



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 1

Neiva, 21 de marzo de 2017

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Kevin Steven Ortiz Fernández, con C.C. No.1075254197,

Alexandra Manios Rojas, con C.C. No. 1075276367,

Autores del trabajo de grado titulado: Plataforma móvil para adquisición de datos meteorológicos controlada remotamente por medio de una aplicación Android

Presentado y aprobado en el año 2017 como requisito para optar al título de ingeniero electrónico;

Autorizamos al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

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

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

EL AUTOR/ESTUDIANTE:

EL AUTOR/ESTUDIANTE:

Firma: Kevin Steven Ortiz

Firma: Alexandra

Vigilada Mineducación



TÍTULO COMPLETO DEL TRABAJO: Plataforma móvil para adquisición de datos meteorológicos controlada remotamente por medio de una aplicación Android

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Ortiz Fernández	Kevin Steven
Manios Rojas	Alexandra

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Molina Mosquera	Johan Julián

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
----------------------------	--------------------------

PARA OPTAR AL TÍTULO DE: Ingeniero electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Electrónica

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2017

NÚMERO DE PÁGINAS: 85

TIPO DE ILUSTRACIONES (Marcar con una X):

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

Vigilada mieducación

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



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

MATERIAL ANEXO:

Anexo E. Archivos.php

Anexo F. Archivos.py

Anexo G. Codigo_Arduino

Anexo H. Servicio.java

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

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>
1. plataforma	platform
2. móvil	mobile
3. Android	Android
4. aplicación	application
5. meteorológicos	meteorological

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El proyecto plataforma móvil para adquisición de datos meteorológicos controlada remotamente por medio de una aplicación para dispositivos Android es un sistema cuya función principal es realizar un monitoreo constante de distintas variables climáticas entre las cuales se encuentran: temperatura, humedad, presión atmosférica, velocidad del viento, radiación UV; y otras como latitud, longitud, temperatura y humedad del suelo. Dicho monitoreo se realiza a través de una aplicación móvil Android que brinda además diversas prestaciones entre las cuales se encuentra la transmisión de video, base de datos y graficas de cada una de las variables.

Los sensores y demás componentes se han puesto sobre una plataforma robótica, que por sus características de movilidad y tamaño reducido la hacen fácil de manipular y transportar; basta con iniciar la aplicación y dando algunas órdenes estará puesta en marcha. El control de la plataforma al igual que el de cada uno de los sensores y demás funciones del sistema se realiza a través de la aplicación móvil dentro o fuera de la red local.



ABSTRACT: (Máximo 250 palabras)

The mobile platform project for acquisition of meteorological data remotely controlled by an application for Android devices is a system whose principal function is do a constant monitoring of different weather variables among which are: temperature, humidity, atmospheric pressure, wind speed, UV radiation; and others such as latitude, longitude, temperature and soil moisture. This monitoring is done through an Android application that offers some services between which are video streaming, database and graphics of each of the variables.

The sensors and other components are placed on a robotic platform, which by their mobility characteristics and small size is easy to handle and transport; simply start the application and with some orders is initiated. The control of the platform, sensors and other system functions are executed through the mobile application inside or outside the local network.

APROBACION DE LA TESIS

Nombre Jurado: Faiber Ignacio Robayo

Firma:

Nombre Jurado: José de Jesus Salgado

Firma:

**PLATAFORMA MÓVIL PARA ADQUISICIÓN DE DATOS METEOROLÓGICOS
CONTROLADA REMOTAMENTE POR MEDIO DE UNA APLICACIÓN ANDROID**

**ALEXANDRA MANIOS ROJAS
KEVIN STEVEN ORTIZ FERNANDEZ**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA ELECTRONICA
NEIVA - HUILA
2016**

**PLATAFORMA MÓVIL PARA ADQUISICIÓN DE DATOS METEOROLÓGICOS
CONTROLADA REMOTAMENTE POR MEDIO DE UNA APLICACIÓN ANDROID**

**ALEXANDRA MANIOS ROJAS
KEVIN STEVEN ORTIZ FERNANDEZ**

**Trabajo de grado para optar al título de
INGENIERO ELECTRÓNICO**

**Director:
Ing. Johan Julián Molina Mosquera**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERIA ELECTRONICA
NEIVA - HUILA
2016**

Nota de aceptación:

Firma del Director del Proyecto

Firma del primer jurado

Firma del segundo jurado

DEDICATORIA

A Dios por darme la sabiduría necesaria para culminar esta etapa de mi vida y alcanzar cada una de las metas que me he propuesto.

A mis padres, Luis Ángel Manios y Adela Rojas quienes son mi mayor motivación y apoyo, que sin importar cuántas veces me equivoque, nunca han dejado de creer en mí como persona y ahora como profesional.

Alexandra Manios Rojas

A mis padres, Edgar Ortiz y Carmen Fernández por su apoyo y comprensión durante toda mi etapa como estudiante y a mi hermano Juan Esteban Ortiz por ser mi compañía y ejemplo de fortaleza.

Kevin Steven Ortiz Fernández

AGRADECIMIENTOS

Agradezco a Dios por darme las fuerzas para superar cada uno de los obstáculos que se presentaron en el transcurso de mi vida universitaria y por poner en mi camino personas importantes para finalizar esta etapa.

A mis padres por brindarme su amor y ser un apoyo incondicional en cada momento de mi vida aun cuando he sentido desfallecer; y a mis hermanos que de una u otra manera han contribuido a mi formación.

Al Ing. Johan Julián Molina por su acompañamiento al dirigir este proyecto y a los ingenieros Faiber Robayo y José Salgado por haber aportado su juicio para el mejoramiento de nuestro trabajo de grado.

A la universidad Surcolombiana y cada uno de sus docentes que me brindaron su conocimiento para contribuir a mi formación personal y profesional; a mis compañeros por compartir junto a mí esta gran etapa, especialmente a mi compañero Kevin Ortiz por su amistad y por el gran trabajo en equipo que realizamos.

Alexandra Manios Rojas

Agradezco a Dios por darme vida, salud y conocimiento para permitirme cumplir con este y cada uno de mis objetivos, a mis padres y hermano por motivarme y darme fortaleza cada vez que fue necesario.

A cada uno de los profesores e ingenieros de la Universidad Surcolombiana, que fueron guías y apoyo en mi vida como universitario, en especial al ingeniero Julián molina, quien fue el encargado de dirigir este proyecto de grado.

A compañeros de universidad y amigos, que de alguna manera contribuyeron con sus consejos y experiencias para que pudiera llegar hasta este punto de mi vida.

A la universidad en general por permitir que me formara como profesional.

Kevin Steven Ortiz Fernández

CONTENIDO

INTRODUCCION	15
1. OBJETIVOS.....	17
1.1 OBJETIVO GENERAL	17
1.2 OBJETIVOS ESPECÍFICOS	17
2. MARCO TEORICO.....	18
2.1 RASPBERRY PI.....	18
2.1.1 ¿Qué es Raspberry pi?	18
2.1.2 Raspbian	18
2.1.3 Raspberry pi 2 modelo B	19
2.1.4 Modulo Camera Raspberry pi.....	20
2.2 ARDUINO.....	21
¿Qué es Arduino?	21
2.2.1 Características	21
2.2.2 Arduino Mega 2560	22
2.3 ACTOBOTICS KIT - NOMAD 4WD OFF-ROAD CHASSIS.....	23
2.4 MODEM MIFI SIERRA AIRCARD 763S.....	24
2.5 MOTORES	25
2.5.1 Motor Paso a Paso.....	25
2.5.2 Motor DC	25
2.6 SERVIDOR.....	26
2.6.1 Servidor LAMP	27
2.6.2 Hamachi	28
2.7 ANDROID.....	29
2.7.1 Service	29
2.7.2 SQLite	30
2.7.3 Androidplot	31

3. DESCRIPCIÓN DEL SISTEMA.....	32
4. SENSADO DE VARIABLES	35
4.1 TEMPERATURA Y HUMEDAD DEL AMBIENTE.....	35
4.2 RADIACION ULTRAVIOLETA.....	35
4.3 PRESIÓN ATMOSFÉRICA	37
4.4 VELOCIDAD DEL VIENTO	37
4.5 LATITUD Y LONGITUD.....	38
4.6 TEMPERATURA Y HUMEDAD DEL SUELO.....	39
Soporte del sensor de temperatura y humedad del suelo.	40
4.7 DISTANCIA	42
5. SERVIDOR.....	44
5.1 ENLACE TCP CON EL CLIENTE	44
5.1.1 Red Local	44
5.1.2 Red VPN	45
5.2 ARDUINO Y RASPBERRY PI.....	47
5.2.1 Comunicación Arduino y Raspberry pi	48
5.2.2 Envío de datos al cliente	48
5.3 CONTROL DE LOS MOTORES DE LA PLATAFORMA	49
5.4 ENVÍO DE VIDEO: MJPGSTREAMER	52
5.5 BASE DE DATOS	54
6. CLIENTE	55
6.1 MODELO VISTA CONTROLADOR (MVC)	55
6.2 VISTA DE INICIO	56
6.3 VISTA PARA EL CONTROL DE LA PLATAFORMA.....	57
6.4 VISTA PARA TALADRAR SUELO	58
6.5 VISTA PARA LA RECEPCIÓN DE LOS DATOS METEOROLÓGICOS.....	60
6.6 VISTAS HISTORIAL DE DATOS	63
6.7 VISTAS DE LOS GRÁFICOS.....	64
6.8 VISTA RECORRIDO AUTONOMO	66
7. RESULTADOS.....	68

7.1 PRUEBA DENTRO DE LA RED LOCAL.....	69
7.2 PRUEBA DESDE UNA RED REMOTA.....	71
6.4 VALIDACION DE LOS DATOS MEDIDOS POR LOS SENSORES.....	74
8. CONCLUSIONES.....	79
9. RECOMENDACIONES	80
BIBLIOGRAFÍA	81
ANEXOS	83

LISTA DE FIGURAS

Figura 1. Raspberry pi.....	19
Figura 2. Módulo Cámara.....	20
Figura 3. Arduinio Mega 2560.....	22
Figura 4. Características Arduino Mega.....	23
Figura 5. Platafoma Robótica.....	23
Figura 6. Modem.....	24
Figura 7. Motor DC.....	26
Figura 8. Diagrama de bloques del sistema.....	32
Figura 9. Diagrama de caso de usos.....	33
Figura 10. DHT11 y su conexión.....	35
Figura 11. Vol/Radiación UV.....	36
Figura 12. ML8511 y su conexión.....	36
Figura 13. BMP180 y su conexión.....	37
Figura 14. Conexión anemómetro.....	38
Figura 15. GPS Venus y su conexión.....	38
Figura 16. Rango Tem/Hum.....	39
Figura 17. Conexión SHT10 y su conexión.....	39
Figura 18. Motor Paso a Paso y engranaje.....	40
Figura 19. Imagen 3D del montaje para el sensor del suelo.....	41
Figura 20. Dimensiones y pieza real para el montaje del sensor del suelo.....	41
Figura 21. Dimensión y broca real.....	42
Figura 22. Motoreductor.....	42
Figura 23. HCSR04 y su conexión.....	43
Figura 24. Registro Hamachi y creación Red VPN.....	45
Figura 25. Clientes VPN.....	46
Figura 26. VPN Dispositivo Móvil.....	47
Figura 27. Comunicación Serial.....	48
Figura 28. Diagrama envío datos al cliente.....	49
Figura 29. Circuito Giro Motores.....	50
Figura 30. Diagrama Dirección de Motores.....	51
Figura 31. Video en el navegador.....	53
Figura 32. Base de Datos Servidor.....	54
Figura 33. Modelo Vista Controlador.....	55
Figura 34. Vista Inicio.....	56
Figura 35. D. Clases Inicio.....	57
Figura 36. Vista Dirección de motores y Diagrama de clases.....	58

Figura 37. Vista Taladrar Suelo.....	59
Figura 38. Pieza Mecánica Sensor Invasivo	59
Figura 39. D. Clases Taladrar Suelo	60
Figura 40. Vista Adquisición de Datos.....	60
Figura 41. D. Clases de la Vista de Datos.....	61
Figura 42. Vista Historial de Datos	63
Figura 43. D. Clases Base de Datos	64
Figura 44. Gráfico de Variables.....	65
Figura 45. Vista Recorrido Autónomo y selección de ruta.....	66
Figura 46. D. Clases Recorrido Autónomo.....	66
Figura 47. Plataforma Meteorológica	68
Figura 48. Aplicación Móvil (Kampla)	68
Figura 49. Prueba1 Velocidad del Viento	69
Figura 50. Prueba1 Presión.....	70
Figura 51. Prueba1 Radiación UV.....	70
Figura 52. Prueba1 Temperatura	71
Figura 53. Conexión Red VPN	72
Figura 54. Prueba2 Temperatura	72
Figura 55. Prueba2 Presión.....	73
Figura 56. Prueba2 Velocidad del Viento	73
Figura 57. Caseta termoregistradora y plataforma robótica	74
Figura 58. Datos registrados en el Servidor	75
Figura 59. Primera medida anemómetros	76
Figura 60. Segunda medida anemómetros	76

LISTA DE ANEXOS

Anexo A. Conexión Motores DC de la plataforma

Anexo B. Conexión Motor DC de la broca

Anexo C. Conexión motor paso a paso

Anexo D. fotografías

Anexo E. Archivos.php

Anexo F. Archivos.py

Anexo G. Código_Arduino

Anexo H. Servicio.java

GLOSARIO

HTML: (*HyperText Markup Language*). Lenguaje de marcas de hipertexto. Es un lenguaje para la elaboración de páginas web.

HTTP: (*Hypertext Transfer Protocol*). Protocolo de transferencia de hipertexto.

IP: (*Internet Protocol*). Protocolo de comunicación de datos digitales.

IPv4: (*Internet Protocol version 4*). Es la cuarta versión del protocolo IP, el cual usa direcciones de 32 bits.

I2C: (*Inter-Integrated Circuit*). Protocolo de circuito integrado. Es un protocolo que permite comunicar múltiples circuitos integrados digitales "esclavos" con uno o más chips "maestro".

JPG: (*Joint Photographic Experts Group*). Formato de imagen más común, utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen.

LAMP: (*Linux Apache MySQL PHP*).

PHP: (*Pre Hypertext -processor*). Lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

PWM: (*pulse-width modulation*). La modulación por ancho de pulsos es una técnica en la que se modifica el ciclo de trabajo de una señal periódica.

SOCKET: Mecanismo que permite la conexión entre distintos procesos, habitualmente se utilizan para establecer comunicaciones entre distintas máquinas que estén conectadas a través de la red.

TCP: (*Transmission Control Protocol*). El protocolo de control de transmisión, es uno de los protocolos fundamentales de la capa de transporte del modelo TCP/IP.

VPN: (*Virtual Private Network*). Red Privada Virtual. Es una tecnología de red que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet.

RESUMEN

El proyecto plataforma móvil para adquisición de datos meteorológicos controlada remotamente por medio de una aplicación para dispositivos Android es un sistema cuya función principal es realizar un monitoreo constante de distintas variables climáticas entre las cuales se encuentran: temperatura, humedad, presión atmosférica, velocidad del viento, radiación UV; y otras como latitud, longitud, temperatura y humedad del suelo. Dicho monitoreo se realiza a través de una aplicación móvil Android que brinda además diversas prestaciones entre las cuales se encuentra la transmisión de video, base de datos y graficas de cada una de las variables.

Los sensores y demás componentes se han puesto sobre una plataforma robótica, que por sus características de movilidad y tamaño reducido la hacen fácil de manipular y transportar; basta con iniciar la aplicación y dando algunas órdenes estará puesta en marcha. El control de la plataforma al igual que el de cada uno de los sensores y demás funciones del sistema se realiza a través de la aplicación móvil dentro o fuera de la red local.

Palabras claves: *plataforma, móvil, Android, aplicación, meteorológicos.*

ABSTRACT

The mobile platform project for acquisition of meteorological data remotely controlled by an application for Android devices is a system whose principal function is do a constant monitoring of different weather variables among which are: temperature, humidity, atmospheric pressure, wind speed, UV radiation; and others such as latitude, longitude, temperature and soil moisture. This monitoring is done through an Android application that offers some services between which are video streaming, database and graphics of each of the variables.

The sensors and other components are placed on a robotic platform, which by their mobility characteristics and small size is easy to handle and transport; simply start the application and with some orders is initiated. The control of the platform, sensors and other system functions are executed through the mobile application inside or outside the local network.

Keywords: platform, mobile, Android, application, meteorological.

INTRODUCCION

Colombia por su posición geográfica no presenta el cambio de las cuatro estaciones pero si variedad de climas y eventos meteorológicos, que con el pasar de los años son más impredecibles; por ser un país en vía de desarrollo, su capacidad para responder ante estas eventualidades es limitada. El constante avance de la tecnología no es ajeno a dichos acontecimientos de ahí que se hayan implementado estaciones meteorológicas en distintos lugares.

Según datos del IDEAM¹, de los 37 municipios que conforman el departamento del Huila, son aproximadamente 19 los que cuentan con una o más estaciones meteorológicas, lo cual es una falencia, pues aproximadamente el 50% de los municipios permanecen a la expectativa de las alteraciones de las variables meteorológicas sin poder prever los cambios y así tomar las precauciones pertinentes, para que la población y la economía sea lo menos afectada posible.

Las estaciones meteorológicas son la mejor alternativa para analizar el comportamiento del clima y su influencia sobre la economía, la salud y demás aspectos donde este desempeña un papel importante. El presente proyecto surge como respuesta a dicha necesidad, brindando no solo la capacidad de tomar las distintas mediciones, ya que también ofrece la alternativa de consultar los cambios que se hayan producido durante la toma de datos.

El sistema está diseñado sobre software y hardware libre en su totalidad, aprovechando las ventajas en cuanto a costos y manejo de licencias para su instalación e implementación. Cuenta con variedad de sensores entre los cuales se encuentran de temperatura, humedad, velocidad del viento, presión atmosférica, Radiación UV, altitud, longitud, temperatura y humedad del suelo; los datos serán almacenados en la memoria del teléfono móvil y en el servidor que en este caso será la tarjeta Raspberry pi.

