



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**TEMA: DESARROLLO DE UN SISTEMA HMI EN LAS
PLATAFORMAS ANDROID E iOS PARA EL MONITOREO DE LA
RED INALÁMBRICA DE SENSORES AGRÍCOLAS UBICADA EN
EL BARRIO RUMIPAMBA DE NAVAS CANTÓN SALCEDO,
PROVINCIA DE COTOPAXI**

AUTORES:

**EVELYN FERNANDA ENDARA CRESPIATA
VÍCTOR ALFONSO VARGAS ORTIZ**

**DIRECTOR: ING. DAVID RIVAS
CODIRECTOR: ING. MAYRA ERAZO**

LATACUNGA

2015

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

CERTIFICADO

Ing. David Rivas (DIRECTOR DE TESIS)

Ing. Erazo Mayra (CODIRECTORA DE TESIS)

CERTIFICAN

Que el trabajo titulado "DESARROLLO DE UN SISTEMA HMI EN LAS PLATAFORMAS ANDROID E IOS PARA EL MONITOREO DE LA RED INALÁMBRICA DE SENSORES AGRÍCOLAS UBICADA EN EL BARRIO RUMIPAMBA DE NAVAS CANTÓN SALCEDO, PROVINCIA DE COTOPAXI", realizado por la señorita Endara Crespata Evelyn Fernanda y el señor Vargas Ortiz Víctor Alfonso, ha sido guiado y revisado periódicamente y cumple normas establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPEL.

Debido a que constituyen un trabajo de alto contenido de investigación y que ayuda a la formación profesional y aplicación de conocimientos, si se recomienda su publicación. El mencionado trabajo consta de dos documentos empastados, dos discos compactos que contienen los archivos en formato portátil de Acrobat (pdf). Autorizan a los señores Endara Crespata Evelyn Fernanda y Vargas Ortiz Víctor Alfonso que lo entregue al ing. Silva Franklin, en su calidad de Director de la Carrera.

Latacunga, Julio 2015.

Ing. David Rivas
DIRECTOR

Ing. Erazo Mayra
CODIRECTORA

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

AUTORIZACIÓN

Nosotros: Evelyn Fernanda Endara Crespata
Víctor Alfonso Vargas Ortiz

Autorizamos a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la Institución del trabajo "DESARROLLO DE UN SISTEMA HMI EN LAS PLATAFORMAS ANDROID E iOS PARA EL MONITOREO DE LA RED INALÁMBRICA DE SENSORES AGRÍCOLAS UBICADA EN EL BARRIO RUMIPAMBA DE NAVAS CANTÓN SALCEDO, PROVINCIA DE COTOPAXI", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, Julio 2015.

Evelyn Fernanda Endara Crespata
C.C.: 050362751-5

Víctor Alfonso Vargas Ortiz
C.C.: 050349598-8

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

DECLARACIÓN DE RESPONSABILIDAD

Nosotros: Evelyn Fernanda Endara Crespata
Víctor Alfonso Vargas Ortiz

DECLARAMOS QUE:

El proyecto de grado denominado "DESARROLLO DE UN SISTEMA HMI EN LAS PLATAFORMAS ANDROID E iOS PARA EL MONITOREO DE LA RED INALÁMBRICA DE SENSORES AGRÍCOLAS UBICADA EN EL BARRIO RUMIPAMBA DE NAVAS CANTÓN SALCEDO, PROVINCIA DE COTOPAXI", ha sido desarrollada en base a una investigación, respetando derechos intelectuales de terceros, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance investigativo del proyecto de grado en mención.

Latacunga, Julio 2015.

Evelyn Fernanda Endara Crespata
C.C.: 050362751-5

Víctor Alfonso Vargas Ortiz
C.C.: 050349598-8

DEDICATORIA

Quiero dedicar este trabajo a mi madre Sonia por haber sido el pilar fundamental en cada etapa de mi vida, quien me ha conducido por el camino correcto y ser un ejemplo de lucha y perseverancia; a mis hermanos Emily y Néstor por ser mi motivación para seguir adelante y no decaer frente a las adversidades; a una persona muy especial en mi vida Darío quien estuvo a mi lado en cada momento brindándome ese apoyo incondicional gracias por todo mi amor, y a mi familia en especial a mi abuelita María y tíos Jaime ,Moisés, Henry, Bety y Silvia quienes me han apoyado siempre y por estar presente cuando más los he necesitado, de todo corazón UN DIOS LE PAGUE.

EVELYN

El presente trabajo es dedicado a los seres que representan el pilar fundamental en la formación de mi persona, me refiero a mis padres Luis y Daniela, quienes con su esfuerzo, apoyo incondicional y sabias enseñanzas, han inculcado en mí los valores éticos, los cuales han servido para vencer las dificultades que en el camino se me han presentado; a mis hermanos Luis y Tatiana, y una persona especial Gaby quienes me acompañaron durante estos años de formación universitaria y brindándome su apoyo incondicional, palabras de aliento y motivación. A ustedes mi familia, reitero mi sentimiento de gratitud, pues con su apoyo decidido han contribuido a que se cristalice la meta que juntos nos hemos propuesto "HOT".

VICTOR

AGRADECIMIENTO

A Dios por habernos acompañado y guiado a lo largo de nuestra carrera, por ser nuestra fortaleza en los momentos de debilidad y por brindarnos una vida llena de aprendizaje, experiencias y felicidad.

A nuestras familias por ser la fuente de nuestra inspiración y motivación para superarnos cada día más, por estar siempre guiándonos y brindándonos todo su amor y comprensión.

A todos los docentes de nuestra carrera, quienes con su paciencia, vocación y respeto han logrado transmitir sus valiosos conocimientos en nosotros, así mismo un agradecimiento especial por la confianza, apoyo y dedicación brindada por nuestros tutores de tesis Ing. David Rivas e Ing. Mayra Erazo por haber compartido con nosotros sus conocimientos y sobre todo su amistad.

EVELYN ENDARA
VICTOR VARGAS

ÍNDICE DE CONTENIDO

| | |
|---|-------------|
| PORTADA..... | i |
| CERTIFICADO..... | ii |
| AUTORIZACIÓN | iii |
| DECLARACIÓN DE RESPONSABILIDAD | iv |
| DEDICATORIA | v |
| AGRADECIMIENTO..... | vi |
| ÍNDICE DE CONTENIDO..... | vii |
| ÍNDICE DE FIGURAS | xi |
| ÍNDICE DE TABLAS..... | xv |
| RESUMEN | xvi |
| ABSTRACT..... | xvii |

CAPÍTULO I

| | |
|--|----------|
| FUNDAMENTOS TEÓRICOS | 1 |
| 1.1 Antecedentes..... | 1 |
| 1.2 Descripción del proyecto | 2 |
| 1.3 Justificación | 2 |
| 1.4 Comunicación móvil | 3 |
| 1.4.1 Tipos de dispositivos móviles | 4 |
| 1.5 Plataforma Android..... | 7 |
| 1.5.1 Arquitectura | 7 |
| 1.5.2 Características principales | 11 |
| 1.5.3 Versiones | 12 |
| 1.5.4 Software para el desarrollo de aplicaciones..... | 15 |
| 1.6 Plataforma iOS | 17 |

| | | |
|--|---|----|
| 1.6.1 | Arquitectura | 17 |
| 1.6.2 | Características principales | 18 |
| 1.6.3 | Versiones | 19 |
| 1.6.4 | Software para el desarrollo de aplicaciones..... | 21 |
| 1.7 | Comparación entre sistemas operativos móviles | 22 |
| 1.8 | Servidor web..... | 24 |
| 1.8.1 | Tipos de servidores | 25 |
| 1.8.2 | Arquitectura cliente servidor | 26 |
| 1.8.3 | Modelo osi..... | 26 |
| 1.8.4 | Tcp/ip | 28 |
| 1.8.5 | Medios de acceso a internet..... | 29 |
| 1.9 | Internet | 30 |
| 1.10 | Tecnología wireless..... | 31 |
| 1.11 | Protocolo 802.11 o wi-fi | 32 |
| 1.11.1 | Los distintos estándares wi-fi..... | 32 |
| 1.11.2 | Rangos ieee 802.11 | 35 |
| 1.12 | Tecnología móvil 3G..... | 36 |
| 1.13 | HMI en telefonía móvil..... | 37 |
| 1.14 | Red de sensores inalámbricos | 37 |
| 1.14.1 | Elementos en una red wsn..... | 38 |
| 1.15 | Estándares utilizados en tecnologías inalámbricas | 39 |
| 1.15.1 | Tecnología zigbee | 39 |
| 1.15.2 | Tecnología digimesh | 41 |
| 1.15.3 | Tecnología wifi | 42 |
| | | |
| CAPÍTULO II | | |
| SELECCIÓN DE SOFTWARE Y COMPONENTES | | |
| 2.1 | Requisitos para el servidor web..... | 45 |

| | | |
|-------|---|----|
| 2.1.1 | Xampp | 45 |
| 2.1.2 | Base de datos en xampp..... | 46 |
| 2.1.3 | Php peticion de consulta mysql..... | 50 |
| 2.2 | Requisitos para acceder a la base de datos..... | 51 |
| 2.2.1 | Mysql connector / odbc..... | 51 |
| 2.2.2 | Labview | 52 |
| 2.3 | Requisitos Android | 54 |
| 2.4 | Requisitos iOS..... | 55 |

CAPÍTULO III

| | | |
|--------------------------------------|---|----|
| DISEÑO E IMPLEMENTACIÓN | 56 | |
| 3.1 | Diagrama de flujo de las aplicaciones | 56 |
| 3.1.1 | Tendencia | 57 |
| 3.1.2 | Históricos | 60 |
| 3.1.3 | Ayuda..... | 62 |
| 3.2 | Crear un proyecto en Android..... | 64 |
| 3.3 | Crear un proyecto en iOS (xcode) | 67 |
| 3.4 | Programación en Android..... | 70 |
| 3.4.1 | Tendencia | 70 |
| 3.4.2 | Históricos | 76 |
| 3.4.3 | Grafica | 79 |
| 3.5 | Programación en xcode..... | 81 |
| 3.5.1 | Tendencia | 81 |
| 3.5.2 | Históricos | 87 |
| 3.5.3 | Grafica | 91 |

CAPÍTULO IV

| | | |
|----------------------------------|------------------------|----|
| ANÁLISIS COMPARATIVO..... | 93 | |
| 4.1 | Consumo de datos | 93 |

| | | |
|--|--------------------------|------------|
| 4.1.1 | Wifi | 93 |
| 4.1.2 | 3G..... | 99 |
| 4.1.3 | Comparación | 105 |
| 4.2 | Tiempo de respuesta..... | 106 |
| 4.2.1 | Wifi | 106 |
| 4.2.2 | 3g..... | 108 |
| 4.2.3 | Comparación | 110 |
| CAPÍTULO V | | |
| CONCLUSIONES Y RECOMENDACIONES..... | | 111 |
| 5.1.1 | Conclusiones..... | 111 |
| 5.1.2 | Recomendaciones..... | 113 |
| LINKGRAFÍA | | 114 |
| ANEXOS..... | | 117 |

ANEXO A Código de los archivos php red (wifi)

ANEXO B Código de programación Android

ANEXO C Código de programación Xcode

ANEXO D Análisis con acceso a wifi

ÍNDICE DE FIGURAS

CAPÍTULO I

| | |
|---|----|
| Figura 1.1 Arquitectura de Android [4] | 11 |
| Figura 1.2 Arquitectura iOS [7] | 17 |
| Figura 1.3 Servidor Web [10]..... | 24 |
| Figura 1.4 Capas del modelo OSI [11]..... | 27 |
| Figura 1.5 Modelo TCP/IP relacionado con el modelo OSI [12] | 29 |
| Figura 1.6 Red de sensores inalámbricos. [19] | 38 |
| Figura 1.7 Dispositivos que conforman ZigBee. | 41 |
| Figura 1.8 Elementos de una red WIFI. | 43 |

CAPÍTULO II

| | |
|--|----|
| Figura 2.1 Etapas del proyecto | 45 |
| Figura 2.2 Crear las tablas de la base de datos | 47 |
| Figura 2.3 Asignar nombres, tipo y longitud de los datos a manejar | 47 |
| Figura 2.4 MySQL ODBC 5.2 Driver..... | 51 |
| Figura 2.5 Asignar los nombres a la conexión | 52 |
| Figura 2.6 Origen de los datos..... | 53 |
| Figura 2.7 Conexión MYSQL..... | 54 |
| Figura 2.8 Añadir la dirección de descarga del plugin | 55 |

CAPÍTULO III

| | |
|--|----|
| Figura 3.1 Menú Principal..... | 56 |
| Figura 3.2 Diagrama de flujo (Principal) | 57 |
| Figura 3.3 Diagrama de flujo (Tendencia) | 58 |
| Figura 3.4 Diagrama de flujo (Digimesh) | 58 |
| Figura 3.5 Diagrama de flujo (Wi-Fi)..... | 59 |
| Figura 3.6 Diagrama de flujo (Zigbee) | 59 |
| Figura 3.7 Diagrama de flujo de históricos | 60 |

| | |
|--|----|
| Figura 3.8 Diagrama de flujo red Zigbee (Históricos) | 61 |
| Figura 3.9 Diagrama de flujo nodo 2 de la red Zigbee (Históricos)..... | 61 |
| Figura 3.10 Diagrama de flujo de la variable temperatura | 62 |
| Figura 3.11 Diagrama de flujo de Ayuda | 63 |
| Figura 3.12 Layout de la Activity (Ayuda) | 63 |
| Figura 3.13 Datos de la nueva aplicación Android..... | 65 |
| Figura 3.14 Selección de icono..... | 66 |
| Figura 3.15 Icono de la aplicación | 66 |
| Figura 3.16 Plantilla de actividades | 67 |
| Figura 3.17 Seleccionar la plantilla de inicio | 68 |
| Figura 3.18 Llenar los campos del proyecto | 69 |
| Figura 3.19 Nuevo proyecto | 69 |
| Figura 3.20 Archivo xml | 71 |
| Figura 3.21 Graphical Layout..... | 71 |
| Figura 3.22 Librerías importadas | 72 |
| Figura 3.23 Clase Principal (Main)..... | 72 |
| Figura 3.24 Hilo de comunicación y direccionamiento de la url | 73 |
| Figura 3.25 Conversión a String | 74 |
| Figura 3.26 Objetos json..... | 74 |
| Figura 3.27 Obtener valor de la variable..... | 74 |
| Figura 3.28 Acción del botón | 75 |
| Figura 3.29 Visualización de datos (Nodo1) | 75 |
| Figura 3.30 Layout de Activity (Fecha) | 76 |
| Figura 3.31 Envío de datos por método POST | 77 |
| Figura 3.32 Respuesta del método Response..... | 77 |
| Figura 3.33 Obtención de los datos de la variable..... | 78 |
| Figura 3.34 Clase cargaListado | 78 |
| Figura 3.35 Visualización de datos en un ListView | 78 |
| Figura 3.36 Envío de datos a la clase Graficar | 79 |
| Figura 3.37 Recuperar los datos con el método Bundle | 80 |

| | |
|--|----|
| Figura 3.38 Almacenar los datos en un vector para ser graficado..... | 80 |
| Figura 3.39 Visualización de la grafica | 81 |
| Figura 3.40 Declarar métodos públicos y privados | 82 |
| Figura 3.41 ViewController.m | 83 |
| Figura 3.42 Visualización..... | 83 |
| Figura 3.43 Función getDataURL | 84 |
| Figura 3.44 Crear el objeto jsonArray | 84 |
| Figura 3.45 Referencia a los objetos json creados | 85 |
| Figura 3.46 Acción del botón para enviar información..... | 85 |
| Figura 3.47 Selección de botón | 86 |
| Figura 3.48 Visualización de datos (Nodo1) | 87 |
| Figura 3.49 Layout de Activity (Fecha) | 88 |
| Figura 3.50 Enviar datos al servidor web..... | 88 |
| Figura 3.51 Función GET para la consulta | 89 |
| Figura 3.52 Conversión a String | 89 |
| Figura 3.53 Código para TableView | 89 |
| Figura 3.54 Objetos json..... | 90 |
| Figura 3.55 Clase dato1Object | 90 |
| Figura 3.56 Visualización en la TableView | 90 |
| Figura 3.57 Clase ViewControllerGrafica..... | 91 |
| Figura 3.58 Recepción de datos de las variables | 91 |
| Figura 3.59 Función FSLineChart..... | 92 |
| Figura 3.60 Visualización de la grafica | 92 |

CAPÍTULO IV

| | |
|---|----|
| Figura 4.1 Consumo de datos de Subida (1 minuto wifi) | 94 |
| Figura 4.2 Consumo de datos de Bajada (1 minuto wifi) | 95 |
| Figura 4.3 Consumo de datos (1 minuto wifi) | 95 |
| Figura 4.4 Consumo de datos de Subida (10 minuto wifi) | 96 |
| Figura 4.5 Consumo de datos de Bajada (10 minutos wifi) | 96 |

| | |
|---|-----|
| Figura 4.6 Consumo de datos (10 minutos wifi)..... | 97 |
| Figura 4.7 Consumo de datos de Subida (1 hora wifi)..... | 98 |
| Figura 4.8 Consumo de datos de Bajada (1 hora wifi)..... | 98 |
| Figura 4.9 Consumo de datos (1 hora wifi)..... | 99 |
| Figura 4.10 Consumo de datos de Subida (1 minuto 3G)..... | 100 |
| Figura 4.11 Consumo de datos de Bajada (1 minuto 3G)..... | 100 |
| Figura 4.12 Consumo de datos (1 minuto 3G)..... | 101 |
| Figura 4.13 Consumo de datos de Subida (10 minutos 3G)..... | 102 |
| Figura 4.14 Consumo de datos de Bajada (10 minutos 3G)..... | 102 |
| Figura 4.15 Consumo de datos (10 minutos 3G)..... | 103 |
| Figura 4.16 Consumo de datos de Subida (1 hora 3G)..... | 104 |
| Figura 4.17 Consumo de datos de Bajada (1 hora 3G)..... | 104 |
| Figura 4.18 Consumo de datos (1 hora 3G)..... | 105 |
| Figura 4.19 Tiempo de respuesta WIFI (iOS)..... | 107 |
| Figura 4.20 Tiempo de respuesta WIFI (Android)..... | 107 |
| Figura 4.21 Tiempo de respuesta 3G (iOS)..... | 109 |
| Figura 4.22 Tiempo de respuesta 3G (Android)..... | 109 |

ÍNDICE DE TABLAS

CAPÍTULO I

| | |
|---|----|
| Tabla 1.1 Versiones de Android..... | 13 |
| Tabla 1.2 Versiones iOS..... | 19 |
| Tabla 1.3 Comparativa de sistemas operativos para Smartphone's..... | 22 |
| Tabla1.4 Tipos de servidores..... | 25 |
| Tabla 1.5 Estándares WIFI | 33 |
| Tabla 1.6 Rangos IEEE 802.11/x..... | 35 |

CAPÍTULO II

| | |
|------------------------------|----|
| Tabla 2.1 RED DIGIMESH | 48 |
| Tabla 2.2 RED WI-FI | 49 |
| Tabla 2.3 RED ZIGBEE | 50 |

CAPÍTULO IV

| | |
|---|-----|
| Tabla 4.1 Tiempo de respuesta WIFI..... | 106 |
| Tabla 4.2 Tiempo de respuesta 3G | 108 |

RESUMEN

El siguiente proyecto se basa en el desarrollo de una aplicación móvil para dispositivos Smartphone, en la actualidad la utilización de dispositivos móviles inteligentes ha permitido optimizar la gestión de los tiempos de trabajo e incrementar la productividad, por esta razón se ha desarrollado una aplicación para las plataformas Android e iOS que permite monitorear el comportamiento de las variables climatológicas de la red inalámbrica diseñada e implementado en un invernadero de producción de tomate de riñón en la ciudad de Salcedo, además permite obtener el histórico de un intervalo de tiempo que se requiere analizar el comportamiento de la variable climatológica. Este proyecto consta de dos etapas principales, la primera etapa es la adquisición y almacenamiento de los datos de cada red de sensores inalámbricos independientes, la segunda etapa está basada en el servidor web encargado de recibir y enviar información hacia la aplicación para visualizar los datos que requiere el cliente. Este proyecto es la etapa complementaria del proyecto titulado “DISEÑO E IMPLEMENTACION DE UNA RED WSN (WIRELESS SENSOR NETWORK) BASADO EN LOS PROTOCOLOS ZIGBEE, WIFI Y ZIGBEE MESH, PARA EL MONITOREO DE VARIABLES CLIMATICAS EN EL INVERNADERO UBICADO EN EL BARRIO RUMIPAMBA DEL NAVAS, CANTON SALCEDO, PROVINCIA DE COTOPAXI” con la fusión de ambos proyectos se pretende dotar de tecnología inteligente a la agricultura con ello mejorar la calidad de producción, evitando pérdidas y producir productos a gran escala con mira de mercados internacionales.

PALABRAS CLAVE:

- **SERVIDOR WEB**
- **TELÉFONO INTELIGENTE**
- **COMUNICACIÓN MÓVIL**
- **PLATAFORMA ANDROID**
- **PLATAFORMA iOS**

ABSTRACT

The next project is based on the development of a mobile application for smartphone devices currently using smart mobile devices has led to better management of working time and increase productivity for this reason has developed an application for Android and iOS platforms that allows monitoring the behavior of the climatic variables of the wireless network designed and implemented in a greenhouse tomato production kidney in the city of Salcedo, also giving the history of a time interval that is required to analyze the behavior of the climate variable. This project consists of two main stages, the first stage is the acquisition and storage of data for each network of independent wireless sensors, the second stage is based on the Web server to receive and send information to the application to display data It requires the customer. This project is the complementary phase of the project entitled "Design and Implementation of a WSN (Wireless Sensor Network) ZIGBEE NETWORK BASED PROTOCOLS, WIFI AND ZIGBEE MESH, FOR MONITORING CLIMATE VARIABLES IN THE GREENHOUSE LOCATED IN THE NEIGHBORHOOD RUMIPAMBA NAVAS, CANTON SALCEDO, COTOPAXI PROVINCE "with the merger of both projects is intended to provide intelligent technology to agriculture thereby improve the quality of production, avoiding losses and produce large-scale look at international markets..

KEYWORDS:

- **WEB SERVER**
- **SMARTPHONE**
- **MOBILE COMMUNICATION**
- **ANDROID PLATFORM**
- **iOS PLATFORM**

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1.1 ANTECEDENTES

Desde la antigüedad el hombre ha desarrollado diversas formas de comunicación, que han evolucionado de manera sorprendente desde señales de humo, en un inicio hasta los innovadores teléfonos inteligentes existentes en la actualidad que acortan distancias, a mayores velocidades.

Años atrás al término teléfono se lo concebía como un “conjunto de aparatos e hilos conductores con los cuales es posible transmitir toda clase de sonidos o voz, por la acción de la electricidad”. Gracias al avance de los sistemas y tecnologías de comunicación se crea el teléfono celular basado en señales electromagnéticas que se han convertido en una herramienta de gran utilidad para las personas, negocios e incluso en las industrias con la finalidad de intercambiar información.

En la actualidad las comunicaciones realizan sus enlaces sin la necesidad de cables, abaratando costos, lo que se denomina “comunicaciones inalámbricas”, las cuales se caracterizan por su fácil instalación, flexibilidad y escalabilidad. Hoy en día se dispone de una variedad de dispositivos móviles capaces de conectarse a cualquier red inalámbrica, con el propósito de acceder a la información existente en el internet.

En el campo de la industria la tecnología móvil es de gran utilidad, permitiendo acceder a ambientes hostiles y remotos, así salvaguardar la integridad de la planta y la seguridad del operador. Muchas empresas han optado por la implementación de este sistema de comunicación inalámbrica con la finalidad de adquirir datos en el menor tiempo posible y a mayor velocidad, ahorrando recursos de infraestructura a la empresa. Todo esto lleva al desarrollo de nuevas formas de integrar la tecnología móvil en diversas industrias como: medicina,

petrolera, agrícola, etc. que permitirán dar solución a problemas de movilidad y alcance.

1.2 DESCRIPCIÓN DEL PROYECTO

El presente proyecto tiene como finalidad desarrollar un HMI móvil en las plataformas Android e IOS, que permita el monitoreo de variables climáticas de la red inalámbrica de sensores de un invernadero de producción agrícola de tomate riñón implementado en la ciudad de Salcedo con visualización de datos históricos y monitoreo en tiempo real.

Los datos de las variables climáticas de las redes ZIGBEE, WI-FI y DIGIMESH serán receptados por el software LabVIEW, seguido a esto se realizará la conexión a una base de datos del servidor web, para la obtención de datos se ejecutarán consultas generadas por un código PHP cuya respuesta será visualizada en un Smartphone o Tablet con acceso a internet.

1.3 JUSTIFICACIÓN

La Universidad de la Fuerzas Armadas ESPE Extensión Latacunga se encuentra desarrollando el proyecto de investigación titulado “MONITOREO DE VARIABLES AMBIENTALES MEDIANTE UNA RED DE ESTACIONES METEOROLÓGICAS EN LA PROVINCIA DE COTOPAXI” que tiene como objetivo monitorear las variables ambientales en diferentes zonas de la provincia.

El proyecto de investigación consta de dos partes, la etapa inicial se está ejecutando titulada “DISEÑO E IMPLEMENTACION DE UNA RED WSN (WIRELESS SENSOR NETWORK) BASADO EN LOS PROTOCOLOS ZIGBEE, WIFI Y ZIGBEE MESH, PARA EL MONITOREO DE VARIABLES CLIMATICAS EN EL INVERNADERO UBICADO EN EL BARRIO RUMIPAMBA DEL NAVAS, CANTON SALCEDO, PROVINCIA DE COTOPAXI” que tiene como objetivo diseñar e implementar una red WSN basado en los protocolos ZigBee, Wifi y

ZigBee Mesh, para el monitoreo de variables climáticas en un invernadero. Como una etapa complementaria a este proyecto se presenta esta propuesta que se enfoca en el **DESARROLLO DE UN SISTEMA HMI EN LAS PLATAFORMAS ANDROID E IOS PARA EL MONITOREO DE LA RED INALÁMBRICA DE SENSORES AGRÍCOLAS.**

La flexibilidad de la tecnología Smartphone permite la instalación de aplicaciones para acelerar el procesamiento de datos. En conjunto con las comunicaciones inalámbricas se han desarrollado aplicaciones con diferente software con o sin licencia creados por el fabricante del dispositivo o por el usuario acorde a su necesidad.

En el Ecuador esta tecnología aún no se implementada en el ámbito de la industria y agrícola, con el cambio de matriz productiva, las empresas requieren sistemas de comunicación inalámbrica.

Con el desarrollo de un sistema HMI móvil en las plataformas Android e iOS permitirá el monitoreo de las variables de un proceso de tal manera estar informado del comportamiento del proceso en tiempo real, con la ayuda de un servidor web se accederá a la información desde cualquier lugar geográfico.

Además evitara que los técnicos y operadores realicen actividades innecesarias al momento de trasladarse hacia un panel remoto que contenga la información del proceso, de esta manera la actividad que se la esté ejecutando en ese momento no deba ser paralizada. Dentro de las ventajas de la comunicación inalámbrica es la no existencia de cables, garantiza la transmisión de datos a mayor velocidad y menor coste de instalación.

1.4 COMUNICACIÓN MÓVIL

Las comunicaciones móviles fueron desarrolladas a partir del crecimiento de la tecnología y durante la segunda guerra mundial, dio origen al primer sistema de comunicación móvil, de tipo cerrado o privado, al establecer una comunicación entre dos usuarios al menos uno de ellos este en movimiento, donde tal

comunicación solo se logran si la frecuencia de radio empleada es la misma para todos los usuarios, cuyo propósito es acortar distancia y en el menor tiempo posible. [1] Además con la presencia del internet o telefonía móvil es posible conectar un dispositivo en cualquier lugar geográfico.

En la actualidad una comunicación móvil se puede llevar a cabo mediante diferentes tecnologías adaptadas como antenas, satélites, repetidoras de señal, para que la comunicación se enlace debe constar con un receptor y un emisor.

1.4.1 TIPOS DE DISPOSITIVOS MÓVILES

Existen varios métodos para realizar una comunicación tanto inalámbrica como alámbrica, la tecnología ha ido evolucionando al pasar de los años lo cual existen diversos tipos de dispositivos que hoy en día se ha convertido en un instrumento muy útil debido a la fácil comunicación entre personas, a continuación se detallara brevemente los diferentes dispositivos:

a) RELOJ INTELIGENTES

Un smartwatch es un reloj pulsera que incorpora un conjunto de componentes electrónicos junto con un microprocesador capaz de ejecutar aplicaciones informáticas gracias a las cuales el propietario puede acceder a internet, consultar su correo electrónico, realizar y recibir llamadas, monitorizar y cuantificar su actividad diaria, ejecutar determinadas aplicaciones es decir, es un reloj computarizado que ofrece mucho más que indicar la hora y el día.

b) NETBOOK

Una netbook es una computadora clasificada como sub-portátil dado que dispone de un bajo costo y dimensiones reducidas, es similar a la laptop o notebook, por tener funcionalidad y dimensiones reducidas dispone de un costo

menor, a la vez permite mayor movilidad y más disponibilidad. Las netbooks se utilizan para navegar la Web y para otras operaciones básicas como procesadores de texto, requieren de un sistema operativo sencillo y confiable sin la alta gama. La netbook más común utiliza un sistema operativo convencional como Windows o Linux. Las principales características de las netbooks son:

- Pantalla: entre 14 o 15 pixeles
- Teclado
- Procesador: entre 1 y 2 GHz
- RAM: 2 GB
- Batería
- Conectividad: inalámbrica Wifi, conexión Bluetooth 2.0
- Puertos: 2 USB como mínimo
- Disco duro: el mínimo sería de 160GB
- Webcam y micrófono integrado

c) TABLET

Un Tablet PC es una computadora portátil que interactúa a través de una pantalla táctil o Multitáctil, sin necesidad de teclado físico, o mouse. Permiten realizar varias tareas simultáneamente, un alto rendimiento multimedia y una navegación por internet continuo y sin riesgos. [2] Las principales características de una Tablet son:

- Lectura de libros electrónicos
- Lectura sin conexión de páginas web.
- Lectura de cómics
- Consulta y edición de documentos de suites ofimáticas
- Navegación web (mediante Wi-Fi, USB o 3G interno).
- Llamadas telefónicas, si son 3G, sustituyendo así al teléfono móvil; se suele utilizar un manos libre bluetooth

- GPS
- Reproducción de música
- Visualización de vídeos y películas, cargadas desde la memoria interna, memoria o disco duro USB o Wi-Drive y con salida mini-HDMI.
- Cámara fotográfica y de vídeo HD
- Videoconferencia

d) SMARTPHONE

Son teléfonos celulares con computadoras integradas, difieren de otros teléfonos celulares por contar con un sistema operativo complejo, capaces de ejecutar una diversidad de aplicaciones de software, permite instalar programas los cuales permite ampliar las funcionalidades y mejorando las características del equipo más allá de lo que entrega el fabricante y proporcionar acceso a la Web. Envían y reciben correos electrónicos, tienen memoria integrada, así como un teclado completo. [2] Las principales características de un Smartphone son:

- Soporta correo electrónico
- Cuenta con GPS
- Permiten la instalación de programas de terceros.
- Utiliza cualquier interfaz para el ingreso de datos.
- Permiten ingresar a Internet y como a redes sociales.
- Poseen agenda digital, administración de contactos
- Permite leer documentos en distintos formatos, entre ellos los PDFs y archivos de Microsoft Office.
- Alta velocidad de procesamiento
- Gran espacio de almacenamiento y memoria ampliable
- Cámara con alta resolución.
- Soporte de banda para mayor cobertura
- Incorporan diversos sensores

1.5 PLATAFORMA ANDROID

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos software de código abierto.

