



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**“DESARROLLO DE UN BOTÓN DE
PÁNICO INTELIGENTE CON
TECNOLOGÍA IOT”**

**TESIS QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

PRESENTA:

IOVANNY OLGUÍN ÁVILA

DIRECTORES DE TESIS:

DRA. BÁRBARA EMMA SÁNCHEZ RINZA

DR. JOSÉ LUIS CARBALLIDO CARRANZA

PUEBLA, PUEBLA

JULIO 2018

DEDICATORIA

A mi madre María del Pilar Ávila Ávila que en paz descanse, por apoyarme y estar siempre conmigo.

A mi tía Piedad Ávila Ávila por el apoyo brindado para la realización de mis estudios.

A mi padre Fernando Olgúin Azpeitia por sus enseñanzas claves y oportunas en las etapas importantes de mi vida.

AGRADECIMIENTOS

A mi asesora, Dra. Barbara Emma Sánchez Rinza, por la amistad brindada y su ayuda en el desarrollo de este proyecto.

A mi comité tutorial, Dra. Claudia Zepeda Cortés, Dr. Ivo Humberto Pineda Torres, Dr. Mario Rossainz López y Dr. José Luís Carballido Carranza, por sus comentarios.

A mis compañeros Stephanie Vázquez González, Cecilia Reyes Peña, Daniel Sánchez Ruiz, Jesús García Ramírez, Rodolfo Ricaño Zuñiga, Alfredo Neri Rojas y Rodolfo Alberto Martínez Torres, por su amistad y por el apoyo brindado en la maestría.

A la Benemérita Universidad Autónoma de Puebla por darme el honor de pertenecer a la institución y brindarme la oportunidad de crecer profesionalmente.

A mi novia Ana Montserrat Jiménez Ramírez por apoyarme en todo momento.

A mi familia por su apoyo y confianza.

Al consejo nacional de ciencia y tecnología (CONACYT) por el apoyo económico que me permitió concluir mis estudios.

ÍNDICE DE CONTENIDO

| | |
|---|-----------|
| INTRODUCCIÓN: | 12 |
| RESUMEN..... | 12 |
| ANTECEDENTES DEL PROYECTO..... | 12 |
| OBJETIVOS GENERALES Y ESPECÍFICOS DEL PROYECTO..... | 12 |
| OBJETIVO GENERAL..... | 12 |
| OBJETIVOS ESPECÍFICOS..... | 12 |
| METODOLOGÍA..... | 13 |
| CAPITULO 1: | 14 |
| MARCO TEÓRICO..... | 14 |
| 1.1 INTERNET DE LAS COSAS..... | 14 |
| 1.1.1 DEFINICIÓN..... | 14 |
| 1.1.2 CARACTERÍSTICAS..... | 16 |
| 1.1.3 FASES..... | 18 |
| 1.1.4 SEGURIDAD..... | 19 |
| 1.1.5 EVOLUCIÓN..... | 21 |
| 1.2 SISTEMAS EMBEBIDOS..... | 21 |
| 1.2.1 DEFINICIÓN..... | 22 |
| 1.2.2 CARACTERÍSTICAS..... | 23 |
| 1.3 WEB SERVICES..... | 24 |
| 1.3.1 DEFINICIÓN..... | 24 |
| 1.3.2 RESTFUL..... | 25 |
| 1.4 TECNOLOGÍAS Y PROTOCOLOS DE RED..... | 27 |
| 1.4.1 IPV6..... | 27 |
| 1.4.2 REDES DE TELEFONÍA MÓVIL..... | 29 |
| 1.5 ESTADO DEL ARTE..... | 33 |