La base es una plataforma robótica que por sus características de tamaño y robustez puede explorar en diferentes terrenos, incluso en los de difícil acceso. El control de dicha plataforma y el sistema en general es realizado a través de una aplicación móvil para dispositivos Android, que cuenta con diferentes vistas entre las cuales están las

¹ IDEAM. *Catalogo de estaciones climatologicas por departamento*. Neiva, Huila 2012.

gráficas de cada una de las variables, la base de datos y la transmisión del video. Todo lo expuesto anteriormente se realiza dentro o fuera de la red local.

Para finalizar el trabajo de grado se realizaron distintas pruebas de funcionamiento, las cuales están expuestas en el presente documento.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Implementar una estación meteorológica sobre una plataforma robótica, capaz de realizar una completa medición de las principales variables ambientales, además de una aplicación móvil para la recepción de datos y el control de la plataforma.

1.2 OBJETIVOS ESPECÍFICOS

Diseñar el control de velocidad para los cuatro motores de la plataforma robótica.

Adaptar a la tarjeta controladora los sensores a implementar para la medición de variables ambientales (temperatura, humedad, velocidad del viento, presión, Radiación UV, latitud y longitud).

Implantar una cámara en la tarjeta Raspberry pi para la transmisión de video de los lugares que recorre la estación móvil.

Instalar sobre la plataforma el sistema de medición de las variables ambientales.

Diseñar una aplicación móvil Android en el IDE de Eclipse para el control remoto de la plataforma, adquisición y visualización de los datos.

Crear una base de datos en la aplicación móvil para el almacenamiento de las variables ambientales y brindar la opción de ser enviadas a un servidor web.

2. MARCO TEORICO

2.1 RASPBERRY PI

2.1.1 ¿Qué es Raspberry pi?. Es una computadora personal pequeña, de bajo costo. A pesar de que carece de la capacidad de expansión de memoria y no puede dar cabida a dispositivos como CD, DVD y discos duros, tiene todo lo que un ordenador personal sencillo requiere. Es decir, tiene dos puertos USB, un puerto Ethernet, HDMI (y compuestos) de vídeo, e incluso un conector de audio para el sonido.

La tarjeta Raspberry Pi tiene una unidad SD que se puede utilizar para arrancar el equipo en cualquiera de varios sistemas operativos Linux. Todo lo que necesita es un monitor HDMI (o DVI con un adaptador de HDMI a DVI), un teclado, un ratón USB y una fuente de alimentación de 5V, y ya está en marcha.²

2.1.2 Raspbian. Es un sistema operativo libre basado en Debian, optimizado para el hardware de Raspberry Pi. Un sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione su Raspberry Pi. Sin embargo, Raspbian ofrece más que un SO puro; viene con más de 35.000 paquetes, software precompilado en un formato que hace más fácil la instalación en su Raspberry Pi.

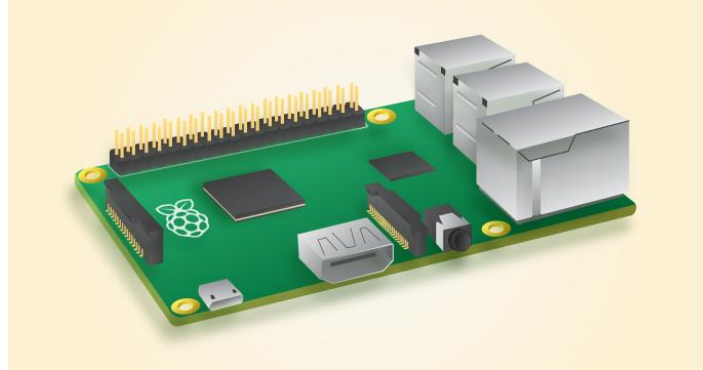
La construcción inicial de más de 35.000 paquetes de Raspbian, optimizado para un mejor rendimiento en el Raspberry Pi, se completó en junio de 2012. Sin embargo, Raspbian todavía está en desarrollo activo con un énfasis en la mejora de la estabilidad y la realización de cuantos paquetes de Debian como sea posible.³

² Bell, Charles. Beginning Sensor Networks with Arduino and Raspberry Pi. Nueva York , 2013.

³ "Welcome to Raspbian", [En línea] [Citado el: 16 de Agosto de 2016.] Disponible en la web: <https://www.raspbian.org/>.

2.1.3 Raspberry pi 2 modelo B.

Figura 1. Raspberry pi



Fuente: Raspberrypi.org

La Raspberry Pi 2 Modelo B (Figura 1) es la segunda generación de Raspberry Pi. Esta sustituye la Raspberry Pi 1 Modelo B+ de febrero de 2015. En comparación con el Raspberry Pi 1 cuenta con:

- Un 900MHz CPU de cuatro núcleos ARM Cortex-A7
- 1 GB de RAM

Al igual que el (Pi 1) Modelo B +, también tiene:

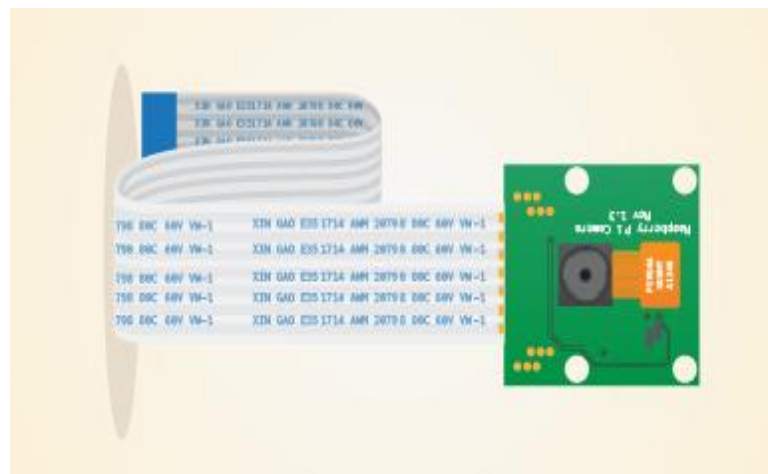
- 4 puertos USB
- 40 pines GPIO
- Puerto lleno de HDMI
- Puerto Ethernet
- Conector de audio de 3,5 mm combinado y vídeo compuesto
- Interfaz de la cámara (CSI)
- Interfaz de pantalla (DSI)
- Ranura para tarjetas micro SD
- VideoCore IV 3D núcleo de gráficos

Debido a que tiene un procesador ARMv7, se puede ejecutar toda la gama de distribuciones ARM GNU / Linux, incluyendo Snappy Ubuntu Core, así como Microsoft Windows 10.

La Raspberry Pi 2 tiene un factor de forma idéntica a la anterior (Pi 1) Modelo B + y tiene una compatibilidad completa Raspberry Pi 1.⁴

2.1.4 Módulo Camera Raspberry pi.

Figura 2. Módulo Cámara



Fuente: Raspberrypi.org

El módulo V1 de Raspberry pi de 5 MP es capaz de capturar vídeo a 1080p (1920x1080) y también imágenes una vez conectada a una placa Raspberry Pi. Para utilizarla, simplemente necesita ser conectada directamente con su cable ribbon al conector SCI (Camera Serial Interfaz) de la Raspberry Pi, cargar la última versión de Raspbian y una vez reiniciada la Pi, se debe activar el soporte de vídeo con raspiconfig; la cámara soporta formato de imagen JPEG, JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888 y de video Raw h.264.

⁴ "Raspberry Pi 2 Model B". [En línea] [Citado el: 30 de Julio de 2016.] Disponible en la web: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.

La placa es de tamaño reducido, 25x24x9 mm y pesa tan solo 3 gramos lo que la hace perfecta para aplicaciones móviles o cualquier otra aplicación donde el tamaño sea importante.⁵

2.2 ARDUINO

¿Qué es Arduino? Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexible y fácil de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.).⁶

2.2.1 Características.

- **Bajo costo:** Las placas Arduino son relativamente económicas comparadas con otras plataformas microcontroladoras.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux.
- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, y más provechoso para usuarios avanzados.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++.

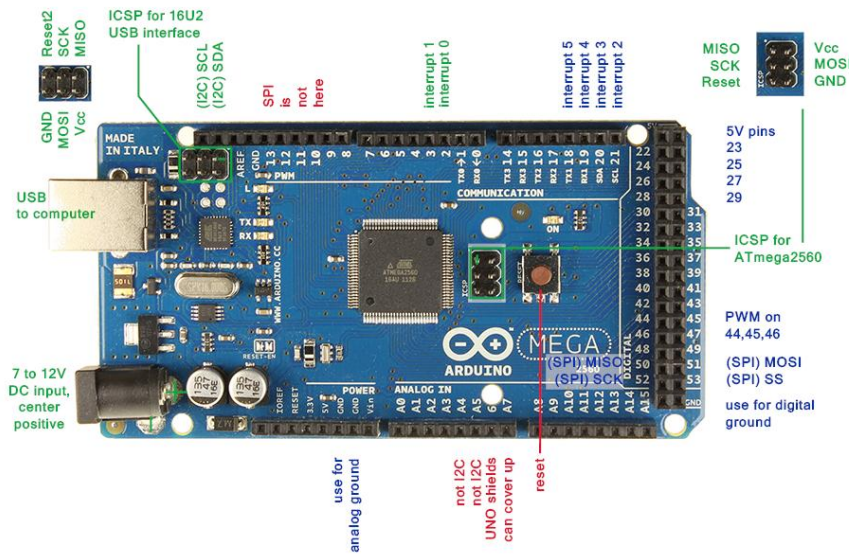
⁵ "Raspberry Pi 2 Model B". [En línea] [Citado el: 30 de Julio de 2016.] Disponible en la web: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.

⁶ "¿Qué es Arduino?". [En línea] [Citado el: 20 de Agosto de 2016.] Disponible en la web: <http://arduino.cl/que-es-arduino/>.

- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo.

2.2.2 Arduino Mega 2560.

Figura 3. Arduino Mega 2560



Fuente: Arduino.org

El Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada/salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 puertos serie, un oscilador de 16MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio.⁷

Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la corriente con un adaptador de CA a CC o una batería para empezar. Las especificaciones técnicas se presentan en la siguiente imagen:

⁷ "Arduino Mega 2560 y Genuino Mega 2560". [En línea] [Citado el: 20 de Agosto de 2016.] Disponible en la web: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>.

Figura 4. Características Arduino Mega

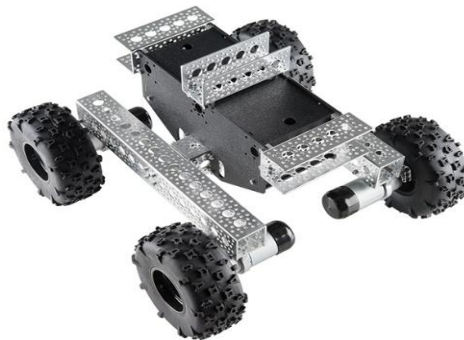
Technical specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Fuente: Arduino.org

2.3 ACTOBOTICS KIT - NOMAD 4WD OFF-ROAD CHASSIS

Figura 5. Plataforma Robótica



Fuente: Sparkfun.com

Este chasis está equipado con cuatro ruedas para carga pesada de 5 pulgadas de diámetro por 2,25 pulgadas de ancho, cuatro motorreductores planetarios con engranajes completamente de metal. También se incluye en el kit un cuerpo central de plástico ABS con dos grandes paneles de acceso que se abren a una cavidad lo suficientemente grande para agregar toda la electrónica cómodamente.⁸

El cuerpo central también tiene recortes en la parte inferior para pasar los cables de motor hacia el interior, sus grandes llantas y la mayoría de partes en aluminio hacen que sea una plataforma resistente a la hora de incursionar en distintos terrenos, y con ello que sea una excelente elección para cumplir con los objetivos del presente proyecto.

2.4 MODEM MIFI SIERRA AIRCARD 763S

Figura 6. Modem



Fuente: Store4G.com

AirCard 763s puede crear un punto de acceso móvil para hasta 10 dispositivos Wi-Fi, con velocidades de descarga de hasta 100 Mbps y velocidades de carga de hasta 50 Mbps. Además el modem mifi posee GPS y dispone de pantalla LCD con lo cual se hace más fácil la gestión de dispositivos.⁹

⁸ "Actobotics Kit - Nomad 4WD Off-Road Chassis". [En línea] [Citado el: 22 de Agosto de 2016.] Disponible en la web: <https://www.sparkfun.com/products/13141>.

⁹ "Sierra Wireless Aircard 763S Hotspot". [En línea] [Citado el: 26 de Agosto de 2016.] Disponible en la web: <http://www.store4g.com/sierra-aircard-763s/>.

Algunas de sus características son:

- LTE/HSPA+/HSPA/UMTS/EDGE/GPRS/GSM
- Ranura para tarjeta de memoria SD
- 2 puertos para antena externa conector TS9
- Pantalla 1.7" LCD
- Dimensiones: 9.75 cm Largo * 5.7 cm Alto y 1.57 cm de ancho
- Peso: 108 gramos con batería

2.5 MOTORES

2.5.1 Motor Paso a Paso. La principal característica de este motor es que puede posicionar su eje en una determinada posición de giro; además, es posible tener un control muy preciso de su velocidad de giro.

El estator de estos motores está constituido por varios electroimanes y el rotor por uno o varios imanes permanentes. A las diferencias de bobinas del estator se alimenta mediante impulsos, proporcionados normalmente por un circuito electrónico, y se consigue que el rotor se posicione paso a paso según sea el avance de dichos impulsos.

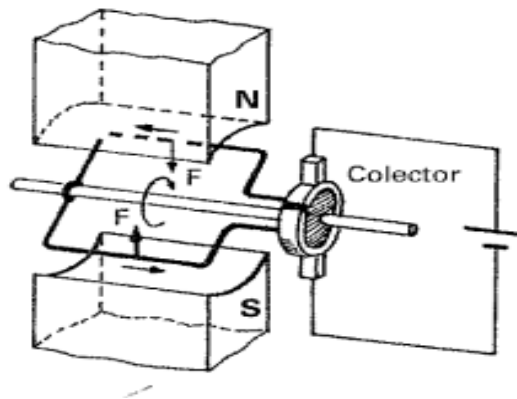
La velocidad de giro del rotor depende de la frecuencia de los impulsos y del número de polos. Se fabrican motores con diferentes ángulos de paso, como, por ejemplo, 24 pasos por revolución, 28, 96, 200, etc. De igual forma que si un motor paso a paso posee 24 pasos, su ángulo de paso será igual a: $360^\circ/24 = 15^\circ$.¹⁰

2.5.2 Motor DC. Como todos los motores eléctricos, su funcionamiento se basa en las fuerzas que aparecen en los conductores cuando son recorridos por corrientes eléctricas y, a su vez, están sometidos a la acción de un campo magnético.

Estos motores necesitan de tres partes fundamentales para su funcionamiento: un circuito que produzca el campo magnético (circuito inductor), un circuito que al ser recorrido por la corriente eléctrica desarrolle pares de fuerza que pongan en movimiento el rotor (circuito inducido) y un colector de delgas con escobillas.

¹⁰ San Miguel, Pablo A. 2014. Electrotecnia. Madrid : Paraninfo, 2014.

Figura 7. Motor DC



Fuente: Electrotecnia. San Miguel, Pablo Alcalde

En la figura se ha representado el aspecto de un motor de corriente continua elemental. Los polos magnéticos del imán, situados siempre en el estator, son los encargados de producir siempre el campo magnético inductor. La espira, que se ha situado en el rotor, es recorrida por una corriente continua que suministra a través de un anillo de cobre cortado por la mitad (colector de delgas).¹¹

2.6 SERVIDOR

Un servidor, en el lenguaje informático, es un ordenador y sus programas, que está al servicio de otros ordenadores. El servidor atiende y responde a las peticiones que le hacen los otros ordenadores. Los otros ordenadores, que le hacen peticiones, serán los "clientes" del servidor.

Precisamente se llaman servidores porque sirven cosas y están al servicio de otros ordenadores. Por ejemplo si se tiene un correo electrónico, lo recibe de un servidor de correo electrónico, si se desea ver una página web, la recibe de un servidor web y así entro otros muchos servicios. El modelo o arquitectura que siguen los servidores es el de cliente-servidor, es decir el cliente/s pide y el servidor proporciona los recursos o servicios.

¹¹ San Miguel, Pablo A. 2014. Electrotecnia. Madrid : Paraninfo, 2014.

2.6.1 Servidor LAMP. Hace referencia a un conjunto de subsistemas de software necesarios para alcanzar una solución general, en este caso configurar sitios Web o servidores dinámicos con un esfuerzo reducido.

En las tecnologías LAMP esto se consigue mediante la unión de las siguientes tecnologías:

- **Linux**, el sistema operativo, que es muy potente, estable, adaptable y robusto; Linux tiene una posición predominante en el ámbito web y su uso se está expandiendo cada vez más a los distintos ámbitos.
- **Apache**, el servidor Web. El Servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. De entre todos los servidores web disponibles Apache es sin duda de lo mejor, muy estable y el más utilizado por mucho tiempo.
- **MySQL**, el gestor de bases de datos. Otorgará a nuestras aplicaciones la capacidad de guardar y acceder a información en forma rápida y precisa. Se caracteriza por su velocidad, escalabilidad y fiabilidad.
- **Perl, PHP o Python**, los lenguajes de programación.

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

El gran parecido que posee PHP con los lenguajes más comunes de Programación estructurada, como Lenguaje de Programación C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.¹²

¹² "PHP". [En línea] [Citado el: 27 de Agosto de 2016.] Disponible en la web: <http://www.ecured.cu/PHP>.

Python es un lenguaje de programación creado por Guido van Rossum en el año 1991, la extensión es .py. Se compara habitualmente con Tcl, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de Código abierto, administrado por la Python Software Foundation.

Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa.¹³

LAMP está considerada como una de las mejores herramientas para desarrollar y ejecutar aplicaciones web, sus tecnologías componentes por separado son de altísima calidad y en conjunto sus virtudes se multiplican. A diferencia gran cantidad de otras tecnologías de uso similar, LAMP no tiene publicidad propia, no hay una corporación detrás de su desarrollo, sino una comunidad de entusiastas y programadores de primer nivel que trabajan con la calidad del producto como premisa. Tanto Linux como Apache, MySQL, PHP y Python son Software de código abierto por lo que si usted escucha hablar de LAMP será verdaderamente por sus virtudes, no por publicidad.

2.6.2 Hamachi. Se trata de una aplicación gratuita con la que se pueden crear redes locales virtuales sin el engorro de tener que configurar firewall y demás parámetros de conexión.