Android es un sistema operativo orientado a dispositivos móviles que usa una versión modificada de kernel¹ de Linux. Esta plataforma permite el desarrollo de aplicaciones a través del SDK², y mediante el uso del lenguaje de programación Java, lo diferencia de los otros sistemas operativos que cualquier persona puede crear nuevas aplicaciones o incluso modificar el sistema operativo, dado que Android es código libre. [3]

1.5.1 ARQUITECTURA

En la Figura 1.1 se observa cada una de las capas que forma parte del Kernel de Linux, donde Android puede acceder a diferentes controladores, librerías creadas, administradores de recursos, y las aplicaciones a las que tiene acceso [3]. A continuación se detalla cada capa de la arquitectura de Android.

¹ Es el responsable de facilitar a los distintos programas el acceso seguro al hardware del ordenador.

² Software Development Kit, es un conjunto de herramientas de desarrollo que permite a un programador desarrollar aplicaciones para un sistema concreto.

a) APLICACIONES

Incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario, nativas y las administradas.

b) MARCO DE TRABAJO DE APLICACIONES

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades, entre las APIs más importantes son las siguientes:

- **Activity Manager.** Se encarga de administrar la pila de actividades de la aplicación así como su ciclo de vida.
- **Windows Manager.** Se encarga de organizar lo que se mostrará en pantalla.
- **Content Provider.** Crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
- **Views.** En Android, las vistas los elementos que ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.
- **Notification Manager.** Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Un dato importante es que esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.
- **Package Manager.** Permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con paquete se refiere a la forma en que se distribuyen

las aplicaciones Android, estos contienen el archivo .apk, que a su vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.

- **Telephony Manager.** Permite realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
- **Resource Manager.** Gestiona todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o layouts.
- **Location Manager.** Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.
- **Sensor Manager.** Manipula los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
- **Cámara:** Con esta librería se puede hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.
- **Multimedia.** Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

c) BIBLIOTECAS

Se sitúan sobre el kernel la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Estas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que

codificarlas cada vez y garantizando que se llevan a cabo de la forma “más eficiente”. Entre las librerías incluidas habitualmente se encuentran:

- **OpenGL** motor gráfico
- **Bibliotecas multimedia** formatos de audio, imagen y video
- **Webkit** navegador
- **SSL** cifrado de comunicaciones
- **FreeType** fuentes de texto
- **SQLite** base de datos

d) **RUNTIME DE ANDROID**

Incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.

e) **KERNEL DE LINUX**

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. También actúa como capa de abstracción entre el hardware y el resto de la pila de software. [4]

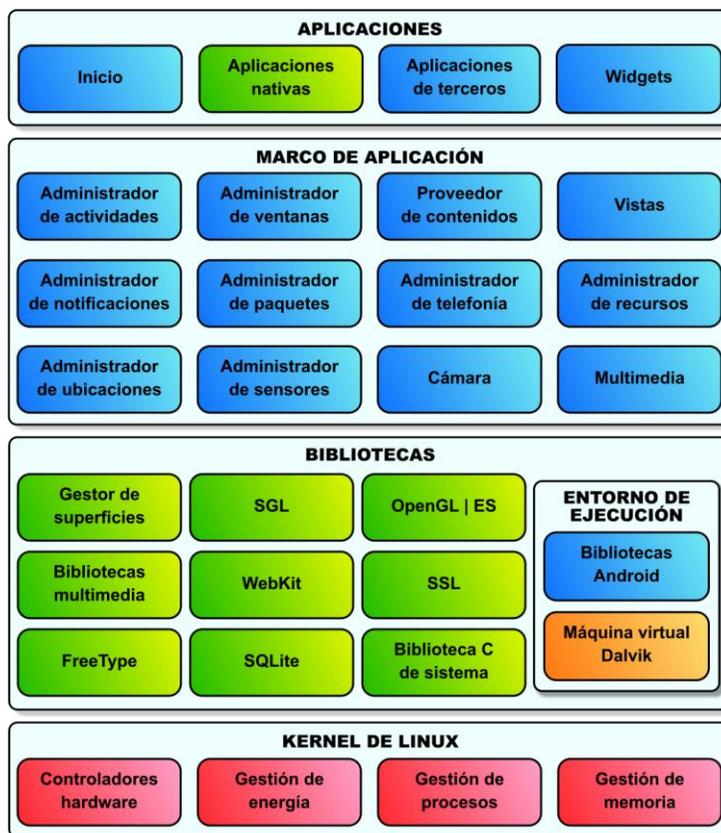


Figura 1.1 Arquitectura de Android [4]

1.5.2 CARACTERÍSTICAS PRINCIPALES

Se trata de un sistema operativo abierto multitarea, permite a los desarrolladores acceder a todas las funcionalidades del dispositivo. Todas las aplicaciones son reemplazables, el sistema operativo no diferencia entre las aplicaciones básicas del teléfono y las aplicaciones de terceros.

Las características principales del sistema operativo Android son:

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.

- Utiliza SQLite³ para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit⁴ incluido.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
- Catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas (Google Play).
- Bluetooth.
- Google Talk desde su versión HoneyComb, para realizar video llamada
- Multitarea real de aplicaciones. [4]

1.5.3 VERSIONES

La Tabla 1.1, muestra como las versiones de Android han evolucionado con cada iteración de Android, un nuevo SDK está disponible para desarrolladores y diversas nuevas características se añaden a dispositivos compatibles.

³ Es un sistema de gestión de bases de datos relacional

⁴ Es una plataforma para aplicaciones que funciona como base para el navegador web Safari, Google Chrome

Tabla 1.1

Versiones de Android

| Versión | Liberación |
|---------------------------|--|
| Astro (1.0) | Fue liberado en septiembre de 2008, sobre el HTC Dream. Mostró varias características básicas del sistema operativo Android, incluyen Android Market, un navegador web, correo electrónico / Gmail, Google Maps, Mensajería, Media Player, YouTube, y varios otros |
| Cupcake (1.5) | Fue liberado el 30 de abril del 2009. Los principales cambios fueron el apoyo a teclados virtuales, soporte para widgets en la pantalla de inicio, animaciones añaden en diversos lugares, y de auto-enlace y apoyo estéreo para dispositivos compatibles con Bluetooth. |
| Donut (1.6) | Fue liberado el 15 de septiembre de 2009, Las principales características incluyen la búsqueda de voz y de texto de contactos, soporte para pantallas WVGA, y mejoras en la funcionalidad de la cámara y velocidad. |
| Éclair (2.0 / 2.1) | Fue liberado el 26 de octubre 2009, que continuó siendo construido sobre el kernel de Linux versión 2.6.29, incluyendo mejoras de velocidad significativas en diferentes aplicaciones. |

CONTINÚA 

| | |
|-----------------------------------|---|
| Froyo (2.2.x) | Fue liberado el 20 de mayo de 2010, la versión de Android SDK 2.2 (Froyo). Sus principales características se han añadido a Froyo, incluyendo Adobe Flash apoyo, Nube Android de mensajería de dispositivos, funcionalidad hotspot Wi-Fi, y significativo optimizaciones de rendimiento. |
| Gingerbread (2.3.x) | Fue liberado el 6 de diciembre de 2010 y se basa en el kernel de Linux 2.6.35. Una de las características más significativas introducidas fue el apoyo abierto de accesorios, permite una función del dispositivo compatible como un periférico USB a plataformas de software compatibles. |
| Honeycomb (3.x) | Fue liberado en febrero de 2011, Honeycomb, fue lanzado en la Xoom Motorola fue creado específicamente para dispositivos de tablet. Esto incluyó un rediseño del teclado en pantalla, una barra del sistema para permitir el rápido acceso a las notificaciones y navegación, múltiples pestañas del navegador. |
| Ice Cream Sandwich (4.0.x) | Fue liberado el 19 de octubre de 2011 y se basó en la Kernel Linux 3.0.1. Las características incluyen un programa de inicio personalizado, con pestañas navegador web, el reconocimiento facial para desbloquear el dispositivo, un editor de fotos integrado, hardware aceleración de la interfaz de usuario. |

CONTINÚA 

| | |
|---------------------------|---|
| Jelly Bean (4.1.x) | Fue liberado el 9 de julio de 2012 y se basa en el kernel de Linux 3.1.10. Jelly Bean dio a conocer mejoras de rendimiento a la interfaz de usuario y de audio dentro de Android. Versión 4.2, características de las versiones Jelly Bean incluyen personalizable distribuciones de teclado, notificaciones expandibles, filtrado de notificación específica de la aplicación, y audio multicanal. |
| KitKat (4.4.x) | Fue liberado el 3 de septiembre de 2013. Sus características incluyen optimizaciones de rendimiento para los dispositivos con menos RAM, se expandieron APIs de accesibilidad, capacidad de impresión inalámbrica, y un nuevo tiempo de ejecución experimental virtuales máquina, llamada ART. |

Fuente: [3]

1.5.4 SOFTWARE PARA EL DESARROLLO DE APLICACIONES

A continuación se enumera diferentes Softwares que existen de tipo comercial gratuito y pagadas para el desarrollo de aplicaciones en la plataforma Android que son:

- a) **Basic4Android:** Es una plataforma de programación para aplicaciones Android cuyo lenguaje base de programación es VisualBasic, el eterno rival de Java. No es el mismo lenguaje de Microsoft, pero su sintaxis es la misma.

- b) **Mono para Android:** Es un lenguaje creado por Microsoft para hacer aplicaciones en C# y .NET, las cuales son muy usados en diferentes ambientes.

- c) App Inventor:** Esta plataforma basada en un lenguaje de desarrollo gráfico no se necesita ni una sola línea de código, tan solo arrastrar los bloques identificados con la acción que se necesita hacer. Usa el navegador como centro principal de trabajo, y almacena en servidores que están disponibles con acceso internet.
- d) Livecode** Herramienta para el desarrollo de aplicaciones potentes para escritorio, base de datos de Internet, y aplicaciones de red. LiveCode incluye un completo constructor de interfaz gráfica de usuario, así como las herramientas que necesarias para producir aplicaciones. Esta plataforma permite desarrollar las aplicaciones para las plataformas tanto para Android, iOS, Windows, Linux, iPhone, iPad, Web y para Servidores con una sola plataforma de trabajo.
- e) Xamarin** Es una herramienta para los desarrolladores de aplicaciones móviles posee la capacidad que el desarrollador escriba su app en lenguaje C# y el mismo código sea traducido para ejecutarse en iOS, Android y Windows Phone, además tiene una implementación libre de la plataforma de desarrollo .NET para dispositivos Android, iOS y GNU/Linux.
- f) Eclipse** Es un software informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE⁵ de Java llamado Java Development Toolkit (JDT)

⁵ Ambiente de desarrollo interactivo

y el compilador (ECJ). Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge [5].

1.6 PLATAFORMA iOS

iOS tiene sus inicios el 29 de junio de 2007, con el lanzamiento del primer iPhone y fue el primer dispositivo en usar iOS, a partir de su lanzamiento ha renovado cada año para aportar nuevas funcionalidades. Sistema operativo móvil de Apple Inc. Desarrollado para sus teléfonos iPhone posteriormente se ha empleado en dispositivos iPod Touch, iPad y Apple TV. iOS no es posible instalar en dispositivos que no pertenezcan a la compañía Apple, debido a que su código es cerrado, es derivado de Mac OS X, que a su vez está basado en Darwin BSD además la interfaz iOS se base en gestos multitáctiles. [6]

1.6.1 ARQUITECTURA

La arquitectura iOS está basada en capas como se muestra en la Figura 1.2, donde las capas más altas contienen los servicios y tecnologías más importantes para el desarrollo de aplicaciones, y las capas más bajas controlan los servicios básicos. A continuación se detalla las capas de la arquitectura de iOS.

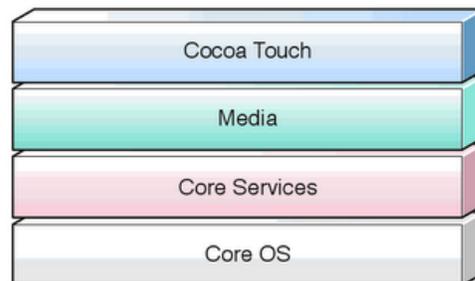


Figura 1.2 Arquitectura iOS [7]

- a) **Cocoa Touch** Es la capa más importante para el desarrollo de aplicaciones iOS. Posee un conjunto de Frameworks que proporciona el API⁶ de Cocoa para desarrollar aplicaciones. Se podría decir que Cocoa Touch proviene de Cocoa, la API ya existente en la plataforma MAC.
- b) **Media** Provee los servicios de gráficos y multimedia a la capa superior.
- c) **Core Services** Contiene los servicios fundamentales del sistema que usan todas las aplicaciones.
- d) **Core Os** Contiene las características de bajo nivel: ficheros del sistema, manejo de memoria, seguridad, drivers del dispositivo. [7]

1.6.2 CARACTERÍSTICAS PRINCIPALES

El sistema operativo es cerrado que no permite a los desarrolladores acceder a todas las funcionalidades del dispositivo, no permite instalar APP que no pertenezcan a la compañía Apple. Las características principales del sistema operativo iOS son:

- Código cerrado
- Interfaz gráfica está diseñada para el touch screen
- Capacidad para gestos multitouch.
- Interfaz está constituida básicamente de sliders, interruptores y botones, con una respuesta inmediata y fluida.
- Soporta acelerómetros internos al dispositivo.
- No tiene soporte para Adobe Flash ni Java
- Emplea unos 500 MB de almacenamiento
- Soporta multitarea

⁶ Application Programming Interface es el conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos.

1.6.3 VERSIONES

En la Tabla 1.2 se indica las versiones de iOS desde sus inicios, y su evolución en cuanto a características y aplicaciones.

Tabla 1.2

Versiones iOS

| Versiones | Liberación |
|----------------|---|
| iOS 1.0 | Fue liberado el 29 de Junio de 2007, con aplicaciones principales: iPod, Calendario, Fotos, reloj, texto, Notas, calculadora, Mail, Maps, Configuración, cámara, YouTube, acciones, Teléfono y safari. |
| iOS 2.0 | Fue liberado el 11 de julio de 2008, el apoyo de App Store y iOS de 2.0 sus características son: la capacidad de abrir documentos de MSOffice, un icono Contactos, capacidad de capturas de pantalla, guardar fotos en Safari. |
| iOS 3.0 | Fue liberado el 17 de Junio de 2009, iOS 3.0 trajo la posibilidad de cortar, copiar y pegar. La precisión del GPS también se ha mejorado en gran medida, y Apple ha añadido una brújula magnética para aplicación Mapas del iPhone. |

CONTINÚA



| | |
|----------------|--|
| iOS 3.2 | Fue liberado el 3 de Abril de 2010 a fin de llevar iOS en el iPad. Esta actualización fue la primera vez que los usuarios podían cambiar su fondo de pantalla de inicio, a de más la pantalla toma el modo horizontal en lugar del modo de inicialización que se manejó en el iPhone. |
| iOS 4.0 | Fue liberado el 17 de Junio del 2010, la principal característica de iOS 4 fue el video chat capacidades de FaceTime, iBooks, que fue la primera vez en ofrecer soporte nativo para libros electrónicos en dispositivos iOS y se añadió un diccionario personalizado. |
| iOS 5.0 | Fue liberado el 6 de Junio de 2011. Con más de 200 nuevas características de iOS 5 la mayor actualización de iOS Apple, sistema de notificación de iOS, iMessage con iOS 5, cámara con posibilidad de editar fotos, ha añadido una función de "leer más tarde" a Safari en iOS 5 sincronización inalámbrica. |
| iOS 6.0 | SD Fue liberado el Septiembre de 2012 con características de Passbook que almacena cupones de los usuarios, tarjetas de embarque, entradas de cine, integración de Facebook, llamadas FaceTime ahora se pueden hacer a través de una red de teléfonos móviles en lugar de Wi-Fi. |
| iOS 7 | Fue liberado el 10 de junio de 2013 con nuevos entornos en el centro de control ha sido una de las adiciones más esperadas con su capacidad para cambiar rápidamente Wifi, Bluetooth, modo avión, y una serie de otros valores, así como poner en marcha rápidamente su linterna, calculadora o la cámara. |

Fuente: [6]

1.6.4 SOFTWARE PARA EL DESARROLLO DE APLICACIONES

A continuación se enumera diferentes Softwares que existen de tipo comercial gratuito y pagado para el desarrollo de aplicaciones en la plataforma iOS que son:

- g) Objective C Y Cocoa:** Son, respectivamente, un lenguaje de programación y una “librería de clases” (conjunto de herramientas que acompaña al lenguaje) creados y proporcionados por Apple. Es lo que usa Apple para crear sus propias aplicaciones, tanto para iOS (iPhone e iPad) como para el mac. El coste de estas herramientas es nulo, Apple lo proporciona de forma gratuita con cada mac. Se trata de un lenguaje muy potente y un conjunto de herramientas muy completas. Para programar en Objective C y Cocoa, se utiliza un entorno de programación llamado Xcode, que también es proporcionado libre de coste por Apple.

- a) Corona Sdk Y Luna:** Es una serie de herramientas creadas por Anasca Mobile para desarrollar apps para iPhone, iPad y Android, con un énfasis especial en los juegos y las apps sencillas. Corona es un SDK alternativo especialmente indicado para juegos sencillos, el lenguaje utilizado es Lua, un lenguaje muchísimo más sencillo que Objective C y muy utilizado para juegos (como World of Warcraft).

- b) Livecode:** Herramienta que permite el desarrollo de aplicaciones potentes. Para escritorio, base de datos de Internet, y aplicaciones de red. LiveCode incluye un completo constructor de interfaz gráfica de usuario, así como las herramientas que necesarias para producir aplicaciones. Esta plataforma permite desarrollar las aplicaciones para las plataformas tanto para Android, iOS, Windows, Linux, iPhone, iPad, Web y para Servidores con una sola plataforma de trabajo.

c) **Xamarin** Es una herramienta para los desarrolladores de aplicaciones móviles, y la novedad de esta herramienta es la capacidad que tiene para que el desarrollador escriba su app en lenguaje C# y el mismo código sea traducido para ejecutarse en iOS, Android y Windows Phone, además tiene una implementación libre de la plataforma de desarrollo .NET para dispositivos Android, iOS y GNU/Linux. Es decir, con Xamarin se podrá evitar tener que utilizar Java para desarrollar una aplicación para Android llamada MonoTouch y MonoDroid [8]

1.7 COMPARACIÓN ENTRE SISTEMAS OPERATIVOS MÓVILES

En la Tabla 1.3 se muestra una breve comparación entre los sistemas operativos para Smartphone's con sus principales características:

Tabla 1.3

Comparativa de sistemas operativos para Smartphone's

| | Android | BlackBerry | iOS | Windows phone | Firefox OS |
|-------------------------------|----------------------|------------|--------|---------------|------------|
| Familia del SO | Linux | Mobile OS | Darwin | Windows Nt | Linux |
| Arquitectura Soportada | ARM, MIPS, X86 | ARM | ARM | ARM | ARM |

CONTINÚA



| | | | | | |
|--|------------------|------------------|------------------------|-----------------------------|--------------------------------------|
| Programado | C++, Java | C, C++ | C; C++, Objective-C | XNA, NET, C++, Vb.net | C++, JavaScript, HTMLs, Css |
| Servicio Remoto de Localización GPS | No dispone | No dispone | Si | Si | No dispone |
| Tipo de Interfaz | Iconos y Widgets | Baldosas e Icoos | Iconos | Baldosas Animadas | Iconos |
| Sincronización con la nube | Google Driver | No de serie | Icloud | SkyDriver | SI |
| Tienda de Aplicaciones | Google Play | App World | App Store | Windows Phone Store | Firefox Market Place |
| Mensajería | Google Talk | BBM | iMessage | Skype | No de serie |

Fuente: [9]

Lo más esencial e importante en un dispositivo Smartphone es su sistema operativo, arquitectura y aplicaciones de accesorios. Las plataformas Android de Google, iPhone de Apple y BlackBerry de RIM se encuentran hoy en día liderando el mercado.

Cada plataforma tiene su filosofía, interfaz y funcionamiento de distribución del software, sus posturas han sido muy similares. Sin embargo, Android ha contado con un apoyo crítico al ser un sistema operativo abierto por usar un núcleo kernel Linux, con una mezcla especial de Java. El iPhone se basa en Darwin, que a su vez es una variante de Unix, uno de los sistemas operativos más poderosos en el mundo de la informática.

1.8 SERVIDOR WEB

Un servidor web recibe peticiones de un cliente o usuario de internet, emitiendo respuestas mediante el envío de ficheros solicitados, implementados con protocolos o binarios y estándares para intercambiar información y aplicaciones. Estos protocolos o estándares permiten escuchar las peticiones de conexión de los clientes en determinados puertos: 80 para HTTP⁷, 443 para el HTTPS⁸, que pertenece a la capa de aplicación del modelo OSI.

La atención a la petición como se muestra en la Figura 1.3 Servidor Web como del cliente consiste en buscar el archivo solicitado, si lo encuentra, está diseñado para transferir hipertextos, páginas web, caso contrario envía un mensaje de error. [10]

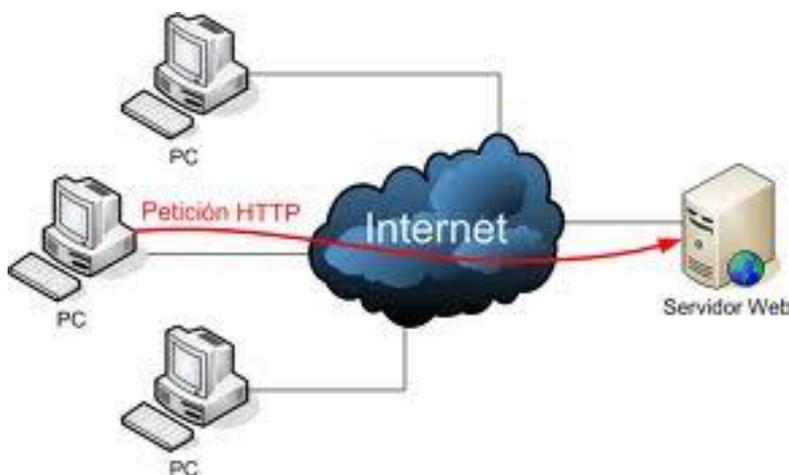


Figura 1.3 Servidor Web [10]

⁷ Hypertext Transfer Protocol permite a los componentes web comunicarse de una forma estándar y definida.

⁸ Hypertext Transfer Protocol Secure destinado a la transferencia segura de datos de Hipertexto

1.8.1 TIPOS DE SERVIDORES

En la Tabla1.4 se describe las características principales de los diferentes tipos de servidores utilizados.

Tabla1.4

Tipos de servidores

| DENOMINACIÓN DEL SERVIDOR | DESCRIPCIÓN |
|----------------------------------|--|
| Servidor de Correo | Es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes. |
| Servidor Proxy | Es el servidor que actúa de intermediario de forma que el servidor que recibe una petición no conoce quién es el cliente que verdaderamente está detrás de esa petición. |
| Servidor Web | Almacena principalmente documentos HTML (son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes), imágenes, videos, texto, presentaciones, y en general todo tipo de información. |
| Servidor de Base de Datos | Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que almacena grandes cantidades de información. |

CONTINÚA



| | |
|-------------------------------|---|
| Servidores Clúster | Son servidores especializados en el almacenamiento de la información teniendo grandes capacidades de almacenamiento y permitiendo evitar la pérdida de la información por problemas en otros servidores. |
| Servidores Dedicados | Son exclusivos para una sola persona o empresa. |
| Servidores de imágenes | Recientemente también se han popularizado servidores especializados en imágenes, permitiendo alojar gran cantidad de imágenes sin consumir recursos del servidor web en almacenamiento o para almacenar fotografías personales. |

Fuente: [10]

1.8.2 ARQUITECTURA CLIENTE SERVIDOR

La arquitectura cliente-servidor se basa en que el cliente realiza una petición para solicitar un servicio a otros programas, el encargado de dar respuesta es el servidor. El cliente no posee control sobre los recursos, al contrario el servidor es el encargado de manejarlos. Por otro lado, el ordenador remoto actúa como servidor evalúa la petición del cliente y decide aceptarla o rechazarla consecuentemente. Una vez que el servidor acepta el pedido la información requerida es suministrada al cliente que efectúa la petición, proporcionando los datos con el formato adecuado [5].

1.8.3 MODELO OSI

En modelo OSI (Open System Interconnection) es el modelo principal para las comunicaciones en red, creado por el ISO (Organización Internacional de Normalización), consiste en siete capas como se indica en la Figura 1.4, donde cada una de cumple una función y proporciona los protocolos con el propósito de

intercambiar información entre varios sistemas. [11] A continuación se detalla la función de cada capa del modelo OSI.



Figura 1.4 Capas del modelo OSI [11]

- **Capa de aplicación** Proporciona la interfaz y servicios que soportan las aplicaciones de usuario, ofrece los servicios de red relacionados como la gestión de mensajes, la transferencia de archivos y las consultas a base de datos.
- **Capa de presentación** Puede considerarse el traductor del modelo OSI, toma los paquetes de la capa de aplicación y los convierte a un formato genérico que pueden leer todas las computadoras.
- **Capa de sesión** Es la encargada de establecer el enlace de comunicación o sesión y también de finalizarla entre las computadoras emisora y receptora. Esta capa ubica puntas de control en la secuencia de datos

además proporciona cierta tolerancia a fallos dentro de la sesión de comunicación

- **Capa de transporte** Es la encargada de controlar el flujo de datos entre los nodos que establecen una comunicación; los datos no solo deben entregarse sin errores, sino además en la secuencia que proceda.
- **Capa de red** Entrega los paquetes y determina la ruta que deben seguir los datos se produce en esta capa, lo mismo que el intercambio efectivo de los mismos dentro de dicha ruta.
- **Capa de enlace de datos** Se encarga de desplazar los datos por el enlace físico de comunicación hasta el nodo receptor, e identifica cada computadora incluida en la red de acuerdo con su dirección de hardware.
- **Capa física** Las tramas procedentes de la capa de enlace de datos se convierten en una secuencia única de bits que puede transmitirse por el entorno físico de la red.

1.8.4 TCP/IP

TCP/IP⁹ es el nombre de un protocolo de conexión de redes y comúnmente utilizados por todos los ordenadores conectados a Internet, la ventajas principales que contiene este protocolo es abierto, compatible con cualquier sistema operativo y con cualquier tipo de hardware. Está diseñado para ser un componente de una red y se encargará de que la comunicación entre todos sea posible. En Internet se diferencian cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI como se indica en la Figura 1.5.

⁹ Protocolo de control de transmisión/Protocolo de Internet

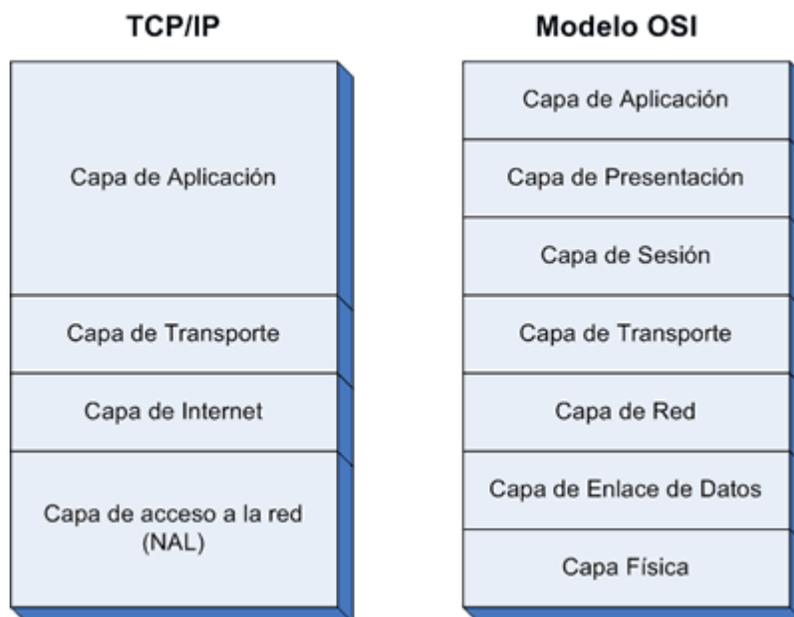


Figura 1.5 Modelo TCP/IP relacionado con el modelo OSI [12]

- **Capa de acceso a la red:** Especifica la forma en la que los datos deben enrutarse, sea cual sea el tipo de red utilizado
- **Capa de internet:** Es responsable de proporcionar el paquete de datos (datagrama);
- **Capa de transporte** Brinda los datos de enrutamiento, junto con los mecanismos que permiten conocer el estado de la transmisión
- **Capa de aplicación** Incorpora aplicaciones de red. [12]

1.8.5 MEDIOS DE ACCESO A INTERNET

Internet es un conjunto descentralizado de redes de comunicación interconectadas. En esta red de redes, existen muchas tecnologías diferentes comunicándose entre sí. Hoy día, los medios de comunicación más modernos, más rápidos y, en algunos casos, más accesibles. A continuación se describe los diferentes medios que se dispone para acceder al internet son [13]:

- **Línea telefónica:** El acceso a internet por medio de las líneas de teléfonos, es la manera más antigua de contactarse. La evolución de ese medio, fue en incrementos de la velocidad de acceso, hasta el límite de 56 K (56,000 bauds por segundo).
- **ADSL:** Este medio es el cumplimiento parcial de una promesa de los años setenta, en que los servicios de telefonía, televisión y radio, se iban a servir por medio de una única línea de fibra óptica. Conocido como Red Digital de Servicios integrados (ISDN, por sus siglas en inglés). En la actualidad, el DSL es el medio de acceso por medio de las líneas telefónicas, más rápidas y más costosas.
- **Cable Digital:** Este medio se logra a través de la misma línea que sirve la televisión por cable. A pesar que puede lograr unas velocidades de conexión muchos más rápidas que el dial-up, es comparable al ADSL.
- **Red Celular:** Para los consumidores domésticos, esta es la forma de acceso que se está popularizando rápidamente, por la conveniencia de la movilidad. Se accede al internet mediante la misma red de telefonía móvil. Originalmente, se logra por medio de un teléfono móvil, pero también se ofrece un dispositivo dedicado al único propósito de conectarse al internet.
- **Modem Banda Ancha:** También conocido como BAM, es un servicio de Internet Móvil que ofrece multitud de Proveedores de Telefonía móvil como Claro, Movistar y CNT en Ecuador. Esta tecnología permite obtener internet en cualquier lugar y momento, siempre que se disponga de cobertura móvil, y pueden ofrecer velocidades equiparables a las velocidades de la Banda Ancha por cable (entre 3 y 42 Mbps dependiendo del operador y del tipo de conexión (GPRS, 3G, HSDPA, HSPA+, LTE).