| | |
|---|-----------|
| CAPITULO 2..... | 35 |
| ANÁLISIS Y DISEÑO..... | 35 |
| 2.1 ANÁLISIS DE LOS REQUERIMIENTOS..... | 35 |
| 2.1.1 REQUERIMIENTOS FUNCIONALES..... | 35 |
| 2.1.2 REQUERIMIENTOS NO FUNCIONALES..... | 40 |
| 2.2 ELEMENTOS DEL SISTEMA..... | 41 |
| 2.3 ARDUINO PRO MICRO..... | 42 |
| 2.3.1 CARACTERÍSTICAS..... | 43 |
| 2.4 SIM800L..... | 46 |
| 2.4.1 CARACTERÍSTICAS..... | 47 |
| 2.5 WEB SERVICE..... | 48 |
| 2.6 MYSQL SERVER..... | 50 |
| 2.7 MODELADO DE CASOS DE USO..... | 51 |
| 2.7.1 DIAGRAMA DE CASOS DE USO..... | 52 |
| 2.7.2 ESPECIFICACIONES DE CASOS DE USO..... | 53 |
| 2.8 DIAGRAMA DE CLASES..... | 59 |
| 2.9 DIAGRAMA DE BLOQUES..... | 69 |
| 2.10 DISEÑO DE LA BASE DE DATOS..... | 70 |
| 2.10.1 DICCIONARIO DE DATOS..... | 70 |
| 2.10.2 MODELO CONCEPTUAL..... | 75 |
| 2.10.3 MODELO RELACIONAL..... | 76 |
| 2.11 MODELO DEL SISTEMA..... | 77 |
| 2.11.1 ENTRADA..... | 77 |
| 2.11.2 PROCESO..... | 77 |
| 2.11.3 SALIDA..... | 78 |
| CAPITULO 3:..... | 79 |
| DESARROLLO E IMPLEMENTACIÓN..... | 79 |
| 3.1 DISPOSITIVO FÍSICO..... | 79 |

| | |
|--|------------|
| 3.1.1 MATERIALES..... | 80 |
| 3.1.2 CONSTRUCCIÓN..... | 81 |
| 3.1.3 SOFTWARE..... | 81 |
| 3.1.4 CONFIGURACIÓN DE LAS HERRAMIENTAS..... | 81 |
| 3.1.6 CIRCUITO FINAL..... | 86 |
| 3.2 WEB SERVICE..... | 87 |
| 3.2.1 SOFTWARE..... | 87 |
| 3.2.2 CONFIGURACIÓN DE LAS HERRAMIENTAS..... | 87 |
| 3.2.3 FRONT-END..... | 95 |
| 3.3 BASE DE DATOS..... | 99 |
| 3.3.1 ESQUEMA FÍSICO..... | 99 |
| 3.4 COMUNICACIÓN..... | 100 |
| 3.4.1 INTRODUCCIÓN..... | 100 |
| 3.4.2 HTTP POST..... | 100 |
| 3.4.3 DIAGRAMAS DE FUNCIONAMIENTO..... | 105 |
| CAPITULO 4..... | 107 |
| CONCLUSIONES..... | 107 |
| REFERENCIAS..... | 108 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 2.1.1: RF01 – Botón funcional..... | 36 |
| Tabla 2.1.2: RF02 – Interpretar respuestas..... | 36 |
| Tabla 2.1.3: RF03 – Notificaciones locales..... | 36 |
| Tabla 2.1.4: RF04 – Inicio de sesión..... | 36 |
| Tabla 2.1.5: RF05 – Permitir al usuario crear, modificar y eliminar sus contactos..... | 37 |
| Tabla 2.1.6: RF06 – Permitir al usuario administrar, activar y desactivar alertas..... | 37 |
| Tabla 2.1.7: RF07 – Permitir al usuario administrar sus dispositivos..... | 37 |
| Tabla 2.1.8: RF08 – Permitir a un administrador crear, editar y eliminar usuarios..... | 37 |
| Tabla 2.1.9: RF09 – Presentar lista de usuarios..... | 38 |
| Tabla 2.1.10: RF10 – Presentar lista de contactos..... | 38 |
| Tabla 2.1.11: RF11 – Presentar lista de dispositivos..... | 38 |
| Tabla 2.1.12: RF12 – Presentar lista de alertas..... | 38 |
| Tabla 2.1.13: RF13 – Mapa de ubicación..... | 39 |
| Tabla 2.1.14: RF14 – El dispositivo físico puede enviar mensajes de texto..... | 39 |
| Tabla 2.1.15: RF15 – El web server puede enviar mensajes de texto..... | 39 |
| Tabla 2.1.16: RF16 – El dispositivo físico puede realizar llamadas de voz..... | 39 |
| Tabla 2.1.17: RF17 – El web server puede realizar llamadas de voz..... | 40 |
| Tabla 2.1.18: RF18 – El web server notificar por internet..... | 40 |
| Tabla 2.1.19: RNF01 – Dispositivo sencillo..... | 40 |
| Tabla 2.1.20: RNF02 – Interfaz sencilla y amigable..... | 41 |
| Tabla 2.1.21: RNF03 – Diseño responsivo..... | 41 |
| Tabla 2.2.1: Características del Arduino Pro Micro..... | 45 |
| Tabla 2.2.3: Características del Módulo SIM800L..... | 48 |
| Tabla 2.7.1: Descripción del caso de uso Administrar alertas..... | 53 |
| Tabla 2.7.2: Descripción del caso de uso Administrar usuarios..... | 54 |
| Tabla 2.7.3: Descripción del caso de uso Login / Logout..... | 55 |
| Tabla 2.7.4: Descripción del caso de uso Administrar contactos..... | 56 |
| Tabla 2.7.5: Descripción del caso de uso Administrar alertas..... | 57 |

