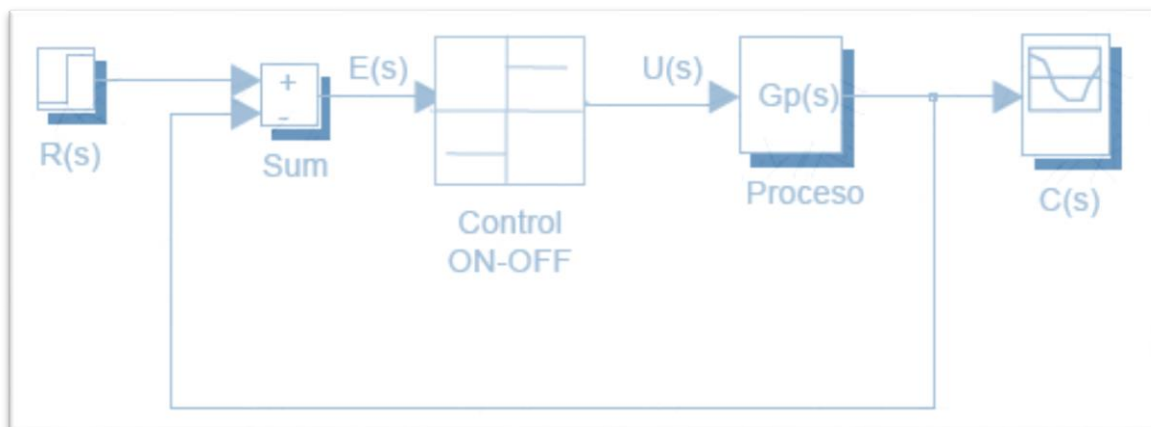
<http://es.slideshare.net/macorca123/cuencas-sedimentarias-de-colombiadoc-1>

### 3.1.3 Control ON/OFF

Un sistema de control consiste en mantener en valores deseados y constantes ciertas variables, en el caso del control de dos posiciones On/Off, es un sistema

no lineal que consiste en activar un mando, proporciona a su salida dos valores fijos que corresponden a conectado/desconectado, funcionando como un interruptor. Inicialmente el error es positivo y el control se activa (ON) hasta llegar al valor deseado, el error se hace negativo y el controlador pasa a OFF. El proceso se repite continuamente.

**Figura 7.** Diagrama de Bloques Control ON/OFF



**Fuente.** Autores

### 3.1.3.1 Sistema Operativo Android

Conocido por ser el sistema operativo con el que cuentan la mayoría de los teléfonos móviles y tabletas, destacado por su interfaz gráfica didáctica y de fácil manejo; es un sistema operativo basado de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, una de sus características más importantes es que Android es una plataforma de software libre diferente a otros sistemas operativos para dispositivos móviles como iOS de Apple o Windows Phone de Nokia, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El hecho de que se pueda tener acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado dispositivo; lo anterior debido a que el código para soportar el hardware (controladores) de cada fabricante normalmente no es de acceso al público, así que faltaría una parte esencial del firmware para poder hacerlo funcionar en dicho terminal, y puesto

que las nuevas versiones de Android suelen requerir más recursos, lo cual hace que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

- **Características**

**Tabla 1.** Características y especificaciones actuales del sistema operativo Android

Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX. GPRS, UMTS, HSPA+ Y HSDPA+
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
Soporte	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4



multimedia	SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.
Google Play	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
Bluetooth	El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.</ref> Los cambios incluyeron:
Video llamada	Android soporta video llamada a través de Google Talk desde su versión HoneyComb.
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose

	en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil android se podría requerir la instalación de software adicional.

**Fuente.** Android. Características y especificaciones actuales.  
<http://es.wikipedia.org/wiki/Android>

### 3.1.3.2 Arduino Leonardo

Arduino es una herramienta de hardware libre, basada en una circuitería que cuenta con un microcontrolador y un entorno de desarrollo, diseñada principalmente para facilitar el uso de la electrónica en diferentes proyectos que pueden incluir múltiples funciones. Cuenta con varias versiones que se diferencia en el microcontrolador que utilizan con respecto al uso que vaya a dársele a la tarjeta. En este proyecto se utilizó la tarjeta Arduino Leonardo, basada en el microcontrolador ATMEGA32U4. Cuenta con 20 pines digitales de entrada / salida (de los cuales 7 puede utilizarse para salidas PWM y 12 entradas como analógicas), un oscilador de cristal de 16 MHz, una conexión micro USB, un conector de alimentación, un botón de reinicio, entre otras características de circuitería. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un computador con cable USB o a una fuente de poder con un adaptador o la batería AC-to-DC para empezar.



**Figura 8.** Tarjeta Arduino Leonardo



**Fuente.** Arduino Leonardo. <http://arduino.cc/en/Main/arduinoBoardLeonardo>

La tarjeta Arduino Leonardo se distingue de todas las placas anteriores de Arduino en que el microcontrolador ATMEGA32U4 se ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario.

**Tabla 2.** Especificaciones Arduino Leonardo.

Microcontroladores	ATMEGA32U4
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital pines I / O	20
Canales PWM	7
Canales de Entrada Analógica	12
Corriente DC por Pin I / O	40 mA
Corriente DC de 3.3V Pin	50 mA
Memoria Flash	32 KB ( ATMEGA32U4 ) de los cuales 4 KB utilizado por gestor de arranque
SRAM	2,5 KB ( ATMEGA32U4 )
EEPROM	1 KB ( ATMEGA32U4 )
Velocidad de reloj	16 MHz

**Fuente.** Arduino Leonardo. Resumen.  
<http://arduino.cc/en/Main/arduinoBoardLeonardo>

## Información General Arduino Leonardo. <sup>4</sup>

### Potencia

El Arduino Leonardo puede ser alimentado a través de la conexión micro USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Potencia (no USB) externo puede venir con un adaptador de CA a CC (pared-verruga) o la batería. El adaptador se puede conectar enchufando un conector de 2,1 mm de centro positivo en el conector de alimentación de la placa. Los cables desde una batería se pueden insertar en los encabezados de pin GND y Vin del conector de alimentación.

La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7 V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y el tablero puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

- VIN. El voltaje de entrada a la placa Arduino cuando se trata de utilizar una fuente de alimentación externa (en contraposición a 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de este pin.
- 5V. La fuente de alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esto puede venir de VIN a través de un regulador de a bordo, o ser suministrada por USB u otra fuente de 5V regulada.
- 3V3. Un suministro de 3,3 voltios generada por el regulador de a bordo. El drenaje actual máximo es de 50 mA.
- GND. patillas de tierra.
- Instrucción IOREF. La tensión a la que los pines de E / S de la placa están operando (es decir VCC de la placa). Esto es de 5V en el Leonardo.

---

<sup>4</sup> Arduino Leonardo. <http://arduino.cc/en/Main/arduinoBoardLeonardo>

## Memoria

El ATMEGA32U4 tiene 32 KB (con 4 KB utilizado por el gestor de arranque). También tiene 2,5 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM ).

## Entrada y salida

Cada uno de los terminales 20 E / S digitales en el Leonardo se puede utilizar como una entrada o salida, utilizando pinMode () , digitalWrite () , y digitalRead () funciones. Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- Serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) TTL datos en serie utilizando el ATmega32U4 capacidad de serie del hardware. Se debe tener en cuenta que en el Leonardo, la serie se refiere a la clase USB (CDC) de comunicación; de serie TTL en los pines 0 y 1, por lo tanto se utiliza el Serial1 clase.
- TWI: 2 (SDA) y 3 (SCL). Apoyar la comunicación TWI usando la librería Wire .
- Interrupciones externas: 3 (interrupción 0), 2 (interrumpir 1), 0 (interrumpir 2), 1 (interrumpir 3) y 7 (interrumpir 4). Estos pines se pueden configurar para activar una interrupción en un valor bajo, un aumento o flanco descendente, o un cambio en el valor. Ver el attachInterrupt () función para más detalles.
- PWM: 3, 5, 6, 9, 10, 11, y 13 para salidas PWM de 8 bits con el analogWrite () función.
- SPI: En la cabecera ICSP estos pines soportan la comunicación SPI utilizando la librería SPI . Es importante tener en cuenta que los pines SPI no están conectados a ninguna de las patillas de E / S digitales, ya que están en el Uno, Sólo están disponibles en el conector ICSP. Esto significa que si se tiene un escudo que utiliza SPI, pero no se tiene un conector ICSP de 6 pines que se conecta a la cabecera ICSP de 6 pines del Leonardo, el escudo no va a funcionar.
- LED: 13 Hay un built-in LED conectado al pin digital 13 Cuando el pin es de alto valor, el LED está encendido, cuando el pasador es bajo, es apagado.
- Entradas analógicas: A0-A5, A6 - A11 (en los pines digitales 4, 6, 8, 9, 10, y 12) El Leonardo tiene 12 entradas analógicas, etiquetados A0 a A11, todos los cuales también se pueden utilizar como i digitales / o. Pasadores A0-A5 aparece en los mismos lugares que en el Uno; Entradas A6-A11 están en E / S digital pines 4, 6, 8, 9, 10 y 12, respectivamente. Cada entrada analógica

proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, la medida de las entradas analógicas de la tierra a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y la analogReference función ().

Hay un par de otras clavijas de la placa:

- AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con analogReference ().
- Restablecer. esta línea LOW es usada para reiniciar el microcontrolador. Normalmente se utiliza para añadir un botón de reinicio a los escudos que bloquean el uno en el tablero.

## Comunicación

El Leonardo tiene una serie de instalaciones para la comunicación con un ordenador, otro Arduino, u otros microcontroladores. El ATmega32U4 ofrece UART TTL (5V) de comunicación en serie, que está disponible en los pines digitales 0 (RX) y 1 (TX). El 32U4 también permite la serie (CDC) de comunicación a través de USB y aparece como un puerto COM virtual para el software en el ordenador. El chip también actúa como un dispositivo de máxima velocidad USB 2.0, el uso de controladores USB COM estándar. En Windows, se requiere un archivo inf. El software de Arduino incluye un monitor de serie que permite a los datos textuales sencillos para ser enviados hacia y desde la placa Arduino. Las RX y TX LED en el tablero parpadean cuando se están transmitiendo datos a través de la conexión USB al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

Una biblioteca SoftwareSerial permite la comunicación en serie en cualquiera de los pines digitales del Leonardo. El ATmega32U4 también soporta I2C (TWI) y la comunicación SPI. El software de Arduino incluye una librería Wire para simplificar el uso del I2C bus.

El Leonardo aparece como un teclado y un ratón genérico, y puede ser programado para controlar estos dispositivos de entrada utilizando el teclado y el ratón clases.

## Programación

El Leonardo se puede programar con el software Arduino. Se selecciona "Arduino Leonardo de Herramientas" Junta de menú (de acuerdo con el microcontrolador en la placa).

Los ATmega32U4 sobre la Arduino Leonardo viene precargado con un *gestor de arranque* que le permite cargar nuevo código a la misma sin el uso de un programador de hardware externo. Se comunica usando el AVR109 protocolo.

También se puede pasar por alto el gestor de arranque y programar el microcontrolador a través del ICSP (In-Circuit Serial Programming) cabeceando utilizando Arduino ISP.

## Automática (Software) Restablecer y Bootloader Iniciación

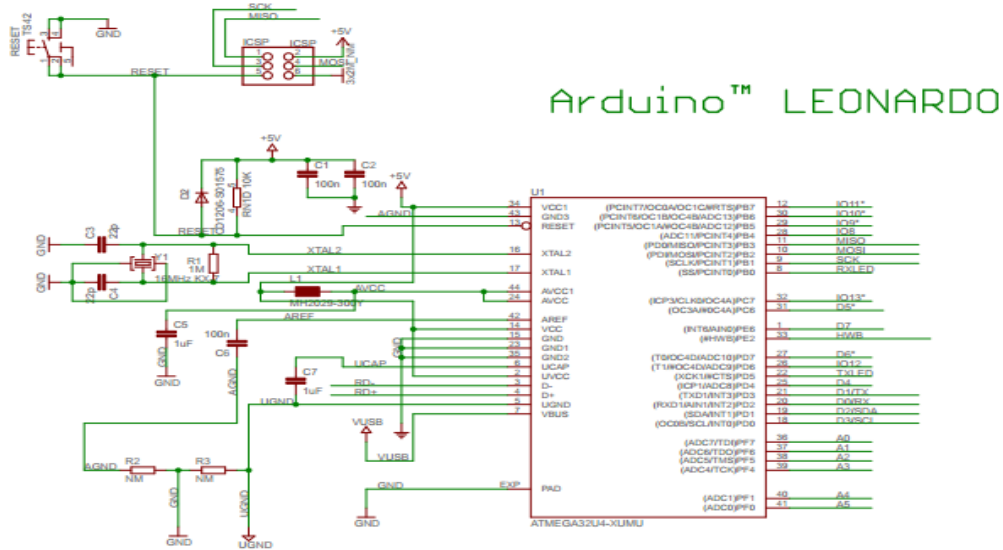
En lugar de requerir una prensa física del botón de reinicio antes de un proceso de carga, el Leonardo está diseñado de una manera que permite que pueda ser restablecido por el software que se ejecuta en un ordenador conectado. El reinicio se activa cuando virtual (CDC) de puerto serie / COM del Leonardo se abre en 1200 baudios y luego es cerrada. Cuando esto ocurre, el procesador se reinicia, rompiendo la conexión USB al ordenador (lo que significa que el puerto serie / COM virtual desaparece). Después de que se restablezca el procesador, el gestor de arranque se inicia, que permanecen activas durante unos 8 segundos. El gestor de arranque también se puede iniciar presionando el botón de reinicio en el Leonardo.