1.9 INTERNET

El internet es una red de redes que permite la interconexión descentralizada de computadoras a través de un conjunto de protocolos denominado TCP/IP,

cuando se hace referencia a Internet se habla de un conjunto de servicios y aplicaciones que permiten hacer un uso provechoso de la misma.

A nivel físico, Internet se compone de un gran número de máquinas distribuidas por todo el mundo y conectadas por los más diversos medios (cable, radio, satélites, etc.). Las máquinas u ordenadores que forman parte de internet no son iguales y pueden tener funciones diferentes. Se puede encontrar desde el PC cuya única función es la de consultar páginas web, el servidor en el que se almacenan las páginas web a consultar, ordenadores intermedios que administran el flujo de información a través de la red.

Actualmente internet es un gran entramado de páginas que se llaman unas a otras y que constituyen una red de relaciones inabarcable e incontenible por cualquier persona o máquina. Aunque resulte paradójico, no existe un directorio en el que se citen todas las páginas de internet. Debido a que Internet es un ente cambiante [14]

1.10 TECNOLOGÍA WIRELESS

La comunicación inalámbrica o wireless es aquella en la que los extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación física, sino que utilizan la modulación de ondas electromagnéticas a través del espacio. Los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales ese encuentra: antenas, notebook, PDA, teléfonos móviles, entre otros. [15] La tecnología wireless posee las siguientes ventajas:

- Brinda mayor comodidad permitiendo que cualquiera que tenga acceso a la red se conecte desde diferentes ubicaciones dentro de un área determinada.
- La instalación y configuración de una infraestructura de red inalámbrica es más sencilla y económica que la de una infraestructura cableada.

La tecnología inalámbrica posee las siguientes desventajas:

- Menor velocidad de transmisión que las redes cableadas a causa de las interferencias y pérdidas de señal.
- Debido a que las transmisiones no se realizan a través un medio físico las mismas son susceptibles a ataques en contra de su seguridad. Un atacante puede ser capaz de capturar paquetes e intentar obtener las contraseñas de red para poder tener acceso a ella.

1.11 PROTOCOLO 802.11 O WI-FI

Wi-Fi son las siglas de Wireless Fidelity y comprende una gran cantidad de estándares para redes de comunicación inalámbrica basados en las especificaciones IEEE 802.11. En sus inicios Wi-Fi fue pensado para conectar redes locales inalámbricas; sin embargo, actualmente se utiliza para el acceso a Internet. [16]

Con Wi-Fi se pueden crear redes de área local inalámbricas de alta velocidad siempre y cuando el equipo que se vaya a conectar no esté muy alejado del punto de acceso. En la práctica, Wi-Fi admite ordenadores portátiles, equipos de escritorio, asistentes digitales personales (PDA) o cualquier otro tipo de dispositivo de alta velocidad con propiedades de conexión también de alta velocidad (11 Mbps o superior) dentro de un radio de varias docenas de metros en ambientes cerrados (de 20 a 50 metros en general) o dentro de un radio de cientos de metros al aire libre. [17]

1.11.1 LOS DISTINTOS ESTÁNDARES WI-FI

El estándar 802.11 en realidad es el primer estándar y permite un ancho de banda de 1 a 2 Mbps. El estándar original se ha modificado para optimizar el ancho de banda (incluidos los estándares 802.11a, 802.11b y 802.11g, denominados estándares físicos 802.11) o para especificar componentes de

mejor manera con el fin de garantizar mayor seguridad o compatibilidad. La Tabla 1.5 se muestra los distintos estándares de 802.11 y sus significados [17].

Tabla 1.5

Estándares WIFI

| Estándar | Descripción |
|-----------------|--|
| 802.11a | El estándar 802.11 (llamado WiFi 5) admite un ancho de banda superior (el rendimiento total máximo es de 54 Mbps aunque en la práctica es de 30 Mbps). El estándar 802.11a provee ocho canales de radio en la banda de frecuencia de 5 GHz. |
| 802.11b | El estándar 802.11 es el más utilizado actualmente. Ofrece un rendimiento total máximo de 11 Mbps (6 Mbps en la práctica) y tiene un alcance de hasta 300 metros en un espacio abierto. Utiliza el rango de frecuencia de 2,4 GHz con tres canales de radio disponibles. |
| 802.11c | El estándar combinado 802.11c no ofrece ningún interés para el público general. Es solamente una versión modificada del estándar 802.1d permite combinar el 802.1d con dispositivos compatibles 802.11 (en el nivel de enlace de datos). |
| 802.11d | El estándar 802.11d es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información en rangos de frecuencia según lo que se permite en el país de origen del dispositivo. |

CONTINÚA



| | |
|----------------|---|
| 802.11 | El estándar 802.11e está destinado a mejorar la calidad del servicio en el nivel de la capa de enlace de datos. El objetivo del estándar es definir los requisitos de diferentes paquetes en cuanto al ancho de banda y al retardo de transmisión para permitir mejores transmisiones de audio y vídeo. |
| 802.11f | El 802.11f es una recomendación para proveedores de puntos de acceso permite que los productos sean más compatibles. Utiliza el protocolo IAPP que le permite a un usuario itinerante cambiarse claramente de un punto de acceso a otro mientras está en movimiento sin importar qué marcas de puntos de acceso se usan en la infraestructura de la red. |
| 802.11g | El estándar 802.11g ofrece un ancho de banda elevado (con un rendimiento total máximo de 54 Mbps pero de 30 Mbps en la práctica) en el rango de frecuencia de 2,4 GHz. El estándar 802.11g es compatible con el estándar anterior, el 802.11b, lo que significa que los dispositivos que admiten el estándar 802.11g también pueden funcionar con el 802.11b. |
| 802.11h | El estándar 802.11h tiene por objeto unir el estándar 802.11 con el estándar europeo (HiperLAN 2, de ahí la h de 802.11h) y cumplir con las regulaciones europeas relacionadas con el uso de las frecuencias y el rendimiento energético. |
| 802.11i | El estándar 802.11i está destinado a mejorar la seguridad en la transferencia de datos (al administrar y distribuir claves, y al implementar el cifrado y la autenticación). Este estándar se basa en el AES (estándar de cifrado avanzado) y puede cifrar transmisiones que se ejecutan en las tecnologías 802.11a, 802.11b y 802.11g. |
| 802.11r | El estándar 802.11r se elaboró para que pueda usar señales infrarrojas. Este estándar se ha vuelto tecnológicamente obsoleto. |
| 802.11j | El estándar 802.11j es para la regulación japonesa lo que el 802.11h es para la regulación europea. |

Fuente: [17]

1.11.2 RANGOS IEEE 802.11

Los estándares 802.11a, 802.11b, 802.11g y 802.11n, llamados "estándares físicos", son modificaciones del estándar 802.11 y operan de modos diferentes, lo que les permite alcanzar distintas velocidades en la transferencia de datos según sus rangos como se muestra en la Tabla 1.6.

Tabla 1.6

Rangos IEEE 802.11/x

| | 802.11A | 802.11B | 802.11G | 802.11N |
|---|----------------|----------------|-----------------|-------------------|
| Fecha de aprobación estándar | 07 1999 | 07 1999 | 06 2003 | Octubre 2009 |
| Velocidad de datos máxima (Mbps) | 54 | 11 | 54 | ~ 600 |
| Modulación | OFDM | CCK o DSSS | CCK, DSSS, OFDM | CCK, DSSS, o OFDM |
| RF Banda (GHz) | 5 | 2.4 | 2.4 | 2,4 o 5 |
| Número de secuencias espaciales | 1 | 1 | 1 | 1, 2, 3, o 4 |
| Ancho de canal nominal (MHz) | 20 | 20 | 20 | 20, o 40 |

Fuente: [17]

1.12 TECNOLOGÍA MÓVIL 3G

Es un servicio de comunicaciones inalámbricas que permite estar conectado de forma permanente a Internet a través del teléfono móvil, el ordenador de bolsillo y el ordenador portátil. La tecnología 3G propone una mejor calidad y fiabilidad, una mayor velocidad de transmisión de datos y un ancho de banda superior. Con velocidades de datos de hasta 384 Kbps, es aproximadamente siete veces más rápida que una conexión telefónica estándar. Este estándar se desarrolló mediante un sistema móvil llamado UMTS (Universal Mobile Telephone System), este a su vez está desarrollado a partir de WCDMA, que es una tecnología móvil inalámbrica que aumenta las tasas de transmisión de datos de los sistemas GSM es por ello que 3G ofrece velocidades mucho más altas de datos en aparatos inalámbricos portátiles. [18] La tecnología 3G posee las siguientes ventajas

- Permite el acceso permanente al Internet en casi cualquier sitio.
- Permite acceso de información en casi cualquier parte de la ciudad donde se tenga cobertura.
- Posee una mejor calidad y fiabilidad, una mayor velocidad de transmisión de datos y un ancho de banda superior que permite tener video llamadas.
- Correo electrónico: Permite convertir los teléfonos celulares en oficinas móviles para recibir y enviar mensajes.
- La transmisión de voz tiene una calidad equiparable a la de las redes fijas.

La tecnología inalámbrica posee las siguientes desventajas:

- Cobertura limitada. Dependiendo de la localización, la velocidad de transferencia puede disminuir drásticamente (o incluso carecer totalmente de cobertura).
- Disminución de la velocidad si el dispositivo conectado está en movimiento
- El costo de infraestructura de la tecnología 3G es elevado.

1.13 HMI EN TELEFONÍA MÓVIL

El HMI¹⁰ móvil permite que el trabajo del operador sea mucho más eficiente. Ahorra tiempo al no tener que caminar hasta el panel de operador para controlar la máquina, para recibir alarmas o para comprobar el estado de la producción. Acciones que anteriormente requerían de múltiple personal para grandes máquinas o aplicaciones de larga distancia se pueden realizar fácilmente con un menor número de operadores.

Disponer de una HMI basada en la nube puede significar menores costes y mejorar la funcionalidad. Los usuarios pueden ver fácilmente los datos a través de los teléfonos inteligentes y Tablet PC. También reciben alertas a través de mensajes de texto SMS y correo electrónico.

1.14 RED DE SENSORES INALÁMBRICOS

Una red inalámbrica de sensores o WSN, es aquella conformada por dispositivos (nodos) de bajo consumo y costo moderado, distribuidos geográficamente como se muestra en la Figura 1.6. Estos tienen la capacidad de obtener información del entorno, procesarla a nivel local y comunicarla de manera inalámbrica a un nodo central que cumple la función de coordinación o administración de la red.

Los sensores instalados pueden ser fijos o móviles y controlan condiciones como temperatura, sonido, vibración, presión, movimiento, aceleración, entre otras.

Una de las cualidades más sobresalientes de este tipo de red es su capacidad para auto restaurarse. Si un nodo sufre algún daño y deja de funcionar de forma

¹⁰ Human Machine Interface se usa para referirse a la interacción entre humanos y máquinas; Aplicable a sistemas de Automatización de procesos.

repentina, la red encontrará nuevas rutas para continuar con el envío de los paquetes. [19]



Figura 1.6 Red de sensores inalámbricos. [19]

1.14.1 ELEMENTOS EN UNA RED WSN

A continuación se detalla los elementos de una red WSN

- a) **Sensores:** De distinta naturaleza y tecnología toman del medio la información y la convierten en señales eléctricas.
- b) **Nodos sensor o procesadores de radio:** Toman los datos del sensor a través de sus puertos de datos, y envían la información a la estación base.
- c) **Gateway:** Elementos para la interconexión entre la red de sensores y una red TCP/IP.
- d) **Estación Base:** Recolector de datos basados en un ordenador común o sistema embebido.
- e) **Red Inalámbrica:** Típicamente basada en el estándar 802.15.4 ZigBee.

1.15 ESTÁNDARES UTILIZADOS EN TECNOLOGÍAS INALÁMBRICAS

Las tecnologías inalámbricas disponen de varios estándares, que se diferencia una de otra en su alcance, su velocidad, su topología, etc. En este caso se mencionara tres tecnologías inalámbricas, a las que se podrá acceder a la información en tiempo real e histórico por la aplicación desarrollada.

1.15.1 TECNOLOGÍA ZIGBEE

a) DESCRIPCIÓN

ZigBee está definido por ZigBee Alliance, que es un grupo de empresas que trabaja en la creación de estándares de redes inalámbricas para radios digitales de baja potencia.

Dado que la especificación tiene por objeto permitir baja potencia y redes de bajo coste, las exigencias impuestas a la especificación son algo diferentes de otras configuraciones de red. Se debe considerar para el diseño, propiedades que afectan el consumo de energía y el coste total de la red. Otro aspecto importante de una red de este tipo es la seguridad [20].

b) CARACTERÍSTICAS

ZigBee es una tecnología inalámbrica relativamente nueva guiada por la IEEE, con el estándar 802.15.4 principalmente para redes de área personal con un alcance de 70 a 100m de distancia. En comparación con otras tecnologías de red, ZigBee tiene algunas características que lo convierten en una opción más adecuada para el sensado y redes de control:

- ZigBee utiliza una velocidad de transferencia de datos baja, pero a diferencia de Wifi, ZigBee fue diseñado principalmente para el monitoreo

de aplicaciones, que no requieren ningún alto tráfico la comunicación como el de video, etc.

- Este estándar opera en tres bandas de 2,4 GHz, 868, y 912 MHz, que tiene la velocidad de datos de 250Kbps, 20kbps y 40kbps respectivamente.
- ZigBee tiene una larga gama de enlaces inalámbricos y soporta un mayor número de dispositivos de Bluetooth. Por otra parte, los dispositivos ZigBee están diseñados para una duración de la batería extremadamente larga debido a ciclos de trabajo bajos.
- Implementado en el estándar ZigBee basado en las capas PHY (física) y MAC, las especificaciones de ZigBee busca introducir topologías de red fiables y seguras, incluyendo topologías como malla, estrella y árbol.
- Igualmente importante a destacar es que ZigBee puede ser considerado relativamente tolerante a fallos, en muchos casos implementa el concepto de red de malla, lo cual asegura que un nodo defectuoso no sea impedimento para que se interrumpa la comunicación de esta manera los datos pueden viajar a través de múltiples caminos [21].

c) DISPOSITIVOS ZIGBEE

Los dispositivos que conforman ZigBee son 3 y en la Figura 1.7 se muestran los tres dispositivos conectados entre si

- **Coordinador:** Es el dispositivo de mayor rango, el coordinador forma la raíz del árbol de la red y permite tender un puente a otras redes. Almacena información sobre la red, incluyendo la actuación como el centro y repositorio de confianza para las claves de seguridad.
- **Router:** Es el dispositivo que actúa como un enrutador, permitiendo la transmisión de datos entre los nodos sensores y coordinador.
- **Nodos sensores:** Es el dispositivo que está en contacto con las variables físicas y recolecta la información que se desea obtener. Un nodo sensor

requiere la menor cantidad de memoria, y por lo tanto puede ser menos caro de fabricar que los otros dos dispositivos mencionados anteriormente.

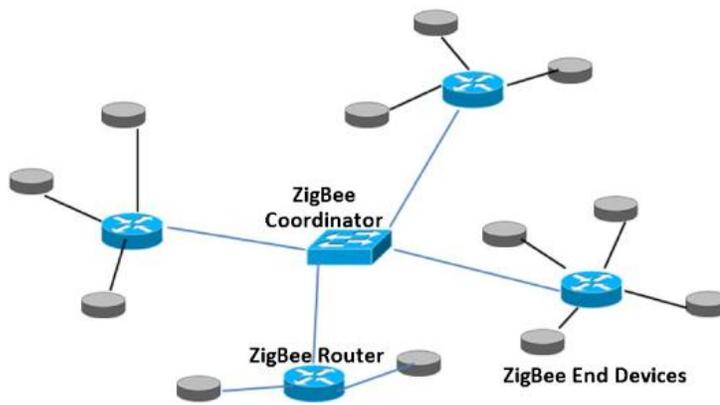


Figura 1.7 Dispositivos que conforman ZigBee.

1.15.2 TECNOLOGÍA DIGIMESH

a) DESCRIPCIÓN

DigiMesh es un derivado de la tecnología ZigBee que trabaja bajo topología malla, para su uso en soluciones de conectividad de punto final inalámbrica. La naturaleza de su arquitectura (Peer to Peer), punto a punto, permite a DigiMesh ser a la vez fácil de usar y disponer de equipos con funciones de red avanzadas, incluyendo soporte para dormir los router y equipos de la red, optimizando así la energía de la batería. [22]

b) CARACTERÍSTICAS

Cuenta con características tales como:

- La creación de redes de malla robusta
- Configuración de la red es más sencillo
- Más flexibilidad para ampliar la red

- Mayor fiabilidad en entornos en los router pueden aparecer y desaparecer debido a la interferencia o daños.
- Un protocolo de consumo de energía optimizado el uso de la batería, así como la potencia suministrada a los distintos dispositivos de la red
- Un protocolo fácil de usar que simplifica la creación de redes de malla (no hay necesidad de definir y organizar los coordinadores, router o nodos finales)
- La capacidad de desplegar soluciones inalámbricas tanto en 900 MHz y 2.4 GHz
- Trabaja bajo una sola topología que es Malla.

c) DISPOSITIVOS DIGIMESH

Sólo tiene un tipo de nodo, como una red homogénea, todos los nodos pueden transmitir datos e intercambiar información. Permitted eliminar el único punto de fallo asociado a depender de un coordinador o puerta de enlace. En su lugar, DigiMesh establece la sincronización de tiempo a través de un proceso de nominación y elección, permite a la red funcionar de forma autónoma. De igual manera que ZigBee dispondrá de nodo sensor, router y coordinador que fueron detallados anteriormente.

1.15.3 TECNOLOGÍA WIFI

Este estándar permite trabajar con velocidades de transmisión de 1 Mbps y 2 Mbps, como se mencionó en el Apartado 1.11 se detalla las características principales como versiones y rangos existentes del estándar IEEE 802.11.

a) DISPOSITIVOS WIFI

Una red WIFI está compuesta por elementos o dispositivos que permiten que la comunicación y envío de datos se realice de manera óptima y eficiente. En la Figura 1.8 se observa dichos dispositivos.



Figura 1.8 Elementos de una red WIFI.

- **Punto de acceso (AP):** Se encarga de proporcionar la unión entre la red WIFI y la red fija, además gestiona toda la información transmitida y la hace llegar a su destino.
- **Antenas:** Es el elemento utilizado para el envío de la información a través del aire en forma de ondas electromagnética, sirven para recepción y transmisión. Existen diversos tipos de antenas, y de acuerdo a la necesidad de la aplicación (alcance, medio físico, datos de transmisión, etc.), se escogerá un tipo.
- **Dispositivo externo WIFI:** Este elemento es una tarjeta de red de área local, que permite la conexión de un terminal de usuario en una red WIFI. Pueden conectarse a ranuras como puertos USB. [24]

CAPÍTULO II

SELECCIÓN DE SOFTWARE Y COMPONENTES

En el capítulo 2 se detalla las etapas principales que constituyen el desarrollo del proyecto (Figura 2.1) titulado “DESARROLLO DE UN SISTEMA HMI EN LAS PLATAFORMAS ANDROID E IOS PARA EL MONITOREO DE LA RED INALÁMBRICA DE SENSORES AGRÍCOLAS” que corresponde a la etapa complementaria del proyecto “DISEÑO E IMPLEMENTACION DE UNA RED WSN (WIRELESS SENSOR NETWORK) BASADO EN LOS PROTOCOLOS ZIGBEE, WIFI Y ZIGBEE MESH, PARA EL MONITOREO DE VARIABLES CLIMATICAS EN EL INVERNADERO UBICADO EN EL BARRIO RUMIPAMBA DEL NAVAS, CANTON SALCEDO, PROVINCIA DE COTOPAXI”.

La etapa inicial hace referencia a la adquisición de datos procedentes de las redes DigiMesh, ZigBee y Wifi; conformadas por varios sensores que captan las señales climatológicas: temperatura, humedad relativa y del suelo, luminosidad, radiación solar, radiación ultravioleta, nivel de lluvia CO₂, CO, velocidad del viento y dirección del viento, por lo cual se requiere llevar un registro del comportamiento de dichas variables que permitirá mejorar la calidad de producción de la planta de tomate de riñón monitoreado en un invernadero del Barrio Rumipamba de Navas del Cantón Salcedo. LabVIEW permite realizar dicho procedimiento por ser un software robusto con un lenguaje intuitivo con la capacidad de adquirir datos, monitorear y controlar cualquier tipo de proceso. La segunda etapa corresponde al almacenamiento del registro de cada variable correspondiente a las redes ya mencionadas.

El software multiplataforma XAMPP constituye la etapa principal del proyecto al integrar un servidor web localhost, lenguaje script y un servidor MySQL, capaz de manejar varias bases de datos, de igual forma interactúa con varios clientes que realizan una petición de consulta de modo que el servidor devuelva una respuesta.

La aplicación desarrollada consta de tres partes principales: la primera integra la función del monitoreo de las variables climatológicas en tiempo real “tendencia”, la segunda etapa integra la función de históricos de cada una de las variables climatológicas y su respectiva gráfica, además cuenta con la facilidad a elegir el intervalo de tiempo a consultar y la tercera proporciona información.

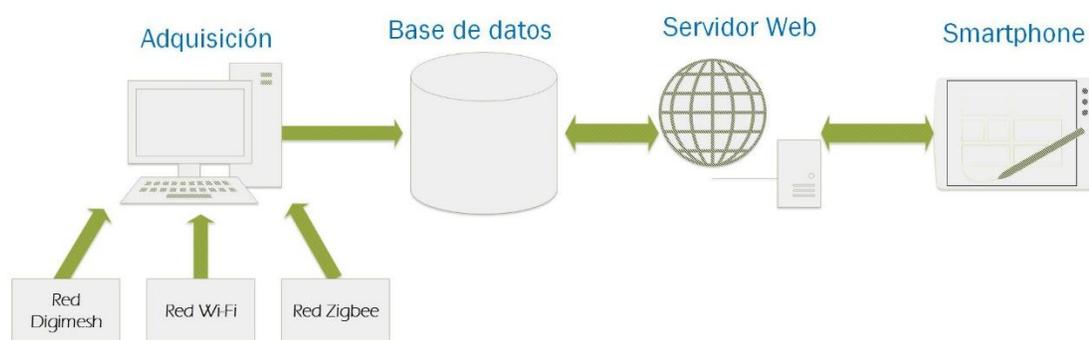


Figura 2.1 Etapas del proyecto

2.1 REQUISITOS PARA EL SERVIDOR WEB

2.1.1 XAMPP

XAMPP¹¹ es un software multiplataforma, totalmente gratuito y libre, constituido por un servidor web que permite el desarrollo de páginas web en código HTML, CSS y PHP, de igual forma está configurado para recibir cualquier tipo de petición de consulta al administrador de base de datos MySQL. Entre las

¹¹ Proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

principales ventajas se puede mencionar: instalación amigable y las configuraciones están predeterminadas a utilizar como un servidor local.

Una vez instalado el servidor XAMPP en este proyecto se requiere configuraciones previas que permita el acceso remoto al servidor web, desde cualquier PC o dispositivo que sea capaz de navegar en la web. Las configuraciones se realizan en la localidad **C:\xampp\apache\conf\extra** en el archivo **httpd-xampp** se requiere cambiar los parámetros de seguridad que permita el acceso, donde se agrega el comando **ALLOW FROM ALL** que habilita al servidor WEB como una página web pública.

2.1.2 BASE DE DATOS EN XAMPP

Xampp permite manejar varias base de datos a la vez y acceder a cada una de ellas mediante peticiones, como requisito principal se activa los módulos del sistema gestor de base de datos (MySQL) y la aplicación de servidor web que procesa y entrega el contenido hacia la base de datos correspondiente, en la localidad phpMyAdmin¹² al crear las bases de datos, se asigna el nombre que las identificará además se elige el tipo de codificación para los valores de ciertos campos.

En la base de datos creada se requieren de tablas que contendrán la información y etiquetas que describe a cada una de las bases >Dar nombre a la tabla>Número de columnas >Continuar (Figura 2.2), Para la creación de cada una de las bases de datos que corresponde a cada red como es en el caso de Zigbee se necesitó 3 tablas llamadas: nodo1, nodo2 y nodo3, en cuanto para las

¹² Herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

bases de datos Wi-fi y Digimesh las tablas fueron: nodo1, nodo2, nodo3 y nodo4 respectivamente.

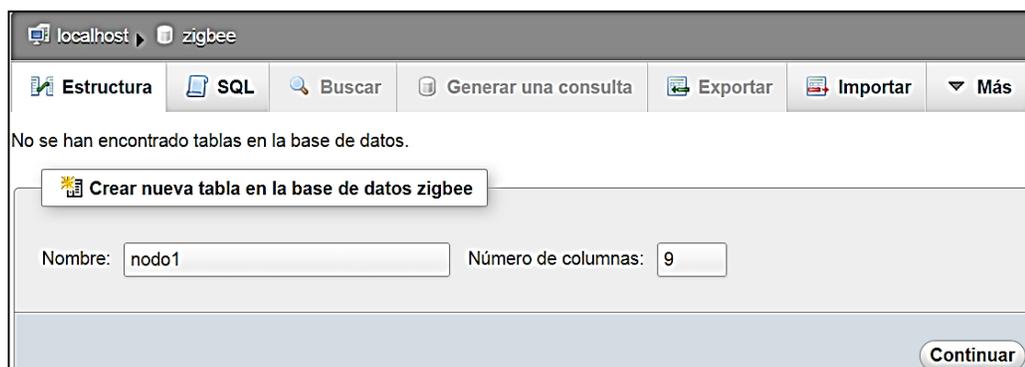


Figura 2.2 Crear las tablas de la base de datos

Cada tabla posee campos a llenar, COLUMNAS se asigna el nombre de las variables a almacenar, TIPO permite elegir el tipo de dato de la variable y LONGITUD el número de dígitos que tendrá cada variable (Figura 2.3).

| Columna | Tipo | Longitud/Valores*1 |
|--------------|-------|--------------------|
| TEMPERATURA | FLOAT | 11 |
| HUEMIDAD | FLOAT | 11 |
| LUMINOSIDAD | FLOAT | 11 |
| ULTRAVIOLETA | FLOAT | 11 |

Figura 2.3 Asignar nombres, tipo y longitud de los datos a manejar

La base de dato Digimesh (Tabla 2.1) posee 4 tablas correspondientes a la información de cada una de las variables climatológicas, los datos son de tipo flotante por tener valores decimales y un longitud de 11 debido a que la variable

luminosidad posee valores altos se optó por mantener esa longitud a todas las variables, a diferencia de FECHA que maneja datos Varchar debido que el contenido a almacenar son caracteres alfanuméricos tendrá una longitud de 25 caracteres.

Tabla 2.1

Red Digimesh

| RED DIGIMESH | | | | | | |
|--------------|--------|--------|--------|--------|---------|----------|
| VARIABLE | NODO 1 | NODO 2 | NODO 3 | NODO 4 | TIPO | LONGITUD |
| TEMPERATURA | X | X | X | | FLOAT | 11 |
| HUMEDAD | X | X | X | | FLOAT | 11 |
| LUMINOSIDAD | X | X | X | | FLOAT | 11 |
| HUM. SUELO | | X | | | FLOAT | 11 |
| ULTRAVIOLETA | | | X | | FLOAT | 11 |
| RADIACION S. | X | | | | FLOAT | 11 |
| CO2 | | | | X | FLOAT | 11 |
| CO | | | | X | FLOAT | 11 |
| BATERIA (v) | X | X | X | X | FLOAT | 11 |
| %BATERIA | X | X | X | X | FLOAT | 11 |
| FECHA | X | X | X | X | VARCHAR | 25 |

La base de dato Wifi (Tabla 2.2) posee 4 tablas correspondientes a la información de cada una de las variables climatológicas, los datos son de tipo flotante por tener valores decimales y un longitud de 11 debido a que la variable luminosidad posee valores altos se optó por mantener esa longitud a todas las variables, a diferencia de FECHA que maneja datos Varchar debido que el contenido a almacenar son caracteres alfanuméricos tendrá una longitud de 25 caracteres.

Tabla 2.2

Red Wi-Fi

| RED WIFI | | | | | | |
|-----------------------|--------|--------|--------|--------|---------|----------|
| VARIABLE | NODO 1 | NODO 2 | NODO 3 | NODO 4 | TIPO | LONGITUD |
| TEMPERATURA | X | X | X | | FLOAT | 11 |
| HUMEDAD | X | X | X | | FLOAT | 11 |
| LUMINOSIDAD | X | X | X | | FLOAT | 11 |
| CANT. LLUVIA (min) | X | | | | FLOAT | 11 |
| CANT. LLUVIA (h) | X | | | | FLOAT | 11 |
| ULTRAVIOLETA | | | X | | FLOAT | 11 |
| HUM. SUELO | | X | | | FLOAT | 11 |
| RADIACION S. | | X | | | FLOAT | 11 |
| CO2 | | | | X | FLOAT | 11 |
| CO | | | | X | FLOAT | 11 |
| BATERIA (v) | X | X | X | X | FLOAT | 11 |
| %BATERIA | X | X | X | X | FLOAT | 11 |
| FECHA | X | X | X | X | VARCHAR | 25 |

La base de datos Zigbee (Tabla 2.3) posee 3 tablas correspondientes a la información de cada una de las variables climatológicas, los datos son de tipo flotante por tener valores decimales y un longitud de 11 debido a que la variable luminosidad posee valores altos se optó por mantener esa longitud a todas las variables, a diferencia de FECHA que maneja datos Varchar debido que el contenido a almacenar son caracteres alfanuméricos tendrá una longitud de 25 caracteres.