Mediante una conexión de IPs segura y siguiendo las estructuras de una red P2P, el programa establece la comunicación VPN entre dos o más ordenadores para poder realizar todo tipo de tareas como si estuvieran conectados localmente, desde jugar a videojuegos hasta disponer de los archivos situados en la carpeta compartida. Una vez se accede a Hamachi y se crea una cuenta, se asignará una dirección IP virtual que será el identificador cada vez que se ingrese al programa desde ese mismo dispositivo.

De esta forma, se puede establecer con plena seguridad la comunicación entre dos ordenadores sin limitaciones espaciales ni cables, solo se necesita este programa, que permitirá la creación de una sala a la que pueden conectarse hasta cinco usuarios en su versión gratuita.

¹³ "Python". [En línea] [Citado el: 27 de Agosto de 2016.] Disponible en la web: <http://www.ecured.cu/Python>.

2.7 ANDROID

Es la plataforma para móviles más popular, cientos de millones de dispositivos móviles en más de 190 países de todo el mundo. Es la mayor base instalada de cualquier plataforma móvil y de rápido crecimiento, cada día otro millón de usuarios da energía a sus dispositivos Android por primera vez y empiezan a buscar aplicaciones, juegos y otros contenidos digitales.

Android le da una plataforma de clase mundial para la creación de aplicaciones y juegos para los usuarios de Android en todas partes, así como un mercado abierto para la distribución de ellos al instante.

Al ser una plataforma libre, Android se ha convertido en un favorito de los consumidores y desarrolladores por igual, impulsando un fuerte crecimiento en el consumo de aplicaciones. Los usuarios de Android descargan miles de millones de aplicaciones y juegos de Google Play cada mes.

Android está empujando constantemente los límites de hardware y software hacia adelante, para traer nuevas capacidades para los usuarios y desarrolladores. Para los desarrolladores, la rápida evolución de la tecnología Android le permite mantenerse al frente con aplicaciones potentes y diferenciadas.

Android le da acceso a las últimas tecnologías e innovaciones a través de una multitud de factores de forma de dispositivos, arquitecturas de conjuntos de chips, y puntos de precio. Desde el procesamiento de múltiples núcleos y gráficos de alto rendimiento para sensores con tecnología de última generación, pantallas táctiles, vibrantes y emergentes tecnologías móviles.¹⁴

2.7.1 Service. Es un componente de la aplicación, que puede realizar operaciones de larga ejecución en segundo plano y no proporciona una interfaz de usuario. Otro componente puede iniciar el service para que se ejecute en segundo plano, incluso si el usuario cambia a otra vista o aplicación. Además, un componente puede vincularse a un service para interactuar con él e incluso realizar la comunicación entre procesos. Por ejemplo, un service puede manejar las transacciones de red, reproducir música,

¹⁴ "Android, la plataforma móvil mas popular del mundo". [En línea] [Citado el: 2 de Septiembre de 2016.] Disponible en la web: <https://developer.android.com/about/android.html>.

ejecutar archivos de E / S, o interactuar con un proveedor de contenidos, todo ello desde segundo plano.

Esencialmente un service puede tomar dos formas:

- **Started Service:** Un service es “started” cuando un componente de la aplicación lo inicia a través del método `startService()`. Una vez iniciado, un service puede ejecutarse en segundo plano indefinidamente, incluso si el componente que lo inicio se destruye.¹⁵
- **Bound Service:** Un service es “bound” cuando un componente de aplicación se une a él a través del método `bindService()`. Un bound service ofrece una interfaz cliente-servidor que contiene los siguientes componentes para interactuar con el service, envía solicitudes, obtiene respuestas, e incluso lo hacen a través de comunicación entre procesos. Múltiples componentes pueden estar ligados al service, pero cuando todos ellos ya no están ligados al service, este es destruido.¹⁶

2.7.2 SQLite: Es el nombre dado a un sistema de bases de datos (SBD) de código abierto cuya principal diferencia es que en lugar de ser un SBD estándar, está contenido en una biblioteca escrita en lenguaje C. un SBD clásico es un programa que crea y gestiona los datos de una base de datos al que se accede externamente por programas que requieren este tipo de datos. SQLite, sin embargo, no es un programa, sino una biblioteca de código que implementa una base de datos y es manipulada por el programa mediante llamadas a funciones.

SQLite no requiere instalación, basta con integrarlo de forma convencional como una biblioteca de código escrito para el programa, lo que permite incorporarlo al sistema. Esto es especialmente útil cuando se tiene que entregar sistemas con bases de datos pero no hay ninguna posibilidad de organizar un servidor de bases de datos para acceso.

¹⁵ "Service". [En línea] [Citado el: 3 de 10 de 2016.] Disponible en la web: <https://developer.android.com/guide/components/services.html?hl=es-419>.

¹⁶ "Service". [En línea] [Citado el: 3 de 10 de 2016.] Disponible en la web: <https://developer.android.com/guide/components/services.html?hl=es-419>.

Por estas razones y por su tamaño relativamente pequeño, SQLite se utiliza en la mayoría de las aplicaciones que requieren almacenar y acceder a datos de manera rápida, pero no pueden acceder a una base de datos externa (por varias razones, incluyendo la velocidad y la conveniencia) como: navegadores, clientes de correo electrónico, Smartphone (Android), los sitios web, los programas que necesitan almacenar datos temporales, etc. SQLite es un SBD simple para la enseñanza de bases de datos (1).

2.7.3 Androidplot. Es una librería para la creación de gráficos dinámicos y estáticos en aplicaciones para Android. Fue diseñada desde un principio para Android, compatible con su versión 1.6 en adelante y actualmente es utilizado por más de 1.000 aplicaciones en Google Play¹⁷. Esta librería es de código abierto y completamente libre; además se mantiene en constante desarrollo; algunos de los gráficos que se pueden realizar son los siguientes:

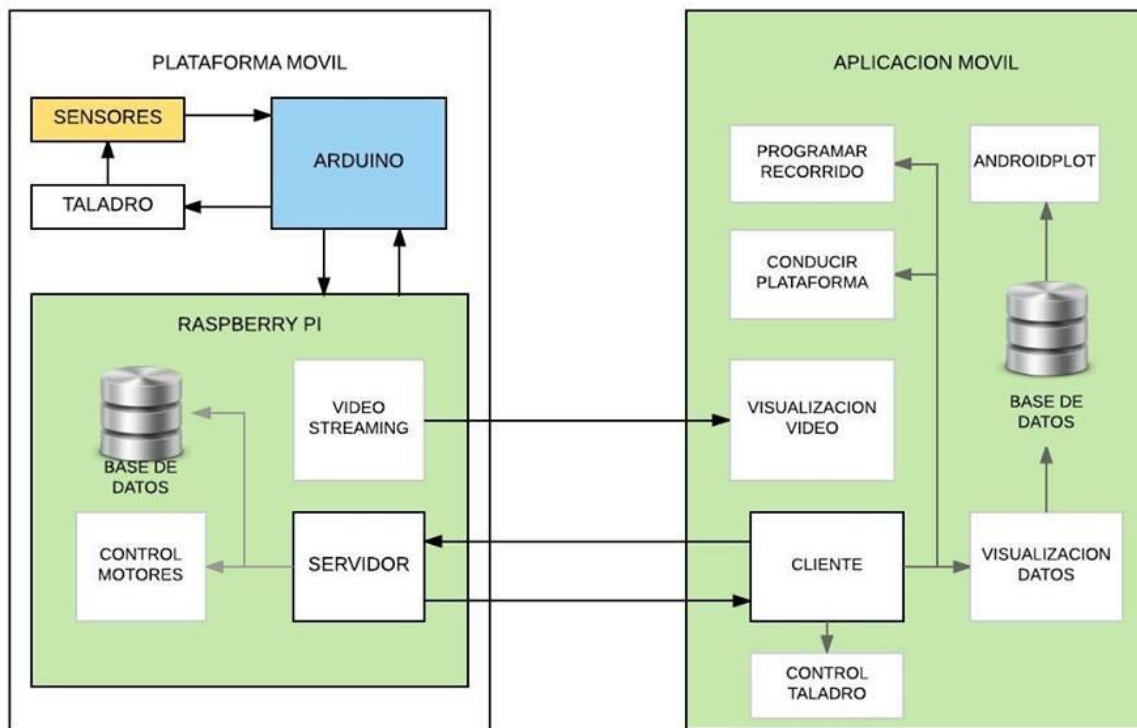
- Gráficos de líneas
- Gráficos de barras
- Gráficos circulares
- Gráficos de dispersión

¹⁷ "Acerca de androidplot". [En línea] [Citado el: 2 de septiembre de 2016.] Disponible en: <http://androidplot.com/about/>.

3. DESCRIPCIÓN DEL SISTEMA

El sistema consiste en una plataforma robótica, que además de realizar la medición de variables meteorológicas, transmite el video del lugar donde se encuentra ubicada, realiza el recorrido que se le ordene al instante y también efectuar un recorrido autónomo que ha sido previamente programado desde el cliente, que es una aplicación móvil llamada Kampla.

Figura 8. Diagrama de bloques del sistema



Fuente: Autor

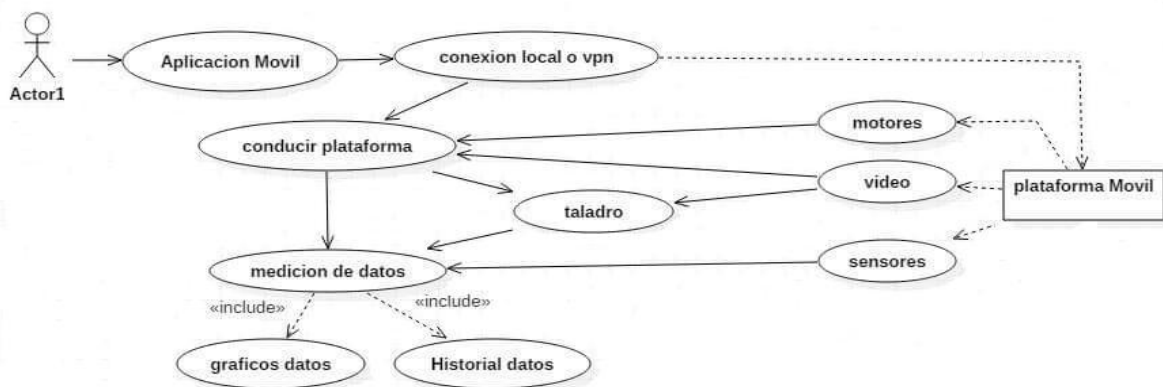
Entre los principales componentes de la plataforma meteorológica, se encuentran; una tarjeta Raspberry pi, una tarjeta micro controladora Arduino Mega, diversos sensores, un módem mifi, entre otros. La Raspberry pi controla la dirección y velocidad de los motores, además funciona como servidor, comunicándose con el cliente (aplicación móvil) mediante protocolo wifi, dentro o fuera de la red local; para lo cual, es necesario

que la plataforma cuente con un módem, en este caso un Mifi. Los sensores son conectados a la tarjeta microcontroladora Arduino Mega, la cual se comunica vía serial a la Raspberry pi, para recibir la orden de activar o desactivar los sensores y enviar la medición de cada una de las variables.

Los sensores fueron ubicados en la parte externa de la plataforma, de tal manera que estén en contacto directo con el ambiente, ya que en su mayoría, así lo requieren, solo hay que tener en cuenta algunas condiciones a la hora de realizar las distintas mediciones. Al momento de medir la velocidad del viento, la plataforma debe estar ubicada en un lugar donde haya yerba corta y lejos de edificios que puedan interrumpir el paso del viento y al igual para tomar mediciones de latitud y longitud, la antena del GPS debe estar a campo abierto de forma que pueda recibir la señal de los satélites.

Por otra parte, la aplicación móvil se diseñó en Eclipse, por ser un software libre y completo que satisface las necesidades del presente proyecto. La aplicación contiene diversas vistas, cada una con una función específica, dentro de las que se encuentran: la vista de inicio para seleccionar el tipo de red, pues el sistema ha sido diseñado para ser manipulado dentro de una red local o a través de una red VPN si se desea una conexión desde otra red, una vista secundaria desde la cual se controla la dirección y velocidad de la plataforma, además de visualizar el video. Desde esa vista secundaria se puede desplegar un menú con las siguientes opciones: recorrido autónomo de la plataforma, control de la pieza mecánica que contiene el sensor de temperatura y humedad del suelo, además de la visualización de los datos obtenidos de las variables meteorológicas.

Figura 9. Diagrama de caso de usos



Fuente: Autor

Durante o después de finalizadas las mediciones, el usuario puede revisar los datos de cada una de las variables meteorológicas en la misma aplicación, además de ello realizar las respectivas graficas donde se observa el comportamiento de las variables con respecto al tiempo; si se desea, se pueden eliminar los datos de la aplicación que también han sido guardados en el servidor y en la memoria del teléfono.

Cada uno de los puntos mencionados, se explican de manera más detallada en los siguientes capítulos del presente documento.

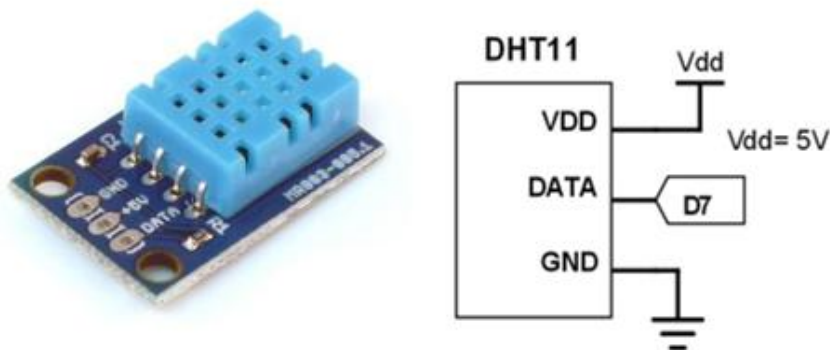
4. SENSADO DE VARIABLES

Se describe el proceso mediante el cual se adaptaron cada uno de los sensores a la tarjeta microcontroladora Arduino, encargada de realizar el control del hardware y enviar los datos medidos hacia el servidor es decir hacia la Raspberry pi. El servidor posteriormente envía dichos datos hacia el cliente, para su visualización, almacenamiento y análisis.

4.1 TEMPERATURA Y HUMEDAD DEL AMBIENTE

La medición de temperatura y humedad del ambiente se realiza con el sensor dht11, el cual posee una señal de salida digital. El sensor es de fácil aplicación, cuenta con tres pines; Vcc, tierra y un pin de datos que en este caso será conectado al D7 de la placa Arduino, mediante el cual se recibirá una señal de respuesta de 40 bits correspondientes datos de la humedad relativa y la temperatura ambiente.

Figura 10.DHT11 y su conexión



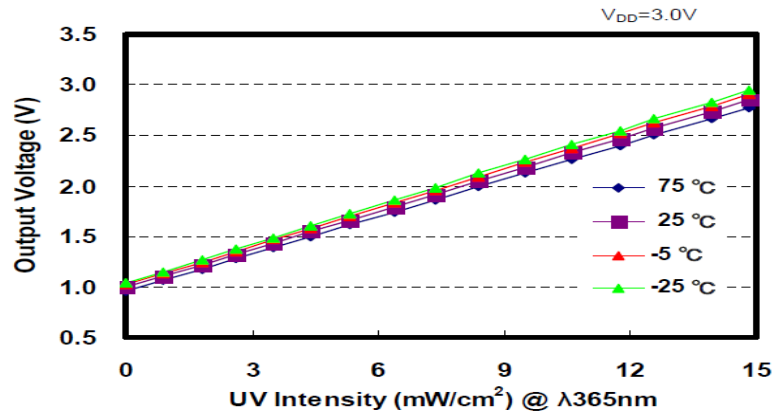
Fuente: microbot.it y autor

4.2 RADIACION ULTRAVIOLETA

El sensor ml8511 es el encargado de medir la radiación ultravioleta, este sensor trabaja con una señal analógica en relación con la cantidad de luz UV que

se detecta; puede ser muy útil en la creación de dispositivos que alerten al usuario de un índice alto de radiación UV, para evitar problemas en la piel o en los ojos.

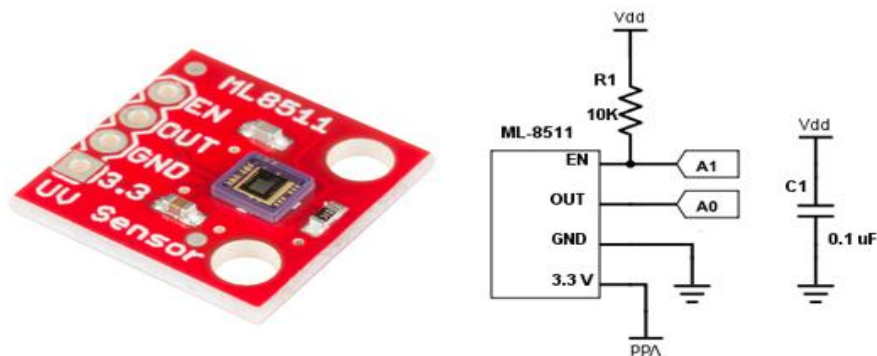
Figura 11. Vol/Radiación UV



Fuente: Sparkfun.com

Por medio del pin analógico cero (A1) de la tarjeta se envía la señal de activación del sensor, el cual responde con un voltaje proporcional a la radiación ultravioleta (ver Figura 12) por el pin A0 con una longitud de onda comprendida entre los 280 y 390nm, rango en el cual se encuentra comprendido tanto el espectro UV-B como el UV-A.

Figura 12. ML8511 y su conexión



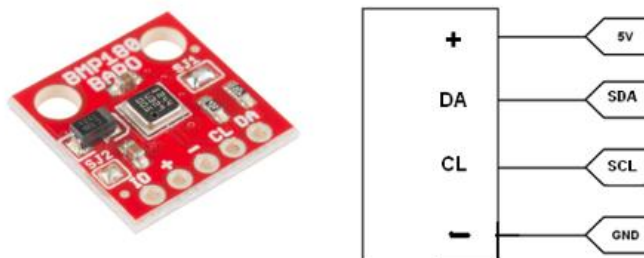
Fuente: Sparkfun.com y autor

4.3 PRESIÓN ATMOSFÉRICA

La medición de la presión atmosférica se realiza con el sensor BMP-180; el cual tiene salida digital y se comunica a través de la interfaz I2C con la tarjeta controladora; tiene gran precisión, y un amplio rango de medida (300 a 1100 hpa).