## Protección de sobrecorriente USB

El Leonardo tiene una polifusible reseteable que protege los puertos USB del ordenador de cortocircuitos y sobrecorriente. Aunque la mayoría de los ordenadores proporcionan su propia protección interna, el fusible proporciona una capa adicional de protección. Si hay más de 500 mA se aplica al puerto USB, el fusible se romperá automáticamente la conexión hasta que la corta o se elimina la sobrecarga

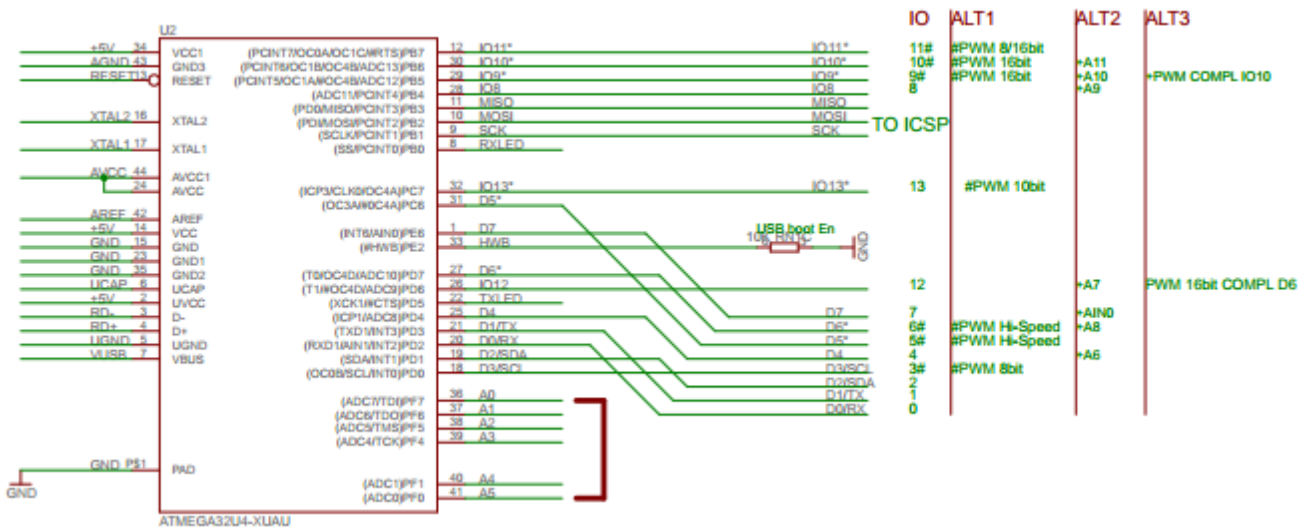
## Esquemático Arduino Leonardo y sus correspondientes PINOUT.

Figura 9. Esquemático ATMEGA32U4 U1.



Fuente. [http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic\\_3b.pdf](http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf)

Figura 10. Esquemático ATMEGA32U4 U2.

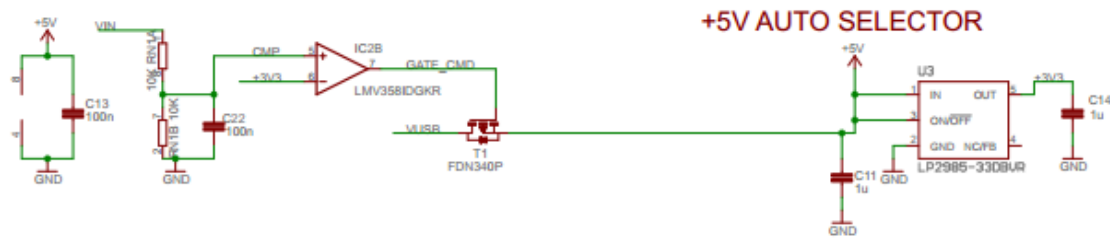


Fuente. [http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic\\_3b.pdf](http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf)





**Figura 13.** +5V Auto Selector – Arduino Leonardo



**Fuente.** [http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic\\_3b.pdf](http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf)

### 3.1.3.3 Actuadores

#### Bomba Sumergible

Una bomba hidráulica básicamente convierte energía mecánica en hidráulica con el fin de hacer una fuerza de empuje y mantener un fluido en movimiento, consiguiendo así aumentar la presión de este. Las bombas sumergibles tiene la característica de mantenerse dentro del fluido a bombear para su funcionamiento, cuentan con un impulsor sellado a la carcasa impidiendo que el líquido en el cual este sumergida pueda pasar a el motor eléctrico. El par se sumerge en el fluido a bombear. La ventaja de este tipo de bomba es que puede proporcionar una fuerza de carga significativa pues no depende de la presión de aire externa para hacer ascender el líquido. Para éste caso se ha utilizado la Bomba Sumergible SP-9000.

**Figura 14.** Bomba Sumergible SP-9000 utilizada en el Prototipo SIMULADOR\_GOS



**Fuente.** [http://www.mispecesdisco.com/components/com\\_virtuemart/shop\\_image/product/Bomba\\_sumergible\\_4daa66da53232.png](http://www.mispecesdisco.com/components/com_virtuemart/shop_image/product/Bomba_sumergible_4daa66da53232.png)



## Especificaciones SP-9000.

MODELO: SP-9000

VOLT. : 100 ~ 120 220 ~ 240V

FREQ. : 50/60

ENERGÍA: 130W

Qmáx: 926GPH-3500 L / hora

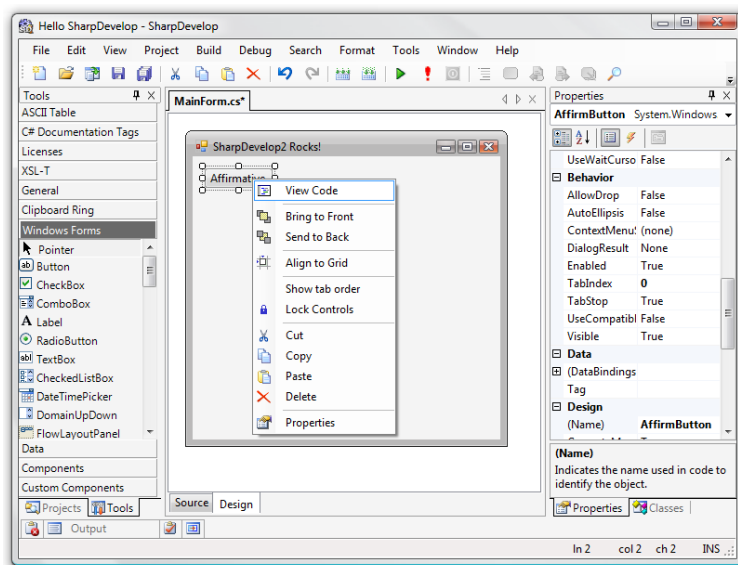
HMAX: 4,5 14.8ft

BOMBA diámetro de salida:  $\varnothing \varnothing 19 \text{ mm } 3 / 4''$

### 3.1.4 Sharp Develop

Es un software compuesto por múltiples herramientas de lenguaje de programación y diseño de programas, categorizado por ser un entorno de desarrollo integrado IDE libre para los lenguajes de programación C#, Visual Basic .NET, F#, Python, Ruby, Boo y C++. AL ser software libre, le permite ser de código abierto y se puede descargar tanto el código fuente como los ejecutables de su página web oficial. Dentro de sus características más destacadas se encuentran que incorpora un diseñador de Windows Forms, el cual es una interfaz de programación de aplicación grafica API que cuenta con un depurador incorporado y al mismo tiempo permite la compilación de código directamente dentro del entorno de desarrollo integrado.

**Figura 15.** Entorno Sharp Develop



**Fuente.** El Medio Ambiente Desarrollo de Código Abierto para NET.  
<http://www.icsharpcode.net/OpenSource/SD/Default.aspx>

### 3.1.4.1 ZedGraph<sup>5</sup>

Es un conjunto de clases, escritas en C #, para crear líneas y gráficos en 2D de conjuntos de datos. Las clases proporcionan un alto grado de flexibilidad, pues casi todos los aspectos de la gráfica pueden ser modificados por el usuario. Al mismo tiempo, el uso de las clases se mantiene simple, proporcionando valores predeterminados para todas las características del gráfico. Las clases incluyen código para la elección de los rangos de escala apropiadas y tamaños de paso en función de la gama de valores de datos que se está trazando.

ZedGraph también incluye una interfaz de control de usuario, lo que permite arrastrar y soltar de edición en el editor de formularios de Visual Studio, además de acceso desde otros lenguajes como C ++ y VB. ZedGraph cuenta con licencia Pública General y se puede adjuntar a Sharp Develop. Esta licencia fue elegida para proporcionar flexibilidad a los usuarios potenciales de ZedGraph, al tiempo que protege ZedGraph si se realiza modificaciones en el propio código ZedGraph, debe hacerse los cambios de código fuente a disposición del público si se quiere la distribución de cualquier aplicación que utilice el código modificado.

### 3.1.5 Medición de Presión

La presión en los fluidos es la fuerza que estos ejercen sobre la superficie que recorre o en la que se encuentra, es por eso que la medición de presión es esencial para optimizar la recuperación de hidrocarburos. Actualmente se pueden determinar las presiones de formación en forma exacta, prácticamente en cualquier momento del ciclo de vida de un pozo petrolero. Ya sea durante la perforación, cuando el pozo alcanza la profundidad final o algunos años después de iniciada la producción, como lo es en nuestro caso para operaciones de estimulación al pozo, la técnicas actuales nos permiten adquirir datos de presión precisos y económicamente efectivos; lo cual ayudan a reducir los riesgos y mejorar la recuperación. La presión de inyección y la que ejerce en el anula Tubing- Coiled Tubing en el Simulador de la mecánica de fluidos en pozos con Coiled Tubing que se desarrolla en este proyecto, se medirán con el sensor de referencia MPX 5010.

---

<sup>5</sup> Página web principal de ZedGraph. <http://zedgraph.dariowiz.com/>

**Sensor de presión con acondicionador de señal y compensación por temperatura integrados MPX 5010.** El funcionamiento está basado en un transductor piezoresistivo, diseñado para un amplio rango de aplicaciones.

**Figura 16.** Sensor de Presión MPX 5010.



**Fuente.**<http://pdf.datasheetcatalog.net/datasheet2/a/0aiqj65ai5zrwcfcadgglgpsu23y.pdf>

### **Especificaciones**

Rango de presión: 0 - 10 kPa (1019.78 mm H<sub>2</sub>O, 1.45psi).

Voltaje de operación: 5 VDC.

Corriente de operación: 5mA.

Sensibilidad: 450mV/mm (4.413 mV/mm H<sub>2</sub>O).

Tiempo de respuesta: 1mS.

Empaque: 867B.

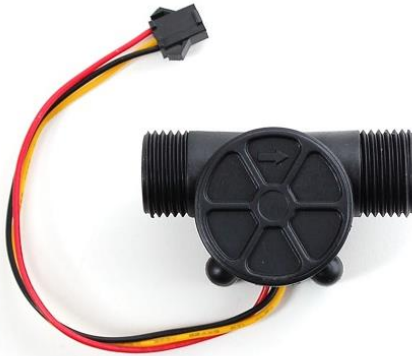
Espacio entre pines: 0.1" (2.54mm).

### **3.1.6 Medición de Caudal a través de Flujo**

En la industria no contamos con variedad de sensores de caudal de dimensiones pequeñas, es por eso que utilizaremos un sensor de flujo, y con dicha medida por medio de ecuaciones matemáticas podemos hallar el caudal. Para la variable de flujo, existen varios métodos según el tipo de caudal al cual se le va hacer la medición, sin embargo, el estudio se centrará en el medidor tipo turbina, el cual por factores económicos y a pesar de su exactitud de  $\pm 1\%$  a comparación con la de otros instrumentos, fue escogido en la implementación.

**Sensor de velocidad tipo turbina.** El funcionamiento de la turbina consiste en un rotor que realiza un movimiento giratorio al detectar el paso del fluido con una velocidad que es directamente proporcional al caudal y es aquella la que ejerce una fuerza de arrastre en el rotor que hace que éste gire por diferencia de presión en su área, sin necesidad de hacer uso de mecanismos de rodamientos que producirían un rozamiento al utilizarlos. (CREUS SOLÉ A. , 1997). La turbina, genera un campo magnético que es leído y codificado por un sensor de efecto Hall.

**Figura 17.** Sensor de Flujo (Caudal) SEN\_0394



**Fuente.** <http://tienda.tdrobotica.co/producto/394>

<sup>6</sup>El sensor trae tres cables, rojo (alimentación de 5-24V), negro (tierra) y amarillo (señal de salida del sensor de efecto hall). Cada pulso equivale aproximadamente a 2.25 mililitros. Éste no es sensor de alta precisión, adicionalmente la frecuencia de los pulsos puede variar dependiendo del flujo, la presión y la orientación del líquido. Va a necesitar una calibración sensible si se necesita una precisión de más del 10%.