Tabla 2.3

Red Zigbee

| RED ZIGBEE | | | | | |
|--------------|--------|--------|--------|---------|----------|
| VARIABLE | NODO 1 | NODO 2 | NODO 3 | TIPO | LONGITUD |
| TEMPERATURA | X | X | X | FLOAT | 11 |
| HUMEDAD | X | X | X | FLOAT | 11 |
| LUMINOSIDAD | X | X | X | FLOAT | 11 |
| ULTRAVIOLETA | | | X | FLOAT | 11 |
| RADIACION S. | | X | | FLOAT | 11 |
| VEL. VIENTO | X | | | FLOAT | 11 |
| DIR. VIENTO | X | | | FLOAT | 11 |
| BATERIA (v) | X | X | X | FLOAT | 11 |
| %BATERIA | X | X | X | FLOAT | 11 |
| FECHA | X | X | X | VARCHAR | 25 |

2.1.3 PHP PETICION DE CONSULTA MYSQL

Cuando el cliente solicite una petición de consulta, el servidor web recibe la petición e inmediato ejecuta un archivo PHP, que contiene código script solicitando una conexión con el servidor MYSQL para obtener un resultado. El resultado de la consulta es enviado por el servidor, quien a su vez lo reenvía al cliente. La creación de estos archivo PHP deben ser guardados en la localidad C:\xampp\htdocs). Una manera de crear los archivos es utilizar el editor de texto “bloc de notas”, la programación de estos registros se encuentra en el ANEXO A.

2.2 REQUISITOS PARA ACCEDER A LA BASE DE DATOS

2.2.1 MYSQL CONNECTOR / ODBC

Permite conectarse a un servidor de base de datos MySQL usando la base de datos ODBC, Para que Mysql conector se pueda enlazar a los datos en LabVIEW, y estos ser almacenarlos en cada base de datos. En la localidad >panel de control > sistemas de seguridad > herramientas administrativas > orígenes de datos ODBC (32bits), el controlador (MySQL ODBC 5.2 Driver) se encargara de gestionar la base de datos. (Figura 2.4)

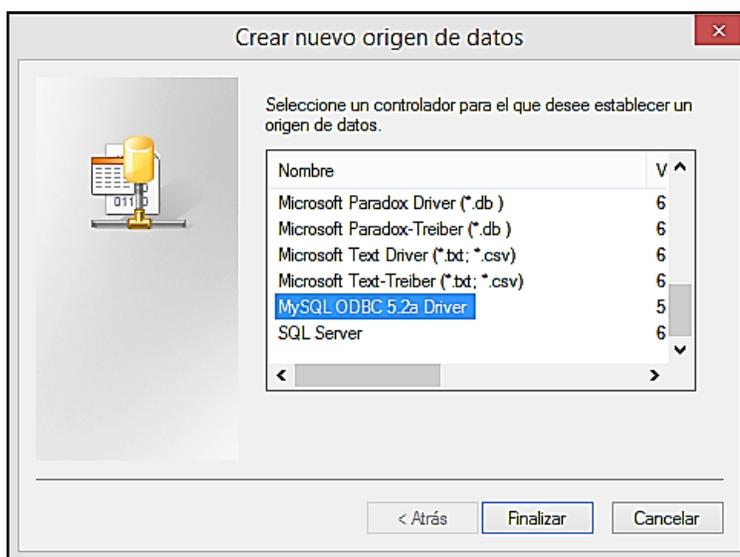


Figura 2.4 MySQL ODBC 5.2 Driver

Para crear el canal de comunicación es necesario llenar los campos: DATA SOURCE BASE que es el nombre de la conexión con la que se identificará, TCP/IP SERVER: el nombre del servidor local, USER: nombre del usuario que tendrá acceso y DATABASE: nombre de la base de datos creada en XAMPP (Figura 2.5), PORT se deja por defecto. Para comprobar que la conexión es satisfactoria se realiza un TEST de verificación, en caso de no existir errores el

mensaje será connection successful, este procedimiento se lo debe realizar para la redes digimesh, wifi y zigbee.

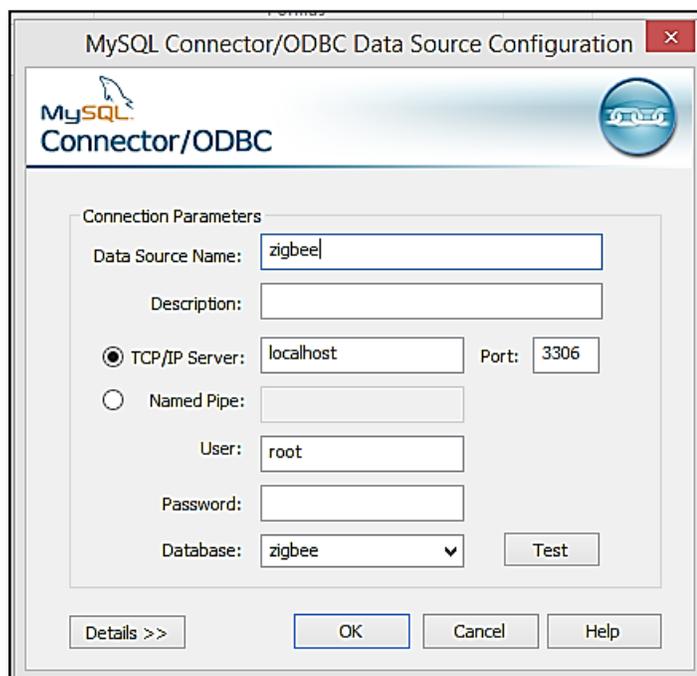


Figura 2.5 Asignar los nombres a la conexión

2.2.2 LABVIEW

LabVIEW es un entorno de desarrollo gráfico con funciones integradas que permite realizar adquisición de datos, y estos puedan ser almacenados en una base de datos, se emplean dos toolkit. HERRAMIENTAS DE BASE DE DATOS Utilizado para conectarse a bases de datos locales, remotas y ejecuta operaciones de bases de datos comunes sin tener que realizar lenguaje de consulta estructurado (SQL) de programación. HERRAMIENTAS REAL TIME Es una herramienta para analizar y verificar la ejecución del código ayuda a optimizar el código de tiempo real para lazos de control más rápidos y mayor rendimiento determinístico. Para que LabVIEW se pueda conectar a la base de datos MYSQL se realiza la configuración en Tools >Create Data Link>Microsoft OLE DB

Provider for ODBC Driver> Siguiente. La opción MICROSOFT OLE FOR DB PROVIDER ODBC DRIVERS permite establecer la conexión a la base de datos generada anteriormente, en Conexión>Especificar el origen de datos > Seleccionar el nombre de la conexión creada en el OBDC (Figura 2.6). A manera de verificar que no exista error en la conexión se realiza una prueba en PROBAR CONEXIÓN. Una vez realizado se procede a generar un archivo donde se guarda la base de datos

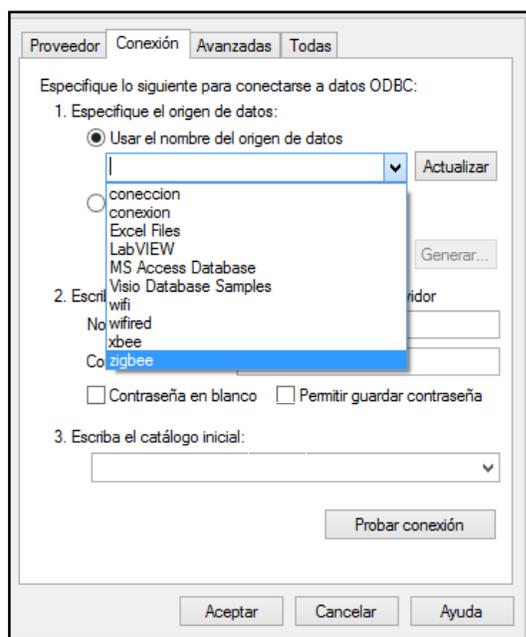


Figura 2.6 Origen de los datos

Al establecer una conexión entre LabVIEW y MYSQL será posible que los datos adquiridos sean enviados a la base, en LabVIEW dirigirse a Connectivity> Database> herramientas de abrir y cerrar la conexión. La Figura 2.7 se muestra los parámetros necesarios para dicho proceso.

- 1) Recopilación de datos de las variables de la red Zigbee (Tabla 2.3).
- 2) Direccionamiento del archivo creado por ODBC en LabVIEW hacia MYSQL base de datos.
- 3) Realizar la conexión hacia el servidor MYSQL con Usuario: "root" y Contraseña:"".

- 4) Hacer el llamado a la tabla nodo1 y a la tabla general creada MYSQL, para el almacenar los datos con sus variables ya definidas en cada una de las tablas.
- 5) Cerrar la conexión establecida entre LabVIEW y MYSQL
- 6) Nombres de las variables que contiene la tabla nodo.

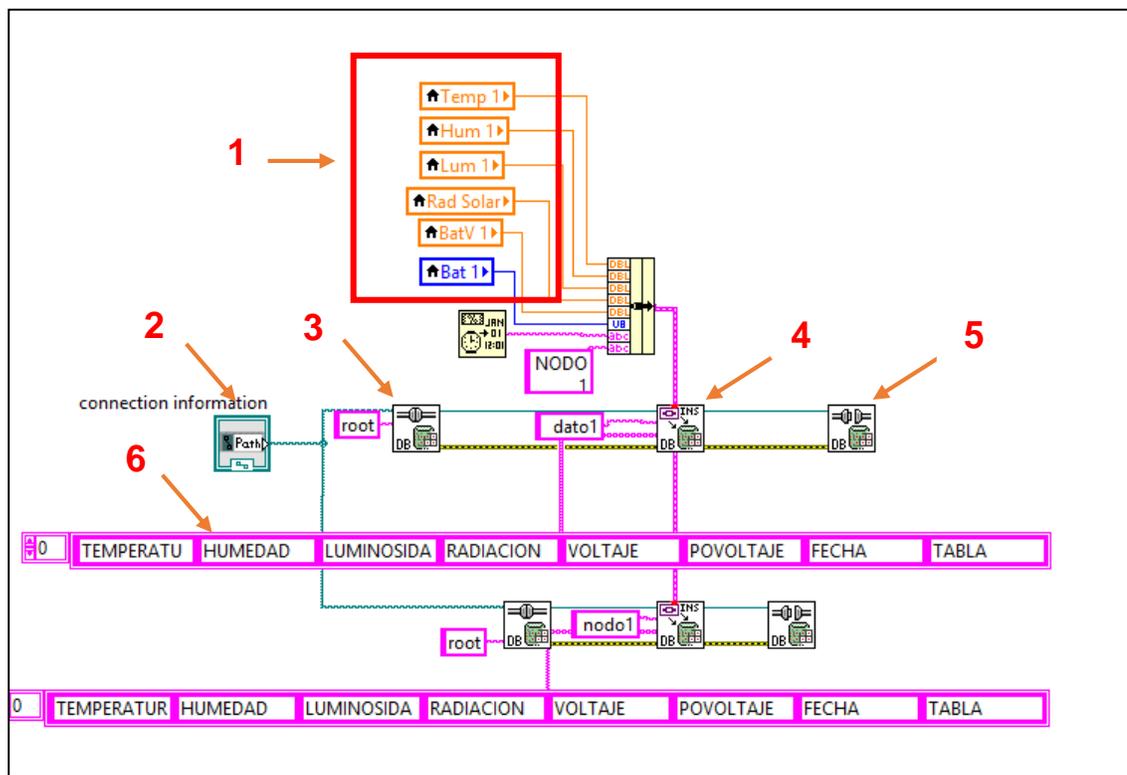


Figura 2.7 Conexión MYSQL

2.3 REQUISITOS ANDROID

Para programar en la plataforma Android se necesita un software que se compatible con las versiones de API de los Smartphone y Tablet al momento de compilarlo con el dispositivo por ello se tomó en cuenta a Eclipse como entorno de desarrollo por ser de código abierto basado en Java, como segunda instancia se necesita el kit de desarrollo Android SDK que proporciona las API, herramienta de desarrollo necesaria para la creación de la aplicación y la plataforma de

simulación mediante los Plung-ins para su correcto funcionamiento de igual forma permite depurar las aplicaciones.

Para el desarrollo de la aplicación se necesita de un plugin que configure los proyectos para Android, crea una interfaz de usuario más amigable, agrega paquetes basado en la API de Android Framework en Eclipse con la asignación de la url <https://dl-ssl.google.com/android/eclipse/> se podrá añadir dicha funcionalidad, esto se lo realiza en Help > Install new Software (Figura 2.8). La localidad que tendrá se asigna en Windows> Preferences

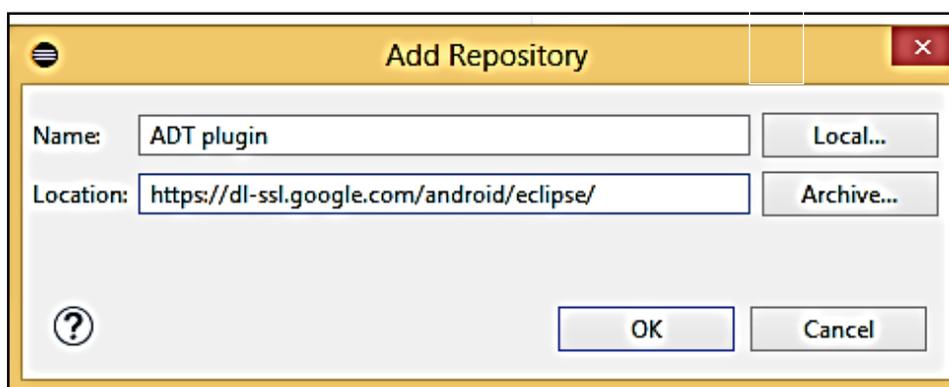


Figura 2.8 Añadir la dirección de descarga del plugin

2.4 REQUISITOS iOS

Para programar en la plataforma iOS se necesita una versión de software actualizable, se eligió a Xcode al ser gratuito permite desarrollar aplicaciones compatibles con versiones de API de los dispositivos Smartphone y Tablet, al momento de compilar la aplicación permite suaves transiciones entre el editor de código fuente, el depurador y el diseño de interfaces, todo ello en la misma ventana posee un entorno de desarrollo basado en c++, Objective-C y java, en la actualidad este software esta migrado a un nuevo lenguaje de programación llamado Swift.

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN

El capítulo 3 comprende el diseño y desarrollo de las aplicaciones móviles en las plataformas Android e iOS, ambas aplicaciones poseen la misma lógica y estructura a continuación se detallará a profundidad cada etapa de la misma

3.1 DIAGRAMA DE FLUJO DE LAS APLICACIONES

Al iniciar la aplicación en la pantalla principal se muestra un menú principal con tres opciones tendencia, históricos y ayuda como se muestra en la Figura 3.1 en cuanto el diagrama de flujo se muestra en la Figura 3.2 cabe de recalcar tanto como la estructura y el diseño están implementados en las 2 plataformas. A continuación se detallará cada uno de los casos del menú con su respectivo diagrama de flujo:



Figura 3.1 Menú Principal

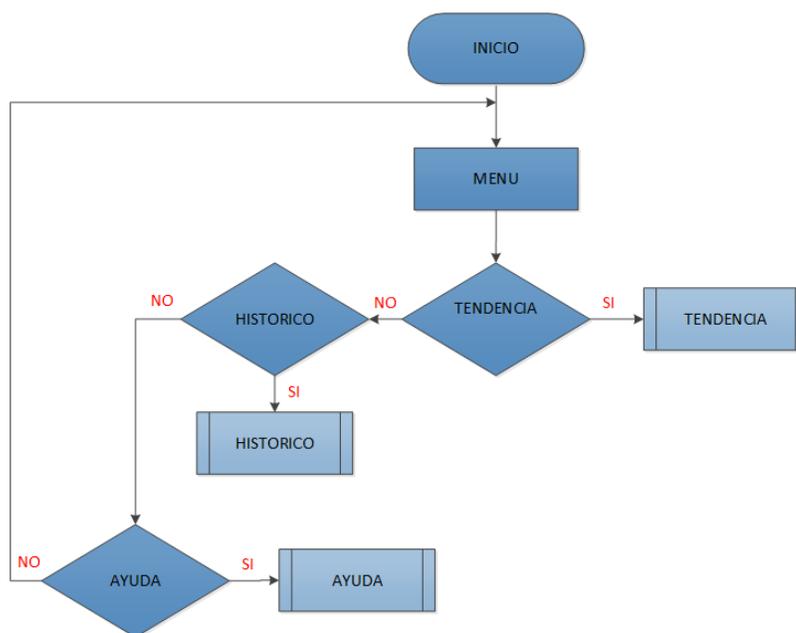


Figura 3.2 Diagrama de flujo (Principal)

3.1.1 TENDENCIA

Brinda acceso a los datos de las variables climatológicas en tiempo real, para tener una interfaz amigable con el usuario consta de un menú con tres opciones de acceso a las redes: Digimesh, Wifi y Zigbee, en la Figura 3.3 se muestra el diagrama de flujo.

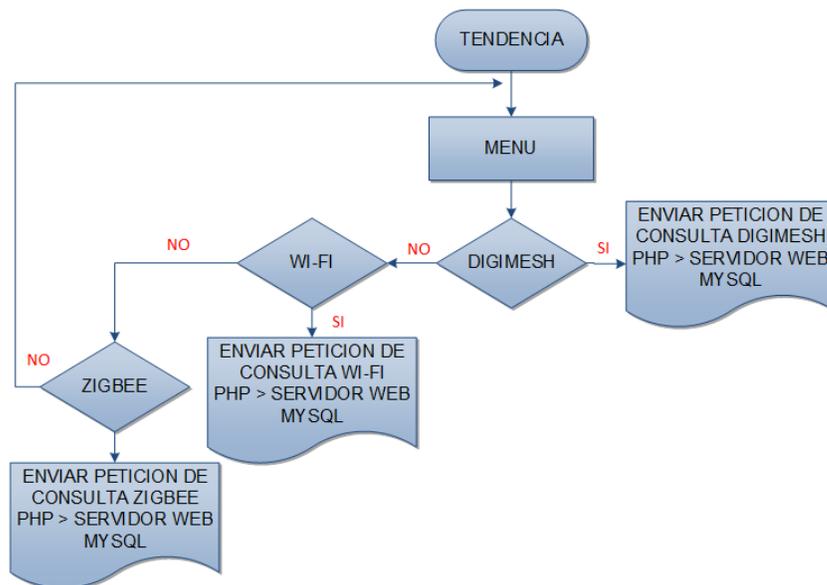


Figura 3.3 Diagrama de flujo (Tendencia)

Cada red está conformada por varios nodos de modo que, al elegir uno de ellos se desplegara toda la información de las variables climatológicas que contenga dicho nodo seleccionado , en la Figura 3.4 ,3.5 y 3.6 se muestra en los diagramas de flujo correspondientes a cada red.

a) DIGIMESH

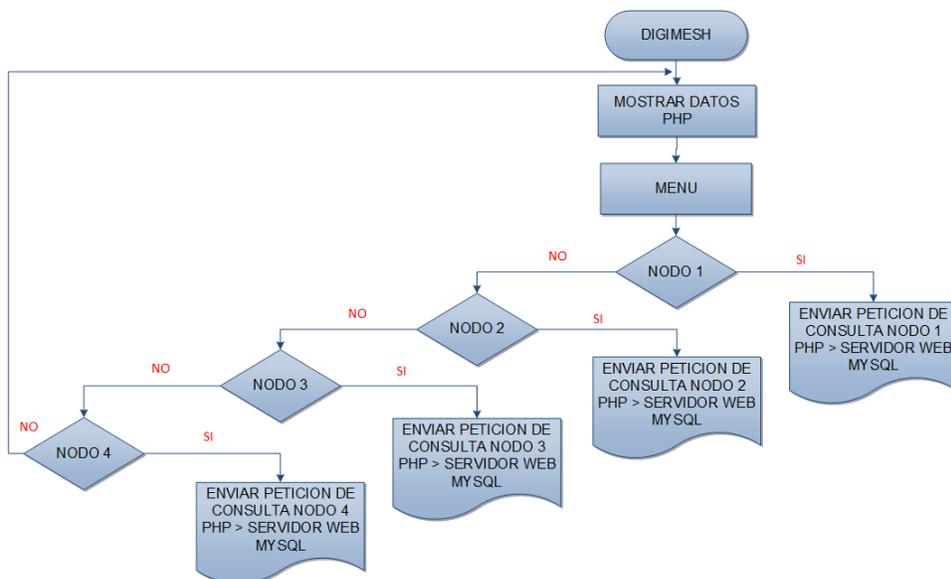


Figura 3.4 Diagrama de flujo (Digimesh)

b) WI-FI

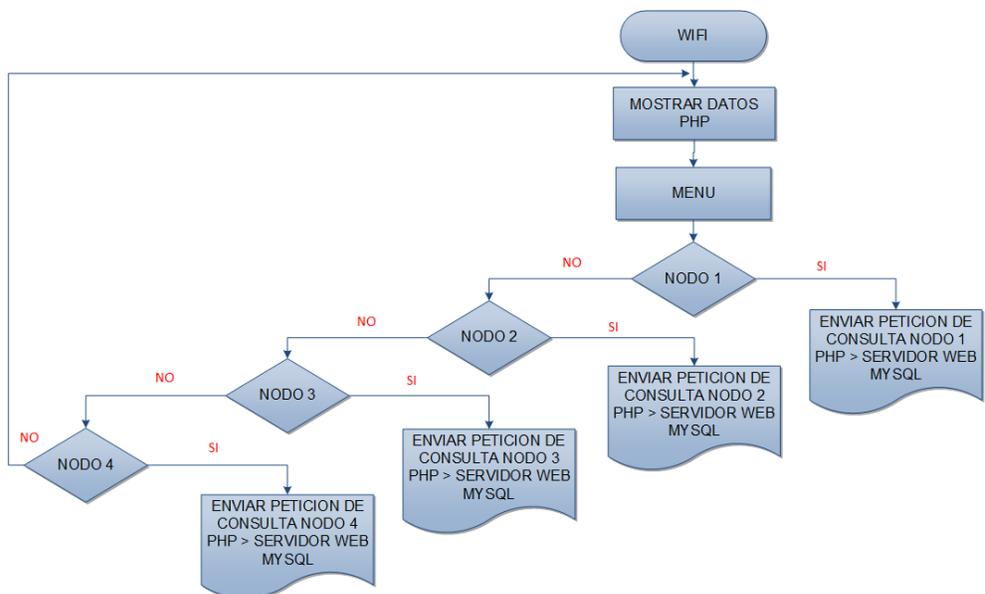


Figura 3.5 Diagrama de flujo (Wi-Fi)

c) ZIGBEE

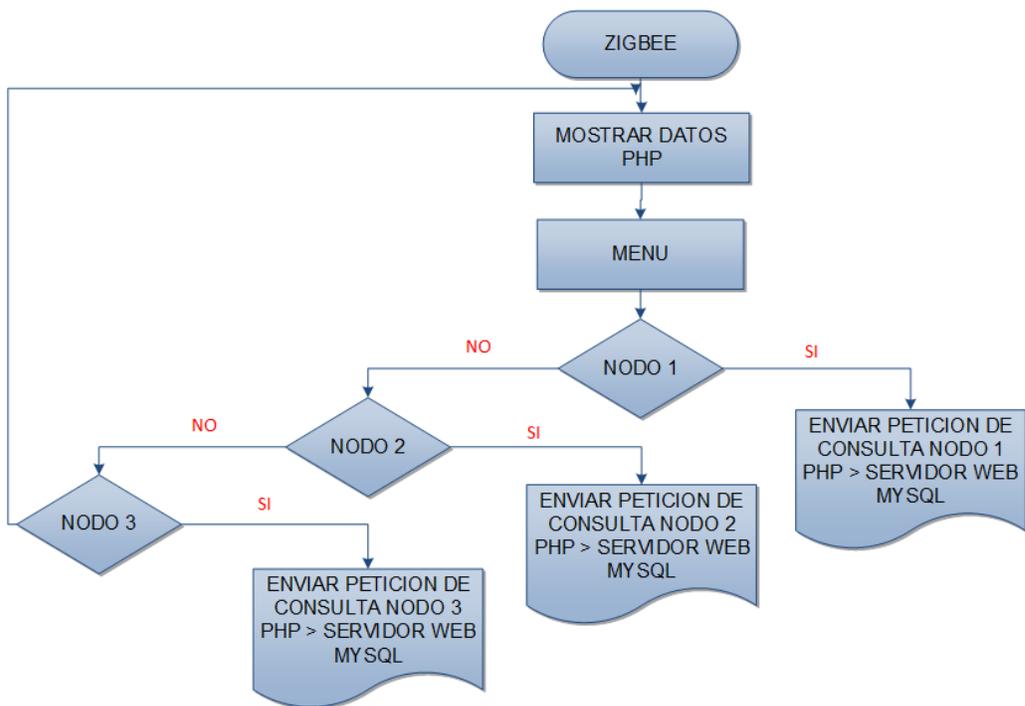


Figura 3.6 Diagrama de flujo (Zigbee)

3.1.2 HISTÓRICOS

Brinda acceso a los datos históricos de las variables climatológicas dentro de un intervalo de tiempo además provee de una gráfica de su comportamiento, posee un menú principal que permite elegir la red a consultar como lo indica el diagrama de flujo de la Figura 3.7.

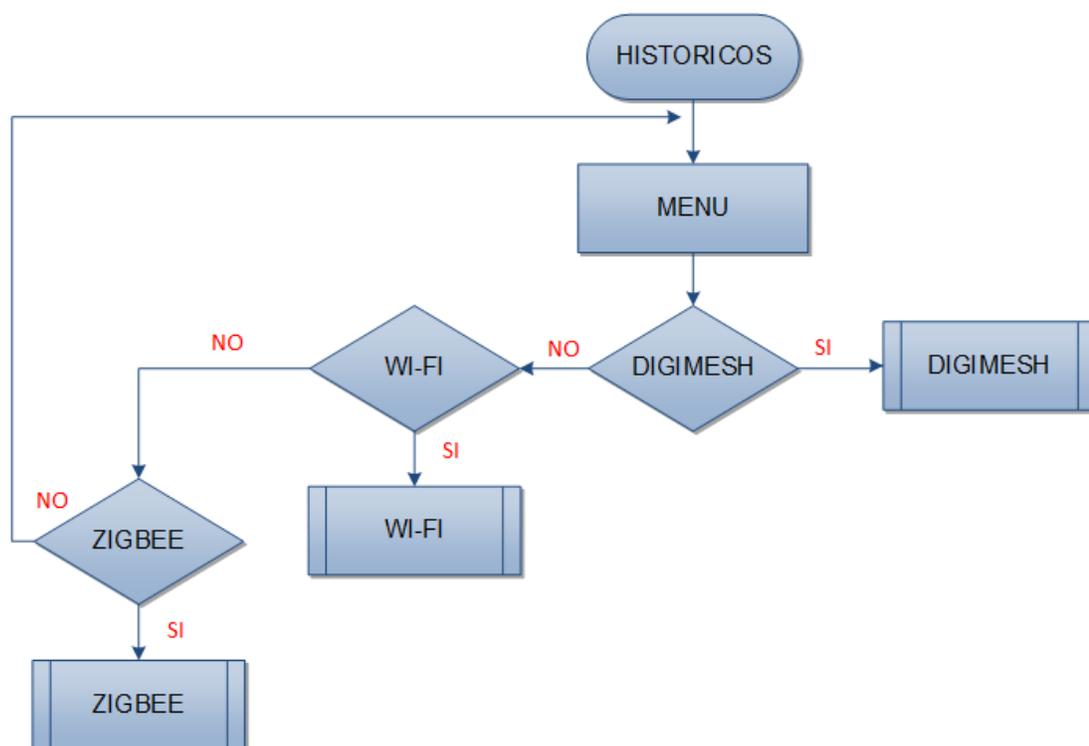


Figura 3.7 Diagrama de flujo de históricos

Las redes zigbee, digimesh y wifi poseen la misma lógica de desarrollo, para un mejor entendimiento se explicará el procedimiento de la red Zigbee. Una vez seleccionada la red (Figura 3.8) permite a consultar el nodo (Figura 3.9) de la misma manera la variable a consultar (Figura 3.10) la cual necesita un rango de fechas ingresadas por teclado para mostrar la información correspondiente.