Los pines de datos y sincronización de reloj están conectados al pin SDA y SCL de la placa Arduino respectivamente. El BMP180 ofrece una precisión absoluta de hasta 0,03 hPa, se basa en la tecnología piezo-resistiva con robustez EMC, alta precisión y linealidad, así como con estabilidad a largo plazo; lo cual lo hace adecuado para la presente aplicación.

Figura 13. BMP180 y su conexión



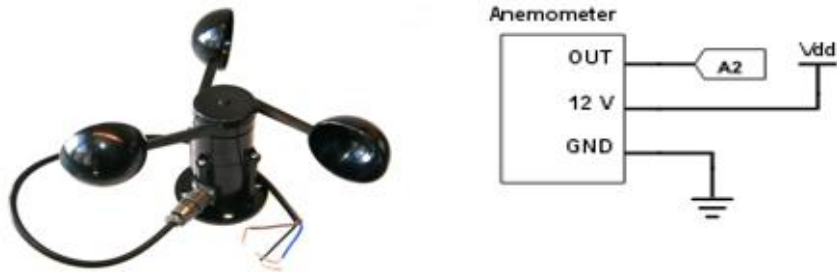
Fuente: Sparkfun.com y autor

4.4 VELOCIDAD DEL VIENTO

Para la medición de la velocidad del viento se hace uso de un anemómetro; el cual es un instrumento común e importante de una estación meteorológica. Para el presente proyecto se implementó un anemómetro diseñado para colocarse a la intemperie y medir la velocidad del viento con facilidad.

La señal analógica emitida es recibida a través del pin A2 de la tarjeta Arduino, en forma de una tensión que oscila entre 0.4V hasta 2.0V para un rango de 0 m/s a 32.4 m/s respectivamente.

Figura 14. Conexión anemómetro



Fuente: Adafruit.com y autor

4.5 LATITUD Y LONGITUD

Venus GPS board basado en el integrado Venus638FLPx es el encargado de realizar la medición de latitud y longitud; el cual posee gran sensibilidad y un conector SMA para antena externa. Para recibir la información, basta con alimentar la board y conectar el pin TX a uno receptor de la tarjeta (RX); en este caso al RX2 y ejecutar el código correspondiente.

Figura 15. GPS Venus y su conexión



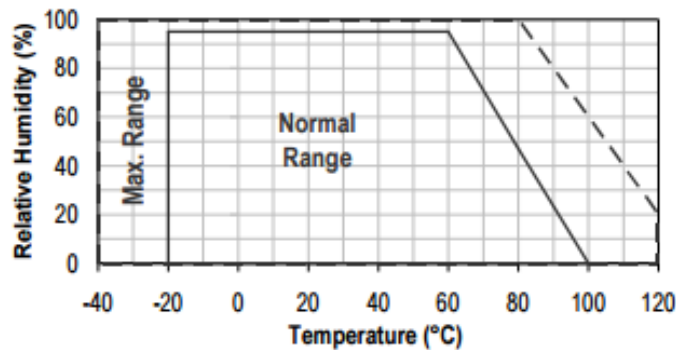
Fuente: Sparkfun.com y autor

Obteniendo los datos de latitud y longitud se puede estimar la ubicación del lugar donde se está realizando la exploración.

4.6 TEMPERATURA Y HUMEDAD DEL SUELO

Para medir la temperatura y humedad del suelo se utiliza el sensor digital SHT10 el cual está en una malla envolvente de metal sinterizado. La carcasa es resistente a la intemperie y evitará que el agua se filtre en el cuerpo del sensor y dañarlo. Posee una precisión de 4.5 % al momento de medir la humedad y de 0.5 °C al medir la temperatura. En la Figura 16 se puede observar que tiene un amplio rango de trabajo para cada una de las variables, característica que permite realizar mediciones en gran variedad de terrenos.

Figura 16. Rango Tem/Hum



Fuente: Adafruit.com

La señal de respuesta del sensor es de adquirida a través del pin digital D10 y la señal de reloj a través del pin D11 de la tarjeta Arduino, la cual ejecuta las respectivas líneas de código para su funcionamiento.

Figura 17. Conexión SHT10 y su conexión



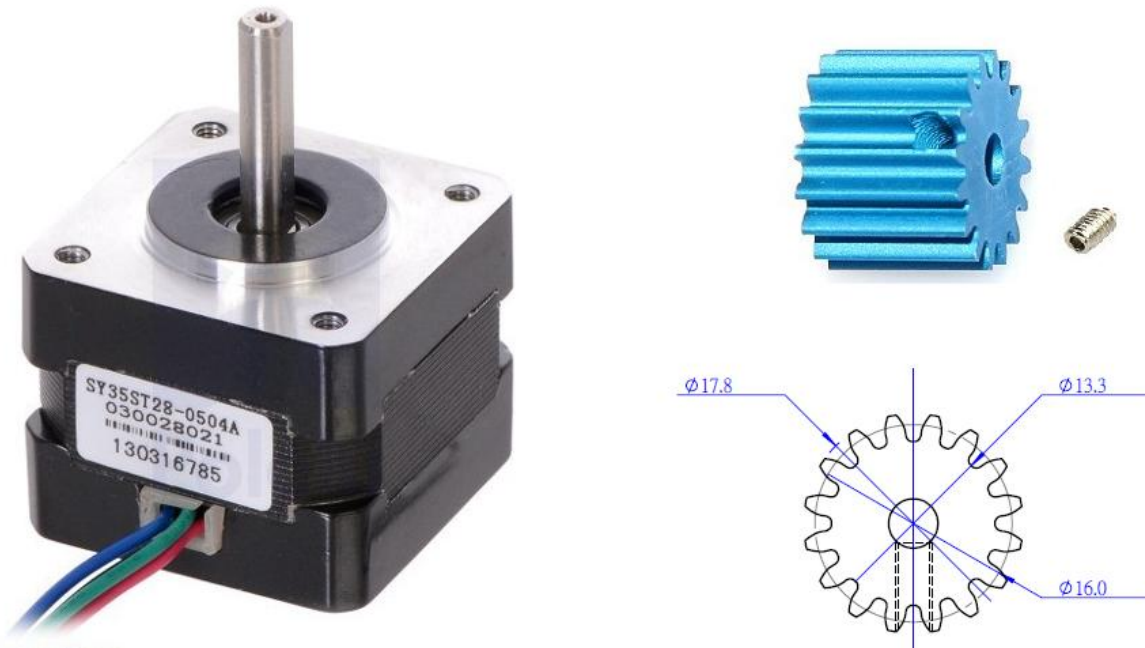
Fuente: Adafruit.com y autor

Soporte del sensor de temperatura y humedad del suelo.

Al momento de realizar la medición de temperatura y humedad del suelo es indispensable introducir el sensor en la tierra, debido a eso, fue necesario diseñar un sistema que realizara dicho proceso. Se trata de un sistema piñón cremallera, el cual se encarga de bajar hasta el suelo la pieza que contiene en su interior el sensor que mide dichas variables; Ya que, también es necesario abrir un hueco en el suelo, este artefacto lleva en la punta un motoreductor que hace girar una broca, diseñada de forma que fuese lo más similar posible, a las brocas que se usan para abrir orificios en la tierra.

Para bajar y subir toda la pieza, se hace uso de un motor paso a paso bipolar de 200 pasos por vuelta y un torque de 1kg por cm, que en su eje lleva un engranaje de aluminio con 16 dientes (Figura 18).

Figura 18. Motor Paso a Paso y engranaje



Fuente: tdrobotica.com

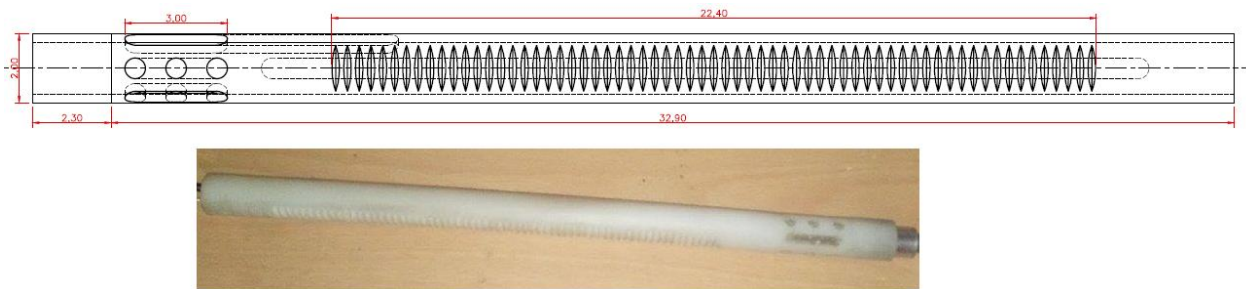
La pieza que lleva en su interior el sensor de temperatura y humedad del suelo (Figuras 19 Y 20), tiene orificios para que el sensor pueda realizar medidas más exactas, un poco más abajo va incrustado el motoreductor, y en la parte superior están las muescas que forman parte de la cremallera, mediante la cual, el engranaje efectúa el recorrido. La pieza tiene una longitud total de 35.2cm y la cremallera mide 22.4 cm, de forma que al momento de bajar toda la pieza, pueda quedar totalmente introducido el sensor en la tierra.

Figura 19. Imagen 3D del montaje para el sensor del suelo



Fuente: autor

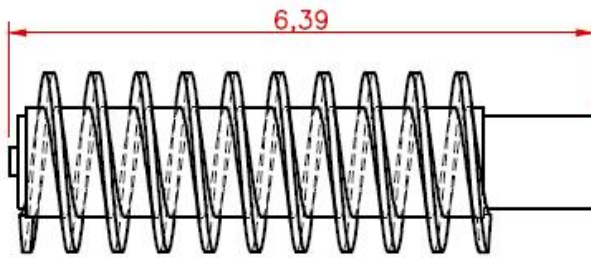
Figura 20. Dimensiones y pieza real para el montaje del sensor del suelo



Fuente: autor

El encargado de hacer girar la broca es un motoreductor de 16 mm de diámetro, que gira a 200 RPM y que tiene un torque máximo de 0.9Kg-cm, La broca está elaborada en aluminio y su diseño se basa en el de las brocas utilizadas para perforar la tierra.

Figura 21. Dimensión y broca real



Fuente: autor

Figura 22. Motoreductor



Fuente: ebay.com

4.7 DISTANCIA

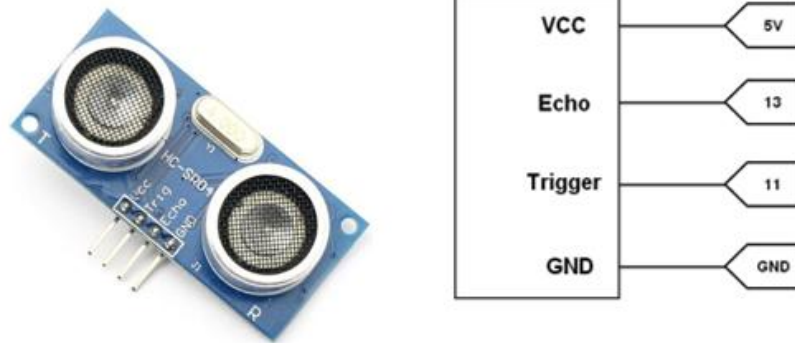
Por ser una plataforma móvil es indispensable contar con uno o más dispositivos para medir la distancia, los cuales alertan la presencia de algún obstáculo y poder impedir que la plataforma sea expuesta a daños por choque.

En el presente proyecto se utiliza un sensor de ultrasonido (HCSR04) ubicado en la parte delantera de la plataforma; su funcionamiento consiste en dos transceptores de ultrasonidos, uno emisor y otro receptor; donde la distancia es proporcional al tiempo

que tarda la señal en ir desde el emisor hasta el objeto obstáculo y volver rebotada desde éste hasta el receptor.

Este sensor tiene la particularidad de manejarse solo con dos hilos, aparte de los de alimentación. Por uno de ellos se le envía el pulso de disparo (trigger) con el que se activa el sensor y por el otro se recibe el pulso de eco (echo) cuya amplitud es directamente proporcional a la distancia a la que se ha detectado un objeto; ambas señales tanto la de envío y recepción están conectadas la tarjeta Raspberry mediante los pines 11 y 13 respectivamente, y desde allí se controla una distancia mínima (50 cm) referente a algún obstáculo para que la plataforma se detenga.

Figura 23.HCSR04 y su conexión



Fuente: Electronilab.co y autor

5. SERVIDOR

El servidor, en este caso la Raspberry pi, es el encargado de escuchar las peticiones de la aplicación Android para dar respuesta con un servicio dependiendo de la solicitud; entre los servicios prestados por el servidor, está la comunicación TCP con el cliente, el control de los motores de la plataforma, la transmisión de video vía streaming, la comunicación serial con la tarjeta Arduino y el posterior almacenamiento de la información en una base de datos MySQL.

Al encender el servidor, este se encarga de ejecutar el código para iniciar la comunicación automáticamente con el cliente, dependiendo del tipo de solicitud se inicia la tarea determinada.

5.1 ENLACE TCP CON EL CLIENTE

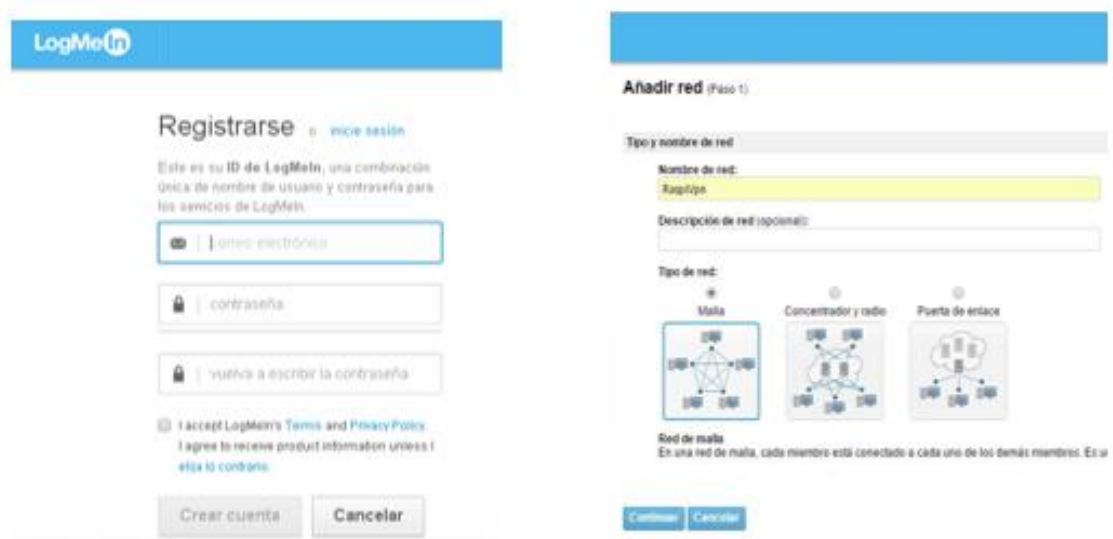
Al iniciar el servidor, éste se encuentra a la escucha de solicitudes de conexión por parte del cliente. Al ejecutar la aplicación cliente y elegir la ruta de comunicación, ya sea en la red local o red VPN; el cliente envía la solicitud de enlace con el servidor a su respectiva dirección IP. Luego de haber establecido la conexión, el cliente envía la solicitud al servidor, y si es el caso, espera su respuesta.

Para la creación de la red se utiliza un modem MiFi, el cual se encarga de establecer la comunicación entre el servidor y el cliente. Dependiendo de la cobertura del prestador de servicios de internet, la comunicación se puede realizar a través de una red externa o en una red local.

5.1.1 Red Local. para la comunicación entre el servidor y el cliente en una red local, se asigna una dirección IP fija al servidor, es decir, siempre que éste se conecte a cualquier red, toma la IP que se le ha asignado previamente; lo cual es una ventaja ya que no se requiere configurar una nueva dirección IP en el cliente cada vez que se produzca un cambio de red en el servidor.

5.1.2 Red VPN. La creación de la Red Privada Virtual se realizó a través de Hamachi, utilizando la versión gratuita con la cual se permite la conexión de hasta cinco usuarios, lo que es suficiente para satisfacer las necesidades del presente proyecto. Se crea la cuenta en **LogMeIn Hamachi** y luego de tener la cuenta activa, se procede a crear la nueva red en forma de malla (Figura 19). Esto permite que todos los equipos de la red se pueden comunicar entre sí.

Figura 24. Registro Hamachi y creación Red VPN



Fuente: Autor

Para la instalación y ejecución de hamachi en el servidor se ejecutan las siguientes líneas de código en el terminal de Raspbian:

Actualizar los repositorios

```
sudo apt-get update
```

Instalar paquetes LSB

```
sudo apt-get -y install --fix-missing lsb lsb-core
```

Descargar Hamachi

```
sudo wget https://www.vpn.net/linux-logmein-hamachi-2.1.0.165-armel.tgz
```

Abrir ubicación del instalador

```
cd logmein-hamachi-2.1.0.165-armel
```

Se instala Hamachi

```
Sudo ./install.sh
```

Se ejecuta Hamachi

```
Sudo /etc/init.d/logmein-hamachi start
```

Se inicia la sesión en Logmein-Hamachi

```
Sudo hamachi login
```

Ingresar usuario y contraseña

```
hamachi attach [Usuario]
```

```
hamachi set-nick [Contraseña]
```

Terminado este proceso, se envía una solicitud para agregar el servidor a la red creada en la cuenta hamachi; allí se unen los clientes con los cuales se desea realizar la conexión (Figura 20), a estos clientes se les asigna una dirección IPv4 virtual en el rango 25.x.x.x, la cual es exclusiva globalmente y se utiliza para acceder al cliente desde cualquier lugar a través de la red VPN.

Figura 25. Clientes VPN

Añadir red (Paso 3)

Red: RaspiVpn2

Seleccionar ordenadores para unir a esta red.