#### **Características:**

- Voltaje de operación: 5 a 24V
- Máximo consumo de corriente: 15mA @5V
- Caudal de trabajo: 1 a 30 Litros/Minuto
- Rango de temperatura de trabajo: -25 a 80°C
- Rango de humedad de trabajo: 35%-80% HR
- Presión de agua máxima: 2.0 MPa
- Ciclo útil de salida: 50% +- 10%

---

<sup>6</sup> Medidor de flujo líquido de 1/2". <http://tienda.tdrobotica.co/producto/394>

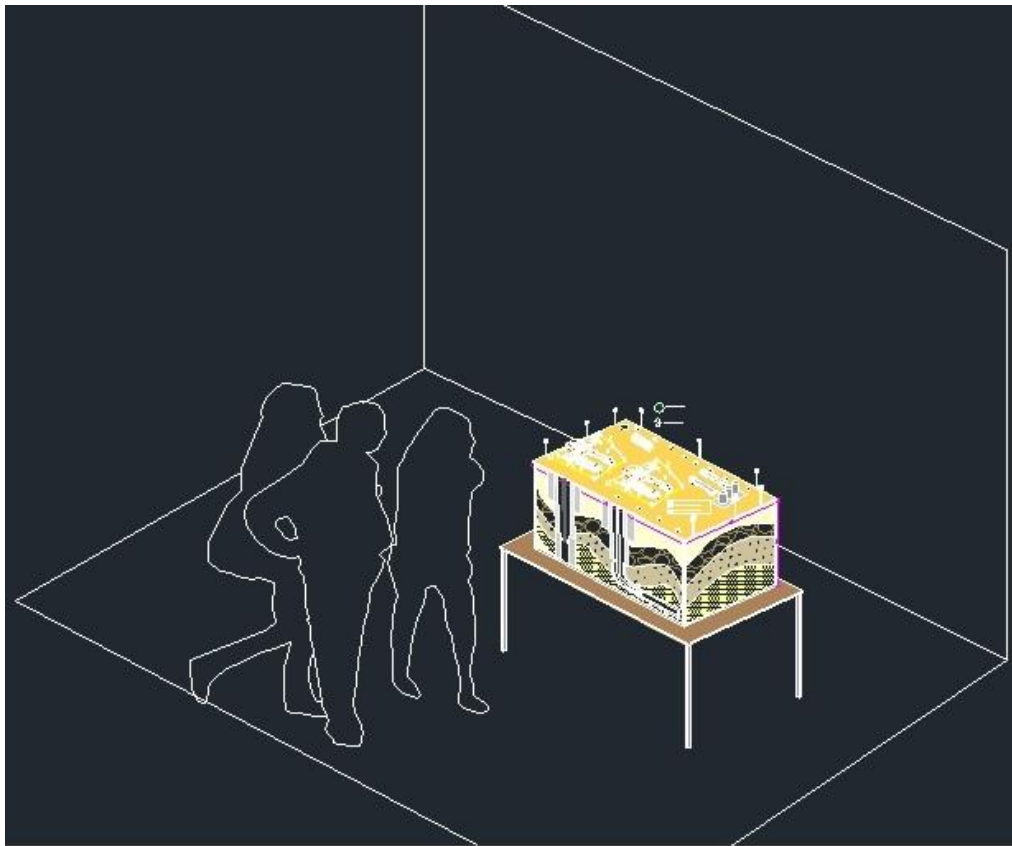
- Tiempo de subida: 0.04us
- Tiempo de bajada: 0.18us
- Características del pulso: Frecuencia (Hz) = 7.5 \* Caudal (L/min)
- Pulsos por litro: 450
- Durabilidad: mínimo 300.000 ciclos
- Conexión de 1/2" nominal, 0.75" de diámetro externo y rosca de 1/2"
- Tamaño: 63.5 mm x 35mm x 35mm

#### 4) FUNCIONAMIENTO GENERAL

**4.1 Diseño del Prototipo Simulador.** Una vez recibidas las capacitaciones y asesorías requeridas por parte de la empresa Guacamaya Oil Services S.A.S, donde se adquirió los conocimientos necesarios acerca de cómo se lleva a cabo en campo una Limpieza de arenas y cómo se evidencia la Separación de Fases teniendo en cuenta tanto la parte operativa como las diferentes acciones preventivas y correctivas que se deben tener en algún suceso inesperado; se procedió al diseño y ejecución de nuestro prototipo Simulador.

En primer lugar, se realizó el diseño del prototipo físico como tal en el software de AUTOCAD lo cual permitió tener una visión más real y atractiva del proyecto.

**Figura 18.** Diseño del Simulador de la Mecánica de Fluidos para operaciones de limpieza y Separación de Fases en pozos verticales y horizontales con tecnología Coiled Tubing.



**Fuente.** Autores

Habiéndose realizado el diseño se procede a la implementación del mismo a escala en material acrílico para el cual se realizan las correspondientes conversiones de pulgadas reales de tuberías correspondientes tanto al pozo vertical como horizontal a 1/10 (escala final).

A continuación;

✓ **Casing de Superficie**

9.5 Pulgadas	X
1 Pulgada	0.0254m

$$X = \frac{0.0254m \times 9.5 \text{ pulgadas}}{1 \text{ pulgada}} = 0.2413 \text{ m} = 24.13 \text{ cm (Perímetro)}$$

$$P = 2\pi \times r;$$

$$r = \frac{24.13}{2\pi} = 3.84$$

Diámetro = 7.68 cm; largo = 5cm

✓ **Casing Intermedio**

7 Pulgadas	x
1 Pulgada	0.0254m

$$X = \frac{0.0254m \times 7 \text{ pulgadas}}{1 \text{ pulgada}} = 0.1778 \text{ m} = 17.78 \text{ cm (Perímetro)}$$

$$P = 2\pi \times r;$$

$$r = \frac{17.78}{2\pi} = 2.82$$

Diámetro = 5.64 cm; largo = 23cm

✓ **Lainer**

5 Pulgadas

x

1 Pulgada

0.0254m

$$X = \frac{0.0254m \times 5 \text{ pulgadas}}{1 \text{ pulgada}} = 0.127 \text{ m} = 12.7 \text{ cm (Perímetro)}$$

$$P = 2\pi \times r;$$

$$r = \frac{12.7}{2\pi} = 2.02$$

Diámetro = 4.04 cm; largo = 22cm

✓ **Tubing**

2 <sup>7</sup>/<sub>8</sub> Pulgadas

x

1 Pulgada

0.0254m

$$X = \frac{0.0254m \times 2 \frac{7}{8} \text{ pulgadas}}{1 \text{ pulgada}} = 0.073 \text{ m} = 7.3 \text{ cm (Perímetro)}$$

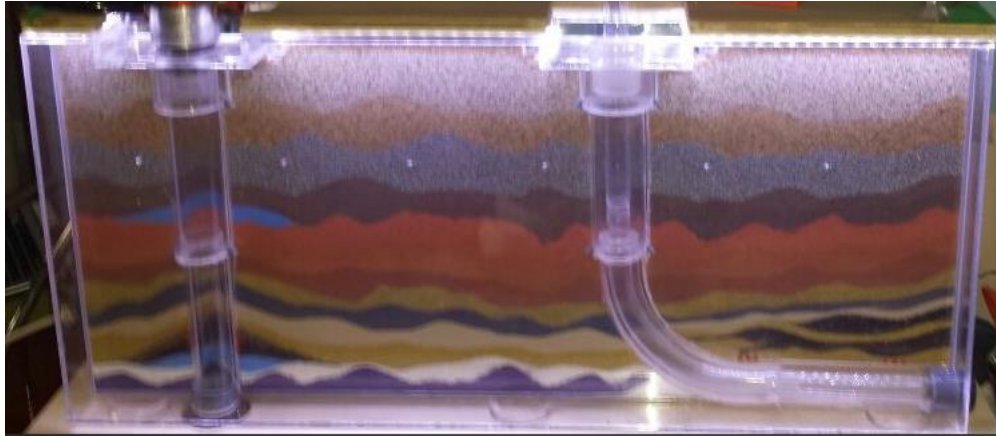
$$P = 2\pi \times r;$$

$$r = \frac{7.3}{2\pi} = 1.16$$

Diámetro = 2.32 cm; largo = 45cm



**Figura 19.** Prototipo a escala de tuberías de pozo horizontal y vertical de petróleo implementado en el Simulador.



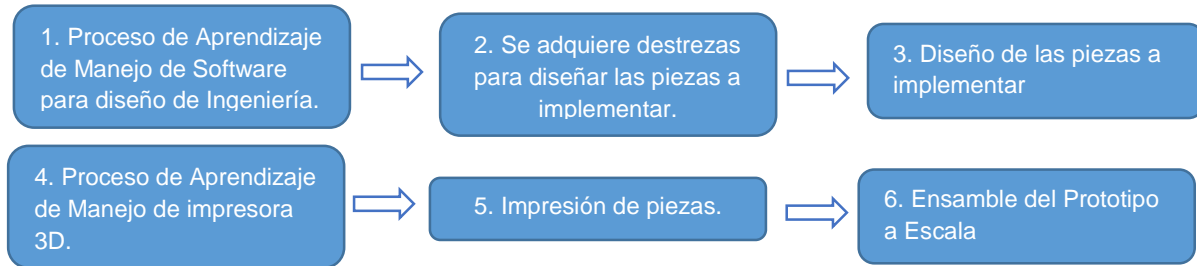
**Fuente.** Autores

Nótese en la Figura 19. el prototipo a escala de las tuberías correspondientes tanto al pozo vertical como horizontal del Simulador al igual que las diferentes formaciones geológicas que se presentan en los yacimientos habitualmente, las cuales se realizaron teniendo en cuenta la Estratigrafía valle superior del Magdalena (correspondiente al departamento del Huila).

A continuación se realizó el modelamiento y elaboración (3D) de piezas ubicadas en la superficie del Prototipo Simulador mediante el Software Solid Works para su posterior impresión en 3D y así brindar material didáctico donde se observe y explique de forma funcional, atractiva y lo más real posible la Mecánica de Fluidos en operaciones de Workover (Limpieza de Arenas) mediante el uso de Tecnología Coiled Tubing.

#### 4.1.1 Componente Ingeniería de Diseño.

**Figura 20.** Diagrama en bloques Ingeniería de Diseño.

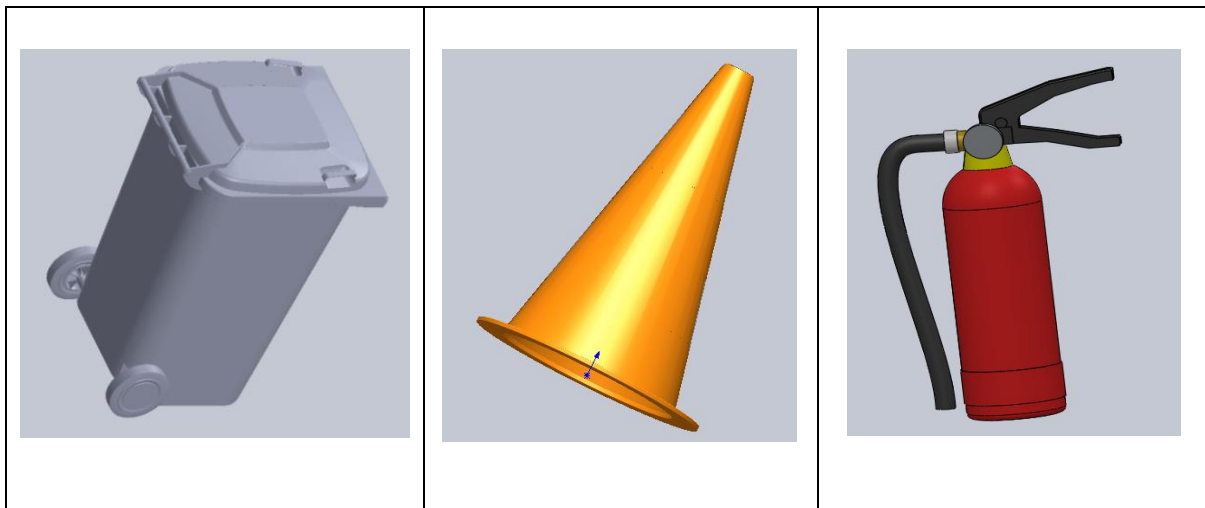


**Fuente.** Autores

#### 4.1.2 Diseños e Impresiones en Solid Works (3D)

##### 4.1.2.1 Diseños HSEQ

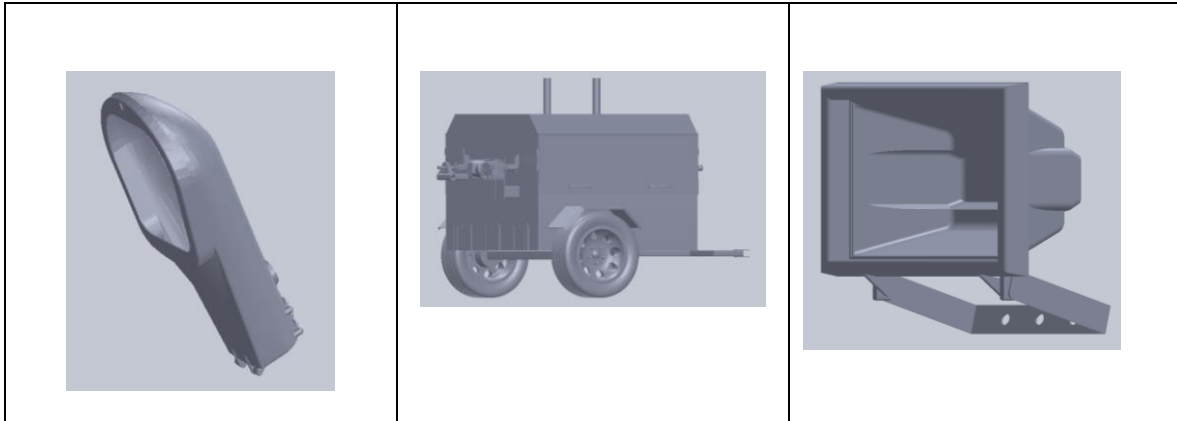
**Tabla 3.** Diseños en Solid Works piezas HSEQ en arme de locación (Caneca de Residuos, cono, extintor Multipropósito)