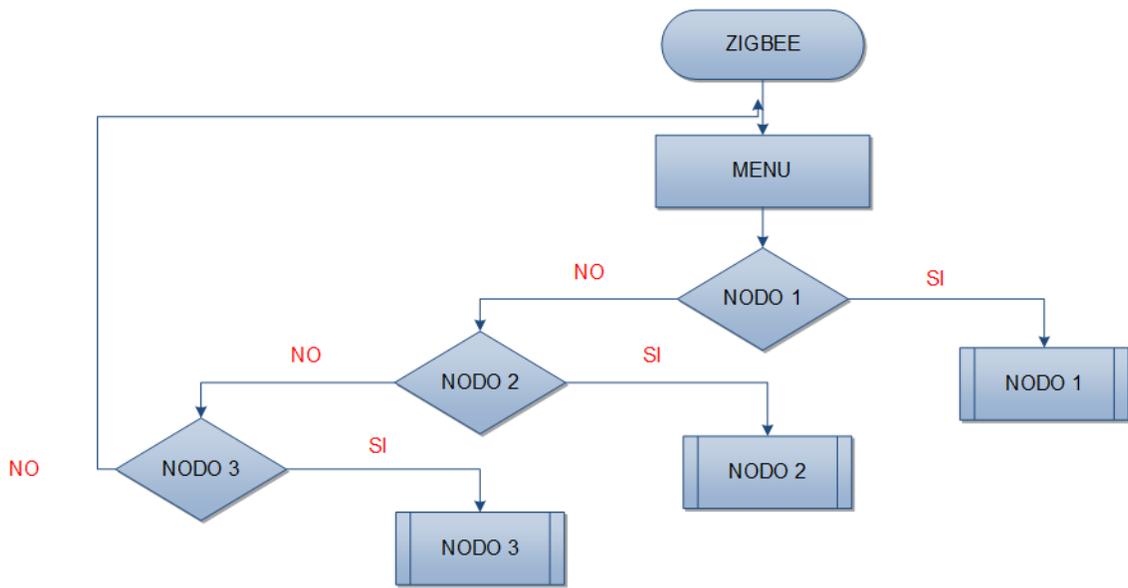


Figura 3.8 Diagrama de flujo red Zigbee (Históricos)

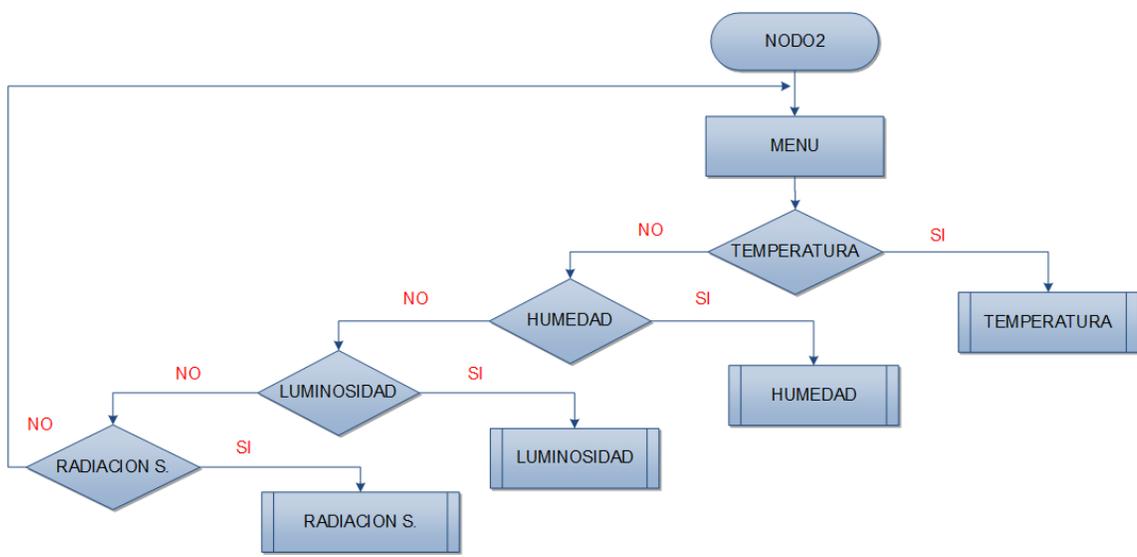


Figura 3.9 Diagrama de flujo nodo 2 de la red Zigbee (Históricos)

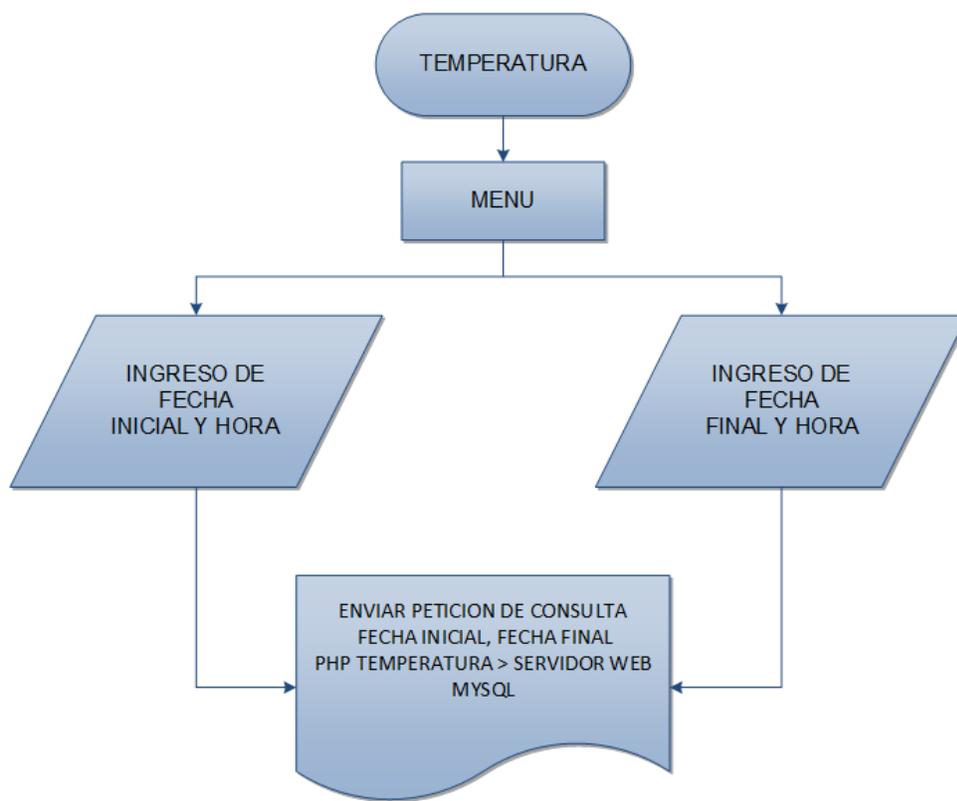


Figura 3.10 Diagrama de flujo de la variable temperatura

3.1.3 AYUDA

Al elegir esta opción se obtiene la información de la aplicación y los autores del proyecto. En la Figura 3.11 se muestra el diagrama de flujo y la Figura 3.12 visualización.

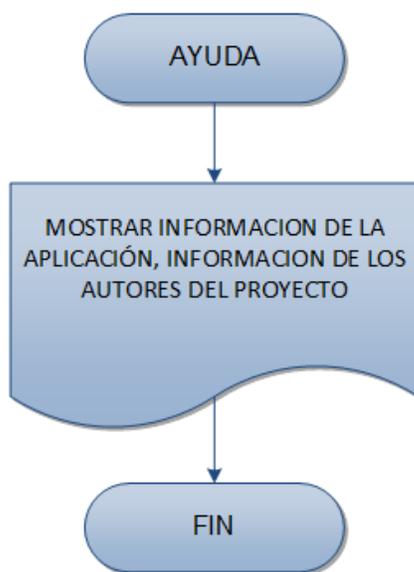


Figura 3.11 Diagrama de flujo de Ayuda



Figura 3.12 Layout de la Activity (Ayuda)

3.2 CREAR UN PROYECTO EN ANDROID

Antes de iniciar con la programación en eclipse se procede a crear un proyecto Android que contendrá todo el código de desarrollado en Nuevo> Carpeta Android>“Proyecto de aplicación de Android”> Siguiente, existe campos a llenar (Figura 3.13) para el nuevo proyecto:

- **Nombre de la Aplicación:** Nombre de la aplicación que aparece a los usuarios.
- **Nombre del Proyecto:** Nombre del directorio del proyecto y el nombre visible en Eclipse
- **Nombre del paquete:** Es el espacio del nombre del paquete para la aplicación, éste nombre debe ser único en todos los paquetes instalados en el sistema Android.
- **SDK Mínimo requerido:** Es la versión más baja de Android que admite la aplicación, indican usando el nivel de API.
- **SDK Target:** Indica la versión más alta de Android (también mediante el nivel de API). A medida que se disponga de nuevas versiones de Android, se debe probar la aplicación en la nueva versión y actualizar este valor para que coincida con el último nivel de API con el fin de aprovechar las nuevas características de la plataforma.
- **Compilar con:** es la versión de la plataforma con cual se compila la aplicación.
- **Tema:** especifica el estilo de interfaz de usuario de Android que tendrá la aplicación.

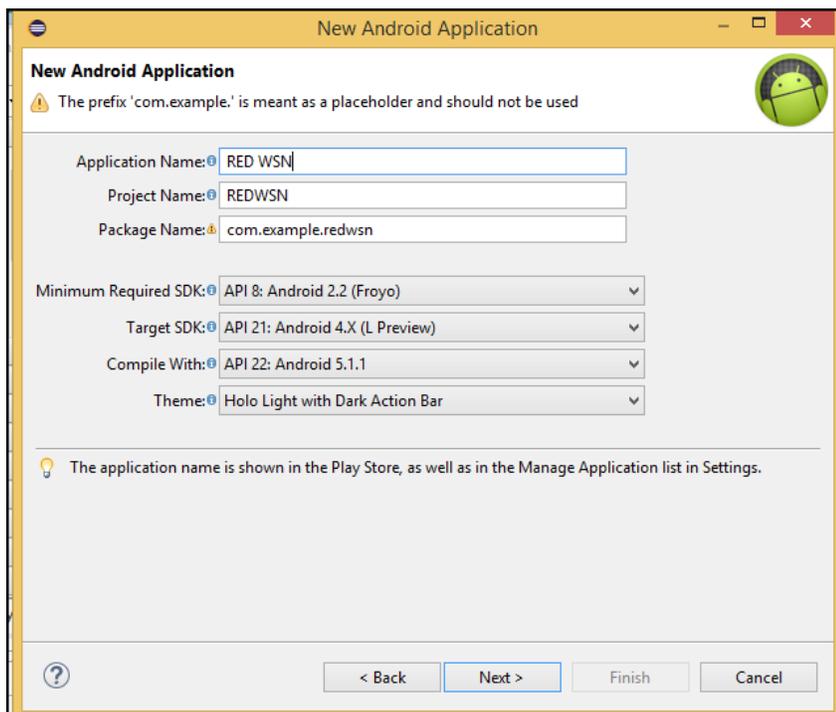


Figura 3.13 Datos de la nueva aplicación Android

Para identificar la aplicación en la pantalla del dispositivo es necesario crear un icono (Figura 3.14) podrá ser personalizado de varias maneras. Si se opta por mantener el icono por defecto o si se utiliza un diseño para la aplicación, el icono creado para el proyecto se muestra en la Figura 3.15.

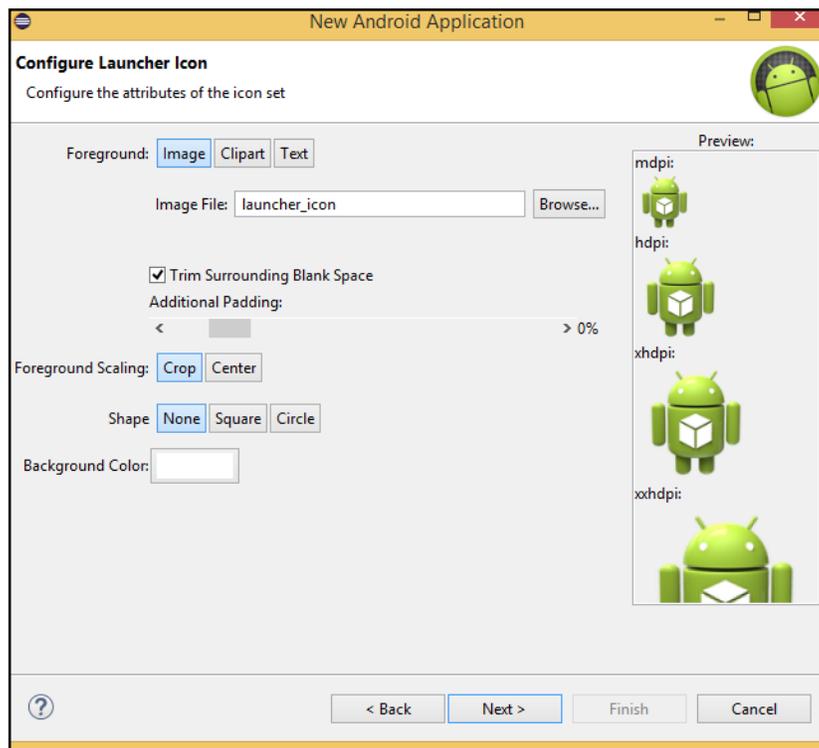


Figura 3.14 Selección del icono



Figura 3.15 Icono de la aplicación

Eclipse al momento de iniciar con un nuevo proyecto brinda la facilidad de utilizar plantillas de diseño de pantallas por comodidad y para añadir los elementos necesarios se elige la opción BlankActivity (Figura 3.16). Además es necesario dotar de nombres a las clases a utilizar.

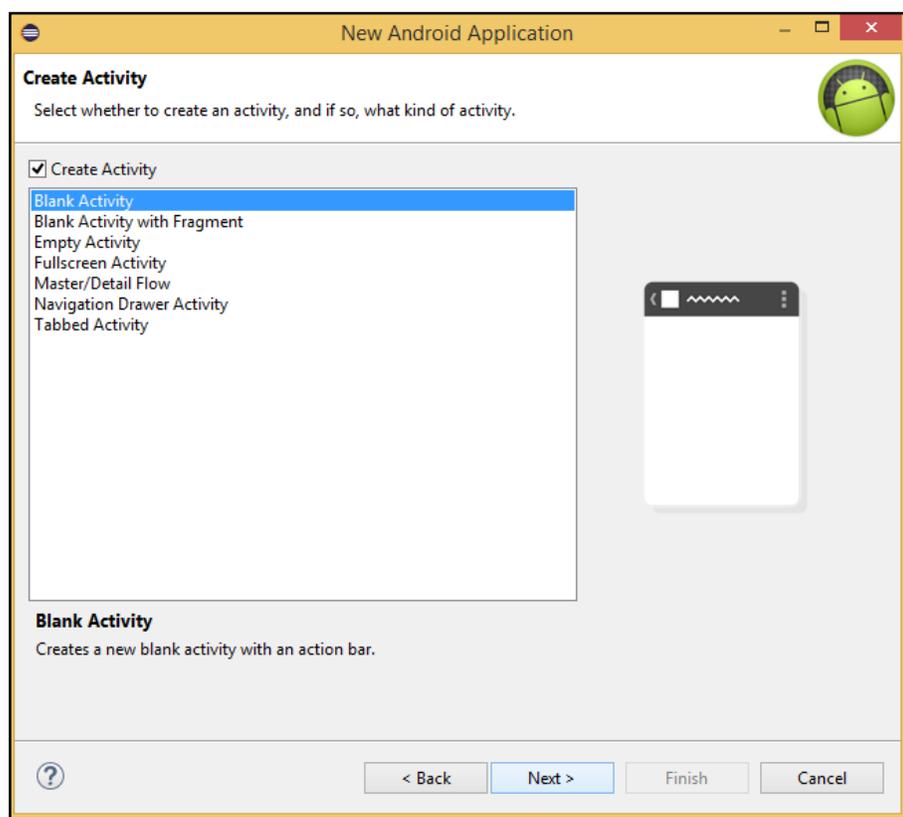


Figura 3.16 Plantilla de actividades

3.3 CREAR UN PROYECTO EN iOS (XCODE)

Para crear una aplicación vaya a File > New > Project, al seleccionar la plantilla “Single View Application” (Figura 3.17) las configuraciones de inicio serán básicas al comenzar a desarrollar la aplicación.

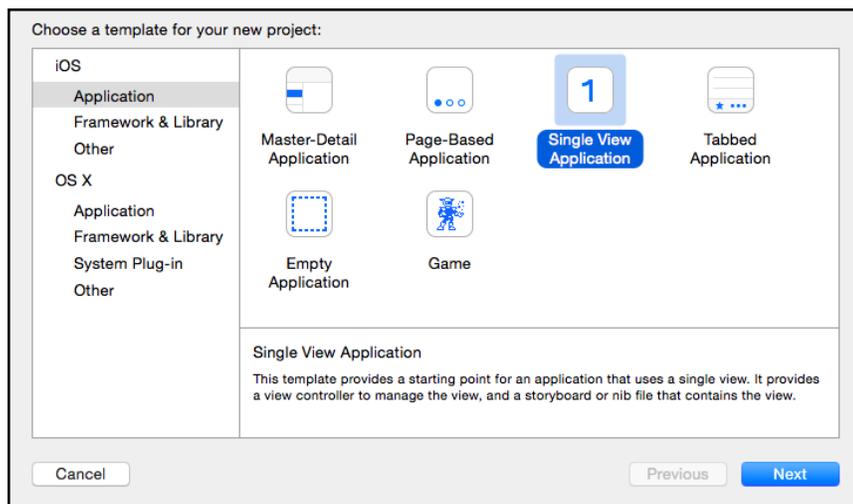


Figura 3.17 Seleccionar la plantilla de inicio

Existen campos que deberán ser llenados para dar inicio al iniciar el proyecto en Xcode (Figura 3.18), añadir la localidad a guardar el proyecto y finalmente crear para visualizar las configuraciones realizadas anteriormente (Figura 3.19).

- **Nombre del proyecto:** Dar nombre al proyecto, es este caso "BlankDemo"
- **Nombre de la Organización:** (Opcional) En la parte superior de todos sus archivos, dirá "CompanyName" en caso de no ser llenado.
- **Identificador de empresa:** Se trata de identificar la aplicación, puede ser com.companyname. de esta manera la aplicación va a resultar ser com.companyname.projectname.
- **Clase Prefijo:** Se puede dejar el campo vacío de ser llenada todos los archivos nuevos que se crean serán precedidos por este prefijo para distinguir los archivos creados a partir de archivos procedentes de otras fuentes.
- **Dispositivos:** Puede dejar este set para iPhone por ahora.

Choose options for your new project:

Product Name: RED SWN

Organization Name: usuario

Organization Identifier: prueba

Bundle Identifier: prueba.RED-SWN

Language: Objective-C

Devices: Universal

Use Core Data

Cancel Previous Next

Figura 3.18 Llenar los campos del proyecto

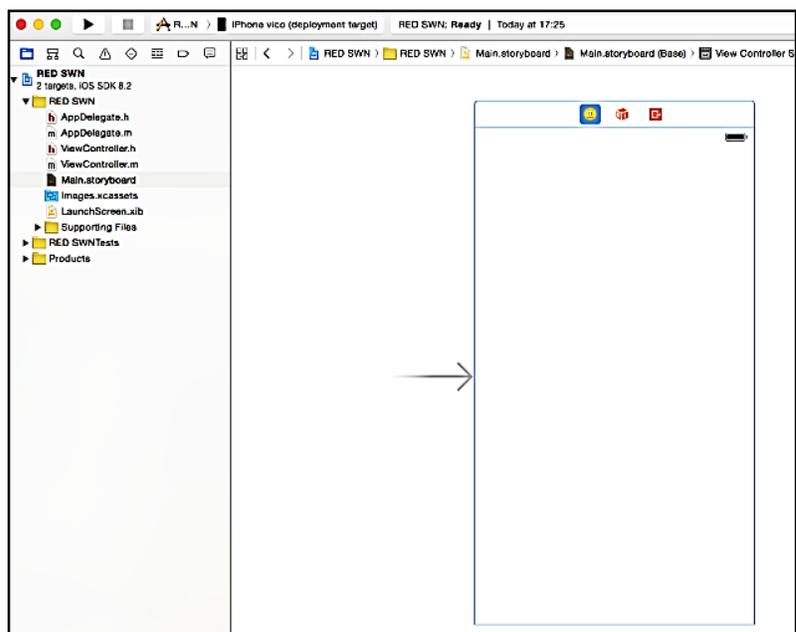


Figura 3.19 Nuevo proyecto

3.4 PROGRAMACIÓN EN ANDROID

A continuación se detalla el desarrollo de aplicación para tendencia, históricos y gráfica, el código de programación de cada etapa se encuentra más detallado en el ANEXO B, como requisito para que la aplicación funcione correctamente es necesario dar los permisos de internet, esto se realiza en el archivo Manifest del proyecto Android.

3.4.1 TENDENCIA

En las actividades REDWIFI, REDGIMESH Y REDZIGBEE se realizó la programación que gestiona la comunicación entre aplicación – servidor, cabe recalcar que la programación en las tres actividades son similares en lo único que difieren es en el direccionamiento de la URL a la base de datos, en este apartado se explicará la programación de una de las tres actividades mencionadas anteriormente. La actividad WIFI consta de dos partes:

- a) **El Layout Activity wifi** donde se realiza la interfaz del menú con el usuario consta del: archivo xml de la Figura 3.20 donde se asigna el nombre, tipo, color y tamaño de los botones, ubicación que tendrá además cada botón contará con un evento onClick que determina la acción que tendrá al ser pulsado y el Graphical Layout de la Figura 3.21 donde se visualiza la programación realizada anteriormente.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo1"
    android:padding="20dp"
    android:textColor="#000000"
    android:textSize="36sp"
    android:textStyle="bold" >

    <TextView
        android:id="@+id/titulo6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="@string/titulo6"
        android:textColor="#FF0000"
        android:textSize="36sp"
        android:textStyle="bold" />

```

Figura 3.20 Archivo xml



Figura 3.21 Graphical Layout

- b) La clase java wifi es la parte principal de la actividad donde se realiza la programación de botones, hilos de comunicación y visualización de datos. Para dar inicio a la programación es necesario añadir librerías de los

diferentes comandos a utilizar en programación (Figura 3.22), en caso de no hacerlo no se podrá realizar la programación.

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONArray;
import org.json.JSONObject;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
```

Figura 3.22 Librerías importadas

- Por la necesidad de mostrar valores y caracteres se declara variables de tipo TextView, String e int, en la clase principal del archivo java se da el acceso a los componentes del Layout Activity vinculándolos a una de las variables declaradas anteriormente, además se desarrolla la programación del hilo de comunicación, para evitar conflictos de tiempo se crea la función getData () de acceso público y tipo void (Figura 3.23).

```
public class Redwifit extends Activity {
    TextView resultado;
    String nodo1t, nodo2t, nodo3t,nodo4t,n1t;
    String valor;
    int k;
    final static String NODO1T = "com.example.monitoreo.NODOSWIFIT";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.red_wifit);
        StrictMode.enableDefaults();
        resultado = (TextView) findViewById(R.id.result);
        getData();
    }
}
```

Figura 3.23 Clase Principal (Main)

- En la función `getData()` se realiza el direccionamiento de la base de datos por medio de la url, en caso de haber un error en la conexión este desplegara un mensaje mediante la librería `ProgressDialog` (Figura 3.24).

```

public void getData() {
    String result = "";
    InputStream isr = null;
    try {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost("http://190.99.73.45:8080/wifi/respuesta.php");
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity entity = response.getEntity();
        isr = entity.getContent();
    } catch (Exception e) {
        Log.e("log_tag", "ERROR EN LA CONEXION " + e.toString());

        final ProgressDialog ringProgressDialog = ProgressDialog.show(
            Redwifit.this, "ACCESO NEGADO...", "REVISE LA CONEXION ",
            true);
        ringProgressDialog.setCancelable(true);
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    Thread.sleep(10000);

                } catch (Exception e) {

                }
                ringProgressDialog.dismiss();
            }
        }).start();
    }
}

```

Figura 3.24 Hilo de comunicación y direccionamiento de la url

- Para poder encriptar de una mejor manera la información de respuesta emitida por `HttpResponse`. (Figura 3.25) se convierten los datos a tipo `String` y estos a su vez son unificados en una sola variable.

```

try {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        isr));
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    isr.close();
    result = sb.toString();
} catch (Exception e) {
    Log.e("log_tag", "ERROR DATO " + e.toString());
}

```

Figura 3.25 Conversión a String

- Los datos almacenados en la base de datos son en formato JSON, al ser un formato ligero para el intercambio de datos y deben ser decodificados para acceder a la información, por ello se crean cuatro objetos tipo JSON como muestra la Figura 3.26 json, json1, json2 y json3 que almacenan la información de los nodos: nodo1, nodo2, nodo3 y nodo4 respectivamente.

```

JSONObject json = jsonArray.getJSONObject(0);
JSONObject json1 = jsonArray.getJSONObject(1);
JSONObject json2 = jsonArray.getJSONObject(2);
JSONObject json3 = jsonArray.getJSONObject(3);

```

Figura 3.26 Objetos json

- Para acceder a los valores de cada nodo se hace referencia al objeto json creando más el comando getString y con el nombre de la variable a obtener la información (Figura 3.27).

```

json.getString("TEMPERATURA")

```

Figura 3.27 Obtener valor de la variable

- La acción del botón del nodo seleccionado permite pasar parámetros de información a la actividad Nodoswifi (Figura 3.28).

```
public void bt_nodo1wifi(View v) {  
  
    Intent act = new Intent(this, Nodoswifit.class);  
    act.putExtra(NODO1T, nodo1t);  
    startActivity(act);  
  
}
```

Figura 3.28 Acción del botón

- En la Figura 3.29 se muestra la visualización de las variables correspondiente al nodo1 de la red Wi-fi.

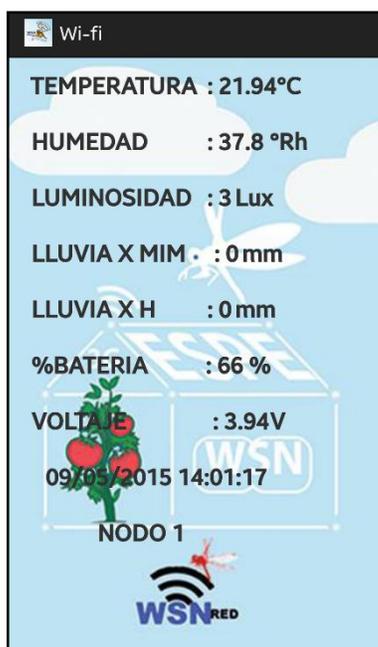


Figura 3.29 Visualización de datos (Nodo1)

Entre los comandos utilizados para el hilo de comunicación se citan los más importantes:

- **HttpResponse** Ejecuta una solicitud al destino con el contexto dado. Devuelve una respuesta final a dicha solicitud. En caso de ocurrir un error muestran mensajes en los que señalan el problema:
- **HttpGet** El método GET significa recuperar toda la información identificado por el Request-URI.
- **HttpEntity** Una entidad puede ser enviada o recibida con un mensaje HTTP, las entidades se pueden encontrar en algunas solicitudes y respuestas.

3.4.2 HISTÓRICOS

Al igual que tendencia requiere de hilos de comunicación de la misma manera de explicará de la red WIFI. Al momento de seleccionar la red, el nodo y la variable a consultar se despliega un Layout denominado FechaN1W ver Figura 3.30, en donde se elige la fecha de inicio y fin denominados textinicio y textfinal respectivamente estos son enviados al servidor para que este devuelva una respuesta en base al intervalo de tiempo requerido.



Figura 3.30 Layout de Activity (Fecha)

- Una vez seleccionadas las fechas mediante el botón CONSULTAR son almacenadas y enviadas a la clase Dato1w donde se realiza la petición de información. En este caso se utilizó la clase privada insertar () de tipo String, donde se envía los datos hacia el servidor por método POST (Figura 3.31).

```
private String insertar() {  
  
    String resultado = null;  
    HttpClient httpClient;  
    httpClient = new DefaultHttpClient();  
    List<NameValuePair> nameValuePairs;  
    HttpPost httppost;  
    httppost = new HttpPost("http://190.99.73.45:8080/wifi1/nodo1.php");  
    nameValuePairs = new ArrayList<NameValuePair>(1);  
    nameValuePairs.add(new BasicNameValuePair("textinicio", textinicio  
        .getText().toString().trim()));  
    nameValuePairs.add(new BasicNameValuePair("textfinal", textfinal  
        .getText().toString().trim()));  
}
```

Figura 3.31 Envío de datos por método POST

- Por el método Response se obtiene la respuesta del servidor, y la almacena en la variable resultado de tipo String (Figura 3.32).

```
httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));  
HttpResponse response = httpClient.execute(httppost);  
HttpEntity entity = response.getEntity();  
resultado = EntityUtils.toString(entity, "UTF-8");
```

Figura 3.32 Respuesta del método Response

- En esta caso se crea un ArrayList de los objetos json, para guardar en un listado todas las variables y así visualizarlos de mejor manera en un

ListView. Además se almacena en un vector llamado series los datos de las variables que estos puedan ser graficados (Figura 3.33).

```
public ArrayList<String> obtDatosJSON(String response) {
    ArrayList<String> listado = new ArrayList<String>();
    try {
        JSONArray json = new JSONArray(response);
        String texto = "";
        for (int i = 0; i < json.length(); i++) {
            if (sensor.getText().equals("TEMPERATURA")) {
                dato = json.getJSONObject(i).getDouble("TEMPERATURA");
                series[i] = dato;
                texto = json.getJSONObject(i).getString("TEMPERATURA")
                    + " " + "°C" + " " + " " + " - " + " "
                    + json.getJSONObject(i).getString("FECHA");
                listado.add(texto);
            }
        }
    }
}
```

Figura 3.33 Obtención de los datos de la variable

- La clase cargaListado (Figura 3.34) recupera los datos almacenados en la variable listado y estos son visualizados en el ListView (Figura 3.35).

```
public void cargaListado(ArrayList<String> datos) {
    ArrayAdapter<String> adaptador = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, datos);
    ListView listado = (ListView) findViewById(R.id.list1);
    listado.setAdapter(adaptador);}
}
```

Figura 3.34 Clase cargaListado



Figura 3.35 Visualización de datos en un ListView

Los comandos utilizados para el hilo de comunicación se citaron en tendencia a diferencia de:

- **Http Post** El método POST se utiliza para solicitar al servidor de origen aceptar a la entidad incluida en la solicitud, está diseñado para permitir un método uniforme para cubrir las siguientes funciones:

3.4.3 GRAFICA

- Se necesita los datos que se obtienen en Históricos estos son enviados a la clase Grafica con el botón Graficar programado con la acción de enviar datos. (Figura 3.36).

```
public void grafica(View v) {  
  
    Bundle text = new Bundle();  
    text.putString("fecha1", textinicio.getText().toString());  
    text.putString("fecha2", textfinal.getText().toString());  
    text.putString("sensor", sensor.getText().toString());  
    Intent act = new Intent(this, Grafica.class);  
    text.putDoubleArray("dato", series);  
    act.putExtras(text);  
    startActivity(act);  
  
}
```

Figura 3.36 Envío de datos a la clase Graficar

- En la clase Grafica los datos son recuperados con el método Bundle que contiene el paquete de datos enviados (Figura 3.37). los mismos que son almacenados en variables de tipo String o vector de acuerdo al tipo de dato enviado.

```

Bundle extras = getIntent().getExtras();
    String f1 = extras.getString("fecha1");
    String f2 = extras.getString("fecha2");
    String s = extras.getString("sensor");
    series1 = extras.getDoubleArray("dato");

```

Figura 3.37 Recuperar los datos con el método Bundle

- Para graficar los datos son guardados en un vector y con la librería AndroidPlot al ser una API para la creación de gráficos dinámicos y estáticos (Figura 3.38), permite obtener la gráfica del comportamiento de las variables.

```

for (int i = 0; i < series1.length; i++) {
    Datox = Datox + 1;
    Vector.add(Datox);
    Datoy = series1[i];
    Vector.add(Datoy);
}

XYSeries series = new SimpleXYSeries(Vector,
    SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, " ");
LineAndPointFormatter seriesFormatter = new LineAndPointFormatter(
    Color.rgb(0, 191, 255), 0x000000, Color.rgb(173, 216, 230),
    null);
plot.clear();
plot.addSeries(series, seriesFormatter);

```

Figura 3.38 Almacenar los datos en un vector para ser graficado

- Donde el eje Y corresponde al valor de la señal y el eje X al número de muestras que se obtuvo (Figura 3.39).



Figura 3.39 Visualización de la grafica

3.5 PROGRAMACIÓN EN XCODE

A continuación se detalla el desarrollo de la aplicación para la etapa de tendencia, histórica y gráfica, el código de programación de cada etapa se encuentra más detallado en el ANEXO C.