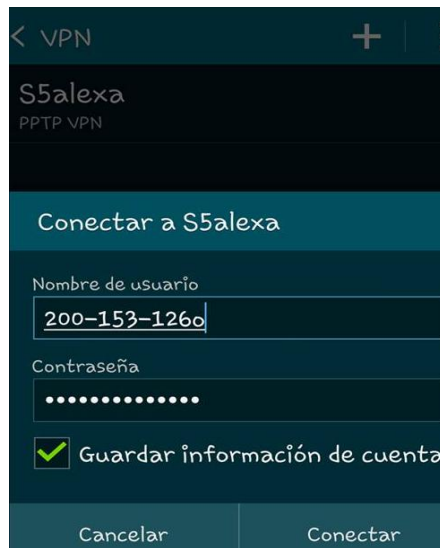
Los clientes de otras cuentas también pueden solicitar unirse a esta red. [Leer más](#)

Seleccionar miembros de red

Miembros	ID de cliente
<input type="checkbox"/> Alexa-PC	200-225-977
<input type="checkbox"/> ARsistemas	200-157-182
<input type="checkbox"/> LgKevin	200-152-964
<input type="checkbox"/> raspberry	200-152-469
<input type="checkbox"/> S5alexa	200-153-126

Fuente: Autor

Figura 26. VPN Dispositivo Móvil



Fuente: Autor

En el dispositivo móvil se añade la red VPN (Figura 21), y se activa para efectuar la conexión con los demás clientes de la red, en este caso con el servidor que también ha sido previamente agregado y asignado una dirección IP virtual.

5.2 ARDUINO Y RASPBERRY PI

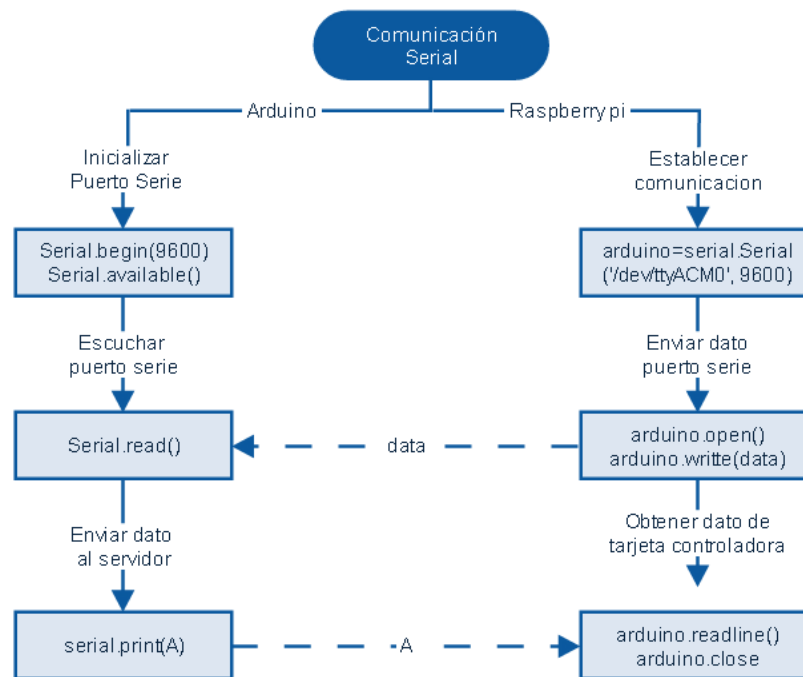
Arduino y Raspberry pi son dos plataformas muy completas con diversidad de aplicaciones. Aunque en algunas ocasiones ambas tarjetas se pueden utilizar para realizar un mismo proyecto; estas son completamente diferentes; la raspberry pi funciona como un computador con su propio sistema operativo, tiene la capacidad de realizar distintas tareas en segundo plano, comunicarse inalámbricamente a internet, entre otras cosas.

Por otra parte Arduino es un microcontrolador, es decir realiza una función específica, cuenta con entradas analógicas lo que es una ventaja pues no requiere de hardware adicional a la hora de implementar sensores que responden con señales análogas, como si debe hacerse al usar la raspberry pi.

Por sus prestaciones individuales ambas son implementadas en el presente proyecto, una como complemento de la otra; la arduino se encarga de activar, desactivar y recibir los datos de cada uno de los sensores mientras que la raspberry pi funciona como servidor, además de controlar la velocidad y funcionamiento de los motores.

5.2.1 Comunicación Arduino y Raspberry pi. La comunicación entre las dos tarjetas se realiza vía serial a través de un cable USB; el mismo que se utiliza para conectar la arduino al computador. Posteriormente se inicializa el puerto serial en arduino y se instalan las librerías correspondientes para el control del puerto serial en la Raspberry pi.

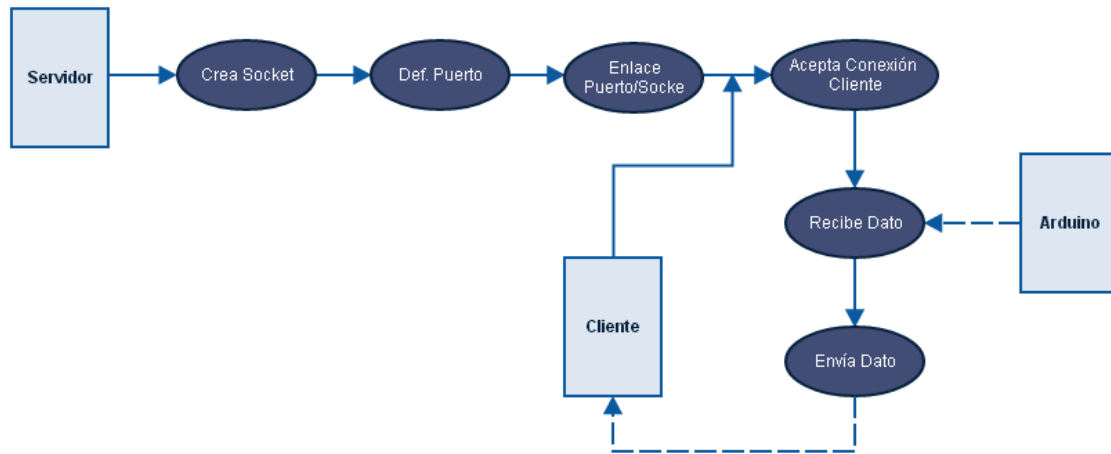
Figura 27. Comunicación Serial



Fuente: Autor

5.2.2 Envío de datos al cliente. La raspberry pi cuenta con un módulo WiFi, a través del cual se conecta con el cliente, ya sea a través de una red local o una red virtual privada. Para el envío de los datos se ejecuta la secuencia presentada en el siguiente diagrama de flujo.

Figura 28. Diagrama envío datos al cliente



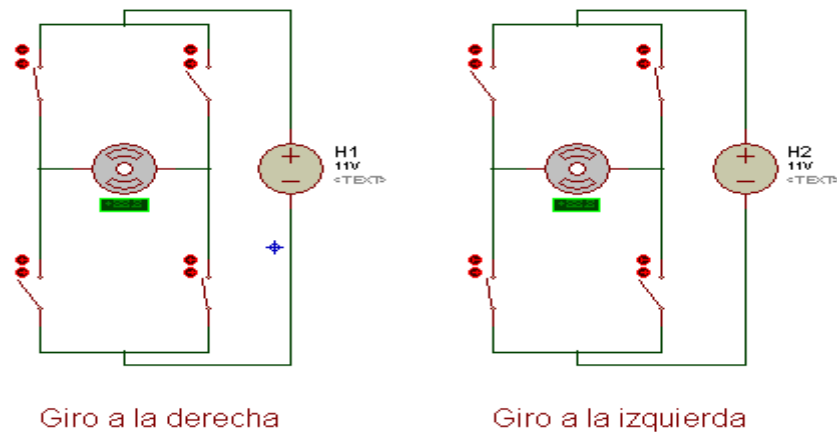
Fuente: Autor

5.3 CONTROL DE LOS MOTORES DE LA PLATAFORMA

La plataforma cuenta con cuatro motores, uno para cada rueda; los cuales funcionan a través de una orden que es enviada desde la aplicación móvil donde se tiene la opción de ir hacia adelante, atrás, giro a la derecha, izquierda y detener. El cliente envía un dato al servidor, el cual es evaluado para dar respuesta a la solicitud y ejecutar la función correspondiente.

El motor básicamente está diseñado para recibir una excitación la cual produce un movimiento del mismo; para atender a la solicitud del cliente en cuanto al sentido de giro, avance o detención de los motores, es necesario implementar una etapa adicional. Para dar una breve explicación de lo que se requiere, se hizo uso de los siguientes circuitos:

Figura 29. Circuito Giro Motores



Fuente: Autor

Como se puede observar los interruptores determinan el camino de la corriente y por lo tanto el sentido de giro del motor, teniendo en cuenta que para que se produzca movimiento debe haber una diferencia de potencial; si los dos interruptores superiores están cerrados y los dos inferiores abiertos o viceversa, no habrá diferencia de potencial y por lo tanto no se produce movimiento en el motor.

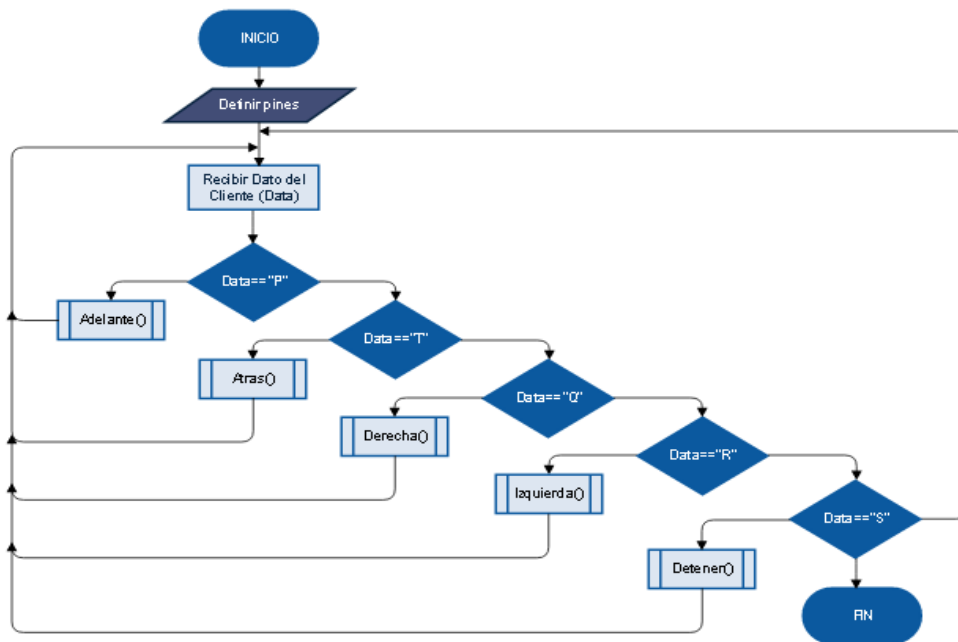
El dispositivo electrónico que realiza esta función se denomina puente H y se consigue como un circuito integrado que en vez de usar interruptores, consta de transistores de potencia operando en modo de corte como un interruptor abierto o en modo de saturación como un interruptor cerrado; al estar construido con transistores se puede controlar con señales de baja potencia, en nuestro caso las señales de habilitación serán suministradas con la tarjeta Raspberry pi. Dentro de la gran variedad de dispositivos que se encuentran en el mercado se hace uso del Full- Bridge L298N, con el cual se conectan dos motores y suministra la corriente necesaria para que estos funcionen a su máxima capacidad.

Los dos motores de adelante están conectados a un driver dual L298N y los motores de la parte trasera a otro, ya que al hacer la prueba de corriente de los motores, cada uno

mostró un consumo máximo de corriente de aproximadamente 2 Amperios y según las especificaciones técnicas del driver L298N¹⁸, este tiene un máximo de corriente de 4 A.

Cuando al servidor le llega la petición de poner en marcha los motores, la tarjeta Raspberry pi envía la orden para activar cada motor, además de una señal PWM que se encarga de controlar la velocidad de giro de cada motor; el diagrama muestra la sentencia de ejecución del código.

Figura 30. Diagrama Dirección de Motores



Fuente: Autor

¹⁸ STMicroelectronics. 2000. "L298 DUAL FULL-BRIDGE DRIVER". [En línea] enero de 2000. [Citado el: 15 de octubre de 2016.] Disponible en la web: http://www.st.com/content/st_com/en/products/motor-drivers/brushed-dc-motor-drivers/l298.html

5.4 ENVÍO DE VIDEO: MJPGSTREAMER

Para la transmisión de video en vivo se utiliza un módulo de cámara Raspberry Pi y la herramienta MJPG-Streamer; con la cual se obtienen frames JPG capturados desde la cámara y son transmitidos mediante el protocolo HTTP. El video puede ser visualizado en navegadores u otras herramientas; en este caso por medio de un WebView en la aplicación móvil.

Para el funcionamiento de la herramienta MJPG-Streamer se requirió de instalar las siguientes librerías en el servidor; por medio de las líneas de comando ejecutadas desde la terminal de Raspbian.

```
sudo apt-get install libjpeg8-dev
```

```
sudo apt-get install imagemagick
```

Para la descarga y compilación de MJPG-Streamer:

```
mkdir mjpg
```

```
cd mjpg
```

```
svncohttps://svn.code.sf.net/p/mjpg-streamer/code/mjpg-streamer/ mjpg-streamer
```

Una vez se han instalado las correspondientes librerías y compilado la herramienta de video; se procede a la ejecución en segundo plano, además de asignar la resolución (-r 420x320) y el puerto de comunicación (-p 8001).

```
./mjpg_streamer -i "/input_uvc.so -d /dev/video0 -y -r 420x320" -o "/output_http.so -w ./www -p 8001"
```

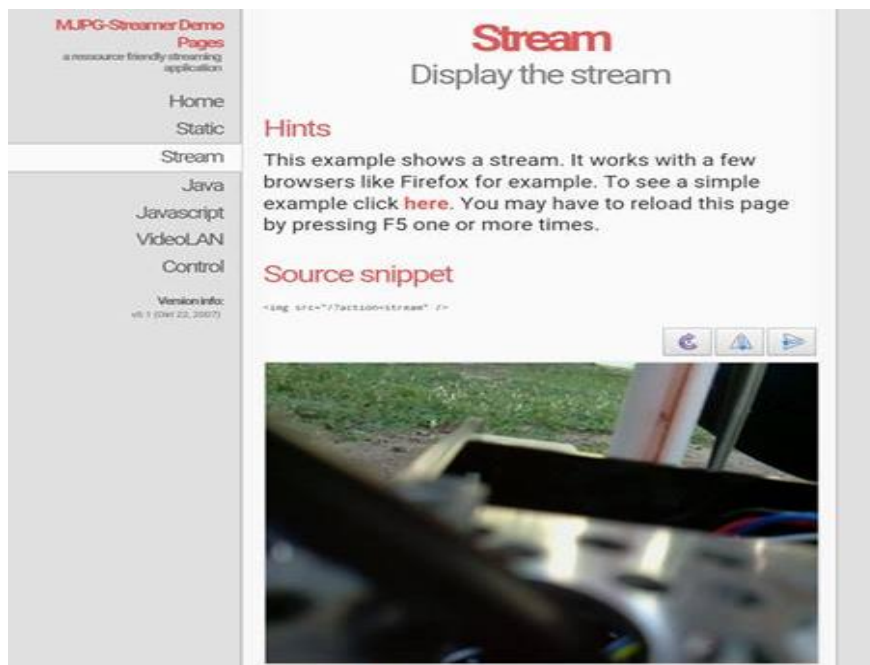
Con la línea de comando anterior, además de iniciar la herramienta, también se muestra en la terminal algunos datos del video y la transmisión.

```
MJPEG Streamer Version: svn rev: 3:172
```


i: Using V4L2 device.: /dev/video0	// Identifica que la cámara está conectada
i: Desired Resolution: 420 x 320	// Resolución del video: 420 x 320
i: Frames Per Second.: 5	// Se toman imágenes a razón de 5 FPS
i: Format.....: YUV	// Formato de color
i: JPEG Quality.....: 80	// Calidad de las imágenes en %
o: www-folder-path....: ./www/	// Directorio de la aplicación Web
o: HTTP TCP port.....: 8001	// Protocolo y Puerto de comunicación
o: username:password.: disabled	// Sin contraseña de acceso
o: commands.....: enabled	

Para visualizar el video desde el navegador en la red local, se accede a la dirección <http://192.168.1.50:8001> y desde una red remota a través de la VPN con la dirección <http://25.126.20.61:8001>; en el caso de la aplicación móvil, por medio del método LoadUrl se carga la dirección para observar el video en el WebView.

Figura 31. Video en el navegador



Fuente: Autor

5.5 BASE DE DATOS

Las mediciones de los sensores además de ser guardadas en la memoria del teléfono móvil, también serán registradas en una base de datos ubicada en un servidor Web. En este caso se instalan cada uno de los elementos que conforman un servidor LAMP; como sus siglas lo indican es una combinación de Linux, Apache, MySQL y PHP o Python; donde Linux es el sistema operativo sobre el que se trabaja, Apache es el servidor Web, MySQL es el gestor de base de datos y; PHP y Python son el lenguaje de programación. Para la instalación de los elementos se ejecutan en el terminal de la raspberry pi las siguientes líneas de comando:

```
sudo apt-get install apache2
```

```
sudo apt-get install php5
```

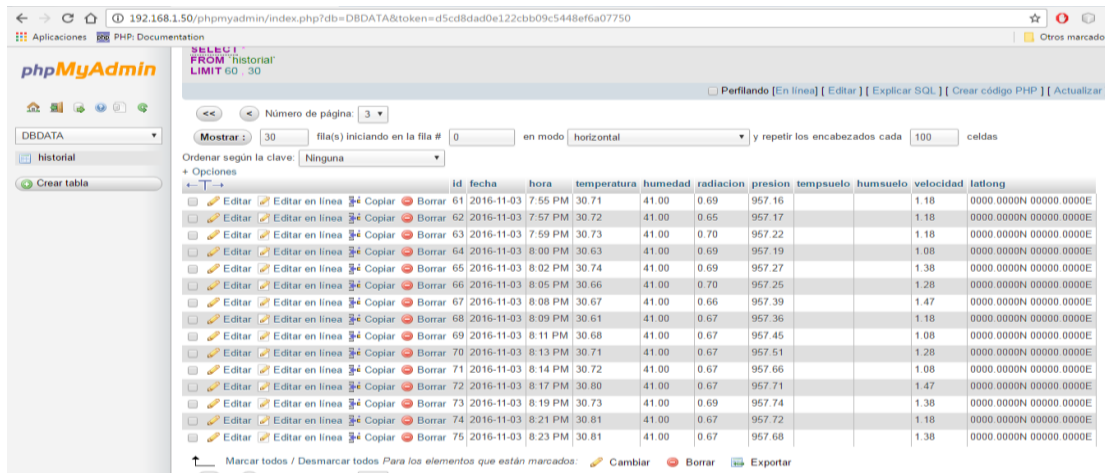
```
sudo apt-get install php5-mysql
```

```
sudo apt-get install mysql-server
```

```
sudo apt-get install phpmyadmin
```

Seguidamente se crea la base de datos en phpMyAdmin (DBDATA) y dentro de ella una tabla que se ha llamado historial, la cual está conformada por once columnas; una para cada variable a medir, además del id, la fecha y la hora como se observa en la Figura 27. En el servidor se ejecutan dos archivos php (ver Anexo C) que contienen las líneas de código necesarias para insertar los datos en la tabla.