**Fuente.** Autores

#### 4.1.2.2 Iluminación Locación

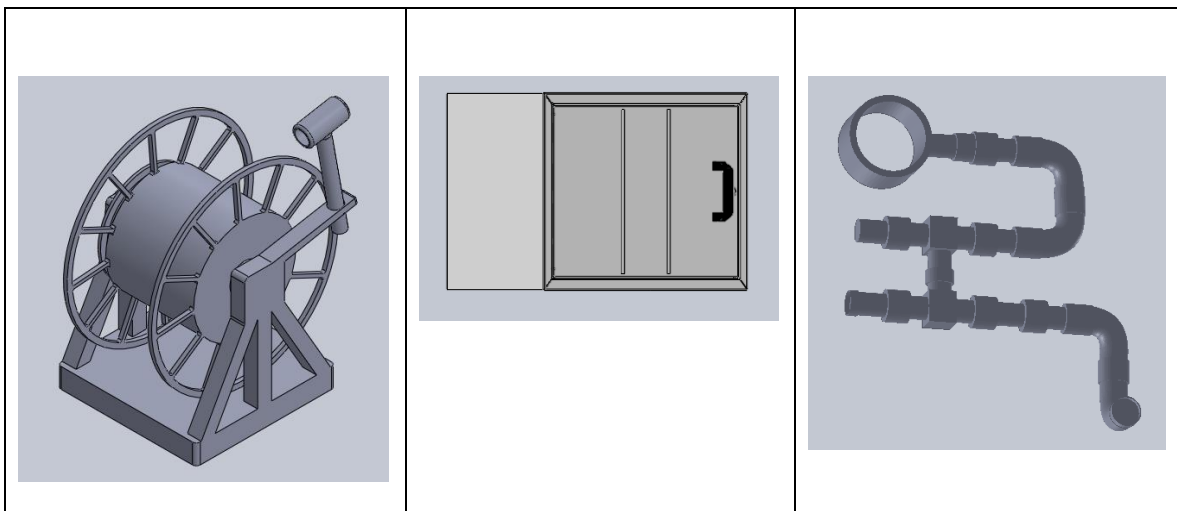
**Tabla 4.** Diseños en Solid Works piezas Iluminación Locación en arme de locación (Luz Posta, Planta Estadio, Reflector)



Fuente. Autores

#### 4.1.2.3 Unidad de Coiled Tubing

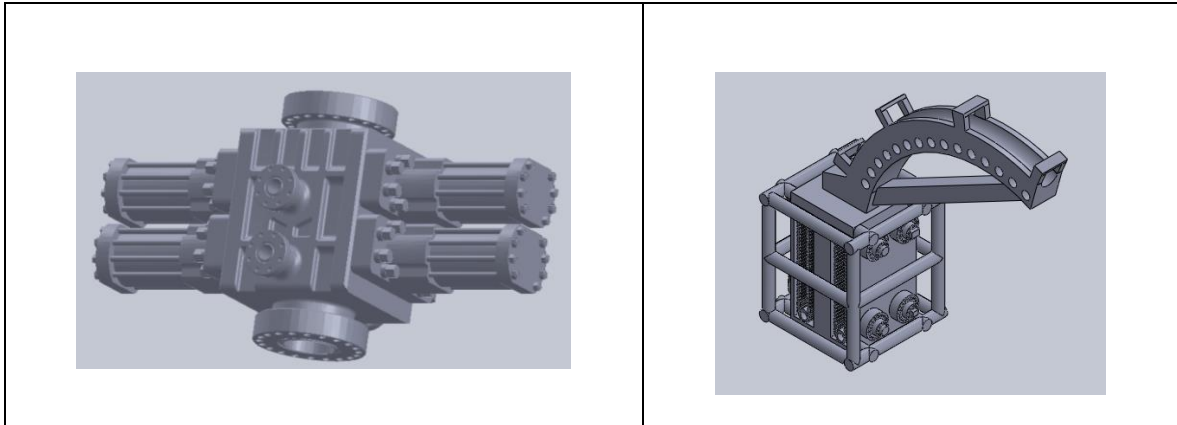
**Tabla 5.** Diseños en Solid Works piezas Unidad de Coiled Tubing (Rollo, caja de herramientas, manifold)



Fuente. Autores

#### 4.1.2.4 Arme sobre Cabezal de Pozo

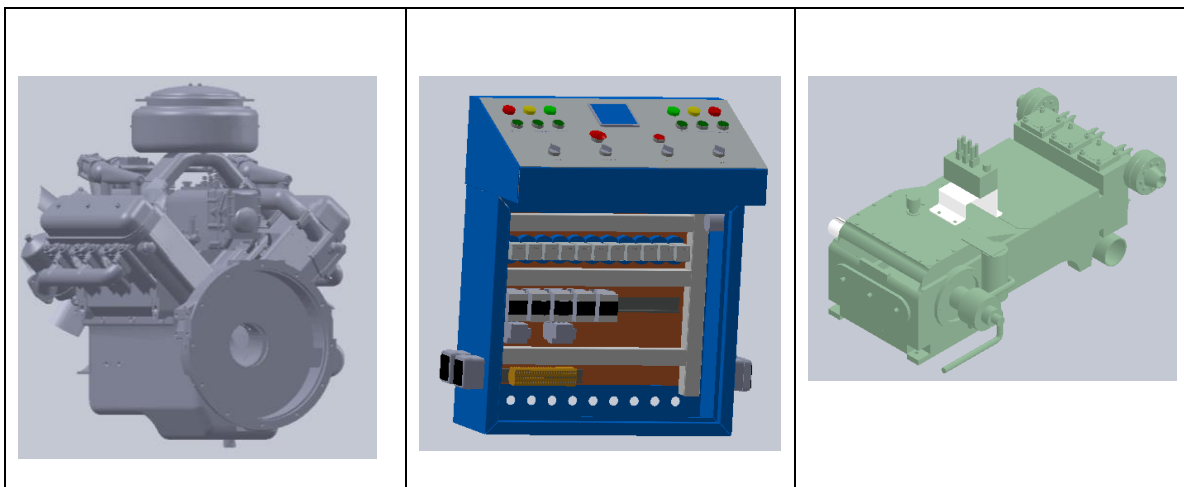
**Tabla 6.** Diseños en Solid Works piezas sobre arme de Cabezal de Pozo (BOP, Inyector de Unidad de Coiled Tubing).

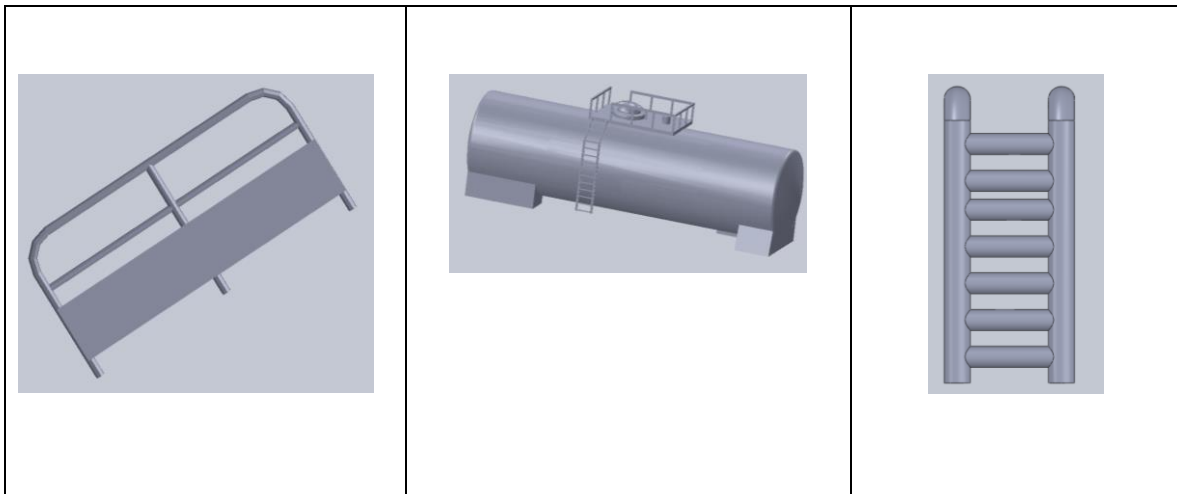


Fuente. Autores

#### 4.1.2.5 Unidad de Bombeo

**Tabla 7.** Diseños en Solid Works piezas Unidad de Bombeo (motor, control de mandos, bomba, reja, tanque de almacenamiento de fluidos, escalera).

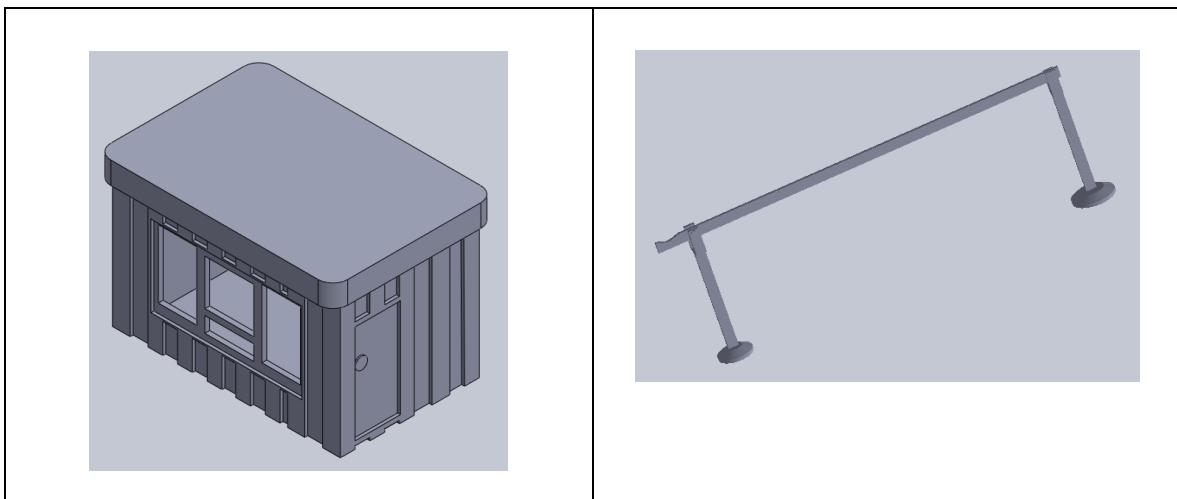




**Fuente.** Autores

#### 4.1.2.6 Locación

**Tabla 8.** Diseños en Solid Works piezas de Locación (caseta, stop parking de entrada a locación).



**Fuente.** Autores

**Figura 21.** Resultados Diseño e implementación de piezas 3D en Prototipo Simulador



**Fuente.** Autores

Como puede observar en la Figura 21 se encuentra gran parte de las piezas diseñadas y elaboradas con software y hardware mediante tecnología 3D implementadas en el Prototipo Simulador.

Es apropiado saber que la escala en la cual se construyeron las piezas referentes a la Seguridad que se debe tener en una Operación como estas (Limpieza de Arenas en pozos petroleros) es diferente a las demás piezas ya que el objetivo es resaltar la gran importancia de los sistemas manejados e implementados HSEQ (Salud, Ambiente, Seguridad, Calidad), ya que es misión de Guacamaya Oil Services S.A.S realizar la prestación de todos sus servicios bajo los lineamientos de un Sistema Integral de Gestión de HSEQ (calidad, seguridad, salud ocupacional y medio ambiente) y buscar el bienestar de su personal y el de sus familias, el de sus asociados y el de la sociedad en general; Logrando así la satisfacción plena de sus clientes en relación a sus servicios, tratando de exceder siempre las expectativas.