3.5.1 TENDENCIA

En las actividades REDWIFI, REDGIMESH Y REDZIGBEE se realizó el diseño y la programación que gestiona la comunicación entre la aplicación y servidor Web Apache, cabe recalcar que la programación en las tres actividades son similares en lo único que difieren es en el direccionamiento de la URL hacia cada base respectivamente, en este apartado se explicará la programación de una de las tres actividades mencionadas anteriormente. La actividad WIFI consta de dos partes:

a) Storyboard: Es una representación visual de la interfaz de usuario que diseña la aplicación, contiene pantallas, texto y transiciones entre pantalla (

Figura 3.40) el diseño de la interfaz contiene TextView, Label, Botones e imágenes que se requiere al momento del desarrollo de la aplicación.

b) ViewController es un controlador de vista y programación, dividida en dos partes principales ViewController.h y ViewController.m, donde el controlador de vista le corresponde a ViewController.h y la programación a ViewController.m, es necesario para cada una de las pantallas que contenga Storyboard, contendrá las acciones que va a realizar.

- **ViewController.h** es la parte principal donde se desarrolla la vista, para cada pantalla de la aplicación. El controlador de vista contendrá todos los métodos IBOutlet y IBAction, UILabel declarados que puede ser métodos privados como públicos (Figura 3.41)

```
#import <UIKit/UIKit.h>
#import "datos1.h"

@interface wiffiViewController : UIViewController
@property (nonatomic, strong) IBOutlet UILabel *datoLabel;
@property (nonatomic, strong) IBOutlet UILabel *dato1Label;
@property (nonatomic, strong) IBOutlet UILabel *dato2Label;
@property (nonatomic, strong) IBOutlet UILabel *dato3Label;
@property (nonatomic, strong) IBOutlet UILabel *dato4Label;
@property (nonatomic, strong) IBOutlet UILabel *dato5Label;
@property (nonatomic, strong) IBOutlet UILabel *horaLabel;
@property (nonatomic, strong) IBOutlet UILabel *tablaLabel;
@property (nonatomic, strong) NSString * nssvari;
@property (weak, nonatomic) IBOutlet UILabel *dato;
@property (weak, nonatomic) IBOutlet UILabel *dato1;
```

Figura 3.40 Declarar métodos públicos y privados

- **ViewController.m** es la parte complementaria de la actividad que se crea como un controlador, donde se realiza la programación requerida utilizando los métodos IBOutlet y IBAction, UILabel declarados en WIFiViewController.h (Figura 3.42)

```

#import "wiffiViewController.h"

@interface wiffiViewController ()

@end

@implementation wiffiViewController
@synthesize datoLabel,dato1Label,dato2Label,dato3Label,dato4Label,dato5Label,horaLabel,currentDato;
@synthesize d,d1,d5,d4,d2,d3,d6,dato6Label,dato6;
- (void)viewDidLoad {
    [super viewDidLoad];
    [self setLabels];
    self.view.backgroundColor = [UIColor colorWithPatternImage:[UIImage imageNamed:@"ffg.png"]];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

```

Figura 3.41 ViewController.m

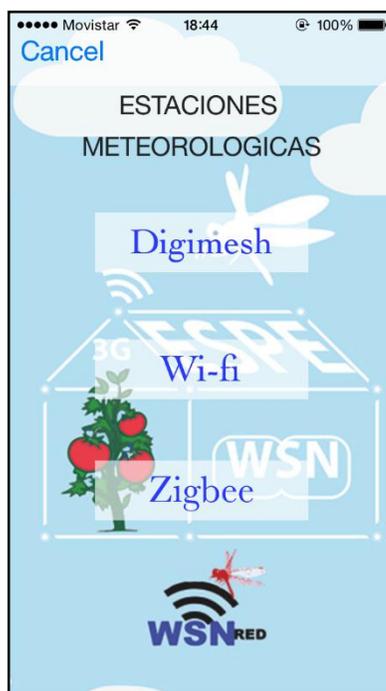


Figura 3.42 Visualización

- En la función `getDataURL` (Figura 3.43) contiene el direccionamiento de la base de datos por medio de una URL, en caso de existir error se desplegará un mensaje de aviso, posteriormente se detallará los comandos empleados.

```

getDataURL @"http://190.99.73.45:8080/wifi/respuesta.php"
NSURL *url = [NSURL URLWithString:getDataURL];
NSData * data = [NSData dataWithContentsOfURL:url];
NSError *error =nil;
if (data == nil) {
    NSLog(@"ERRO");
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"ERROR"
        cancelButtonTitle:@"Ok"otherButtonTitles:nil];
    [alert show];
    return; }

```

Figura 3.43 Función getDataURL

- **GetDataURL:** variable declarada que contiene la dirección de la base que se requiere hacer una petición de consulta.
- **NSURL:** convierte la dirección GetDataURL en formato URLString.
 - **NSData:** potencialmente realiza la conexión de un servidor remoto servidor remoto GetDataURL.
- Los datos almacenados en la base de datos son en formato JSON y deben ser decodificados para acceder a la información, se crea un array de tipo NSMutableArray llamado jsonArray como se muestra en la Figura 3.44 almacenan la información de cada uno de los nodos.

```

jsonArray = [NSJSONSerialization JSONObjectWithData:data

```

Figura 3.44 Crear el objeto jsonArray

- Para obtener los valores de cada variables cada NODO se hace referencia a los nombres de los objeto json creados por PHP, al momento de responder a la petición de consulta (Figura 3.45.), son almacenados en un datos1Array que contiene su propio ViewController el código se encuentra en el ANEXO C para su visualización.

```

for (int i = 0; i < jsonArray.count; i++) {
    NSString * cTEMPERATURA =[[jsonArray objectAtIndex:i] objectForKey:@"TEMPERATURA"];
    NSString * cHUMEDAD =[[jsonArray objectAtIndex:i] objectForKey:@"HUMEDAD"];
    NSString * cLUMINOSIDAD =[[jsonArray objectAtIndex:i] objectForKey:@"LUMINOSIDAD"];
    NSString * cULTRAVIOLETA =[[jsonArray objectAtIndex:i] objectForKey:@"ULTRAVIOLETA"];
    NSString * cRADIACION =[[jsonArray objectAtIndex:i] objectForKey:@"RADIACION"];
    NSString * cLLUVIA =[[jsonArray objectAtIndex:i] objectForKey:@"LLUVIA"];
    NSString * cCO2 =[[jsonArray objectAtIndex:i] objectForKey:@"CO2"];
    NSString * cCO =[[jsonArray objectAtIndex:i] objectForKey:@"CO"];
    NSString * cVOLTAJE =[[jsonArray objectAtIndex:i] objectForKey:@"VOLTAJE"];
    NSString * cPOVOLTAJE =[[jsonArray objectAtIndex:i] objectForKey:@"POVOLTAJE"];
}

```

Figura 3.45 Referencia a los objetos json creados

- La acción de un botón que corresponda a cualquier Nodo (Figura 3.46), realiza la función que permite enviar los parámetros de información del VlewController a WifiVlewController (Figura 3.47)

```

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if ([[segue identifier] isEqualToString:@"pushdeltaView"]) {
        NSIndexPath * indexPath = [self.tableView indexPathForSelectedRow];
        datos1 * object = [datos1Array objectAtIndex:indexPath.row];
        [[segue destinationViewController] getDate:object];
        NSLog(@"KKL%@", datos1Array);
    }
}

```

Figura 3.46 Acción del botón para enviar información

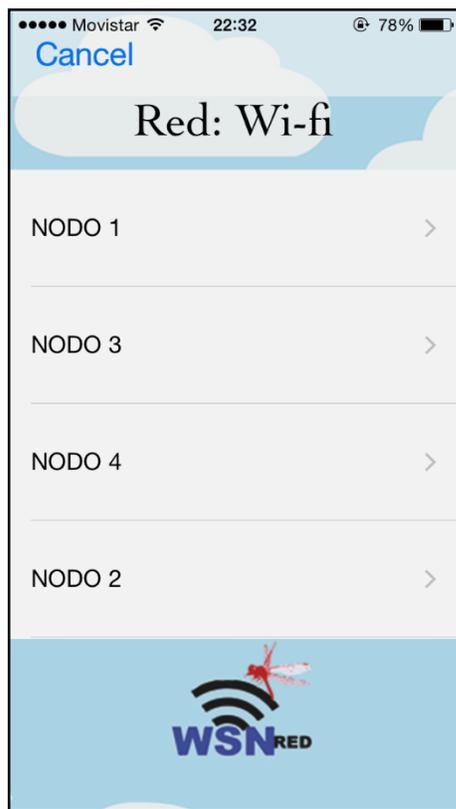


Figura 3.47 Selección de botón

- En la Figura 3.48 se muestra la visualización de las variables correspondiente al nodo1 de la red Wi-fi.

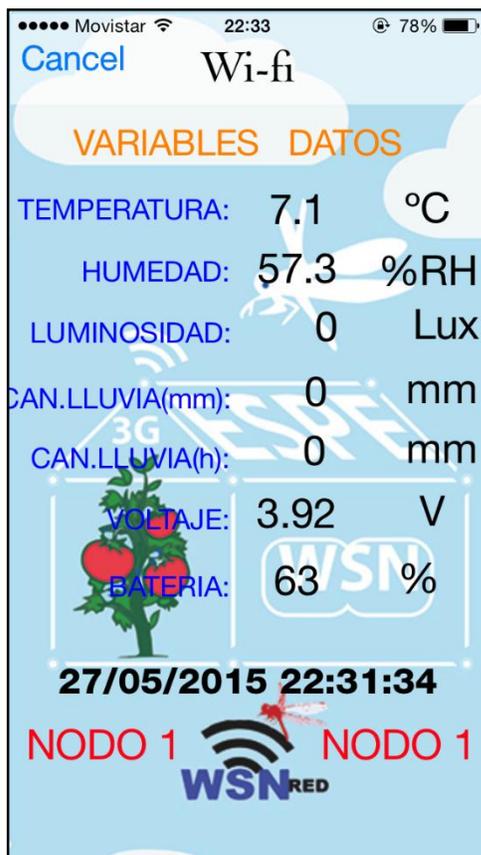


Figura 3.48 Visualización de datos (Nodo1)

3.5.2 HISTÓRICOS

Al igual que tendencia requiere de hilos de comunicación de la misma manera de explicará de la red WIFI. Al momento de seleccionar la red, el nodo y la variable a consultar se despliega una pantalla denominado Fecha ver Figura 3.49 en donde se elige la fecha de inicio y fin denominados textinicio y textfinal respectivamente estos son enviados al servidor Web para que este devuelva una respuesta.

| Cancel | | Digimesh | |
|-------------|-------|------------|----------|
| Nodo 1 | | GRAFICA | CONSULTA |
| TEMPERATURA | | | |
| INICIO | | FINAL | |
| FECHA | HORA | FECHA | HORA |
| Jun 25 may | 20 31 | Jun 25 may | 20 31 |
| mar 26 may | 21 32 | mar 26 may | 21 32 |
| hoy | 22 33 | hoy | 22 33 |
| jue 28 may | 23 34 | jue 28 may | 23 34 |
| vie 29 may | 0 35 | vie 29 may | 0 35 |

Figura 3.49 Layout de Activity (Fecha)

- Una vez seleccionadas las fechas mediante el botón CONSULTAR son almacenadas y enviadas para la petición de información, que se encripta con la dirección de la base de datos.
- En este caso se utilizó la clase NSMutableString de tipo String que concatena la URL con la fecha de inicio y la final, donde envía los datos hacia el servidor Web (Figura 3.50) donde la dirección de php contiene la función GET que se encuentra en espera de dos parámetros para realizar la consulta en la base de datos requerida la petición(Figura 3.51)

```
NSMutableString *url =[[NSMutableString alloc] initWithFormat:
    @"http://1190.99.73.45:8080/wifi/nodo1.
    self.nssinicioLabel,self.nssfinalLabel]
[url setString:[url stringByAddingPercentEscapesUsingEncoding:]
```

Figura 3.50 Enviar datos al servidor web

```

$conexion = mysql_select_db($bd, $servidor) or die ("pro_select_db");
$textinicio = $_GET["textinicio"];
$textfinal = $_GET["textfinal"];

```

Figura 3.51 Función GET para la consulta

- Por el método NSMutableString se obtiene la respuesta del servidor Web, almacena en la variable resultado de tipo String y se le convierte en tipo NSMutableArray almacenado en jsonArray(Figura 3.52)

```

NSData *data = [NSData dataWithContentsOfURL:[NSURL URLWithString:url]
NSMutableString *matresult =[[NSMutableString alloc] initWithData:data
                            :NSUTF8StringEncoding];
jsonArray = [ NSMutableArray arrayWithObject : matresult ];

```

Figura 3.52 Conversión a String

- La visualización de los datos de las variables se lo realiza en una TableView (Figura 3.53) , para obtener los valores de cada variable se hace referencia a los nombres de los objeto json creados por PHP, al momento de responder la petición de consulta (Figura 3.54.), son almacenados en un datos1Array para su visualización

```

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath
static NSString *cellIdentifier = @"Cell";
UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier
                        cellIdentifier forIndexPath:indexPath];

datos1 * dato10object;
dato10object = [datos1Array objectAtIndex:indexPath.row];

```

Figura 3.53 Código para TableView

```

for (int i = 0; i < jsonArray.count; i++) {
    NSString * cTEMPERATURA =[[jsonArray objectAtIndex:i] objectForKey:@"TEMPERATURA"];
    NSString * cHUMEDAD =[[jsonArray objectAtIndex:i] objectForKey:@"HUMEDAD"];
    NSString * cLUMINOSIDAD =[[jsonArray objectAtIndex:i] objectForKey:@"LUMINOSIDAD"];
    NSString * cULTRAVIOLETA =[[jsonArray objectAtIndex:i] objectForKey:@"ULTRAVIOLETA"];
    NSString * cRADIACION =[[jsonArray objectAtIndex:i] objectForKey:@"RADIACION"];
    NSString * cLLUVIA =[[jsonArray objectAtIndex:i] objectForKey:@"LLUVIA"];
    NSString * cCO2 =[[jsonArray objectAtIndex:i] objectForKey:@"CO2"];
    NSString * cCO =[[jsonArray objectAtIndex:i] objectForKey:@"CO"];
    NSString * cVOLTAJE =[[jsonArray objectAtIndex:i] objectForKey:@"VOLTAJE"];
    NSString * cPOVOLTAJE =[[jsonArray objectAtIndex:i] objectForKey:@"POVOLTAJE"];
}

```

Figura 3.54 Objetos json

- La clase dato1Object (Figura 3.55) creada permite recupera los datos almacenados en el Array y realiza la visualización en la TableView (Figura 3.56)

```

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
static NSString *cellIdentifier = @"Cell";
UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:
cellIdentifier forIndexPath:indexPath];

datos1 * dato1Object;
dato1Object = [datos1Array objectAtIndex:indexPath.row];

```

Figura 3.55 Clase dato1Object

| Cancel | | Digimesh | |
|-----------------------------|------------|----------------------------|--|
| INICIO: 27/05/2015 19:33:27 | | FINAL: 27/05/2015 22:33:27 | |
| TEMPERATURA | FECHA | HORA | |
| 18.7 °c | 27/05/2015 | 1:00:46 | |
| 19.03 °c | 27/05/2015 | 1:01:22 | |
| 19.03 °c | 27/05/2015 | 1:02:00 | |
| 19.03 °c | 27/05/2015 | 1:02:32 | |

Figura 3.56 Visualización en la TableView

3.5.3 GRAFICA

- Utilizando la misma estructura que se obtuvo los datos en Históricos, se realiza el mismo criterio en el ViewControllerGrafica(Figura 3.57)

```
NSMutableDictionary *url = [NSMutableDictionary alloc] initWithFormat:@"http://190.99.73.45:8080
nodo1.php?textinicio=%@&textfinal=%@", self.nssinicioLabel, self.n

[url setString:[url stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]];

NSData *data = [NSData dataWithContentsOfURL:[NSURL URLWithString:url]];
NSMutableDictionary *matresult = [NSMutableDictionary alloc] initWithData:data encoding:NSUTF8St
```

Figura 3.57 Clase ViewControllerGrafica

- Para obtener los valores de cada variable se hace referencia a los nombres de los objeto json creados por PHP, al momento de responder la petición de consulta (Figura 3.58), almacenados en un datos1Array para su visualización

```
for (int i = 0; i < jsonArray.count; i++) {

    NSString * cTEMPERATURA = [[jsonArray objectAtIndex:i] objectForKey:@"TEMPERATURA"];
    NSString * cHUMEDAD = [[jsonArray objectAtIndex:i] objectForKey:@"HUMEDAD"];
    NSString * cLUMINOSIDAD = [[jsonArray objectAtIndex:i] objectForKey:@"LUMINOSIDAD"];
    NSString * cULTRAVIOLETA = [[jsonArray objectAtIndex:i] objectForKey:@"ULTRAVIOLETA"];
    NSString * cRADIACION = [[jsonArray objectAtIndex:i] objectForKey:@"RADIACION"];
    NSString * cLLUVIA = [[jsonArray objectAtIndex:i] objectForKey:@"LLUVIA"];
    NSString * cCO2 = [[jsonArray objectAtIndex:i] objectForKey:@"CO2"];
    NSString * cCO = [[jsonArray objectAtIndex:i] objectForKey:@"CO"];
    NSString * cVOLTAJE = [[jsonArray objectAtIndex:i] objectForKey:@"VOLTAJE"];
    NSString * cPOVOLTAJE = [[jsonArray objectAtIndex:i] objectForKey:@"POVOLTAJE"];
```

Figura 3.58 Recepción de datos de las variables

- Para obtener los datos almacenados en jsonArray se requiere de un vector, una función For que guarda cada dato en la posición del vector con ayuda de un FSLineChart al ser una API para la creación de gráficos dinámicos y estáticos dentro de la aplicación iOS (Figura 3.59) en código de FSLineChart ANEXO D.

```

NSMutableArray* chartData1 = [NSMutableArray arrayWithCapacity:500000000];

if([self.nssestacion isEqualToString:@"HUMEDAD"] ){
    for(int i=0;i<jsonArray.count;i++) {
        chartData1[i] = [[jsonArray objectAtIndex:i] objectForKey:@"HUMEDAD"];
        _unidad.text= @"%RH";
        NSLog(@"kp%@", chartData1);
    }
}

```

Figura 3.59 Función FSLineChart

- Donde el eje Y corresponde al valor de la señal y el eje X al número de muestras que se obtuvo (Figura 3.60).

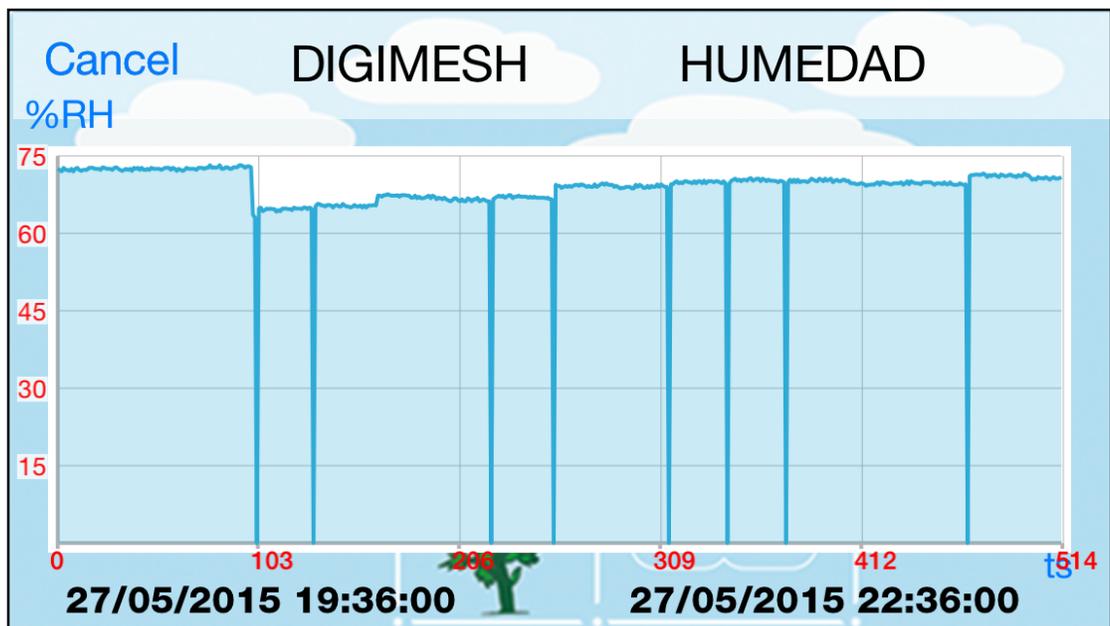


Figura 3.60 Visualización de la grafica

CAPÍTULO IV

ANÁLISIS COMPARATIVO

En este capítulo se describe el análisis del funcionamiento de las aplicaciones móviles en las plataformas Android e iOS, aplicado al monitoreo de las variables climatológicas de un invernadero de tomate de riñón en la barrio Rumipamba de Navas cantón Salcedo provincia de Cotopaxi. El objetivo de estas pruebas permitió determinar las diferencias entre aplicaciones en el consumo de datos, tiempo de respuesta y acceso a la red mediante Wifi y 3G.

4.1 CONSUMO DE DATOS

Al momento de realizar una petición desde la aplicación hacia el servidor web se requiere una conexión a internet ya sea mediante WIFI o 3G para obtener la respuesta, esta petición requerida por el cliente demanda un consumo de datos que serán analizados con el software libre TRAFFIC MONITOR que permite conocer la velocidad de los datos tanto de SUBIDA como de BAJADA de la información desde el servidor web hacia el dispositivo, a continuación se detalla su análisis:

4.1.1 WIFI

Para determinar el consumo con acceso wifi se realizó la toma de datos en los siguientes intervalos de tiempo y así determinar cuando existe mayor demanda de datos:

a) 1 MINUTO

Consumo de datos en el rango de una hora con un intervalo de 1 minuto, la velocidad de subida como se muestra en la Figura 4.1 es directamente

proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada (Figura 4.2) se observa un incremento exponencial al intervalo. En el ANEXO D se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.3):

Subida: 500,43 KB
Bajada: 17286,309 KB
Total: 17785,96 KB

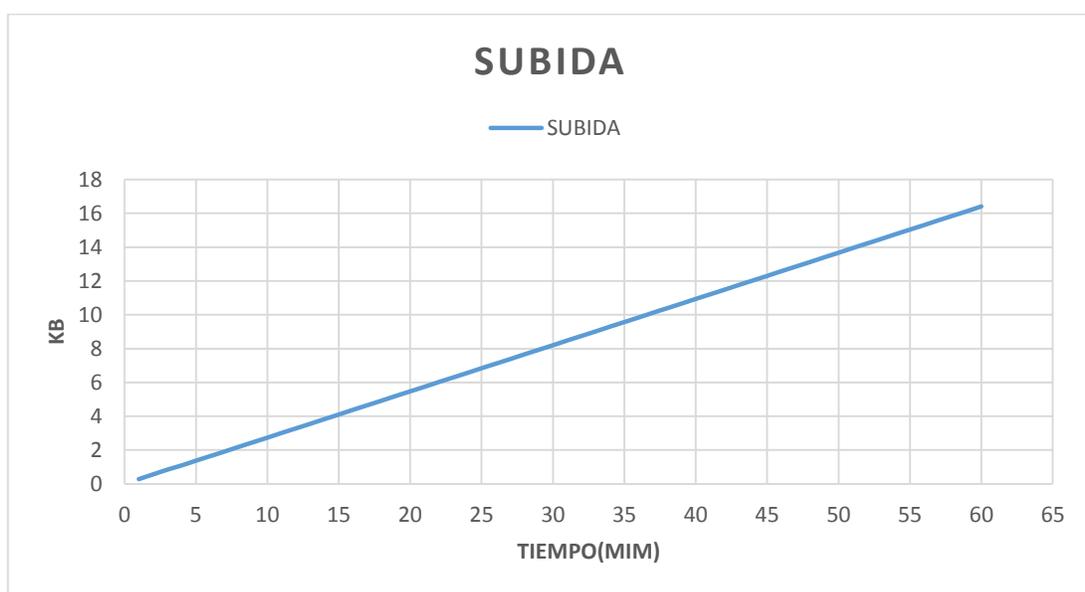


Figura 4.1 Consumo de datos de Subida (1 minuto wifi)

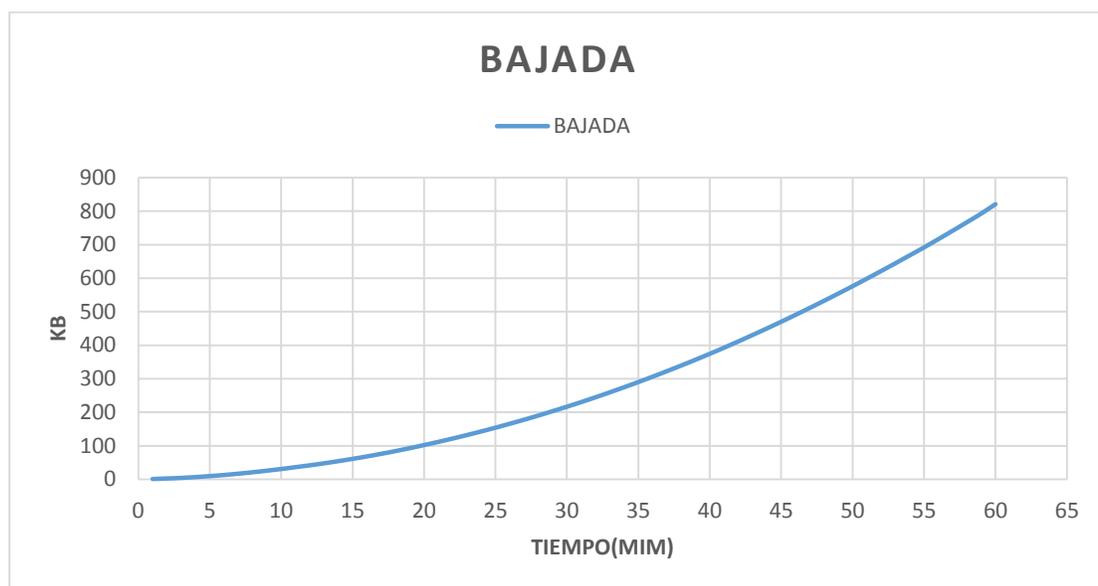


Figura 4.2 Consumo de datos de Bajada (1 minuto wifi)

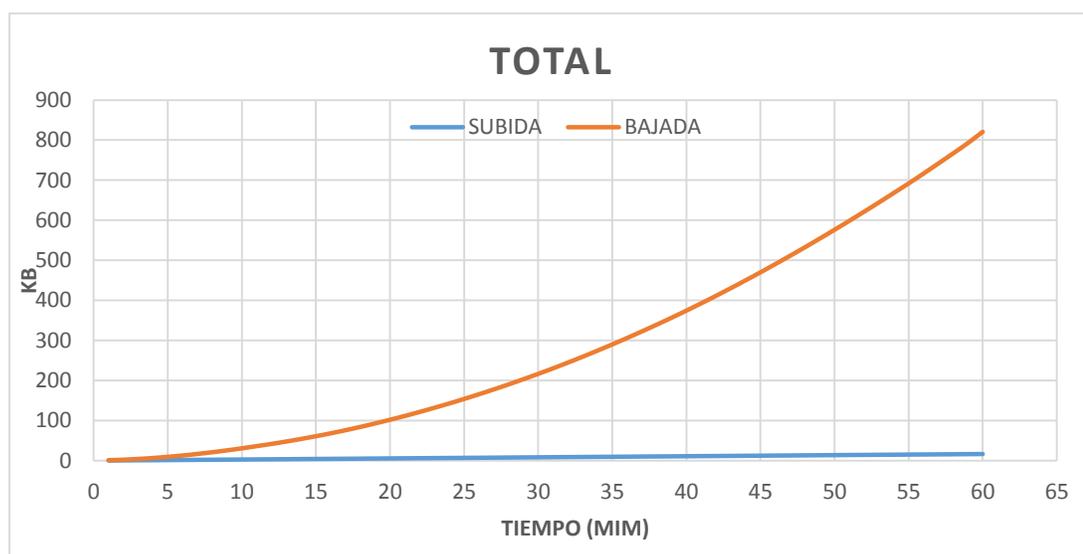


Figura 4.3 Consumo de datos (1 minuto wifi)

b) 10 MINUTOS

Consumo de datos en una hora con un intervalo de 10 minutos, la velocidad de subida como se muestra en la Figura 4.4 es directamente proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada(Figura 4.5) se observa un incremento exponencial al intervalo. En el

ANEXO D se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.6):

Subida: 9,979 KB

Bajada: 311,1 KB

Total: 321,05KB

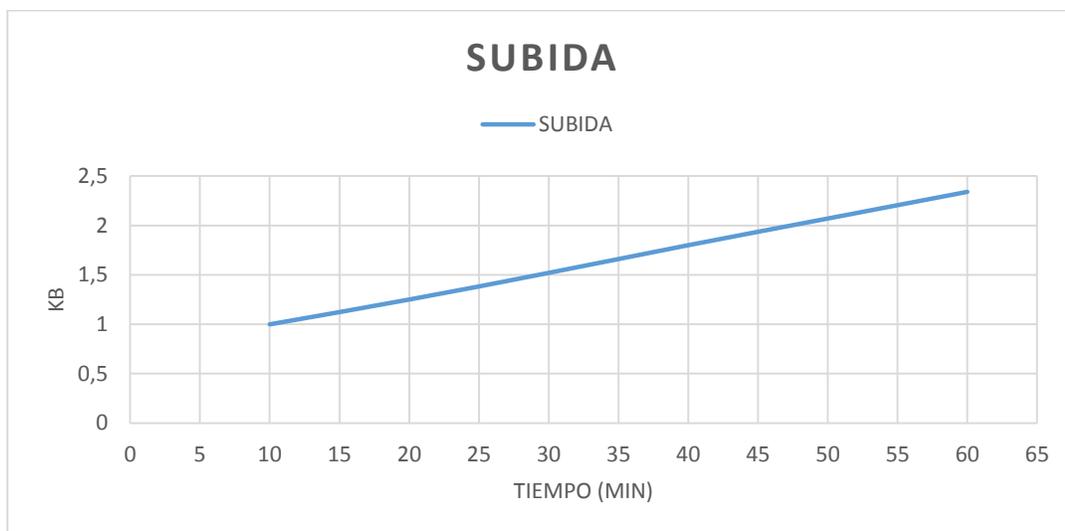


Figura 4.4 Consumo de datos de Subida (10 minuto wifi)