Figura 32. Base de Datos Servidor



	id	fecha	hora	temperatura	humedad	radiacion	presion	tempsuelo	humsuelo	velocidad	latlong
Editar	61	2016-11-03	7:55 PM	30.71	41.00	0.69	957.16			1.18	0000.0000N.00000.0000E
Editar	62	2016-11-03	7:57 PM	30.72	41.00	0.65	957.17			1.18	0000.0000N.00000.0000E
Editar	63	2016-11-03	7:59 PM	30.73	41.00	0.70	957.22			1.18	0000.0000N.00000.0000E
Editar	64	2016-11-03	8:00 PM	30.63	41.00	0.69	957.19			1.08	0000.0000N.00000.0000E
Editar	65	2016-11-03	8:02 PM	30.74	41.00	0.69	957.27			1.38	0000.0000N.00000.0000E
Editar	66	2016-11-03	8:05 PM	30.66	41.00	0.70	957.25			1.28	0000.0000N.00000.0000E
Editar	67	2016-11-03	8:08 PM	30.67	41.00	0.66	957.39			1.47	0000.0000N.00000.0000E
Editar	68	2016-11-03	8:09 PM	30.61	41.00	0.67	957.36			1.18	0000.0000N.00000.0000E
Editar	69	2016-11-03	8:11 PM	30.68	41.00	0.67	957.45			1.08	0000.0000N.00000.0000E
Editar	70	2016-11-03	8:13 PM	30.71	41.00	0.67	957.51			1.28	0000.0000N.00000.0000E
Editar	71	2016-11-03	8:14 PM	30.72	41.00	0.67	957.66			1.08	0000.0000N.00000.0000E
Editar	72	2016-11-03	8:17 PM	30.80	41.00	0.67	957.71			1.47	0000.0000N.00000.0000E
Editar	73	2016-11-03	8:19 PM	30.73	41.00	0.69	957.74			1.38	0000.0000N.00000.0000E
Editar	74	2016-11-03	8:21 PM	30.81	41.00	0.67	957.72			1.18	0000.0000N.00000.0000E
Editar	75	2016-11-03	8:23 PM	30.81	41.00	0.67	957.68			1.38	0000.0000N.00000.0000E

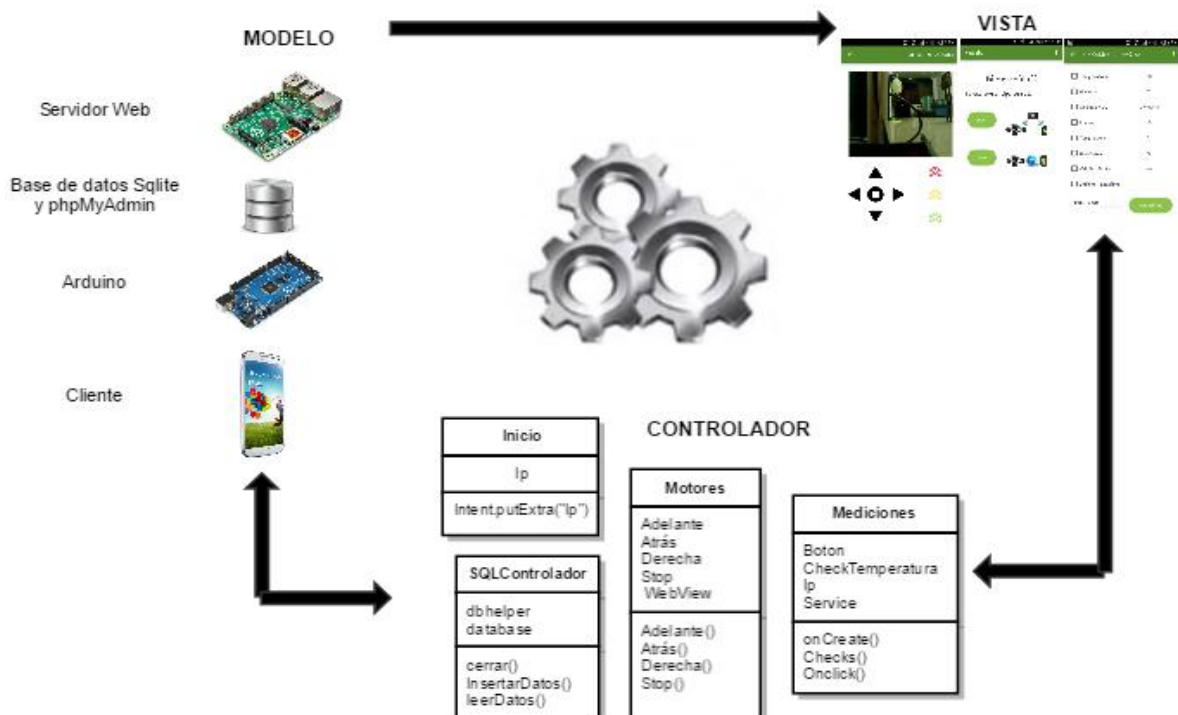
Fuente: Autor

6. CLIENTE

El cliente es una aplicación móvil (Kampla) diseñada en eclipse, mediante lenguaje java; capaz de establecer comunicación con el servidor mediante protocolo wifi en una red local o a través de internet por medio de una red VPN. La aplicación móvil cuenta con distintas vistas, cada una de ellas compuesta por una o varias clases, que influye de alguna manera en el funcionamiento del sistema.

6.1 MODELO VISTA CONTROLADOR (MVC)

Figura 33. Modelo Vista Controlador



Fuente: Autor

Con la estructura del modelo vista controlador se divide el sistema en tres componentes que interactúan entre sí. En el modelo se asignan las variables de entrada y salida tanto en el servidor (Raspberry pi) como en la tarjeta Arduino, se establece la conexión

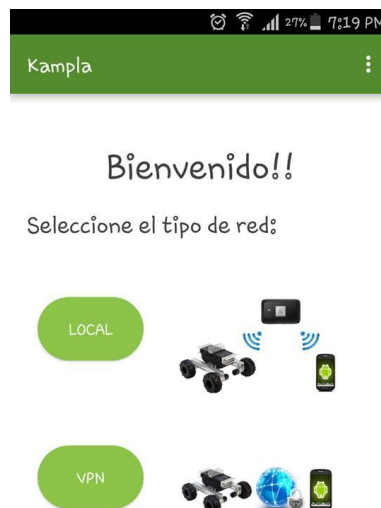
WIFI con el cliente y la comunicación serial entre las tarjetas, se accionan los motores y sensores, además se crean las bases de datos.

La aplicación Kampla tiene variedad de vistas, entre las cuales se encuentran: inicio, direccionamiento de la plataforma, adquisición de datos, grafica de cada una de las variables, entre otras; cada una de ellas con una interfaz amigable con el usuario lo que permite una fácil manipulación.

Por otra parte en el controlador se establecen las clases principales de la aplicación con sus respectivos métodos, los cuales ejecutan las distintas tareas tales como, conexión cliente servidor, envío y recepción de información, almacenamiento en las bases de datos y streaming de video.

6.2 VISTA DE INICIO

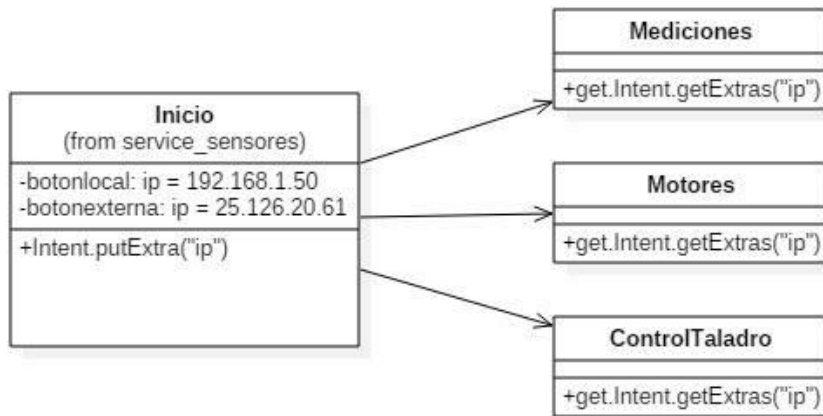
Figura 34. Vista Inicio



Fuente: Autor

En esta vista se encuentran las dos opciones de conexión; allí el usuario podrá seleccionar el tipo de red (Local o VPN) para realizar la comunicación entre el cliente y el servidor.

Figura 35. D. Clases Inicio



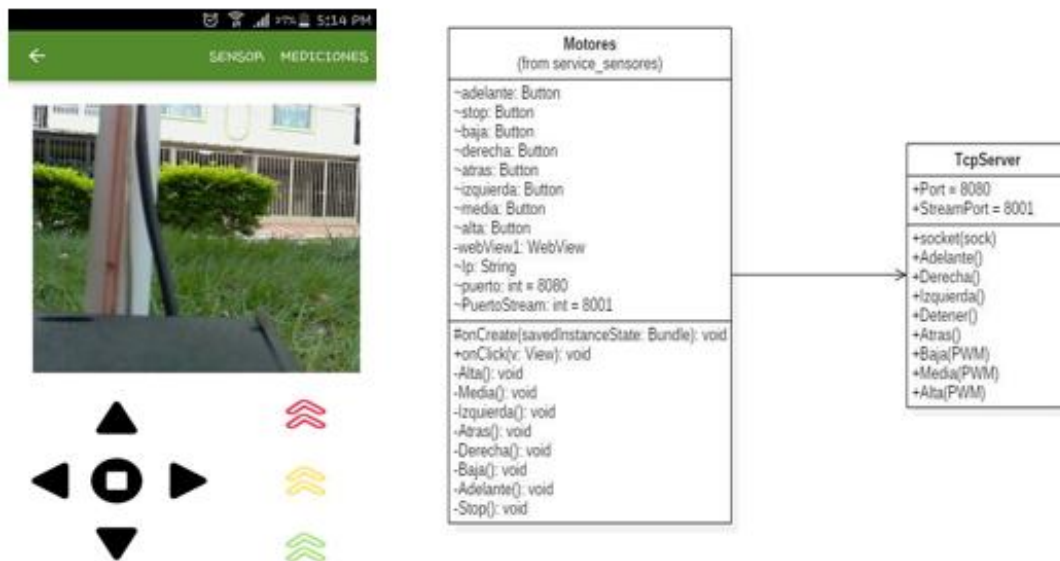
Fuente: Autor

La vista está conformada por dos botones; al presionar sobre el primero, el usuario indica que desea realizar una conexión de red local, con lo cual se envía a la siguiente vista la dirección IP fija 192.168.1.50 que se ha asignado previamente al servidor. Cuando el usuario presione sobre el segundo botón, se enviara a la siguiente vista la dirección IP 25.126.20.61, la cual le corresponde al servidor dentro de la red Virtual creada en Hamachi.

6.3 VISTA PARA EL CONTROL DE LA PLATAFORMA

Esta vista cuenta con un Webview mediante el cual se puede observar la transmisión de video que realiza el servidor a través de MJPG- streamer, además tiene ocho botones que corresponden a las direcciones y los tres niveles de velocidad.

Figura 36. Vista Dirección de motores y Diagrama de clases



Fuente: Autor

En esta vista se recibe la dirección IP que ha sido enviada desde la vista de inicio para realizar la comunicación TCP entre el cliente y el servidor. El cliente envía caracteres al servidor para realizar las distintas peticiones; dependiendo de la solicitud, el servidor activa los motores proporcionándole movimiento a la plataforma en distintas direcciones y velocidades.

6.4 VISTA PARA TALADRAR SUELO

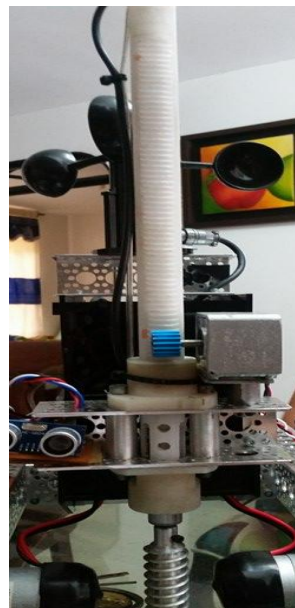
Dentro de las variables a medir se encuentran la temperatura y humedad del suelo, lo que requiere de introducir el sensor respectivo en la tierra de forma automática. La vista desde la cual se realiza dicho proceso además de mostrar el video, cuenta con los botones para taladrar, e introducir el sensor en una pieza mecánica, que se ha adaptado previamente a la plataforma.

Figura 37. Vista Taladrar Suelo



Fuente: Autor

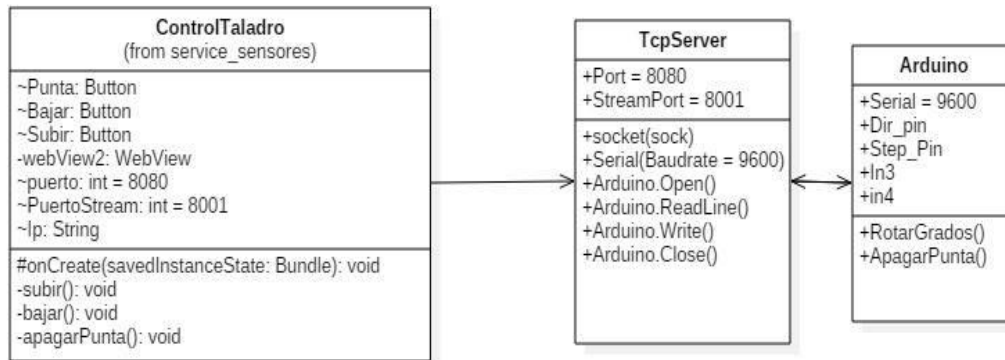
Figura 38. Pieza Mecánica Sensor Invasivo



Fuente: Autor

En primera instancia se establece la conexión con el servidor; y a través de los botones se envía un carácter a Arduino, donde se ejecutan las respectivas subrutinas de acuerdo a la petición.

Figura 39. D. Clases Taladrar Suelo



Fuente: Autor

6.5 VISTA PARA LA RECEPCIÓN DE LOS DATOS METEOROLÓGICOS

Figura 40. Vista Adquisición de Datos

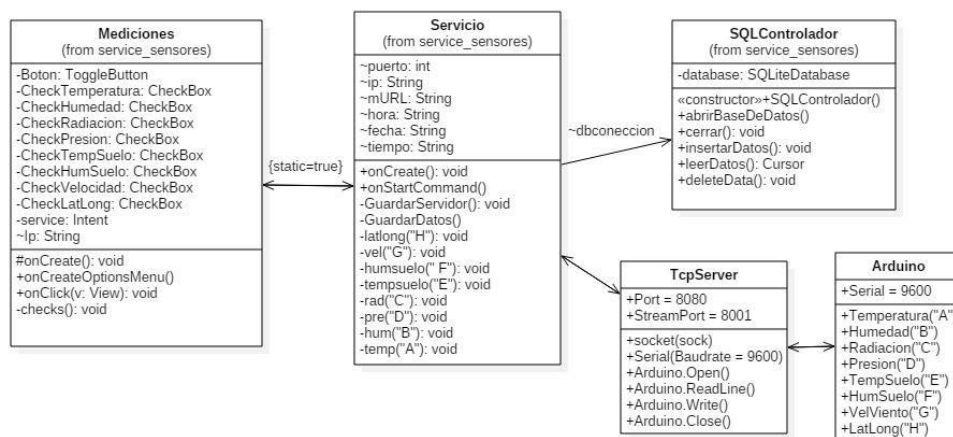


Fuente: Autor

En esta vista el cliente recibe los datos obtenidos por cada uno de los sensores, para luego ser visualizados, almacenados y graficados; además brinda la posibilidad de elegir las mediciones a realizar por medio de CheckBox.

La clase principal se extiende a un servicio que contiene toda la programación fundamental, tal como la conexión TCP con el servidor y el almacenamiento de los datos tanto en el servidor como en el teléfono móvil.

Figura 41. D. Clases de la Vista de Datos



Fuente: Autor

El servicio (Service) inicia su ejecución al momento de presionar el ToggleButton, cambiando así al estado ON para empezar la recepción de los datos; cuando el ToggleButton pasa a estado OFF, el servicio deja de ejecutarse y se detiene la recepción de la información. Cabe resaltar que una de las principales características del servicio es que se puede ejecutar en segundo plano, evitando así, que todas las tareas que realiza se detengan al momento de cambiar de vista o de aplicación.

La clase servicio cuenta con los métodos StartService y stopService para iniciar y detener el servicio desde la clase principal, la cual cuenta con un Intent que envía los parámetros al servicio en el momento de iniciar su ejecución; antes de evaluar el Intent e iniciar el servicio, se verifica que haya al menos un checkbox seleccionado; esta tarea se realiza a través de un condicional y la operación booleana Or, dado el caso que ningún checkbox este seleccionado el servicio no se inicia.

Para la creación de la clase Servicio hay que implementar los métodos onCreate(), onDestroy(), onBind() y onStartCommand() para su correcto funcionamiento. En el método onCreate() se ubican las funciones que se van a ejecutar al momento de iniciar el servicio, tales como la apertura de la base de datos SQLite, la definición de las variables globales y la configuración del handler para la actualización de los datos en la vista principal. Por otra parte el método onStartCommand es el más complejo de todos, ya que a través de él se llevan a cabo todas las operaciones que van a estar ejecutándose continuamente, tales como la conexión con el servidor y el almacenamiento de los datos.