## 4.2 Diseño del Control de la Bomba Sumergible e Iluminación Simulador

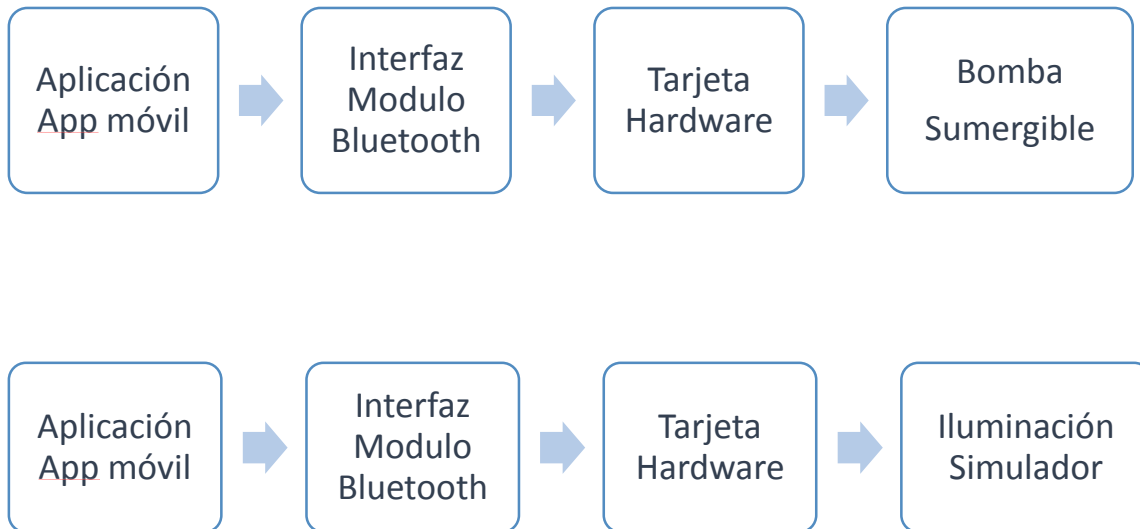
Ésta fase del proyecto nos permite que la Bomba Sumergible se active o desactive por medio de la orden que recibe a través de un dispositivo móvil Android (Software Libre). Gracias a las diversas plataformas software para el desarrollo de aplicaciones móviles que se están implementando en la actualidad, que tienen las características de poder comunicarse, la cual en este caso sería vía BLUETOOTH con una tarjeta Hardware Arduino basada en microcontroladores con entornos de desarrollo que implementan Wiring (lenguajes propios de programación) y que permiten la ejecución de múltiples diseños y proyectos electrónicos ejecutables en la misma tarjeta hardware con sencillez y bajo costo, y con la posibilidad de trabajar con comunicación inalámbrica que es lo que se evidencia en nuestra aplicación.

El diagrama del diseño del control de la Bomba Sumergible e Iluminación del Simulador vía Bluetooth desde una aplicación móvil es como se presenta en la figura 11. Y consiste en un sistema de activación controlado desde un teléfono celular o dispositivo móvil con sistema operativo Android mediante una aplicación móvil que se instala en los mismos.

A continuación se programa la tarjeta Hardware para que reciba los datos enviados desde el teléfono celular, los interprete y proceda a efectuar las salidas digitales correspondientes para ejecutar la operación indicada; ya sea la iluminación del prototipo en su exterior (Iluminación Locación, Planta Estadio) o la acción de controlar la bomba sumergible que permitirá el paso o no paso de los fluidos a inyectar tanto al pozo vertical como al horizontal para de ésta manera iniciar el proceso de Limpieza de Arenas donde se evidencia Separación de fases, tal como sucede en un procedimiento habitual de trabajo en Campo, o todas las tareas al mismo tiempo.



**Figura 22.** Diagrama de Bloques Control ON/OFF

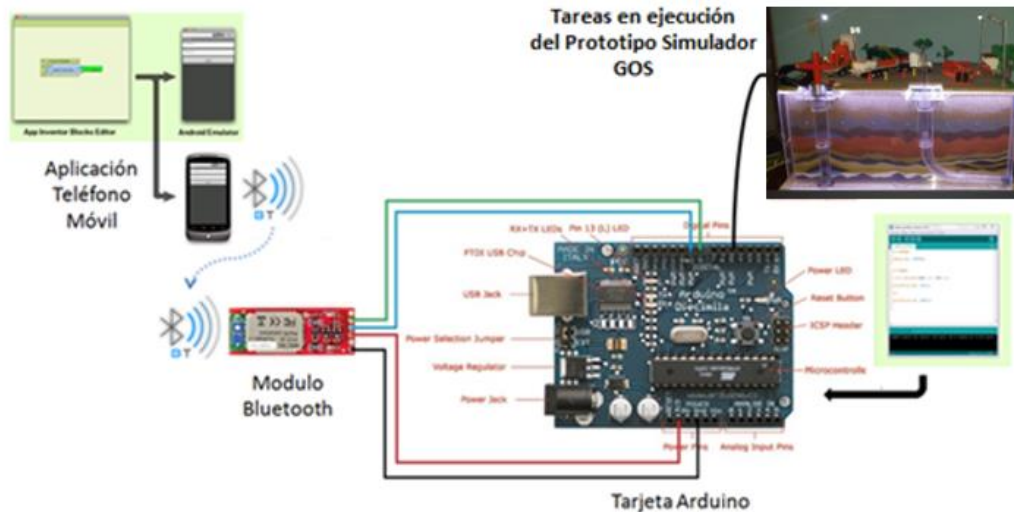


**Fuente.** Autores

La aplicación 'SIMULADOR\_GOS' ofrece una interfaz sencilla y agradable, donde el usuario realiza las tareas de activación y desactivación con sencillos comandos táctiles; estos comandos son enviados mediante la conexión del celular o dispositivo móvil hacia el servidor o módulo Arduino vía Bluetooth, el cual procesa la información y lleva a cabo la correspondiente tarea sugerida.



**Figura 23.** Representación gráfica del Funcionamiento de la Aplicación Móvil



**Fuente.** Autores

## 4.2.1 Aplicación Móvil

### 4.2.1.1 Componentes de una aplicación Android en App Inventor 2 <sup>7</sup>

**Editor de Bloques.** Con el Editor de bloques se define cómo se comportará la aplicación. Aquí se establece lo que los componentes deben hacer y cuándo hacerlo. Por ejemplo, que debe ocurrir cuando el usuario pulsa un botón.

El Editor de bloques se ejecuta en una ventana independiente.

El Editor de bloques tiene dos fichas en la esquina superior izquierda: Built-In (incorporados) y My Blocks (mis bloques). Los botones de debajo de cada ficha se amplían y se muestran los bloques cuando se hace clic. La lista bloques Built-In son el conjunto estándar de bloques que están disponibles para cualquier

<sup>7</sup> App Inventor en Español. Conceptos. Antonio Ricoy Riego.  
<https://sites.google.com/site/appinventormegusta/conceptos>

aplicación a construir. Los bloques My Blocks contienen bloques específicos que están vinculados al conjunto de componentes que se eligen para cada aplicación.

**Eventos y Controladores de eventos.** Con App Inventor, toda la actividad se produce en respuesta a un evento. Las aplicaciones no deben contener bloques fuera de una pieza "when do" de un evento.

**Propiedades.** Los componentes tienen propiedades que se pueden ajustar para cambiar la forma en que el componente aparece o actúa dentro de la aplicación. Para ver y cambiar las propiedades de un componente, primero se debe seleccionar el componente deseado en la lista de componentes.

**Componentes.** Los componentes son los elementos básicos que se utilizan para hacer aplicaciones Android. Algunos componentes son muy simples, como una Etiqueta (Label), que sólo muestra el texto en pantalla, o un botón (Button) que se pulsa para iniciar una acción. Otros componentes son más elaborados: un lienzo (Canvas) que puede almacenar imágenes fijas o animaciones, un sensor de movimiento (Accelerometor Sensor) que funciona como un mando de Wii y detecta cuando se mueve o agita el teléfono, los componentes para hacer o enviar mensajes de texto, los componentes que para música y video, los componentes para obtener información de sitios Web, y así sucesivamente.

A continuación se explica la estructura de Bloques implementada para el correcto funcionamiento de la Aplicación móvil del presente proyecto **SIMULADOR\_GOS (Guacamaya Oil Services)**

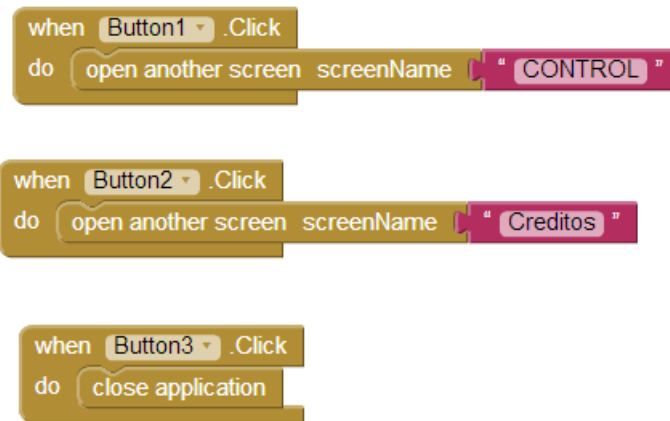
#### 4.2.1.2 Programación Diagramas de Bloques App Móvil 'SIMULADOR\_GOS'

*SIMULADOR\_GOS* cuenta con tres pantallas en su interfaz de la app móvil por medio de las cuales el usuario puede acceder y controlar vía BLUETOOTH el total funcionamiento del presente Prototipo; a continuación se describe la programación en bloques de cada una de ellas.

- **Screen1.** Los bloques Button 1 (*Control ON/OFF*), Button 2 (*Creditos*) y Button 3 (*Salir*) se ejecutan al inicio de la aplicación. En ellos se incluye la orden al proceso de abrir la siguiente pantalla deseada por el usuario y de ésta manera poder acceder ya sea a controlar el funcionamiento del

Simulador como tal, visualizar y conocer los autores de la app móvil o en tal caso salir por completo de ésta y acceder en otra ocasión.

**Figura 24.** Diagrama de Bloques Screen1 de la app móvil *SIMULADOR\_GOS*



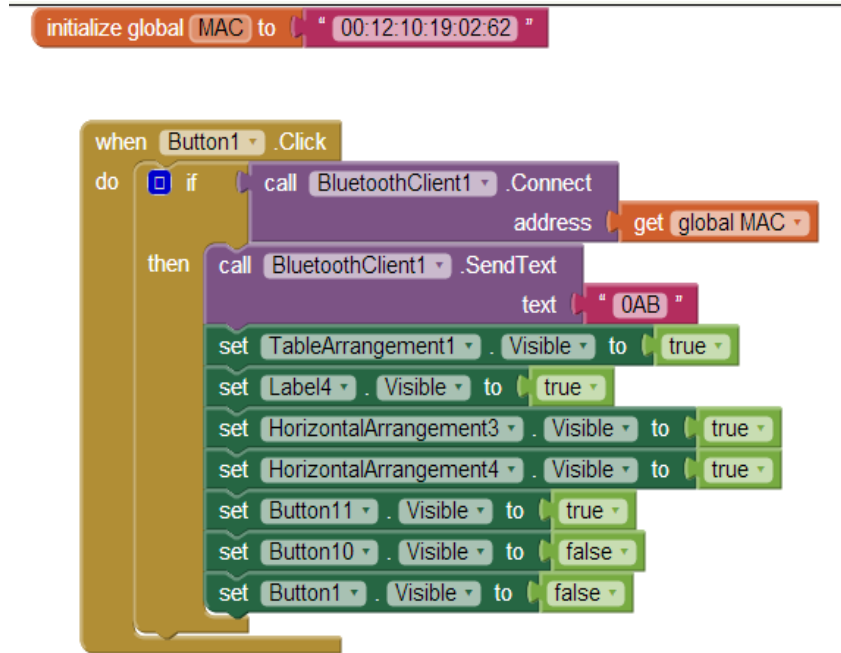
**Fuente.** Autores

- **CONTROL.** El bloque *initialize global MAC* se ejecuta al inicio de ésta pantalla y crea una variable global a través de la dirección MAC del dispositivo a enlazar (Módulo Bluetooth) permitiendo el funcionamiento de los demás bloques conectados entre sí a éste.

Lo anterior se puede realizar cuando el usuario da click en “*Conectar Bluetooth*”, ya que realiza la llamada al proceso de Conectar a la dirección del cliente Bluetooth asignado mediante el Arreglo establecido previamente en la programación de los Bloques.

Una vez ejecutado este comando se permite la visibilidad o no de los botones encargados de controlar el funcionamiento del Prototipo Simulador (Iluminación Locación, Planta Estadio de Pozo, Bombeo – Mecánica de Fluidos), lo cual depende de la exitosa conexión con el servidor que para este caso es bastante rápida.