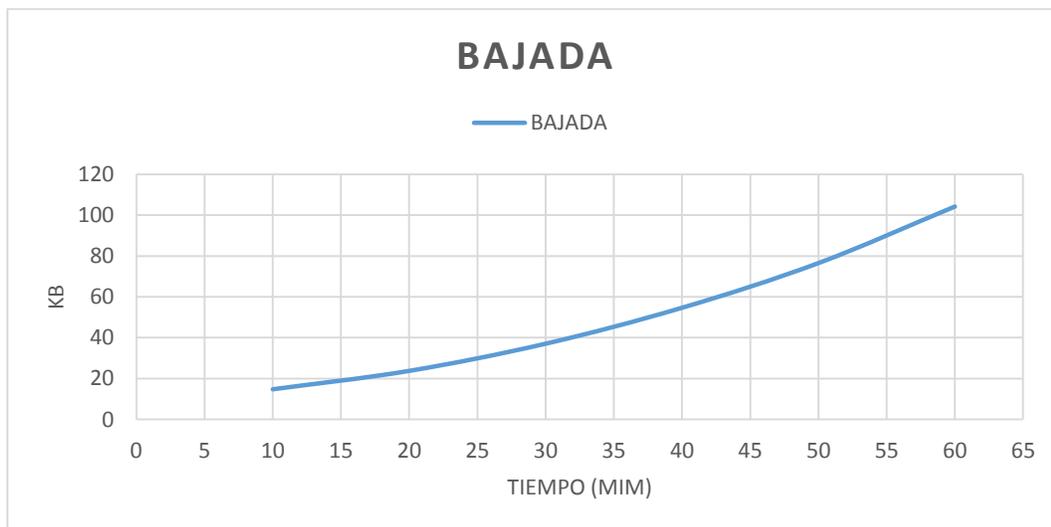


Figura 4.5 Consumo de datos de Bajada (10 minutos wifi)

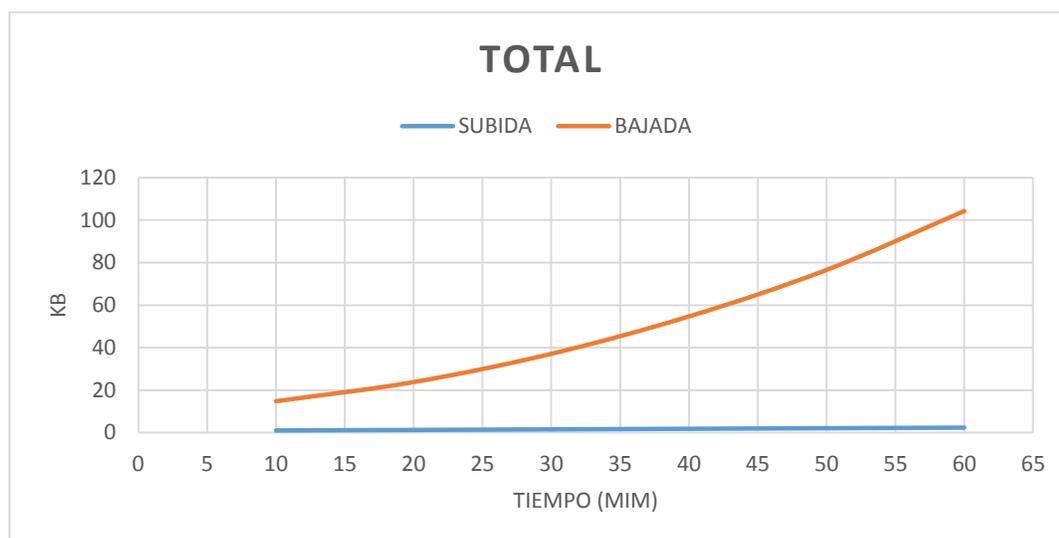


Figura 4.6 Consumo de datos (10 minutos wifi)

c) HORA

Consumo de datos en 9 horas con un intervalo de 1 hora, la velocidad de subida como se muestra en la Figura 4.10 es directamente proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada (Figura 4.11) se observa un incremento exponencial al intervalo. En el ANEXO D se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.9):

Subida: 12,34 KB

Bajada: 3715,32 KB

Total: 3726,91 KB

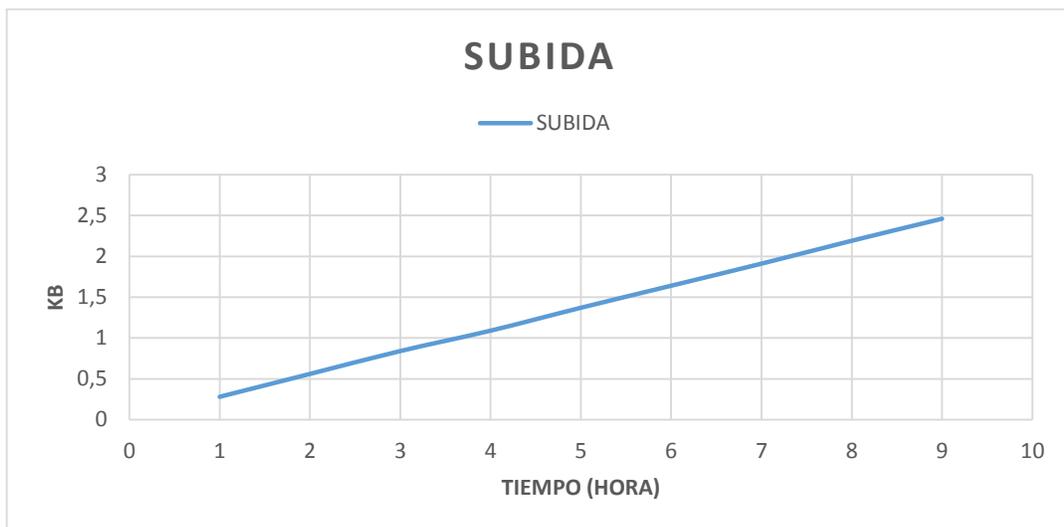


Figura 4.7 Consumo de datos de Subida (1 hora wifi)

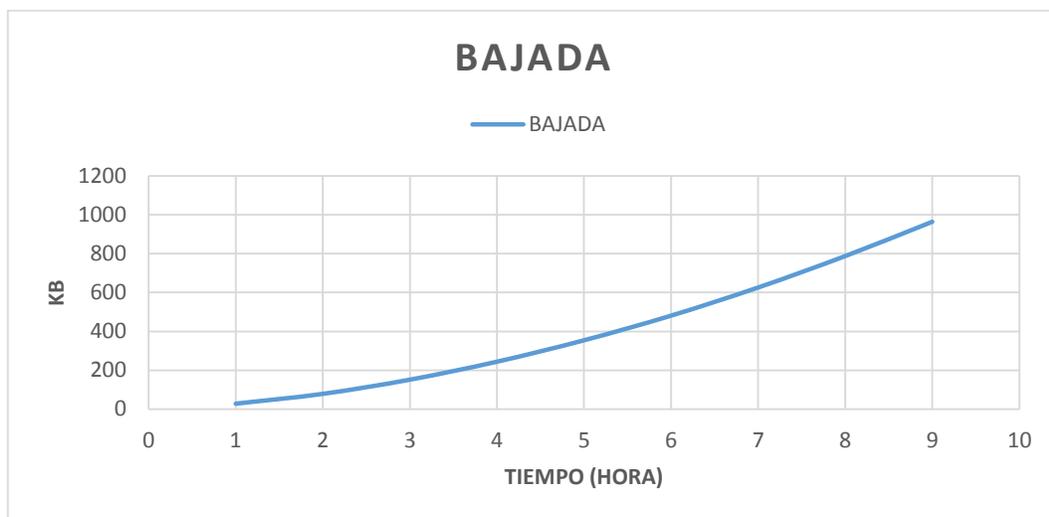


Figura 4.8 Consumo de datos de Bajada (1 hora wifi)

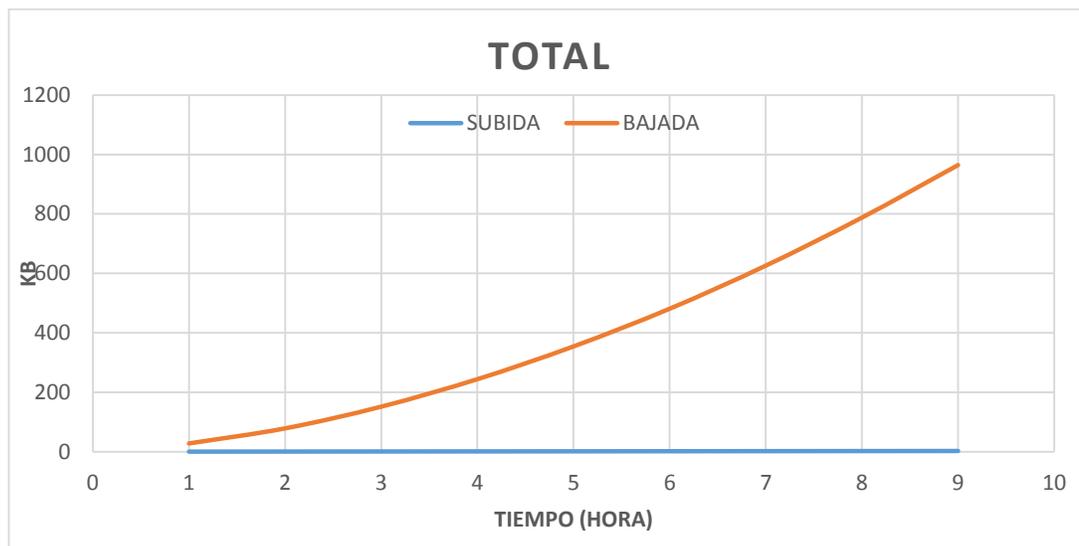


Figura 4.9 Consumo de datos (1 hora wifi)

4.1.2 3G

Para determinar el consumo con acceso 3G se realizó la toma de datos en los siguientes intervalos de tiempo y así determinar cuando existe mayor demanda de datos:

a) 1 MINUTO

Consumo de datos en una hora con un intervalo de 1 minuto, en la velocidad de subida como se muestra en la Figura 4.10 es directamente proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada (Figura 4.11) se observa un incremento exponencial al intervalo. En el ANEXO D se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.12).

Subida: 500,44KB

Bajada: 17426,168 KB

Total: 17925,55 KB

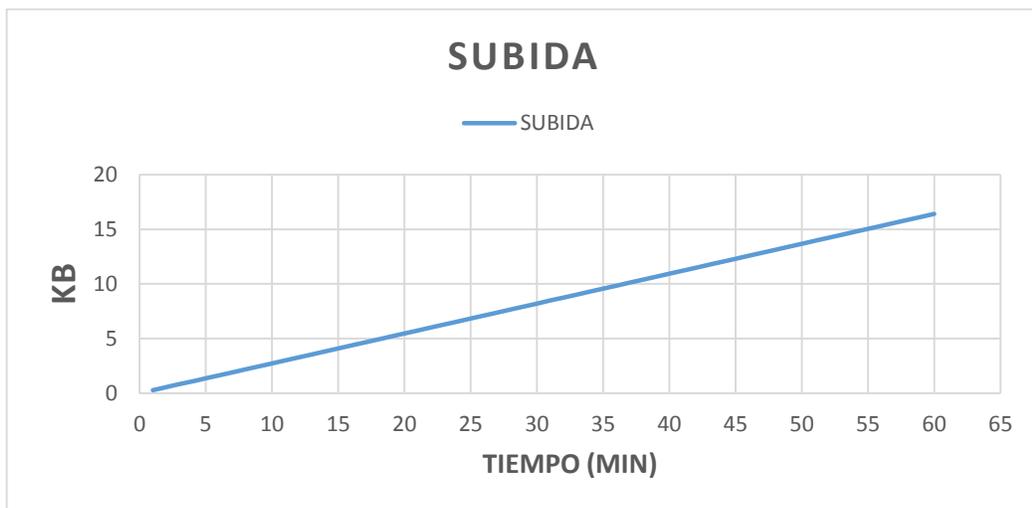


Figura 4.10 Consumo de datos de Subida (1 minuto 3G)

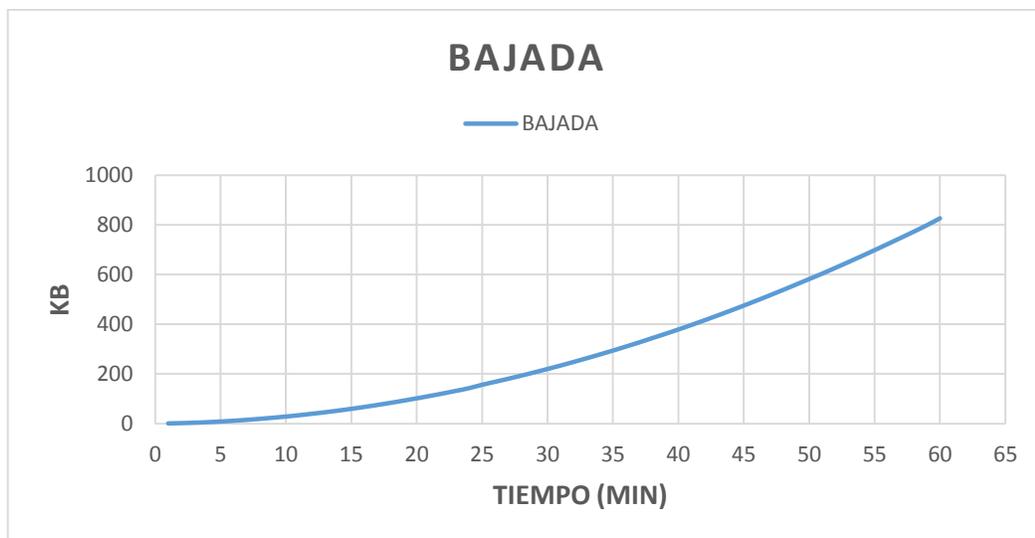


Figura 4.11 Consumo de datos de Bajada (1 minuto 3G)

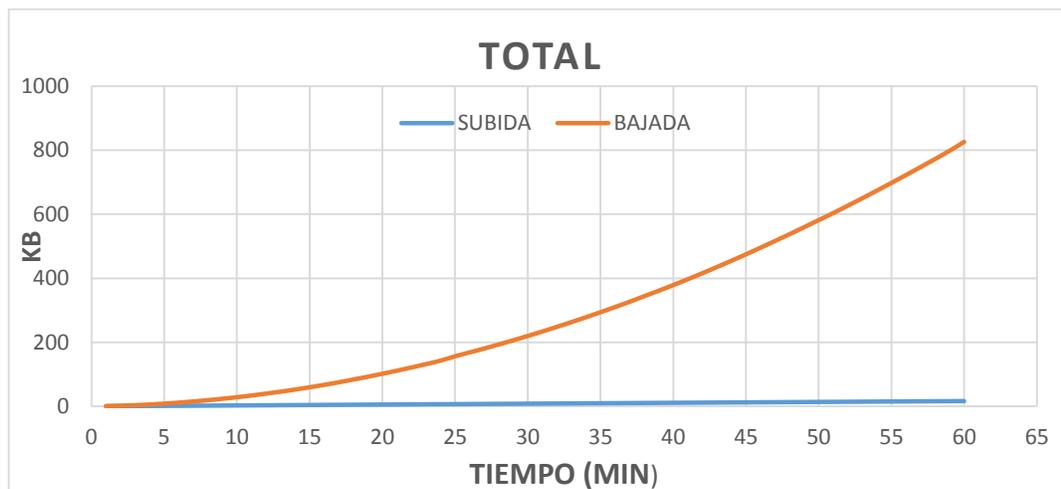


Figura 4.12 Consumo de datos (1 minuto 3G)

b) 10 MINUTOS

Consumo de datos en una hora con un intervalo de 10 minutos, la velocidad de subida como se muestra en la Figura 4.13 es directamente proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada (Figura 4.14) se observa un incremento exponencial al intervalo. En el ANEXO D se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.15):

Subida: 6,727KB

Bajada: 239,25KB

Total: 245,93KB

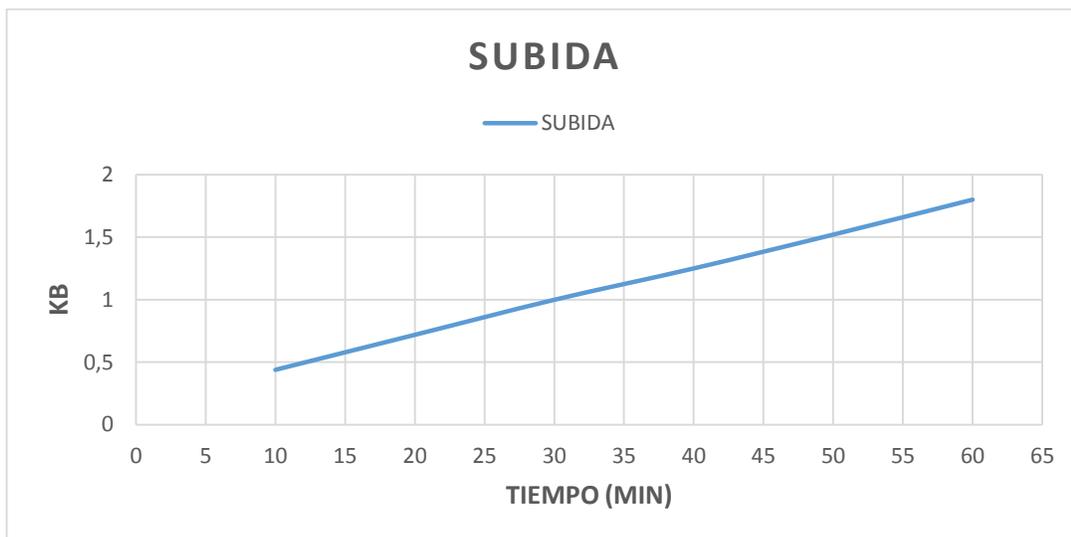


Figura 4.13 Consumo de datos de Subida (10 minutos 3G)

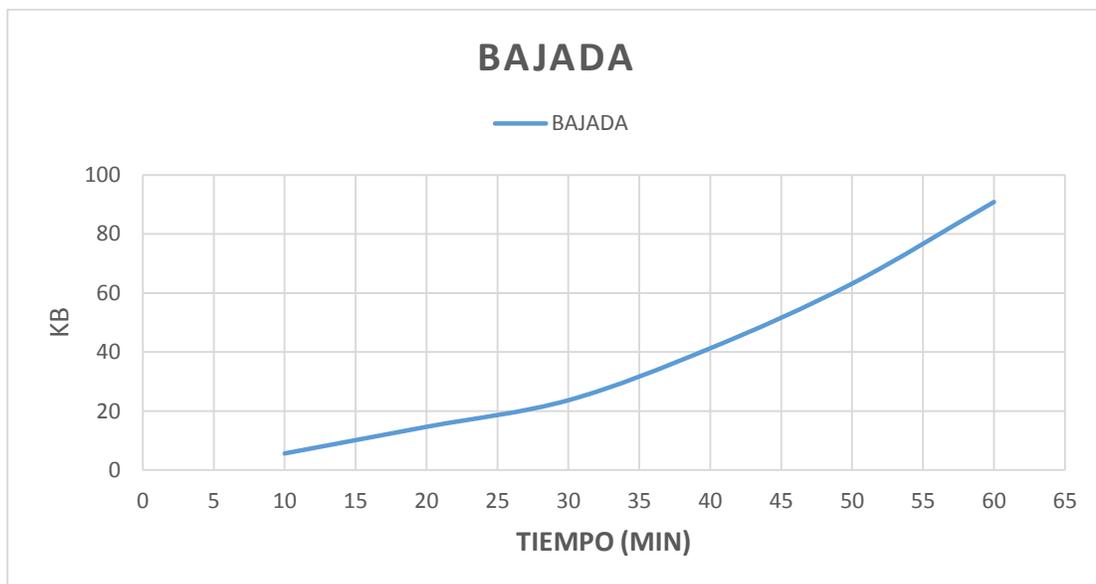


Figura 4.14 Consumo de datos de Bajada (10 minutos 3G)

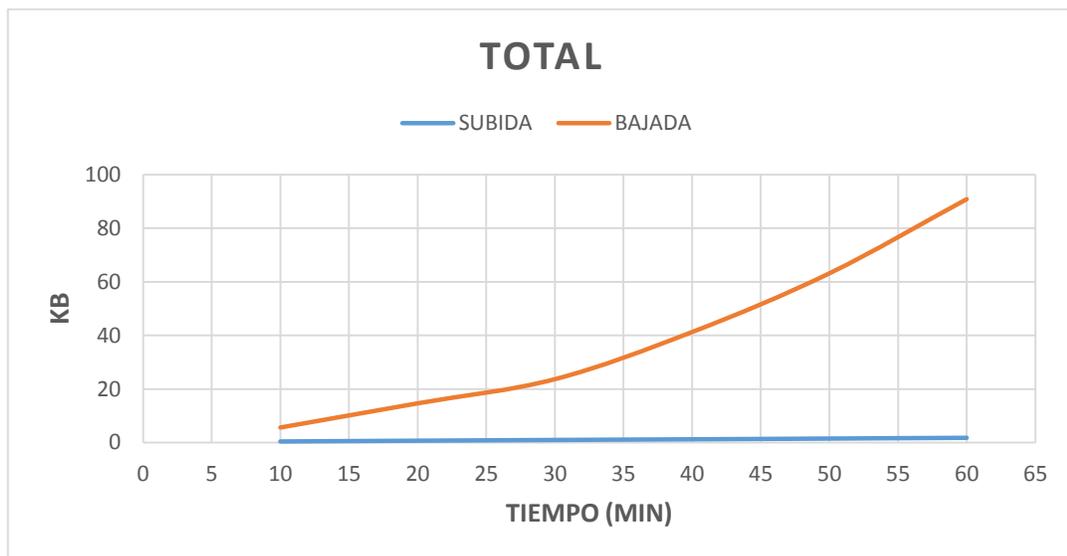


Figura 4.15 Consumo de datos (10 minutos 3G)

a) HORA

Consumo de datos en 9 horas con un intervalo de 1 hora, en la velocidad de subida como se muestra en la Figura 4.16 es directamente proporcional al intervalo con un incremento aproximado a 0,28 KB en cuanto para la velocidad de bajada (Figura 4.17) se observa un incremento exponencial al intervalo. En el se muestra la tabla de análisis de este intervalo con un consumo de (Figura 4.18):

Subida: 12,34 KB

Bajada: 3288,83 KB

Total: 3301,74 KB

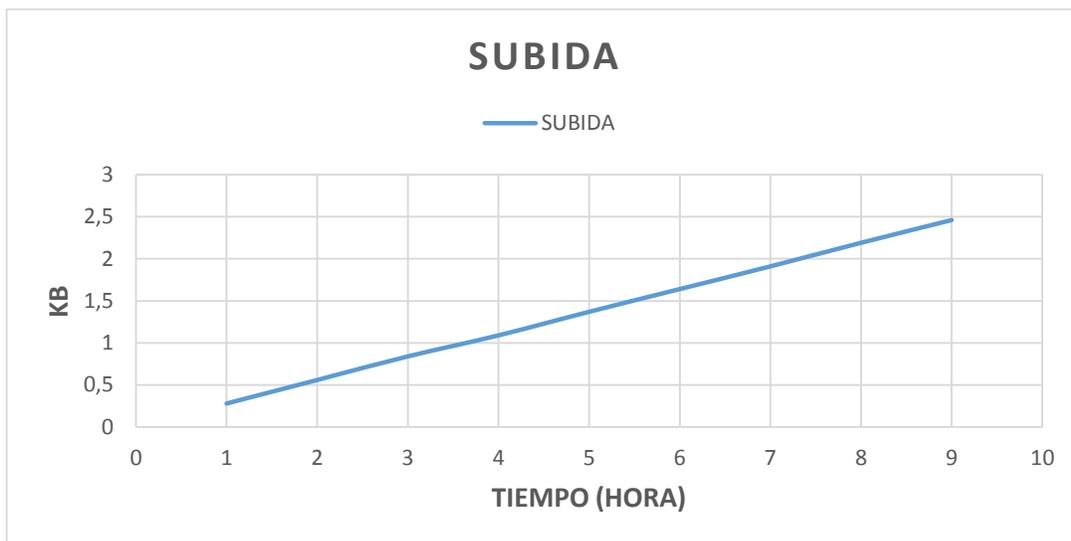


Figura 4.16 Consumo de datos de Subida (1 hora 3G)

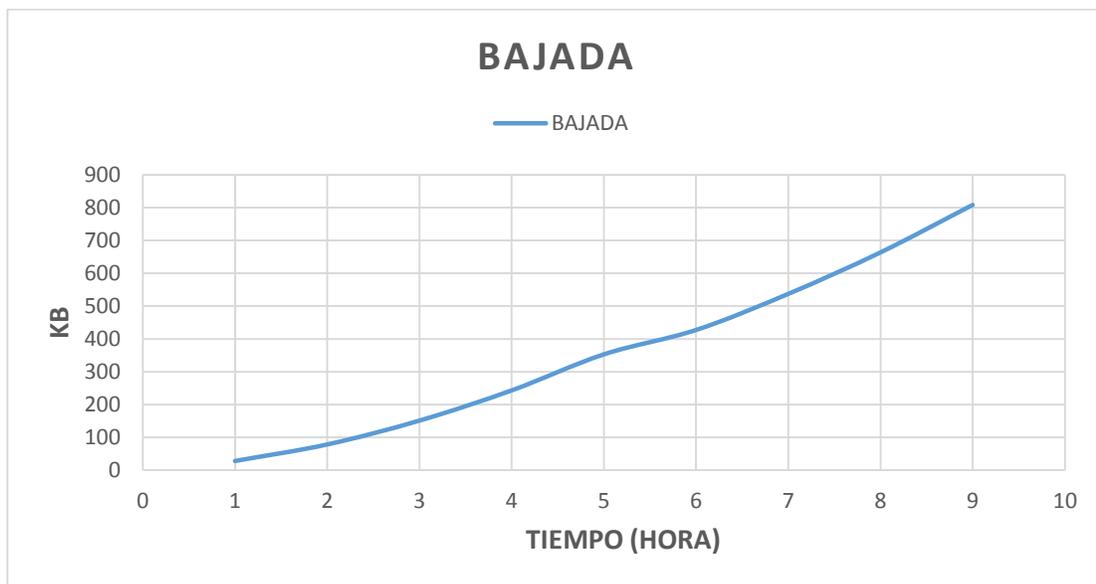


Figura 4.17 Consumo de datos de Bajada (1 hora 3G)

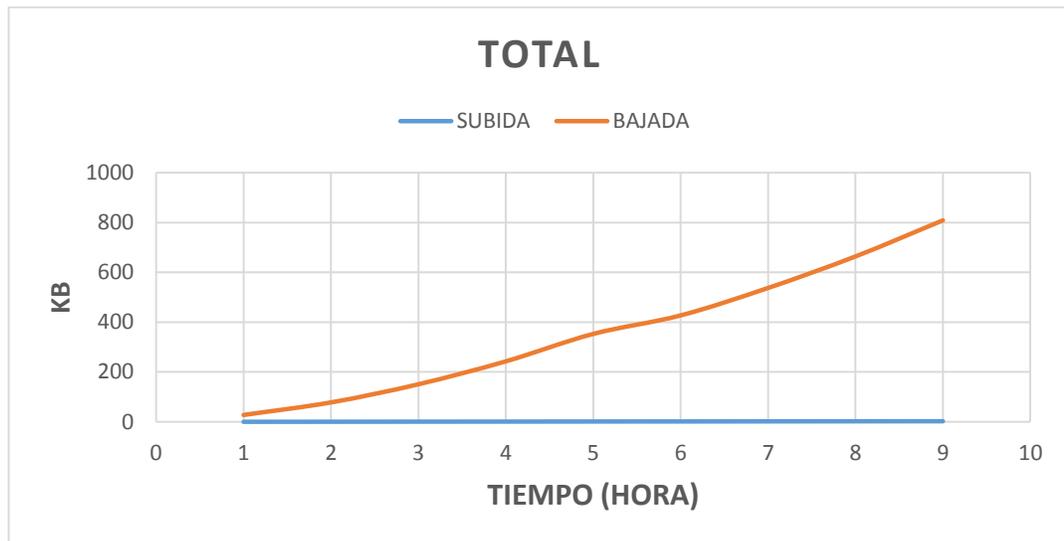


Figura 4.18 Consumo de datos (1 hora 3G)

4.1.3 COMPARACIÓN

- En los intervalos de minuto a minuto no existe gran diferencia con el consumo de datos mediante el acceso a WIFI y 3G, al tratarse de un rango pequeño con 3 datos por minuto de información no existe mayor demanda de datos en dichos intervalos, por otra parte el consumo total de datos es considerable por lo que se recomienda hacer las peticiones de consulta en intervalos mayores.
- En cuanto al intervalo de 10 minutos es mayor el consumo al existir un número aproximado de 27 datos de información, con los resultados obtenidos anteriormente el acceso a Wifi proporciona una mejor velocidad de datos que el acceso 3G.
- En el intervalo de 1 hora existe un aproximado de 170 datos de información, siendo el acceso a wifi el que proporcione mayor velocidad de datos que 3G.

4.2 TIEMPO DE RESPUESTA

Fue evaluado en ambas plataformas para determinar su velocidad de respuesta en diferentes intervalos de datos.