- **Conexión TCP con el servidor:** La comunicación con el servidor se realiza por medio de sockets, uno para la solicitud de cada variable meteorológica. La petición al servidor se hace a través de un carácter, dependiendo de este, el servidor responde con el dato del respectivo sensor.

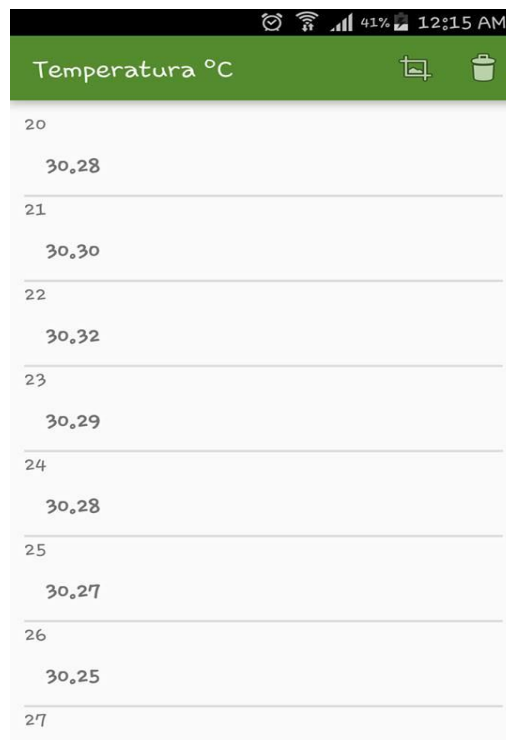
La clase socket de java es una alternativa de fácil uso cuando se quiere establecer una conexión, al momento de usarla solo se necesita conocer la IP del servidor y el puerto por el cual se va a realizar la comunicación. Write y ReadLine son métodos de la clase socket a través de los cuales se envían y se recibe la información; lo cual se garantiza con el método flush.

- **Almacenamiento de datos en SQLite:** Para almacenar los datos en SQLite, se crea un método que hace referencia a la clase SqlControlador, la cual es la encargada de gestionar la base de datos; luego esta procede a insertar cada uno de ellos a través del método dbconexcion.insertarDatos().
- **Almacenamiento de datos en el servidor:** Para almacenar los datos desde el cliente en la base de datos ubicada en el servidor, se hace uso de la clase HttpClient, por medio de esta se pueden realizar solicitudes HTTP, a través de las cuales se puede acceder a los scripts php que se encuentran ubicados en el servidor; dichos scripts contienen las sentencias SQL que gestionan la base de datos.

Cada dato antes de ser enviado al servidor es codificado en formato utf-8, ya que este es uno de los formatos contemplados para la codificación de textos en protocolos de internet.

6.6 VISTAS HISTORIAL DE DATOS

Figura 42. Vista Historial de Datos



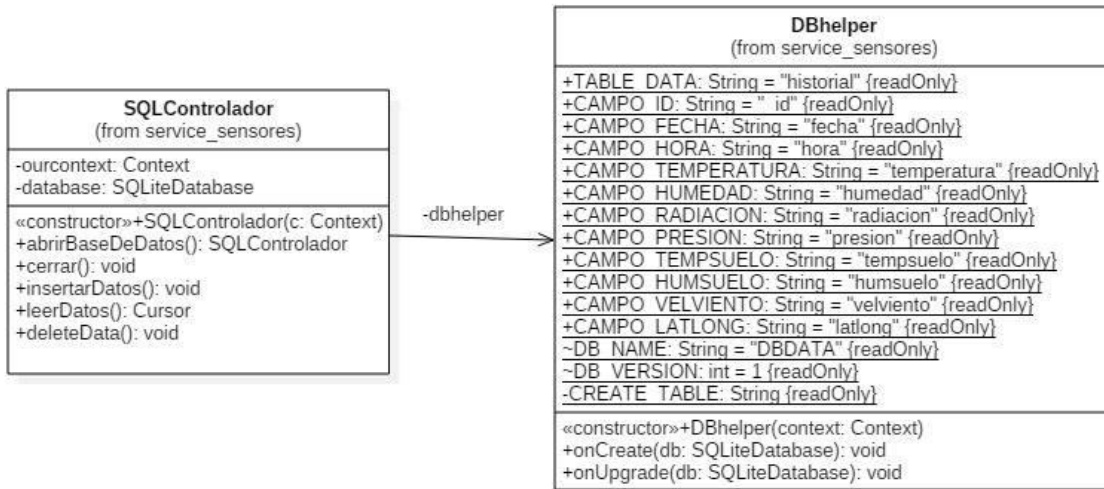
Fecha	Temperatura (°C)
20	30,28
21	30,30
22	30,32
23	30,29
24	30,28
25	30,27
26	30,25
27	

Fuente: Autor

Para la visualización de los datos almacenados en SQLite se diseñaron vistas para cada una de las variables meteorológicas. Por medio de ListViews se pueden observar todos los datos que han sido guardados, pero si se desea obtener un informe más detallado de los datos se puede hacer desde la base de datos del servidor, ya que en ella se puede visualizar hora y fecha de cada una de las mediciones.

La forma más cómoda de integrar SQLite a una aplicación Android es usando dos clases auxiliares que contengan los métodos y constructores para la creación y edición de la base de datos.

Figura 43. D. Clases Base de Datos



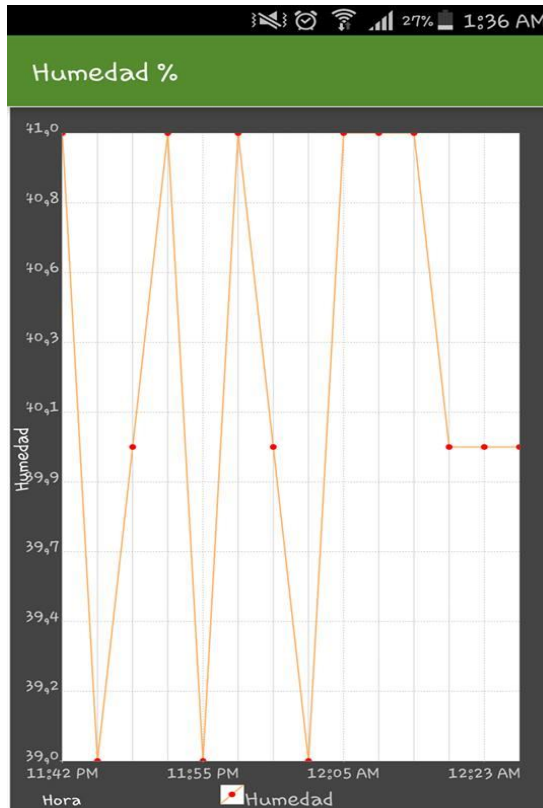
Fuente: Autor

La clase DBHelper es la encargada de la creación de la base de datos; en ella se definen las tablas y sus respectivos campos, a través de sentencias SQL ; por otra parte la clase SQLControlador se ocupa de la gestión de la base de datos, es decir, en ella están ubicados los métodos para la edición de la tabla y cada uno de los campos, un método para la inserción de los datos y otro para la eliminación de los mismos; cada una de estas acciones también se ejecuta a través de comandos SQL.

6.7 VISTAS DE LOS GRÁFICOS

Para realizar los gráficos de cada una de las variables meteorológicas se extraen los datos de la base de datos de SQLite, y mediante la librería AndroidPlot se hace cada una de las gráficas.

Figura 44. Gráfico de Variables



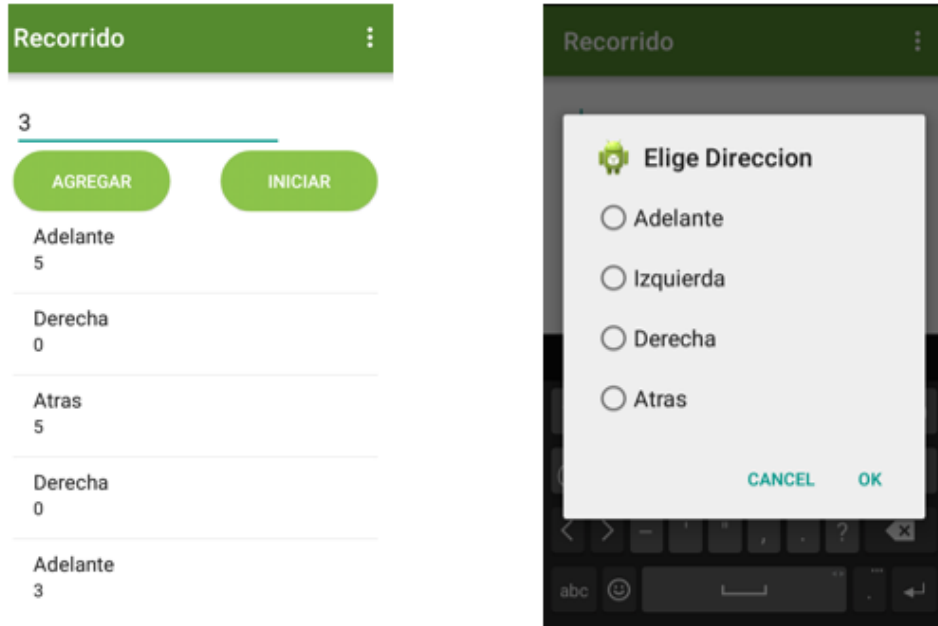
Fuente: Autor

Ya que se desea realizar un grafica sobre dos ejes, se hace uso de la clase XYPlot, por medio de ella se pueden tomar datos de una fuente y graficarlos.

Los datos del eje X corresponden a la hora en que se ha tomado cada medición, la cual es obtenida del teléfono móvil. Los datos del eje Y son extraídos de cada columna de la base de datos para luego se organizan en arreglos. Tanto los datos del eje X como los del eje Y son convertidos en una serie propia de AndroidPlot a través del método simplexseries para finalmente ser graficados.

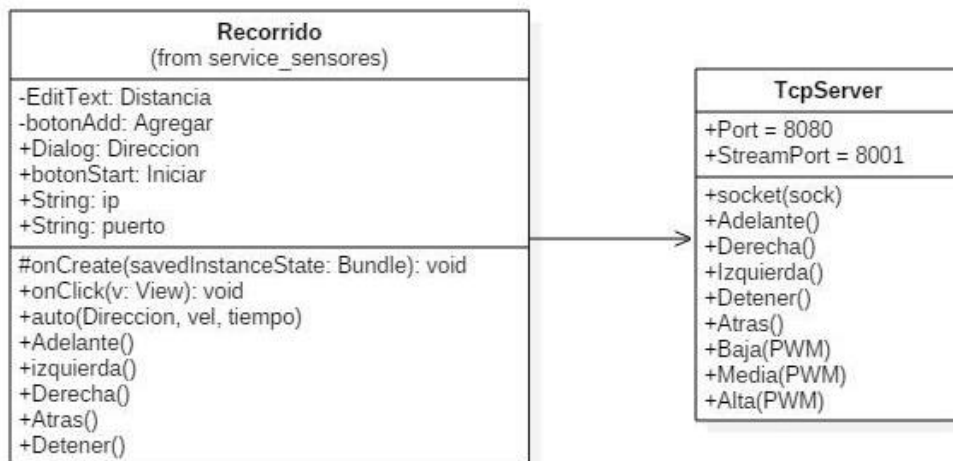
6.8 VISTA RECORRIDO AUTONOMO

Figura 45. Vista Recorrido Autónomo y selección de ruta



Fuente: Autor

Figura 46. D. Clases Recorrido Autónomo



Fuente: Autor

Por medio de esta vista el usuario puede programar una serie de movimientos para que la plataforma los realice automáticamente. La vista contiene un EditText mediante el cual el usuario puede ingresar la distancia en metros a recorrer, un botón para dar dirección y agregar el segmento de ruta, un botón para iniciar el recorrido y un listview por medio del cual se puede observar todos los movimientos que va a realizar la plataforma. Al igual que en la clase, por medio del cual el usuario puede conducir la plataforma, se va a enviar un carácter al servidor para indicarle la dirección que va a recorrer, en este caso cada carácter va a ser enviado luego de determinado tiempo, dependiendo de la distancia que va a avanzar.

7. RESULTADOS

Figura 47. Plataforma Meteorológica



Fuente: Autor

Figura 48. Aplicación Móvil (Kampla)



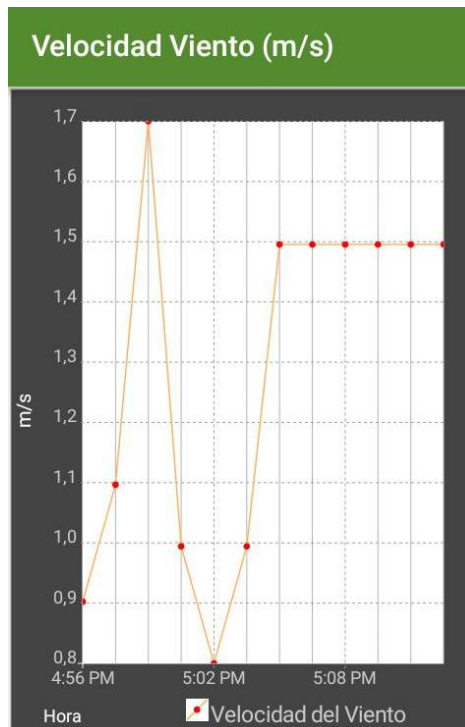
Fuente: Autor

7.1 PRUEBA DENTRO DE LA RED LOCAL

Para realizar esta prueba se conectó tanto la raspberry pi como el teléfono móvil al módem AirCard 763s, permitiendo una conexión mediante el protocolo wifi para enviar y recibir los datos. La toma de mediciones se realizó cada minuto, pero el tiempo entre cada medición se puede variar en la interfaz de la aplicación; teniendo en cuenta que este tipo de datos no sufre variaciones bruscas en un corto tiempo.

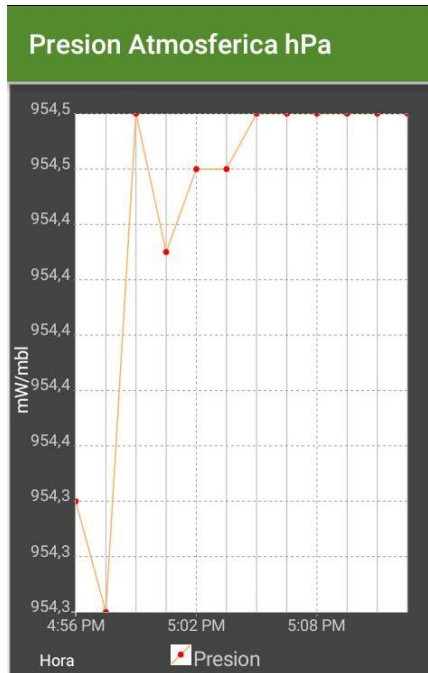
Además activar los sensores, recibir y almacenar los datos meteorológicos, también se comprobó el correcto funcionamiento del streaming de video y el movimiento de la plataforma. Algunas de las gráficas de las variables meteorológicas se muestran a continuación:

Figura 49. Prueba1 Velocidad del Viento



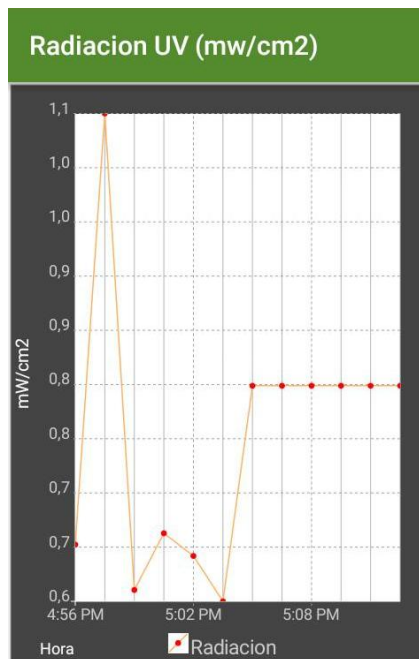
Fuente: Autor

Figura 50. Prueba1 Presión



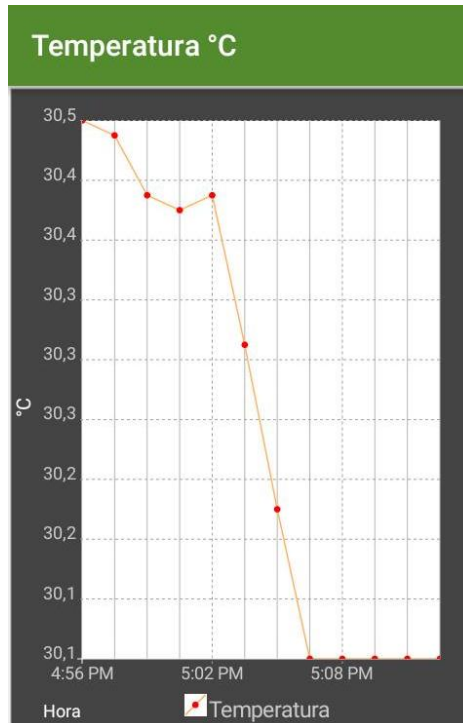
Fuente: Autor

Figura 51. Prueba1 Radiación UV



Fuente: Autor

Figura 52. Prueba1 Temperatura



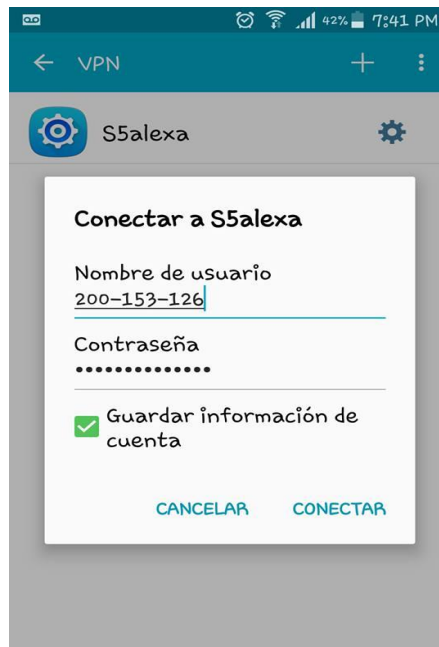
Fuente: Autor

7.2 PRUEBA DESDE UNA RED REMOTA

La tarjeta raspberry pi se conectó al modem AirCard 763s y el teléfono móvil en el cual se ejecutó la aplicación, se conectó a un módem distinto y se activó la red VPN configurada previamente. Bastó con seleccionar la opción del botón VPN de la vista de inicio de la aplicación y el funcionamiento del sistema es similar que dentro de la red local, la única diferencia es el retraso en la llegada de los datos que van desde la plataforma hacia la aplicación móvil.