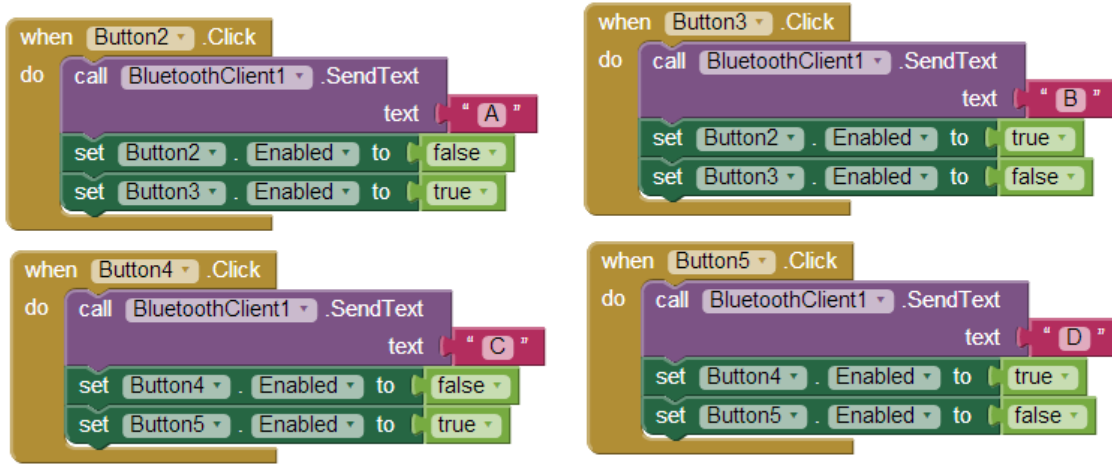
**Figura 25.** Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*



**Fuente.** Autores

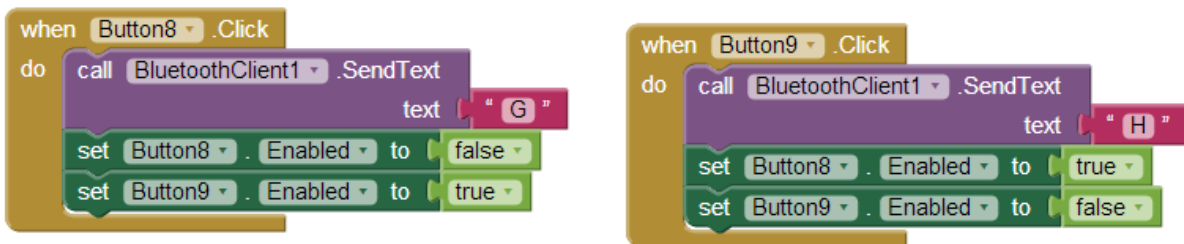
Los bloques siguientes llaman al proceso de Encender o Apagar la Iluminación en locación, la Planta Estadio del pozo en funcionamiento y el bombeo del Fluido a inyectar al pozo para realizar la operación de Limpieza de Arenas donde se evidencia la Separación de fases del Prototipo.

**Figura 26.** Declaraciones variables A, B, C y D Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*



**Fuente.** Autores

**Figura 27.** Declaraciones variables G y H Diagrama de Bloques CONTROL de la app móvil *SIMULADOR\_GOS*

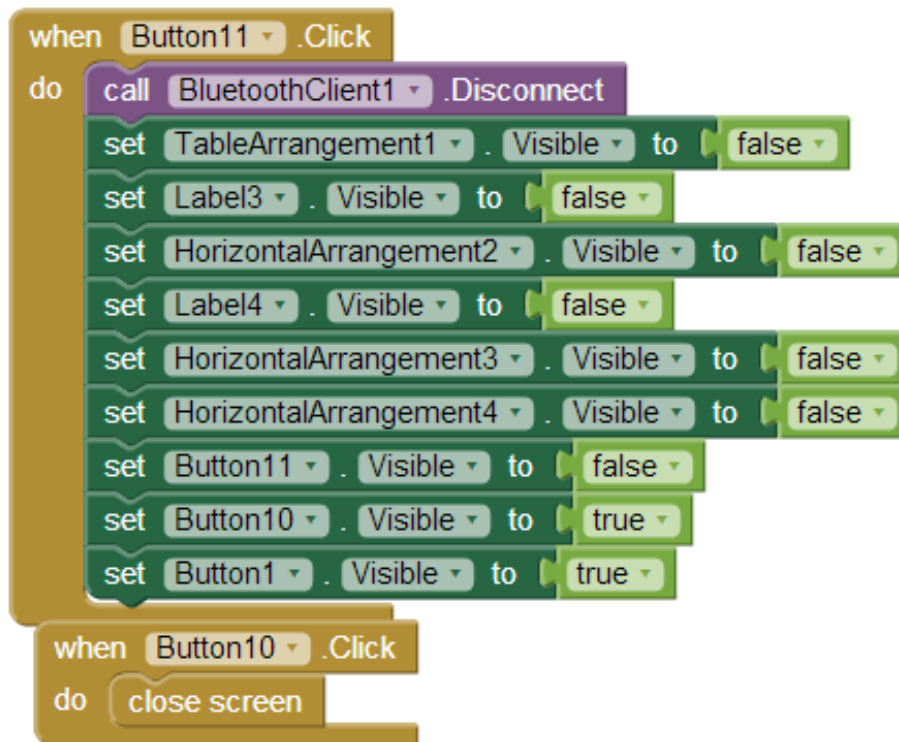


**Fuente.** Autores

Posteriormente al usuario presionar el Botón “Desconectar Bluetooth” le permite la acción de llamar al cliente Bluetooth para realizar la Desconexión y por ende deshabilitación y No visibilidad de los botones anteriormente declarados y explicados.

Por último a lo que ésta pantalla concierne le brinda la posibilidad al usuario de Salir de la interfaz de “CONTROL” para poder acceder a las demás.

**Figura 28.** Diagrama de Bloques “Desconectar Bluetooth” y “Salir” de la pantalla de CONTROL de la app móvil SIMULADOR\_GOS.

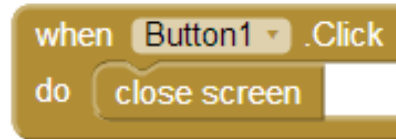


**Fuente.** Autores

- **Creditos**

Permite al usuario identificar los autores diseñadores de la Aplicación Móvil “SIMULADOR\_GOS”

**Figura 29.** Creditos de la app móvil *SIMULADOR\_GOS*.



**Fuente.** Autores

#### 4.2.1.3 Vistas Pantallas de App Móvil

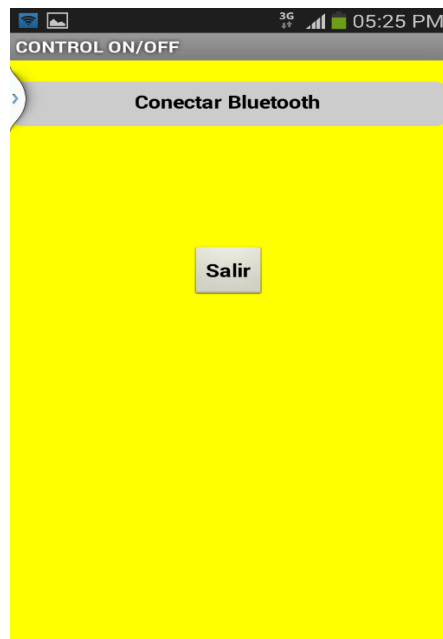
Con el fin de organizar los elementos visibles de la aplicación móvil *SIMULADOR\_GOS* y presentar una interfaz llamativa, agradable y sobre todo de fácil manejo e interpretación al usuario; a continuación se muestran las diferentes vistas con las cuales éste puede interactuar y poner en funcionamiento el Prototipo.

**Figura 30.** Vista “Screen1” *SIMULADOR\_GOS*.



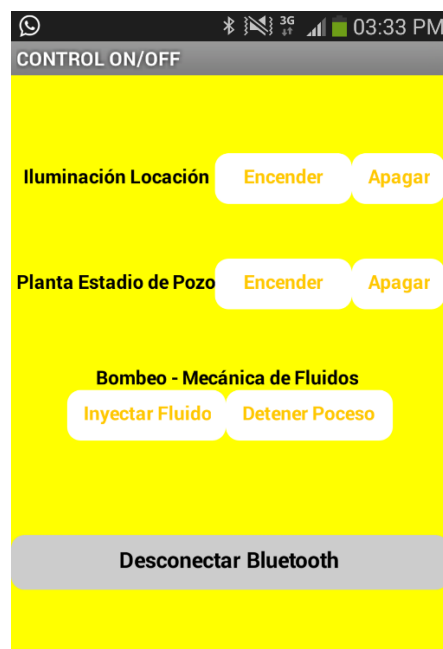
**Fuente.** Autores

**Figura 31.** Vista “CONTROL” SIMULADOR\_GOS



**Fuente.** Autores

**Figura 32.** Vista “CONTROL ON/OFF” Screen SIMULADOR\_GOS



**Fuente.** Autores



Figura 33. Vista “Creditos” SIMULADOR\_GOS



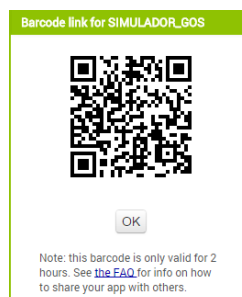
Fuente. Autores

#### 4.2.1.4 Configuración del archivo Android

App Inventor 2 genera automáticamente el archivo ejecutable *App* (*provide QR code for .apk*), en este caso **SIMULADOR\_GOS.apk**, que se ubica en el navegador de paquetes y especifica cuál es la actividad principal.

El código QR o código de respuesta rápida almacena la información en este caso la aplicación móvil, en una matriz de puntos que se caracteriza por los tres cuadros que se encuentran en las esquinas superior derecha, superior izquierda e inferior izquierda y que permiten detectar la posición del código al lector.

Figura 34. Vista del QR de la App móvil “SIMULADOR\_GOS”



Fuente. Autores

### **4.3 Diseño de la interfaz gráfica**

Como ya hemos visto, la aplicación móvil permite hacer el control sobre el prototipo SIMULADOR, sin embargo, el prototipo cuenta con sensores que permitirán el monitoreo en tiempo real de variables de presión y caudal del fluido que se inyecta a los correspondientes pozos del simulador; para dicha tarea se diseña e implementa una interfaz gráfica informática, un ejecutable creado en lenguaje C# en SharpDevelop que permitirá la adquisición, procesamiento y graficado de los datos correspondientes a las variables de Presión de Inyección, Presión en el Anular Tubing – Coiled Tubing, y Caudal.

#### **4.3.1 Diseño de la ventana (Formulario)**

SharpDevelop cuenta con formularios de Windows que permiten el diseño gráfico de la ventana.

El control ComboBox de Windows Forms se utiliza para mostrar datos en un cuadro combinado desplegable. De forma predeterminada, el control ComboBox es un cuadro de lista que muestra una lista de elementos, de los cuales el usuario puede seleccionar uno. Este control nos permite seleccionar el puerto al cual está conectada la tarjeta Arduino Leonardo que para esta aplicación actúa como tarjeta de adquisición y envía por Puerto Serial los datos tomados de cada sensor.

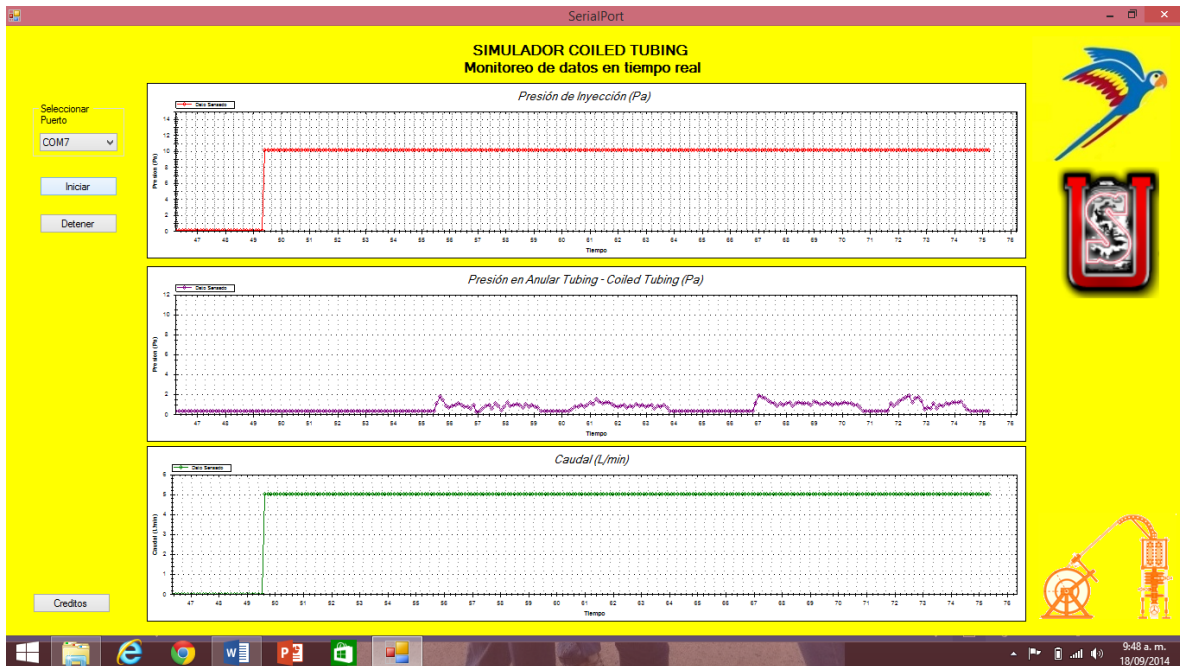
El control Button de Windows Forms permite al usuario hacer clic en él para ejecutar una acción. Utilizamos controles Button para 'Iniciar' y 'Detener' el procesamiento y graficado de los datos, y a su vez un Button de 'Créditos' sobre el ejecutable.

Los PictureBox son Clases que representan un control de cuadro de imágenes de Windows para mostrar las imágenes correspondientes a los logos de la empresa Guacamaya Oil Services S.A.S y de la Universidad Surcolombiana, junto con una imagen que identifica Coiled Tubing.