4.2.1 WIFI

En Tabla 4.1 se determinan el tiempo de respuesta de la aplicación en ambas plataformas, al tener iOS (Figura 4.19) un mejor procesador de memoria responde con mayor rapidez a la petición que Android (Figura 4.20)

Tabla 4.1

Tiempo de respuesta WIFI

| DATOS | RESPUESTA(ms) | |
|-------|---------------|-------|
| | ANDROID | iOS |
| 3 | 0,716 | 1,116 |
| 31 | 1,196 | 1,35 |
| 81 | 1,42 | 1,462 |
| 171 | 1,53 | 1,551 |
| 339 | 1,91 | 1,712 |
| 498 | 2,035 | 1,93 |
| 672 | 2,54 | 2,251 |
| 843 | 2,815 | 2,66 |
| 1026 | 3,706 | 3,258 |
| 1197 | 5,531 | 3,817 |
| 1368 | 6,282 | 4,116 |
| 1439 | 6,78 | 4,273 |

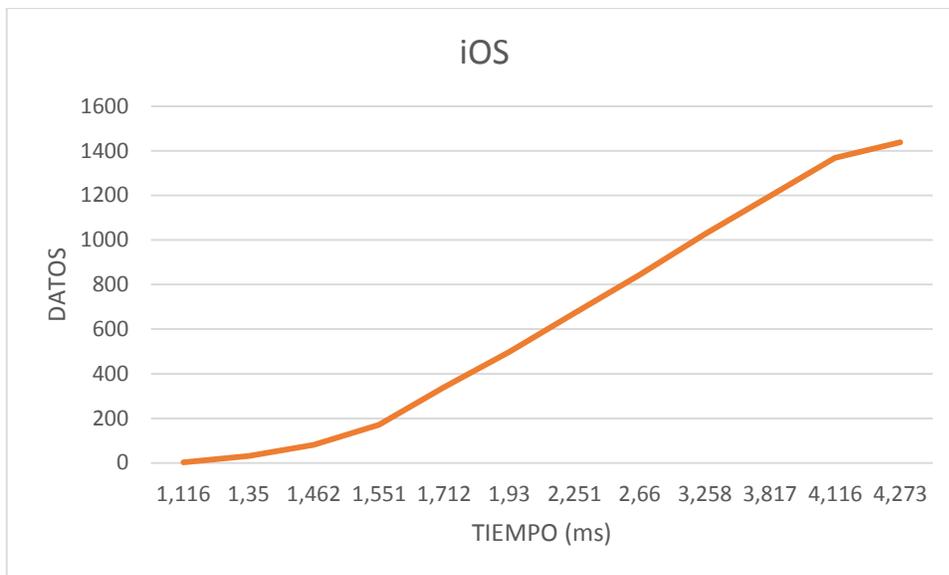


Figura 4.19 Tiempo de respuesta WIFI (iOS)

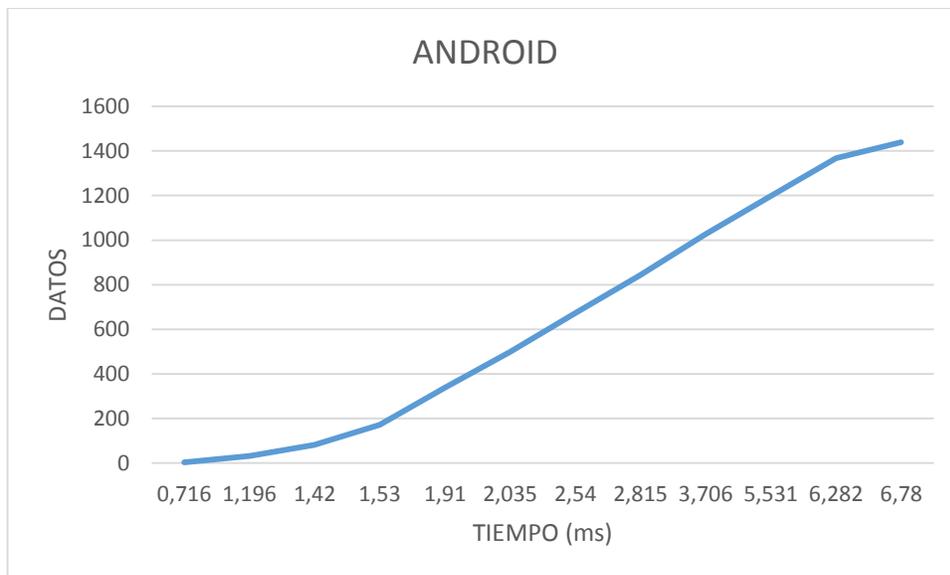


Figura 4.20 Tiempo de respuesta WIFI (Android)

4.2.2 3G

En Tabla 4.2 se determina el tiempo de respuesta de la aplicación en las dos plataformas, al tener iOS (Figura 4.21) un mejor procesador de memoria responde con mayor rapidez a la petición que Android (Figura 4.22).

Tabla 4.2

Tiempo de respuesta 3G

| DATOS | RESPUESTA(ms) | |
|-------------|---------------|-------|
| | ANDROID | iOS |
| 3 | 3,23 | 3,057 |
| 31 | 3,82 | 3,676 |
| 81 | 4,341 | 4,047 |
| 171 | 4,745 | 4,334 |
| 339 | 5,216 | 4,924 |
| 498 | 5,321 | 5,016 |
| 672 | 6,02 | 5,9 |
| 843 | 6,98 | 6,8 |
| 1026 | 7,84 | 7,34 |
| 1197 | 8,823 | 8,12 |
| 1368 | 9,967 | 9,164 |
| 1439 | 11,761 | 10,32 |

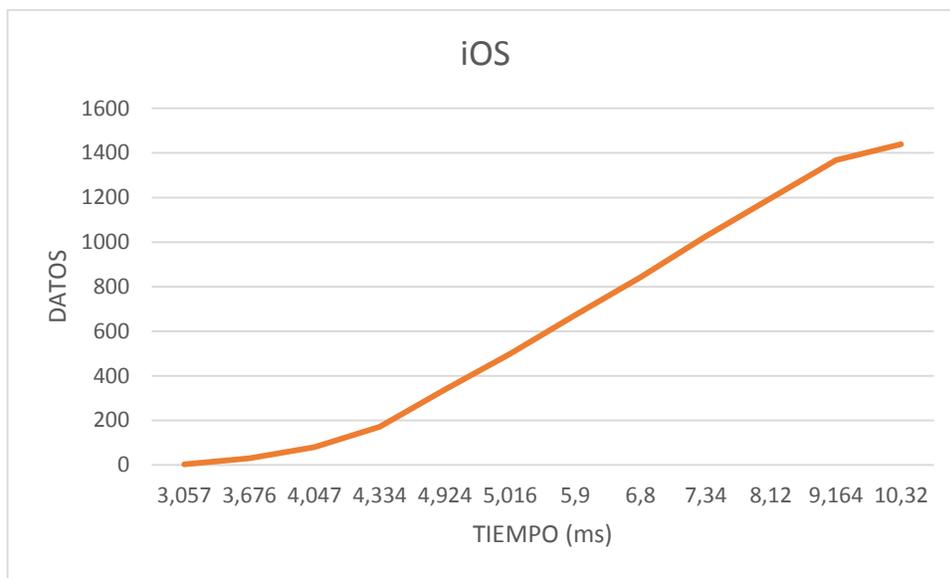


Figura 4.21 Tiempo de respuesta 3G (iOS)

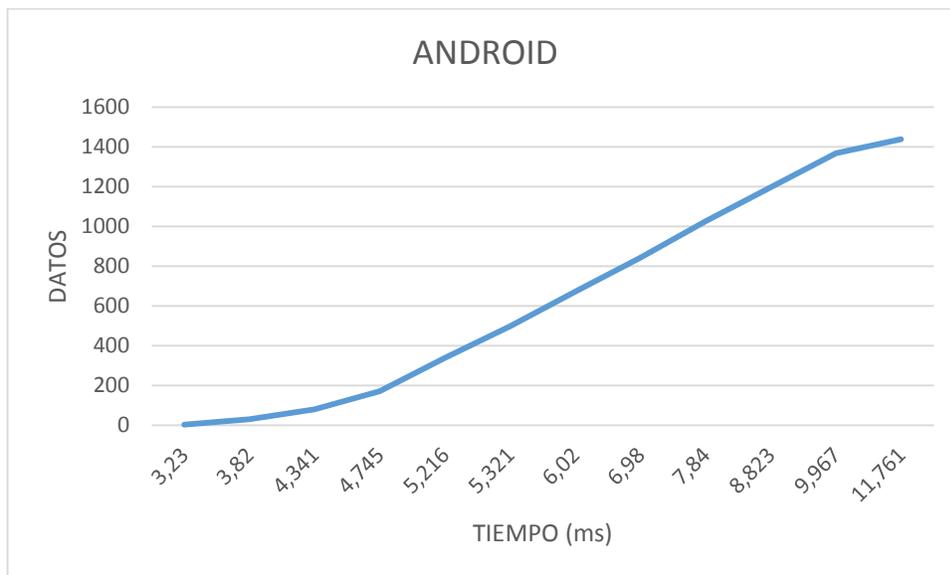


Figura 4.22 Tiempo de respuesta 3G (Android)

4.2.3 COMPARACIÓN

- En intervalos cortos de consulta no existe mayor diferencia en respuesta de las dos plataformas con acceso WIFI y 3G.
- Al realizar consultas superiores a las 3 horas con el acceso a 3G su tiempo de respuesta tarda más que con el acceso a wifi.
- La velocidad de respuesta con acceso 3G depende del plan contratado y del área de cobertura, en cuanto a acceso WIFI depende del número de dispositivos que se encuentren conectados a la red.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1.1 CONCLUSIONES

- Se cumplió con el objetivo propuesto en este proyecto, el cual consiste en el desarrollo de un sistema HMI en las plataformas Android e iOS para el monitoreo de la red inalámbrica de sensores agrícolas ubicada en el barrio Rumipamba de Navas cantón Salcedo, provincia de Cotopaxi.
- La aplicación desarrollada en iOS utiliza menor código de programación que la creada en Android; cumpliendo ambas con las mismas funcionalidades.
- La infraestructura de la comunicación de datos desplegada permite que dispositivos móviles con tecnología (Wifi, 3G, 4G). pueden acceder, desde cualquier parte del mundo a la aplicación creada.
- El consumo de datos en redes móviles está directamente relacionado con el volumen de datos a ser descargados desde de la base, por tal razón a fin de no saturar el servidor se limitó la descarga directa de datos del día, y solo se accederá a datos anteriores si el usuario lo requiere.
- La velocidad de trasmisión de la tecnología de datos móvil está limitado por su área de cobertura e infraestructura de comunicación de la operadora que presta el servicio.

- El análisis de las pruebas realizadas determinó que el tiempo y el consumo de descarga de datos, en la plataforma iOS es más eficiente por su arquitectura cerrada en hardware y software a diferencia de Android al ser una arquitectura abierta.
- El desarrollo del presente proyecto es una iniciativa que para los productores agrícolas a gran escala que implementen un monitoreo móvil mediante los dispositivos Smartphone que permita la visualización del comportamiento de las variables climatológicas que interfieren en la producción agrícola.
- En la actualidad Epsilon es la única empresa en la región de Latinoamérica que comercializa equipos para procesos agrícolas, su interfaz de monitoreo se restringe a ser visualizado vía web, no disponen de aplicaciones móviles que permitan el monitoreo de dichos procesos.
- La ventaja principal de utilizar software libre es que su costo de licencia es nulo, a diferencia del software privado que necesita de una licencia altamente costosa para poder desarrollar aplicaciones.

5.1.2 RECOMENDACIONES

- El trabajo a futuro en este proyecto sería realizar la etapa de control mediante dispositivos Smartphone y de esta manera conseguir un mayor impacto en la agricultura inteligente.
- Para poder acceder al servidor web desde cualquier lugar geográfico, se requiere conocer su dirección web configurada por un dominio privado.
- En la plataforma Android requiere de una versión de Android 4.0.3 o superior, en cuanto la plataforma iOS requiere de una versión iOS 8.1 o superior.
- Al desarrollar la programación en Eclipse es necesario asignar los permisos de conexión a internet en el archivo manifest.
- Para crear un proyecto nuevo de Android e iOS es importante indicar con que versiones va a ser compatible la aplicación, y que estas se encuentre actualizadas mediante el SDK.
- Para depurar la aplicación desarrollada en iOS, se realizó JailBreak al software de desarrollo Xcode y al dispositivo Smartphone.

LINKGRAFÍA

- [1] S. Marcombo, Telecomunicaciones móviles, España: Marcombo, 1998.
- [2] S. Institute, «SANS,» 2011. [En línea]. Available: http://www.securingthehuman.org/newsletters/ouch/issues/OUCH-201102_sp.pdf. [Último acceso: 28 Noviembre 2014].
- [3] D. Sanz, M. Saucedo y P. Torralbo, «Introducción a Android,» [En línea]. Available: <http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/android.pdf>. [Último acceso: 28 Octubre 2014].
- [4] B. Castillo, «Android OS,» 05 11 2012. [En línea]. Available: <http://androidos.readthedocs.org/en/latest/data/caracteristicas/>. [Último acceso: 12 10 2014].
- [5] B. Campderrich , Ingeniería del Software, Madrid : Pearson Educación, 2005. [Último acceso: 08 Febrero 2015].
- [6] G. Fernández, iOS, Todo Lo Que Siempre tiene querido sobre Sobre tu iPhone y iPad, 2013. [Último acceso: 08 Febrero 2015].
- [7] «Tecnología iOS,» [En línea]. Available: <https://sites.google.com/site/tecnologiaiostm/desarrollo-de-aplicaciones/arquitectura-ios>. [Último acceso: 12 Abril 2015].
- [8] R. Fernando, «Lenguaje programación iOS,» cocoaosx, 27 01 2012. [En línea]. Available: <http://www.cocoaosx.com/2012/01/27/lenguaje-programacion-iphone/>. [Último acceso: 12 Abril 2015].
- [9] A. Garcia-Cabot y L. Oto, «Accesibilidad en Smartphones para el acceso a contenidos e-learning,» esvial. [Último acceso: 22 Mayo 2015].
- [10] E. Lara, «Protocolo HTTP y Servidor Web,» [En línea]. Available: <http://personals.ac.upc.edu/elara/documentacion/INTERNET%20-%20UD8%20-%20Protocolo%20HTTP%20y%20servidores%20WEB.pdf>. [Último acceso: 08 Octubre 2014].

- [11] UNICEN. [En línea]. Available: <http://www.exa.unicen.edu.ar/catedras/comdat1/material/ElmodeloOSI.pdf>. [Último acceso: 05 10 2014].
- [12] F. Ortiz y F. Candelas, *Práctica de redes*, San Vicente: Club Universitario, 2002. [Último acceso: 12 Abril 2015].
- [13] B. Guzman, «Tecnosaurios,» Medios de acceso a internet, 17 11 2011. [En línea]. Available: <http://tecnosaurios.blogspot.com/2011/10/medios-de-acceso-internet.html>. [Último acceso: 12 Mayo 2015].
- [14] R. Alemañ Cabrera y D. García Copete , «Práctica de NTAE,» [En línea]. Available: <http://www.alu.ua.es/r/rac6/Buscadores/indice.html>. [Último acceso: 12 Diciembre 2014].
- [15] . G. Bianchi, «Desarrollo de Aplicaciones para dispositivos Móviles sobre la plataforma Android de Google,» 13 07 2011. [En línea]. Available: <http://www.dit.ing.unp.edu.ar/graduate/bitstream/123456789/206/1/Informe.pdf>. [Último acceso: 12 Diciembre 2014].
- [16] L. Desarro. , «Scielo,» Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina, 23 06 2009. [En línea]. Available: http://www.scielo.co/scielo.php?script=sci_arttext&pid=S0122-34612009000100011&lng=es&nrm=iso. [Último acceso: 22 Diciembre 2014].
- [17] «kioskea.net,» Introducción a Wi-Fi (802.11), 12 03 2015. [En línea]. Available: <http://es.kioskea.net/contents/789-introduccion-a-wi-fi-802-11-o-wifi>. [Último acceso: 27 Junio 2015].
- [18] «EDUred,» Telefonía móvil 3G., [En línea]. Available: http://www.ecured.cu/index.php/Telefon%C3%ADa_m%C3%B3vil_3G.
- [19] S. Escolar, WIRELESS SENSOR NETWORKS. [Último acceso: 27 Junio 2015].

- [20] F. Sandell, «ZigBee wireless sensor networks ideas for further development,» Gotemburgo, 2009. [Último acceso: 27 Junio 2015].
- [21] P. Singh, Analysis of ZigBee (IEEE 802.15.4 standard), Allahabada: IJITEE, 2013. [Último acceso: 17 Febrero 2015].
- [22] DigiMesh, «The DigiMesh™ Networking Protocol,» 1996-2014. [En línea]. Available: <http://www.digi.com/technology/digimesh/>. [Último acceso: 12 06 2015].
- [23] F. Andreu, Fundamentos y aplicaciones de seguridad en redes WLAN, España: Marcombo, 2006. [Último acceso: 25 Enero 2015].
- [24] C. Falcón, Wi-Fi : lo que se necesita conocer, Madrid: Grupo RC, 2010. [Último acceso: 25 Junio 2015].

ANEXOS

ANEXO A

CÓDIGO DE LOS ARCHIVOS PHP RED
(WI-FI)

CÓDIGO NODO 1

```
<?php
    $host = "localhost";//host
    $usuario = "root";//usuarioBD
    $pass = "";//Contraseña
    $bd = "wifi";//Base deDatos

    $servidor = mysql_connect($host, $usuario, $pass);
    mysql_set_charset("utf8", $servidor);
    $conexion = mysql_select_db($bd, $servidor) or die ("pro_select_db");
    $textinicio = $_POST["textinicio"];
    $textfinal = $_POST["textfinal"];

    $cons= "SELECT TEMPERATURA,HUMEDAD,LUMINOSIDAD,RADIACION,
    DIVIENTO,VOLTAJE,POVOLTAJE,FECHA,TABLA FROM nodo1 WHERE
    FECHA BETWEEN '$textinicio' AND '$textfinal' ";

    $sql=mysql_query( $cons);
        $records= array();
        while ($obj = mysql_fetch_assoc($sql)) {
            $records []= $obj;
        }
    echo json_encode($records);

?>
```

CÓDIGO GENERAL

```
<?php
```

```
$host = "localhost";//host
```

```
$usuario = "root";//usuarioBD
```

```
$pass = "";//Contraseña
```

```
$bd = "zigbee";//Base deDatos
```

```
$servidor = mysql_connect($host, $usuario, $pass);
```

```
$conexion = mysql_select_db($bd, $servidor);
```

```
$consulta = "SELECT TEMPERATURA ,HUMEDAD, LUMINOSIDAD,  
RADIACION, ULTRAVIOLETA, VIENTO, DIVIENTO, VOLTAJE,  
POVOLTAJE, FECHA, TABLA FROM dato1 ORDER by id DESC LIMIT  
3";
```

```
$sql = mysql_query($consulta);
```

```
if ( ! $sql ) {
```

```
    echo "La conexion no se logró".mysql_error();
```

```
    die;
```

```
}
```

```
$records= array();
```

```
while ($obj = mysql_fetch_assoc($sql)) {
```

```
    $records []= $obj;
```

```
}
```

```
    echo json_encode($records);
```

```
?>
```

ANEXO B

CÓDIGO DE PROGRAMACIÓN ANDROID

Para el acceder al código de programación en Android diríjase al CD que se encuentra a disposición en la biblioteca de la institución.

ANEXO C

CÓDIGO DE PROGRAMACIÓN XCODE

Para el acceder al código de programación en iOS diríjase al CD que se encuentra a disposición en la biblioteca de la institución

ANEXO D

**ANÁLISIS DE RESULTADOS CON ACCESO
A WIFI**

INTERVALO DE 1 MINUTO

| HISTORICOS | | | | |
|------------|-------|------------|----------------|-----------|
| MINUTO | DATOS | SUBIDA(KB) | BAJADA (KB) | TOTAL(KB) |
| 1 | 3 | 0,28 | 0,929 | 1,18 |
| 2 | 6 | 0,56 | 2,29 | 2,84 |
| 3 | 9 | 0,84 | 4,16 | 4,98 |
| 4 | 12 | 1,09 | 6,49 | 7,58 |
| 5 | 16 | 1,37 | 9,47 | 10,82 |
| 6 | 18 | 1,64 | 12,74 | 14,38 |
| 7 | 21 | 1,91 | 16,66 | 18,57 |
| 8 | 24 | 2,19 | 20,89 | 23,08 |
| 9 | 27 | 2,46 | 25,61 | 28,07 |
| 10 | 29 | 2,73 | 30,64 | 33,37 |
| 11 | 32 | 3,01 | 36,14 | 39,15 |
| 12 | 34 | 3,28 | 41,65 | 44,93 |
| 13 | 37 | 3,55 | 47,64 | 51,19 |
| 14 | 39 | 3,83 | 54,1 | 57,93 |
| 15 | 43 | 4,1 | 60,89 | 64,99 |
| 16 | 45 | 4,38 | 68,15 | 72,52 |
| 17 | 47 | 4,65 | 75,88 | 80,53 |
| 18 | 52 | 4,92 | 84,11 | 89,03 |
| 19 | 54 | 5,2 | 92,81 | 98 |
| 20 | 56 | 5,47 | 101,98 | 107,45 |
| 21 | 59 | 5,74 | 111,48 | 117,22 |

CONTINÚA 

| | | | | |
|-----------|-----------|-------------|---------------|---------------|
| 22 | 61 | 6,02 | 121,45 | 127,46 |
| 23 | 63 | 6,29 | 131,89 | 138,18 |
| 24 | 66 | 6,56 | 142,66 | 149,22 |
| 25 | 69 | 6,84 | 153,9 | 160,74 |
| 26 | 71 | 7,11 | 165,62 | 172,73 |
| 27 | 74 | 7,38 | 177,66 | 185,04 |
| 28 | 77 | 7,66 | 190,18 | 197,84 |
| 29 | 79 | 7,93 | 203,17 | 211,1 |
| 30 | 81 | 8,2 | 216,49 | 224,69 |
| 31 | 85 | 8,48 | 230,28 | 238,75 |
| 32 | 88 | 8,75 | 244,54 | 253,29 |
| 33 | 91 | 9,02 | 259,27 | 268,3 |
| 34 | 97 | 9,3 | 274,49 | 283,78 |
| 35 | 99 | 9,57 | 290,02 | 299,59 |
| 36 | 102 | 9,84 | 306,03 | 315,87 |
| 37 | 105 | 10,12 | 322,51 | 332,63 |
| 38 | 107 | 10,39 | 339,32 | 349,71 |
| 39 | 110 | 10,66 | 356,6 | 367,26 |
| 40 | 111 | 10,94 | 374,35 | 385,29 |
| 41 | 114 | 11,21 | 392,59 | 403,8 |
| 42 | 117 | 11,48 | 411,3 | 422,79 |
| 43 | 120 | 11,76 | 430,49 | 442,25 |
| 44 | 123 | 12,03 | 450,03 | 462,03 |
| 45 | 125 | 12,3 | 469,99 | 482,29 |
| 46 | 128 | 12,58 | 490,45 | 503,03 |
| 47 | 130 | 12,85 | 511,23 | 524,09 |
| 48 | 133 | 13,12 | 532,49 | 545,62 |
| 49 | 135 | 13,4 | 554,07 | 567,47 |

CONTINÚA



| | | | | |
|--------------|------------|--------------|---------------|---------------|
| 50 | 138 | 13,67 | 576,13 | 589,13 |
| 51 | 141 | 13,95 | 598,5 | 612,45 |
| 52 | 143 | 14,22 | 621,35 | 635,57 |
| 53 | 146 | 14,49 | 644,52 | 659,01 |
| 54 | 148 | 14,77 | 668,16 | 682,92 |
| 55 | 151 | 15,04 | 691,8 | 706,84 |
| 56 | 153 | 15,31 | 716,39 | 731,71 |
| 57 | 155 | 15,59 | 741,46 | 757,05 |
| 58 | 158 | 15,86 | 766,84 | 782,7 |
| 59 | 161 | 16,13 | 792,69 | 808,83 |
| 60 | 162 | 16,41 | 820,69 | 837,1 |
| TOTAL | | 500,43 | 17286,309 | 17785,96 |

INTERVALO DE 10 MINUTOS

| MINUTO | HISTORICOS | | | |
|--------------|------------|------------|------------|-----------|
| | DATOS | SUBIDA(KB) | BAJADA(KB) | TOTAL(KB) |
| 10 | 27 | 0,999 | 14,75 | 15,73 |
| 20 | 55 | 1,25 | 23,77 | 25,02 |
| 30 | 82 | 1,52 | 37,08 | 38,6 |
| 40 | 109 | 1,8 | 54,68 | 56,47 |
| 50 | 136 | 2,07 | 76,57 | 78,64 |
| 60 | 161 | 2,34 | 104,25 | 106,59 |
| total | | 9,979 | 311,1 | 321,05 |

INTERVALO DE 1 HORA

| HORA | HISTORICOS | | | |
|--------------|------------|------------|------------|-----------|
| | DATOS | SUBIDA(KB) | BAJADA(KB) | TOTAL(KB) |
| 1 | 171 | 0,28 | 27,83 | 28,11 |
| 2 | 317 | 0,56 | 78,53 | 79,07 |
| 3 | 460 | 0,84 | 151,59 | 152,42 |
| 4 | 582 | 1,09 | 244,03 | 245,03 |
| 5 | 695 | 1,37 | 354,26 | 355,62 |
| 6 | 801 | 1,64 | 481,06 | 482,06 |
| 7 | 913 | 1,91 | 625,99 | 627,91 |
| 8 | 1021 | 2,19 | 787,61 | 789,8 |
| 9 | 1130 | 2,46 | 964,42 | 966,89 |
| TOTAL | | 12,34 | 3715,32 | 3726,91 |

ANÁLISIS CON ACCESO A 3G

INTERVALO DE 1 MINUTO

| MINUTO | HISTORICOS | | | |
|--------|------------|------------|------------|-----------|
| | DATOS | SUBIDA(KB) | BAJADA(KB) | TOTAL(KB) |
| 1 | 3 | 0,28 | 0,768 | 1,02 |
| 2 | 6 | 0,56 | 1,98 | 2,52 |
| 3 | 8 | 0,84 | 3,52 | 4,34 |
| 4 | 12 | 1,09 | 5,7 | 6,79 |
| 5 | 16 | 1,37 | 8,35 | 9,72 |
| 6 | 18 | 1,64 | 11,47 | 13,12 |
| 7 | 21 | 1,91 | 15,08 | 16,99 |
| 8 | 24 | 2,19 | 19,16 | 21,34 |
| 9 | 27 | 2,46 | 23,55 | 26,01 |

CONTINÚA



| | | | | |
|-----------|-----------|-------------|--------------|--------------|
| 10 | 29 | 2,73 | 28,42 | 31,16 |
| 11 | 32 | 3,01 | 33,77 | 36,78 |
| 12 | 34 | 3,28 | 39,6 | 42,88 |
| 13 | 37 | 3,55 | 45,75 | 49,3 |
| 14 | 39 | 3,83 | 52,37 | 56,2 |
| 15 | 43 | 4,1 | 59,47 | 63,57 |
| 16 | 45 | 4,38 | 67,05 | 71,42 |
| 17 | 47 | 4,65 | 75,1 | 79,74 |
| 18 | 52 | 4,92 | 83,64 | 88,56 |
| 19 | 54 | 5,2 | 92,49 | 97,69 |
| 20 | 56 | 5,47 | 101,67 | 107,14 |
| 21 | 59 | 5,74 | 111,32 | 117,06 |
| 22 | 61 | 6,02 | 121,29 | 127,31 |
| 23 | 63 | 6,29 | 131,74 | 138,03 |
| 24 | 66 | 6,56 | 142,66 | 149,23 |
| 25 | 69 | 6,84 | 156,29 | 163,12 |
| 26 | 71 | 7,11 | 168,33 | 175,43 |
| 27 | 74 | 7,38 | 180,52 | 187,9 |
| 28 | 77 | 7,66 | 193,2 | 200,85 |
| 29 | 79 | 7,93 | 206,35 | 214,28 |
| 30 | 81 | 8,2 | 219,82 | 228,02 |
| 31 | 85 | 8,49 | 233,76 | 242,24 |
| 32 | 88 | 8,75 | 248,18 | 256,93 |
| 33 | 91 | 9,02 | 263,08 | 272,08 |
| 34 | 97 | 9,3 | 278,29 | 287,58 |
| 35 | 99 | 9,57 | 293,97 | 303,54 |
| 36 | 102 | 9,84 | 310,14 | 319,98 |
| 37 | 105 | 10,12 | 326,62 | 336,74 |

CONTINÚA



| | | | | |
|---------------|------------|--------------|---------------|---------------|
| 38 | 107 | 10,39 | 343,98 | 353,98 |
| 39 | 110 | 10,66 | 361,02 | 371,02 |
| 40 | 111 | 10,94 | 378,78 | 389,72 |
| 41 | 114 | 11,21 | 397,01 | 408,22 |
| 42 | 117 | 11,48 | 415,88 | 427,37 |
| 43 | 120 | 11,76 | 435,07 | 446,83 |
| 44 | 123 | 12,03 | 454,74 | 466,77 |
| 45 | 125 | 12,3 | 474,88 | 487,19 |
| 46 | 128 | 12,58 | 495,5 | 508,08 |
| 47 | 130 | 12,85 | 516,44 | 529,29 |
| 48 | 133 | 13,12 | 537,86 | 550,98 |
| 49 | 135 | 13,4 | 559,59 | 572,99 |
| 50 | 138 | 13,67 | 581,65 | 595,32 |
| 51 | 141 | 13,95 | 603,7 | 617,65 |
| 52 | 143 | 14,22 | 626,71 | 640,93 |
| 53 | 146 | 14,49 | 650,03 | 664,53 |
| 54 | 148 | 14,77 | 673,83 | 688,6 |
| 55 | 151 | 15,04 | 697,95 | 712,95 |
| 56 | 153 | 15,31 | 722,54 | 737,85 |
| 57 | 155 | 15,59 | 747,45 | 763,03 |
| 58 | 158 | 15,86 | 772,53 | 788,53 |
| 59 | 161 | 16,13 | 798,36 | 814,5 |
| 60 | 162 | 16,41 | 826,2 | 842,61 |
| TOTAL: | | 500,44 | 17426,168 | 17925,55 |

INTERVALO DE 10 MINUTOS

| MINUTO | HISTORICOS | | | |
|--------------|------------|------------|------------|-----------|
| | DATOS | SUBIDA(KB) | BAJADA(KB) | TOTAL(KB) |
| 10 | 27 | 0,439 | 5,64 | 6,07 |
| 20 | 55 | 0,719 | 14,66 | 15,36 |
| 30 | 76 | 0,999 | 23,67 | 24,65 |
| 40 | 109 | 1,25 | 41,27 | 42,52 |
| 50 | 136 | 1,52 | 63,17 | 64,69 |
| 60 | 172 | 1,8 | 90,84 | 92,64 |
| TOTAL | | 6,727 | 239,25 | 245,93 |

INTERVALO DE 1 HORA

| HORA | HISTORICOS | | | |
|--------------|------------|------------|------------|-----------|
| | DATOS | SUBIDA(KB) | BAJADA(KB) | TOTAL(KB) |
| 1 | 172 | 0,28 | 27,68 | 27,95 |
| 2 | 316 | 0,56 | 78,05 | 78,6 |
| 3 | 458 | 0,84 | 150,8 | 151,62 |
| 4 | 581 | 1,09 | 242,92 | 244,92 |
| 5 | 695 | 1,37 | 352,83 | 354,2 |
| 6 | 799 | 1,64 | 427,15 | 428,49 |
| 7 | 913 | 1,91 | 537,22 | 539,13 |
| 8 | 1014 | 2,19 | 663,7 | 665,89 |
| 9 | 1107 | 2,46 | 808,48 | 810,94 |
| TOTAL | | 12,34 | 3288,83 | 3301,74 |

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

CERTIFICACIÓN

Se certifica que el presente proyecto de investigación fue desarrollado por los señores: Evelyn Fernanda Endara Crespata y Víctor Alfonso Vargas Ortiz, bajo nuestra supervisión:

Ing. Rivas David
DIRECTOR DEL PROYECTO

Ing. Erazo Mayra
CODIRECTORA DEL PROYECTO

Ing. Silva Franklin
DIRECTOR DE LA CARRERA

Dr. Vaca Rodrigo
SECRETARIO ACADÉMICO