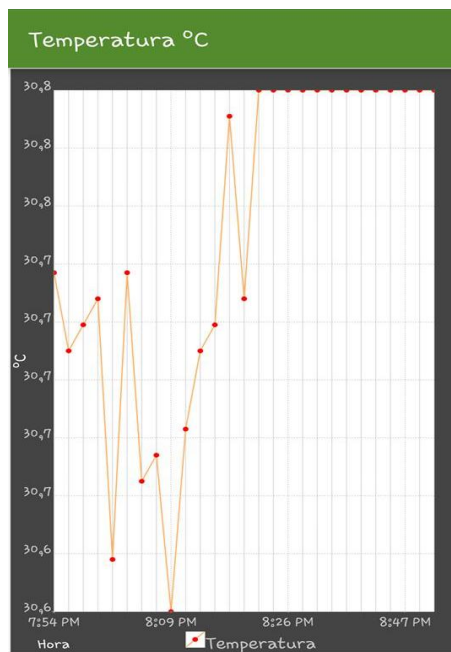
Igualmente se tomaron mediciones durante un lapso de tiempo y algunos los resultados se presentan en los siguientes gráficos, que hacen parte de la aplicación diseñada:

Figura 53. Conexión Red VPN



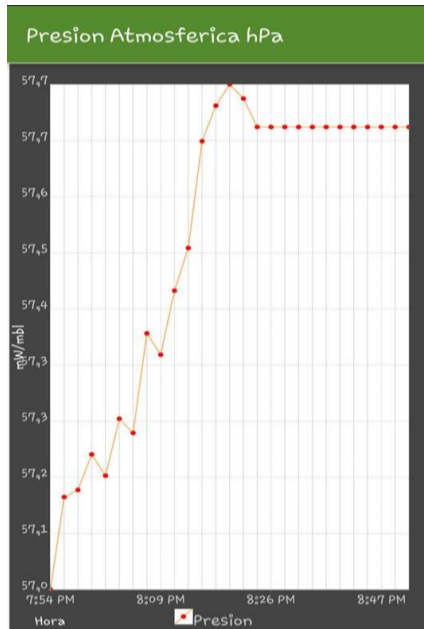
Fuente: Autor

Figura 54. Prueba2 Temperatura



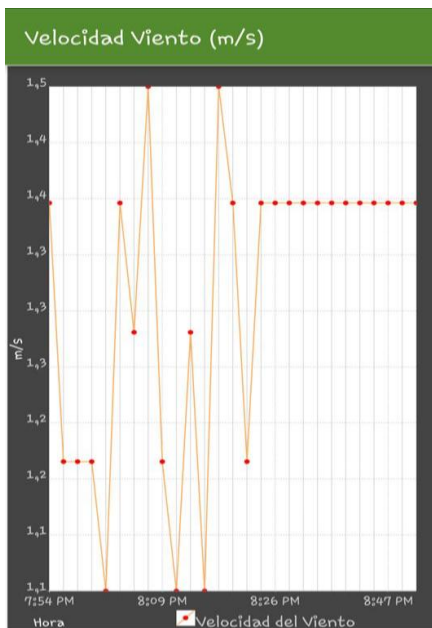
Fuente: Autor

Figura 55. Prueba2 Presión



Fuente: Autor

Figura 56. Prueba2 Velocidad del Viento



Fuente: Autor

6.4 VALIDACION DE LOS DATOS MEDIDOS POR LOS SENSORES

Para validar los datos que miden los sensores de la plataforma robótica, se realizó una visita a la estación meteorológica del aeropuerto Benito salas, perteneciente al IDEAM; allí se registraron las mediciones de velocidad del viento, temperatura y humedad del ambiente para hacer la respectiva comparación.

Figura 57. Caseta termoregistradora y plataforma robótica



Fuente: Autor

Se tomaron mediciones de temperatura y humedad, ubicando la plataforma cerca a la caseta termoregistradora (Figura 52) y dentro de ella; los datos obtenidos tanto en la plataforma robótica como en la estación convencional se consignan en el siguiente cuadro.

Cuadro 1. Datos temperatura y humedad

HORA	ESTACION AEROPUERTO		PLATAFORMA ROBOTICA	
	Temperatura (°C)	Humedad (%)	Temperatura (°C)	Humedad (%)
9:45 am	27.3	68	29.78	61
10:40 am	29	63	32.47	55
10:43 am	29.1	59	31.54	53
10:46 am	30	58	31.72	52
10:49 am	29.8	57	31.67	52

Fuente: Autor

En la medición de la humedad se observó una variación que se puede considerar dentro del rango de tolerancia; la cual según lo observado y analizado por el funcionario del IDEAM, se puede deber a la ubicación y cubrimiento del medidor de la estación, además de que el haz de cabellos, el cual es el elemento que detecta la humedad; lleva tiempo sin ser cambiado y por lo tanto va perdiendo sensibilidad, además de que el sensor de humedad utilizado en la plataforma tiene una precisión del 5%. Por otra parte la temperatura es similar en ambos medidores.

El anemómetro de la estación meteorológica del aeropuerto funciona como un tacómetro análogo, donde la velocidad del viento se obtiene calculando el promedio del recorrido durante el día. Para validar los datos de la velocidad; se realizaron mediciones durante cinco minutos, de acuerdo al recorrido que se observó en el lector y multiplicado x 100m como indica el instrumento, se dividió en el tiempo de la muestra que fueron aproximadamente 5 minutos.

Figura 58. Datos registrados en el Servidor

id	fecha	hora	temperatura	humedad	radiacion	presion	tempsuelo	humsuelo	velocidad	latlong
1	2017-01-13	9:45 AM	29.78	61.00					0.89	
2	2017-01-13	10:39 AM	32.47	55.00					1.28	
3	2017-01-13	10:42 AM	31.54	53.00					1.08	
4	2017-01-13	10:45 AM	31.72	52.00					0.89	
5	2017-01-13	10:49 AM	31.67	52.00					0.69	
6	2017-01-13	10:56 AM	38.04	47.00	3.33	961.04			1.57	0256.7853N 07517.4166W
7	2017-01-13	11:01 AM	32.79	47.00	3.00	961.02			1.18	0256.7853N 07517.4166W
8	2017-01-13	11:07 AM	30.39	48.00	3.17	960.47			1.57	0256.7853N 07517.4166W
9	2017-01-13	11:13 AM	42.92	48.00	7.61	955.85			1.77	0256.7853N 07517.4166W
10	2017-01-13	11:18 AM	33.23	37.00					1.08	
11	2017-01-13	11:20 AM	32.33	42.00						
12	2017-01-13	11:27 AM								
13	2017-01-13	11:27 AM	30.06	44.00						
14	2017-01-13	11:35 AM	29.95	44.00						

Fuente: Autor

Figura 59. Primera medida anemómetros



Fuente: Autor

Figura 60. Segunda medida anemómetros



Fuente: Autor

Con la plataforma en el suelo y el anemómetro convencional a 2 metros (Figura 54) se obtuvo la siguiente relación. Para el anemómetro de la estación convencional:

2818911,5 al iniciar

2818916 después de cinco minutos (300 s)

Para una diferencia de 4,5*100m

$$velocidad\ del\ viento = \frac{450\ m}{300s} = \frac{1.5m}{s}$$

Y en la plataforma meteorológica se tomaron dos mediciones durante esos cinco minutos con las cuales se hace el promedio:

1,57 m/s y 1,18 m/s

$$velocidad\ del\ viento = \frac{1.57m/s + 1.18m/s}{2} = \frac{1.375m}{s}$$

Como se observa, las dos mediciones difieren un poco, teniendo en cuenta que hay una diferencia en la altura de los anemómetros de aproximadamente 1.5 metros lo cual justifica la variación.

Se dispuso la plataforma a la altura aproximada a la del anemómetro de la estación del aeropuerto (Figura 55). Para el anemómetro de la estación convencional:

2818922,8 al iniciar

2818927,8 después de cinco minutos (300 s)

Para una diferencia de 5*100m

$$velocidad\ del\ viento = \frac{500\ m}{300s} = \frac{1.66m}{s}$$

Y en la plataforma meteorológica se tomaron dos mediciones durante esos cinco minutos con las cuales se hace el promedio:

1,57 m/s y 1,77 m/s

$$velocidad\ del\ viento = \frac{1,57m/s + 1,77m/s}{2} = \frac{1.67m}{s}$$

Con lo cual se observa que a una misma altura, los anemómetros presentan mediciones prácticamente iguales.

8. CONCLUSIONES

- Arduino y Raspberry pi son dos plataformas que se complementan y son muy útiles para proyectos que requieren gran capacidad de hardware como de software. Para el presente proyecto Arduino se ocupa de la parte sensorica y actuadores, ya que cuenta con entradas analógicas y digitales. Mientras la Raspberry pi se encarga de la parte de software por la velocidad de procesamiento y su memoria RAM de 1GB.
- Por sus características de movilidad, tamaño y peso reducido; la plataforma no es apta para realizar mediciones de temperatura y humedad del suelo en terrenos de textura rígida.
- Raspberry pi es una placa muy completa, la cual permitió realizar de manera satisfactoria operaciones como: la conducción de la plataforma, conexión serial con la tarjeta controladora, transmisión de video a través de streaming y almacenamiento de datos en phpMyAdmin.
- La estructura del sistema cuenta con las propiedades necesarias para ser adaptada a otros proyectos que requieran de un control remoto de sensores y actuadores; ya que puede ser manipulado desde una red local o una red VPN a través de una App Android.
- La ejecución en segundo plano de la App es una de las características más relevantes y se logra a través de la clase Service, gracias a esto continúa recibiendo los datos de las variables meteorológicas incluso con el teléfono bloqueado o si el usuario está interactuando con otra aplicación.
- Con la práctica en el aeropuerto se validaron los datos medidos con los sensores de la plataforma meteorológica y una estación certificada, como lo es la estación del aeropuerto Benito Salas perteneciente al IDEAM.

9. RECOMENDACIONES

- Aunque el funcionamiento del sistema es intuitivo, se requiere de una previa asesoría para que el usuario pueda manipular la aplicación de forma correcta; ya que a la hora de trabajar en una red VPN se requiere que con anterioridad el usuario se haya registrado en la red creada en Hamachi LogMeIn.
- Para futuras mejoras a la plataforma se recomienda adecuarla de tal manera que todos sus componentes puedan ser expuestos a la lluvia sin correr el riesgo de sufrir algún daño.
- Se recomienda que para un estudio riguroso de las distintas variables meteorológicas, se haga uso de la información almacenada en la base de datos del servidor, ya que esta cuenta con un reporte más completo. El historial de datos que se guarda a través de la aplicación, es demasiado básico y no cuenta con toda la información que se pueda llegar a necesitar.

BIBLIOGRAFÍA

Arias, Angel. 2016. Curso de programación de Apps. Android y iPhone. 2016. 2 Edición.

Bell, Charles. 2013. "Beginning Sensor Networks with Arduino and Raspberry Pi". Nueva York : s.n., 2013.

IDEAM. 2012. *Catálogo de estaciones climatológicas por departamento*. Neiva, Huila : s.n., 2012.

San Miguel, Pablo A. 2014. *Electrotecnia*. Madrid : Paraninfo, 2014.

STMicroelectronics. 2000. "L298 DUAL FULL-BRIDGE DRIVER". [En línea] enero de 2000. [Citado el: 15 de octubre de 2016.]

http://www.st.com/content/st_com/en/products/motor-drivers/brushed-dc-motor-drivers/l298.html.

"Acerca de androidplot". [En línea] [Citado el: 2 de septiembre de 2016.] Disponible en: <http://androidplot.com/about/>.

"Actobotics Kit - Nomad 4WD Off-Road Chassis". [En línea] [Citado el: 22 de Agosto de 2016.] Disponible en la web: <https://www.sparkfun.com/products/13141>.

"Android, la plataforma móvil mas popular del mundo". [En línea] [Citado el: 2 de Septiembre de 2016.] Disponible en la web: <https://developer.android.com/about/android.html>.

"Arduino Mega 2560 y Genuino Mega 2560". [En línea] [Citado el: 20 de Agosto de 2016.] Disponible en la web: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>.

"PHP". [En línea] [Citado el: 27 de Agosto de 2016.] Disponible en la web: <http://www.ecured.cu/PHP>.

"Python". [En línea] [Citado el: 27 de Agosto de 2016.] Disponible en la web: <http://www.ecured.cu/Python>.

"¿Qué es Arduino?". [En línea] [Citado el: 20 de Agosto de 2016.] Disponible en la web: <http://arduino.cl/que-es-arduino/>.

"Raspberry Pi 2 Model B". [En línea] [Citado el: 30 de Julio de 2016.] Disponible en la web: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.

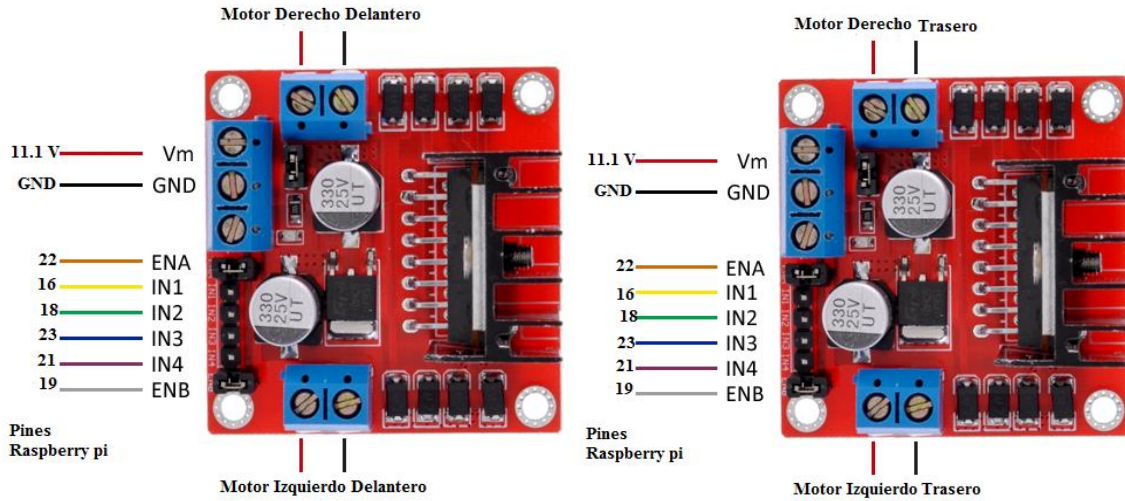
"Service". [En línea] [Citado el: 3 de 10 de 2016.] Disponible en la web:
<https://developer.android.com/guide/components/services.html?hl=es-419>.

"Sierra Wireless Aircard 763S Hotspot". [En línea] [Citado el: 26 de Agosto de 2016.]
Disponible en la web: <http://www.store4g.com/sierra-aircard-763s/>.

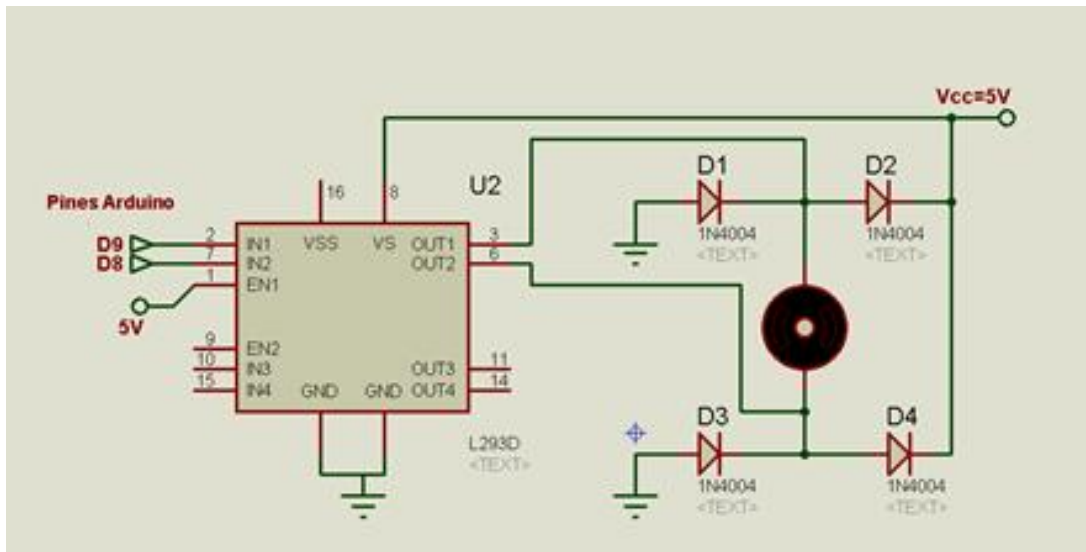
"Welcome to Raspbian". [En línea] [Citado el: 16 de Agosto de 2016.] Disponible en la
web: <https://www.raspbian.org/>.

ANEXOS

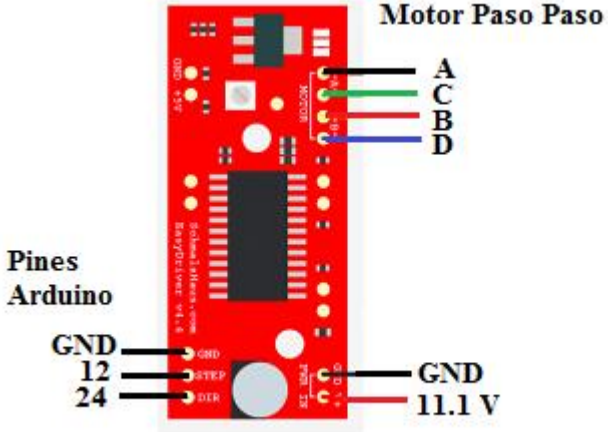
Anexo A. Conexión Motores DC de la plataforma



Anexo B. Conexión Motor DC de la broca



Anexo C. Conexión motor paso a paso



Anexo D. fotografías