Los Label son componentes que se utilizan para desplegar textos o mensajes estáticos dentro de las formas, textos tales como encabezados, solicitud al usuario del programa para que proporcione algún dato o información. El encabezado y los títulos de cada gráfica corresponden a Labels.

Finalmente la ventana cuenta con tres gráficas correspondientes a las variables en medición, estas graficas corresponden a un *ZedGraphControl* cada una, que permite implementar charts o gráficos de tipo 2D en nuestro formulario.

**Figura 35.** Respuesta Data Sensores Electrónicos (Interfaz gráfica)



Fuente. Autores

### 4.3.2 Descripción del Código Fuente

#### 4.3.2.1 Instrucciones (Uso de objetos IDisposable)

Los objetos IDisposable proporcionan un mecanismo para liberar recursos no administrados, mediante la instrucción 'using'.

```
using System;
using System.IO.Ports;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using System.Text;
```

```
using ZedGraph; //Para poder usar los comandos relacionados con el gráfico
using System.Linq;
using System.Threading;
```

#### 4.3.2.2 Componentes de inicialización

Corresponde a los componentes e instrucciones a ejecutar inmediatamente se accede al ejecutable.

```
namespace Tesis
{
    public partial class MainForm : Form
    {
        int tickStart = 0;
        public MainForm()
        {
            InitializeComponent();

            // Obtiene una matriz con los nombres de los puertos serie del equipo actual y los pone
            // en el ComboBox
            foreach(string s in SerialPort.GetPortNames())
            {
                comboBox1.Items.Add(s);
            }

            // Configuración de cada grafico (es este caso el que corresponde a Presión de Inyección)
            GraphPane myPane1 = zedGraphControl1.GraphPane;
            myPane1.Title.Text = "Presión de Inyección"; //Titulo
            myPane1.XAxis.Title.Text = "Tiempo"; //Nombre del eje X
            myPane1.YAxis.Title.Text = "Presion (Pa)"; //Nombre del eje Y
            // Grilla
            myPane1.YAxis.MajorGrid.IsVisible = true;
            myPane1.YAxis.MinorGrid.IsVisible = true;

            //Se guardan 1.200 puntos para cada lista. El RollingPointPairList es una clase de
            //almacenamiento eficiente, que siempre mantiene un conjunto de rodadura de punto de
            //datos sin necesidad de cambiar los valores de datos (Determinamos la capacidad para las
            //tres variables, en este caso list1 corresponde a los utilizados en Presión de Inyección)
```

```
RollingPointPairList list3 = new RollingPointPairList(1200);

// Se determinan los parámetros de la curva (curve3 relaciona a la curva de Caudal)
LineItem curve3 = myPane3.AddCurve("Dato Sensado", list3, Color.Green,
SymbolType.Circle);
// Se configura manualmente el rango de cada grafica (myPane1 – Presión de Inyección)
myPane1.XAxis.Scale.Min = 0; // Se determina el valor mínimo inicial para el eje X.
myPane1.XAxis.Scale.Max = 30; // Se determina el valor máximo inicial para el eje X.
myPane1.XAxis.Scale.MinorStep = 0; // Espacios mínimo en la escala
myPane1.XAxis.Scale.MajorStep = 1; // Espacios máximo en la escala
myPane1.YAxis.Scale.Min = 0; // Se determina el valor mínimo inicial para el eje Y.
myPane1.YAxis.Scale.Max = 15; // Se determina el valor máximo inicial para el eje Y.
myPane1.YAxis.Scale.MinorStep = 0; // Espacios mínimo en la escala
myPane1.YAxis.Scale.MajorStep = 1; // Espacios máximo en la escala

// Permite escalar los ejes (zedGraphControl2 asigna la presión Tubing-Coiled Tubing)
zedGraphControl2.AxisChange();

// Obtiene el número de milisegundos transcurridos desde que se inició el sistema.

tickStart = Environment.TickCount;
}
```

#### 4.3.2.3 Configuración de los botones 'Iniciar' y 'Detener'

Con los botones abrimos y cerramos el puerto y a su vez, activamos y detenemos el timer de operación.

```
// Boton 'Iniciar'
void Button1Click(object sender, EventArgs e)
{
    try{
        serialPort1.PortName = comboBox1.Text;
        serialPort1.Open(); //Abre la conexión de puerto serie.
        timer1.Interval = 10; // Intervalo de temporizador se determina (10)
```

```
timer1.Enabled = true; // Timer se habilita.
timer1.Start(); // Timer se inicia.
}
// En caso de no haber seleccionado previamente el puerto
catch {
    MessageBox.Show("Error! Seleccione puerto serial");
}
}
// Boton 'Detener'
void Button3Click(object sender, EventArgs e)
{
    timer1.Enabled = false; // Timer se desactiva
    timer1.Stop(); // Timer se detiene
    serialPort1.Close(); //Cierra el puerto serial
}
```

#### 4.3.2.4 SetSize

Cambia el número de columnas y filas de datos, así como el número de niveles de etiquetas de columnas y de filas de una cuadrícula de datos asociada a un gráfico en un momento dado.

```
void SetSize()
{
    Rectangle formRect = this.ClientRectangle;
    formRect.Inflate(-10, -10);
    // Se debe hacer una condicion para cada zedGraphControl
    if (zedGraphControl1.Size != formRect.Size)
    {
        zedGraphControl1.Location = formRect.Location;
        zedGraphControl1.Size = formRect.Size;
    }
}
```

#### 4.3.2.5 Temporizador (Timer)

El Timer genera eventos recurrentes en una aplicación, es un temporizador basado en servidor que permite especificar un intervalo recurrente en el que se provoca el evento en la aplicación. Una vez hecho esto, podrá controlar este evento para proporcionar un procesamiento regular. El 'timerTick' lo usamos para condicionar con un If la función 'LeerDatos' pues mientras el puerto este abierto, el timer me permite ejecutar la función de lectura de datos.

```
// Timer
void Timer1Tick(object sender, EventArgs e)
{
    if( serialPort1.IsOpen ) {
        LeerDatos();
    }
}
```

#### 4.3.2.6 Función 'Leer Datos'

Esta función permite adquirir o recuperar los datos provenientes del puerto serial y asignarlos a la lista que los almacena remotamente para luego graficarlos en sus respectivos ZedGraphControl.

```
// Función 'LeerDatos'
void LeerDatos()
{
    // Se asegura que curvelist tiene al menos una curva (Un if para cada grafica)
    if (zedGraphControl1.GraphPane.CurveList.Count <= 0)
        return;

    //Obtener Curveltem por primera vez en el gráfico (se repite para cada grafica)
    Lineltem curve1 = zedGraphControl1.GraphPane.CurveList[0] as Lineltem;
    if (curve1 == null)
        return;

    //Obtener el PointPairList para casa list (Lista de puntos)
    IPointListEdit list1 = curve1.Points as IPointListEdit;
```

```
if (list1 == null)
return;

// Recuperando datos del puerto serie (tipo Double)
double veri1 = Convert.ToDouble(serialPort1.ReadLine());
double veri2 = Convert.ToDouble(serialPort1.ReadLine());
double veri3 = Convert.ToDouble(serialPort1.ReadLine());
double time = (Environment.TickCount - tickStart) / 1000.0;

// Añadimos los datos a cada lista (list1, list2 y list3)
list1.Add(time, veri1/100);

Scale xScale1 = zedGraphControl1.GraphPane.XAxis.Scale; // Escala
if (time > xScale1.Max - xScale1.MajorStep)
{
    xScale1.Max = time + xScale1.MajorStep;
    xScale1.Min = xScale1.Max - 30.0;
}

//Esto realiza un comando de cambio de eje en el GraphPane.
zedGraphControl1.AxisChange();

// Invalida la región especificada del control (agrega a la región de actualización del
control, que es el área que volverá a pintarse en la siguiente operación de dibujo).
zedGraphControl1.Invalidate();

}
```

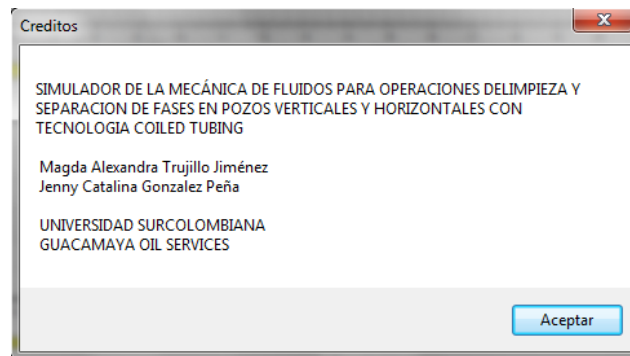
#### 4.3.2.7 Creditos

Al hacer clic en el botón 'Créditos' se mostrará un MessageBox que es un cuadro de mensaje que puede contener texto, botones, y símbolos que informan y pide al usuario, en nuestro caso, muestra al usuario información correspondiente al nombre del proyecto y autores.



```
// Boton 'Creditos'
void Button4Click(object sender, EventArgs e)
{
    MessageBox.Show("SIMULADOR DE LA MECÁNICA DE FLUIDOS PARA OPERACIONES DE"
+"LIMPIEZA Y SEPARACION DE FASES EN POZOS VERTICALES Y HORIZONTALES " +CON
TECNOLOGIA COILED TUBING \r\n\r\n Magda Alexandra Trujillo Jiménez \r\n Jenny Catalina
Gonzalez Peña \r\n\r\n UNIVERSIDAD SURCOLOMBIANA \r\n GUACAMAYA OIL
SERVICES","Creditos");
}
```

**Figura 36.** MessageBox Creditos



**Fuente.** Autores

#### 4.4 Código Arduino

La tarjeta Arduino Leonardo en el prototipo cumple dos funciones principales que son la ejecución del control sobre el Simulador y a su vez como tarjeta de adquisición de datos de los sensores implementados. El código con el que fue programada la tarjeta y que es ejecutado por la misma, reúne ambas funciones paralelamente y permiten que se desarrollen conjuntamente.

Para programar la placa es necesario contar con el entorno de desarrollo (IDE) y en el caso de disponer de una placa USB es necesario instalar los drivers FTDI. Estos drivers vienen incluidos en el paquete de Arduino mencionado anteriormente. Existen en la web versiones para distintos sistemas operativos.

#### 4.4.1 Declara variables e Inicializa

La primera parte del código corresponde a la declaración de variables y a la ejecución de la función 'rpm()' que corresponde a la medición de los flancos provenientes del sensor de efecto Hall que nos suministrara la medida de Caudal.

```
// volatil es una palabra clave conocida como variable calificador, se utiliza por lo general antes de que el tipo de datos de una variable, para modificar la forma en que el compilador y el programa posterior trata a la variable.
```

```
volatile int rpmcont; // para medir los flancos ascendentes de la señal
```

```
// int es variable entera. Los enteros son el tipo de datos principal para el almacenamiento de números.
```

```
int caudal;
```

```
int sensorCaudal = 2; // pin 2 corresponde a la entrada del sensor de efecto hall
```

```
// Esta es la función llama a la interrupción y conteo del líquido (sensor de efecto hall)
```

```
void rpm () {
```

```
    rpmcont++; // Esta función mide el flanco ascendente y descendente de los sensores de efecto Hall señal
```

```
}
```

#### 4.4.2 Función de Configuración o setup

La función setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

```
// El método setup() se ejecuta una vez, cuando se inicia el programa
```

```
void setup() {
```

```
    Serial.begin(9600); // función de configuración donde se inicializa el puerto serie que comunica a la tarjeta con el PC
```

```
// Especifica un llamado de rutina de servicio de interrupción (ISR) para llamar cuando se produce una interrupción. Sustituye cualquier función anterior que se adjuntó a la interrupción. Cada vez que se genera una interrupción se incrementara la variable rpm. RISING detecta flanco de subida.
```

```
attachInterrupt(1, rpm, RISING); // la interrupción se adjunta (1 trabaja el pin2)
```

```
Serial1.begin(9600); // función de configuración donde se inicializa el puerto
                      serie1 que comunica a la tarjeta con la App Movil

// pinMode configura el pin especificado a comportarse ya sea como una
// entrada o una salida, en este caso configuramos los pines 3, 4 y 5 como
// salidas
pinMode(4, OUTPUT); // Iluminación locación
pinMode(3, OUTPUT); // Planta Estadio de Pozo
pinMode(5, OUTPUT); // Activa Bombeo - Mecánica de Fluidos
}
```

#### 4.4.3 Función de Ciclo o Loop

Después de llamar a setup(), la función loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa. El método loop () se ejecuta una y otra vez, mientras la tarjeta Arduino Leonardo este alimentada (conectada por USB al PC.)

```
void loop() {
  rpmcont=0; // Se establece el contador en 0, listo para calcular
  sei(); // Habilita las Interrupciones
  delay (100); // Pausa de 100 milisegundos
  cli(); // Deshabilita las interrupciones
  // Cálculo de las rpm. (Pulso de frecuencia x 60seg) / 7.5Q, = Caudal
  // (litros/hora). La bomba sumergible maneja 350 litros/minuto. Ecuación
  // definida experimentalmente
  caudal = (rpmcont * 60 / 350);

  // Definimos como variable tipo entero las entradas análogas de los sensores
  // de presión
  int sensorPresion1 = analogRead(A0);
  Serial.println(sensorPresion1); // Imprime los datos al puerto serie como texto
  ASCII
```

```
int sensorPresion2 = analogRead(A1);
Serial.println(sensorPresion2);
Serial.println(caudal);

attachInterrupt(1, rpm, RISING); // la interrupción se adjunta
delay(100); // Pausa de 100 milisegundos para mejorar la estabilidad
```

#### 4.4.4 Ejecución del control

La aplicación móvil para cada caso ON/OFF envía por medio del módulo Bluetooth un carácter que va de la 'A' a la 'D' para la iluminación de la locación y la planta estadio, y usa los caracteres 'G' y 'H' para el control sobre la bomba sumergible. Hacemos llamar 'dato' al byte que llega por Bluetooth al puerto Serial1 de la tarjeta Arduino Leonardo y con un control tipo Switch comparamos el valor de la variable a los valores especificados en las declaraciones de cada caso. Cuando se encuentra una sentencia case cuyo valor coincide con el de la variable, se ejecuta el código en esa sentencia case, es decir, se activa o desactiva el pin al que corresponde.

```
byte dato; // Un byte almacena un número sin signo de 8 bits, de 0 a 255.

// available() permite obtener el número de bytes (caracteres) disponibles para
// su lectura desde el puerto serie Serial1. Se trata de datos que ya llegaron y
// se almacenan en el búfer de recepción (que tiene 64 bytes).
if (Serial1.available()) { // Si el Serial1 está disponible (tiene datos)
  dato=Serial1.read(); // dato es igual a la lectura del puerto serie

  // Comparamos el valor de la variable a los valores especificados de carácter.
  switch(dato){
  // Si el caracter que llega es 'A', entonces active (o ponga en alto 5 V) el pin 4
  // que corresponde a encender la iluminación de la locación.
  case 'A':
  {
  digitalWrite(4, HIGH); // digitalWrite escribe un alto o bajo a un pin digital.
  break; // Se usa para salir de la case en la sentencia switch.
  }
}
```

```
case 'B': // Apaga iluminación de la locación
{
digitalWrite(4, LOW);
break;
}

// Comparamos igual el valor de la variable a los valores de los caracteres
case 'C': // Enciende planta estadio de pozo
{
digitalWrite(3, HIGH);
break;
}
case 'D': // Apaga planta estadio de pozo
{
digitalWrite(3, LOW);
break;
}
case 'G': // Activa el bombeo (enciende bomba e inicia mecánica de fluidos)
{
digitalWrite(5, HIGH);
break;
}
case 'H':
{
digitalWrite(5, LOW);
break;
}
} // Cierra el switch(dato)
} // Cierra el if correspondiente al serial Serial1.available()
}
```

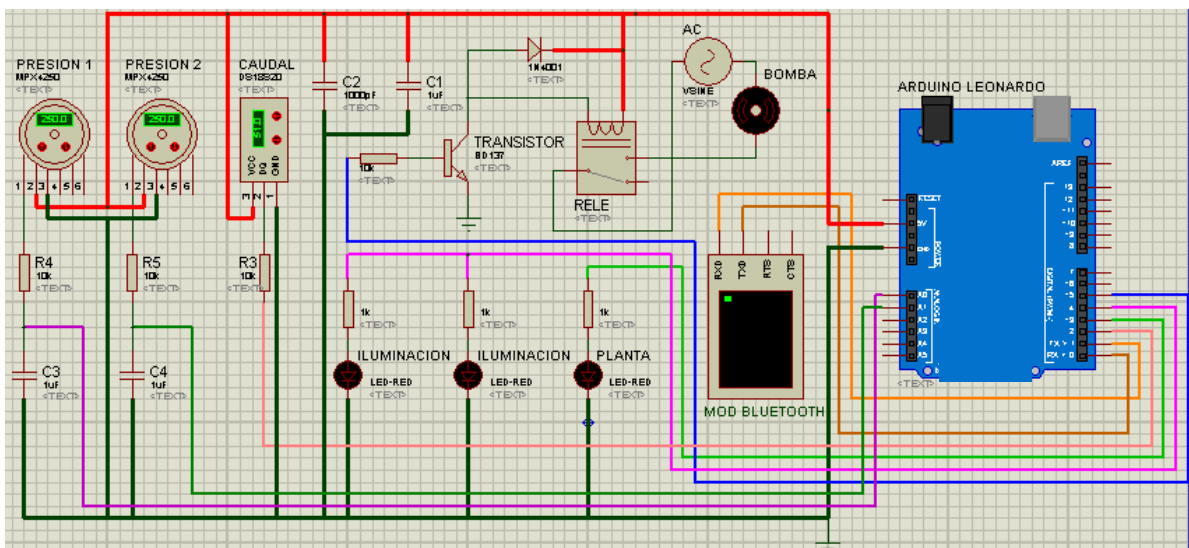
## 4.5 Diseño del circuito

El circuito cuenta con dos funciones importantes, una que se encarga del ingreso de los datos de los sensores a la tarjeta Arduino Leonardo, y la otra que maneja la parte del control sobre la iluminación, la planta estadio y el bombeo, que a su vez cuenta con una etapa de potencia de 5 voltios DC (Arduino) a 120 voltios AC (red pública).

### 4.5.1 Simulación

La Figura 37 corresponde al diseño del plano electrónico en su correspondiente simulación del circuito en ISIS Proteus. Los diseños realizados en ISIS pueden ser simulados en tiempo real, mediante el módulo VSM (Sistema Virtual de Modelado), asociado directamente con ISIS.

Figura 37. Simulación

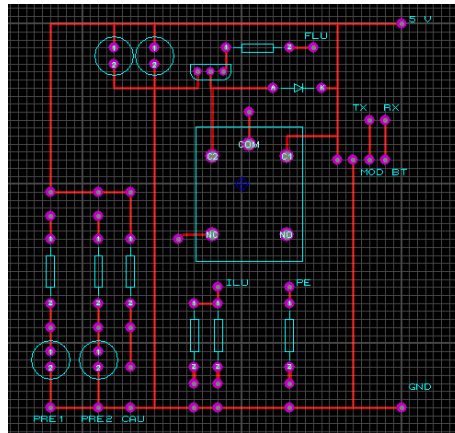


Fuente. Autores

#### 4.5.2 Diseño del circuito impreso PCB

Este diseño nos permite crear el esquema del circuito electrónico y a su vez obtuvimos el diseño del circuito impreso.

**Figura 38.** Circuito impreso



**Fuente.** Autores

## CONCLUSIONES

De manera satisfactoria se ha presentado el diseño e implementación del primer Prototipo Simulador de la Mecánica de Fluidos para operaciones de Limpieza y Separación de Fases en pozos verticales y horizontales con Tecnología Coiled Tubing en Colombia, el cual permite que los operarios, personal en general de la empresa Guacamaya Oil Services S.A.S y cualquiera que esté interesado en capacitarse y observar estos procesos, pueda evidenciar de manera clara en qué consisten éstos mediante la inyección de fluidos a través de tecnología Coiled Tubing.

Teniendo en cuenta lo anterior se ha logrado facilitar el manejo, funcionamiento y comprensión del Prototipo gracias al correcto desempeño de la Aplicación creada en software libre Android para dispositivos móviles con interfaz agradable y fácil de utilizar para el usuario, la cual permite controlar exitosamente la bomba sumergible de inyección de fluidos y la iluminación de éste de manera ON/OFF mediante comunicación Bluetooth.

Igualmente la respuesta de cada uno de los sensores electrónicos implementados; sensor presión de inyección de fluido, sensor presión de anular Tubing – Coiled Tubing y sensor caudal de inyección de fluido es óptima ya que permite monitorear el comportamiento de dichas variables durante la simulación de éstas operaciones y brinda la posibilidad a la empresa de mejorar dichos procesos con unidades Coiled Tubing, ya que a través de éste se les permite a sus trabajadores (Técnicos e Ingenieros) poseer una idea más clara y concreta a parte de la adquirida con la experiencia laboral y académica de la operación a realizar en Campo a sus clientes, y así generar rápidas soluciones y respuestas en tales procesos; lo anterior teniendo en cuenta la concientización e importancia de utilizar en todo momento de trabajo los implementos de seguridad HSEQ.



## RECOMENDACIONES

- Para reducir la deterioración o daño de las piezas elaboradas en material de impresión 3D (Plástico ABS) no exponga la Superficie del Prototipo (Locación Petrolera) a derrames de líquidos, y no coloque o deje caer objetos llenos con líquidos sobre éste a parte de los que están diseñados dentro del mismo para soportar o transmitir fluidos, tales como el FracTank y Arme sobre Cabezal de Pozo.
- Fíjese que la Cajetilla de Formaciones de Arenas está elaborada en material Acrílico y contiene arenas dentro de la misma, lo cual la hace más pesada y por ende debe introducirse o retirarse sin ocasionar movimientos bruscos a ésta en caso de ser requerido removerla.
- Asegúrese de mantener siempre fuera de contacto con líquidos las salidas de los Sensores y demás dispositivos electrónicos.
- No conecte la Bomba Sumergible a la toma corriente mientras esté fuera del agua. Nunca deje funcionar en seco éste aparato, Ni deje que el nivel de agua del tanque esté más bajo que la parte alta de la Bomba, puesto que dañaría el impulsor y se recalentaría demasiado con peligro de ignición.
- Evitar el contacto corporal directo con los pines de la Tarjeta Arduino Leonardo (ubicada en la caja negra detrás de la Cajetilla de Formaciones) debido a las descargas electrostáticas que puedan dañar el dispositivo.
- Para que el sistema en conjunto tenga un desempeño satisfactorio y permanezca en buen estado, es necesario realizar el mantenimiento preventivo para cada uno de los instrumentos del Prototipo Simulador (Sensores, Bomba Sumergible).
- Siendo un proyecto novedoso en el ámbito educativo-ingeniería y con posibles puertas a la investigación se propone finalmente continuar con la implementación de múltiples funciones al prototipo a medida que avanza la tecnología para realizar trabajos de estimulación a pozos petroleros con el fin de fortalecer la industria quien va de la mano a través de excelentes capacitaciones e implementos educativos.

## BIBLIOGRAFIA

- *Android. Características y especificaciones actuales.*  
<http://es.wikipedia.org/wiki/Android>
- *Android. Diseño y Desarrollo.* <http://es.wikipedia.org/wiki/Android>
- *App Inventor en Español. Conceptos.* Antonio Ricoy Riego.  
<https://sites.google.com/site/appinventormegusta/conceptos>
- *Arduino Leonardo.* <http://arduino.cc/en/Main/arduinoBoardLeonardo>
- *CREUS SOLÉ, A. (1997). Instrumentación Industrial: Turbinas.* México D.F. ALFAOMEGA GRUPO EDITOR, S.A de C.V. 6 e (pág 150).
- *Cuencas Sedimentarias de Colombia.*  
<http://es.slideshare.net/macorca123/cuencas-sedimentarias-de-colombiadoc-1>
- *El Medio Ambiente Desarrollo de Código Abierto para NET.*  
<http://www.icsharpcode.net/OpenSource/SD/Default.aspx>  
<http://es.wikipedia.org/wiki/SharpDevelop>
- *Facultad Regional Santa Fe - UTN. Tecnologías Básicas. Electrónica y Sistemas de Control. Documento CONTROLADORES ON – OFF.*  
[http://www.frfsf.utn.edu.ar/matero/visitante/bajar\\_apunte.php?id\\_catedra=156&id\\_apunte=4276](http://www.frfsf.utn.edu.ar/matero/visitante/bajar_apunte.php?id_catedra=156&id_apunte=4276)
- [http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic\\_3b.pdf](http://arduino.cc/en/uploads/Main/arduino-leonardo-schematic_3b.pdf)
- <http://www.oilproduction.net/files/coiledtubing-sanantonio.pdf>
- *Medidor de flujo liquido de 1/2".* <http://tienda.tdrobotica.co/producto/394>
- *Oilfield Review. Las presiones de las operaciones de perforación y producción.*  
[http://www.slb.com/~media/Files/resources/oilfield\\_review/spanish05/win05/p26\\_47.pdf](http://www.slb.com/~media/Files/resources/oilfield_review/spanish05/win05/p26_47.pdf)

- *Oilfield Review. Sistemas Integrados de Limpieza de Pozos: Mejoramiento de la eficiencia y Reducción del riesgo.*  
*file:///C:/Users/ACER/Desktop/01\_integrated\_wellbore\_cleanout.pdf*
- *Rolovic R, Weng X, Hill S, Robinson G, Zemlak k y Najafov J: “An Integrated System Approach to Wellbore Cleanouts with Coiled Tubing”, artículo de la SPE 89333, presentado en la conferencia y Exhibición sobre Tubería Flexible de las SPE/ICoTA, The Woodlands, Texas, EUA, 23 al 24 de marzo de 2004.*
- *Sistemas Industriales de Equipo de Bombeo.*  
*<http://www.sideb.com.mx/pagina/bombas-para-agua-sumergibles/>*
- *Solid Works.* *<http://es.wikipedia.org/wiki/SolidWorks>*
- *ZedGraph.* *<http://zedgraph.dariowiz.com/>*



## ANEXOS

### **ANEXO A. Código completo desarrollado en el Software Sharep Develop**

Este documento se encuentra en el CD que acompaña este libro

### **ANEXO B. Código completo desarrollado en el Software Arduino**

Este documento se encuentra en el CD que acompaña este libro