| | |
|--|----|
| Tabla 2.7.6: Descripción del caso de uso Push button..... | 58 |
| Tabla 2.8.1: Descripción de la tabla index.php..... | 61 |
| Tabla 2.8.2: Descripción de la tabla registro.php..... | 61 |
| Tabla 2.8.3: Descripción de la tabla login.php..... | 61 |
| Tabla 2.8.4: Descripción de la tabla usuario..... | 62 |
| Tabla 2.8.5: Descripción de la tabla dispositivo..... | 63 |
| Tabla 2.8.6: Descripción de la tabla alerta..... | 64 |
| Tabla 2.8.7: Descripción de la tabla contacto..... | 65 |
| Tabla 2.8.8: Descripción de la tabla comando..... | 66 |
| Tabla 2.8.9: Descripción de la tabla sms..... | 66 |
| Tabla 2.8.10: Descripción de la tabla llamadacasa..... | 67 |
| Tabla 2.8.11: Descripción de la tabla llamadacelular..... | 67 |
| Tabla 2.8.12: Descripción de la tabla facebook..... | 67 |
| Tabla 2.8.13: Descripción de la tabla gps..... | 68 |
| Tabla 2.8.14: Descripción de la tabla email..... | 68 |
| Tabla 2.10.1: Tabla usuario..... | 71 |
| Tabla 2.10.2: Tabla contacto..... | 72 |
| Tabla 2.10.3: Tabla dispositivo..... | 73 |
| Tabla 2.10.4: Tabla alerta..... | 73 |
| Tabla 2.10.5: tabla gps..... | 74 |
| Tabla 3.1.1: Materiales necesarios para la construcción del dispositivo..... | 80 |
| Tabla 3.1.2: Software necesario para el desarrollo del dispositivo físico..... | 81 |
| Tabla 3.1.3: Fritzing – Descripción de los paquetes instalados..... | 83 |
| Tabla 3.1.4: Arduino – Descripción de los paquetes instalados..... | 86 |
| Tabla 3.2.1: Herramientas utilizadas en el desarrollo del web service..... | 87 |
| Tabla 3.2.1: Servidor web Apache - Descripción de los paquetes instalados..... | 89 |
| Tabla 3.2.2: PHP - Descripción de los paquetes instalados..... | 90 |
| Tabla 3.2.3: MySQL - Descripción de los paquetes instalados..... | 92 |
| Tabla 3.2.4: phpMyAdmin - Descripción de los paquetes instalados..... | 93 |

Tabla 3.2.5: Sublime-text - Descripción de los paquetes instalados.....95
Tabla 3.4.1: HTTP POST – Secuencia de comandos AT.....102

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 2.2.1: Diagrama del sistema..... | 42 |
| Figura 2.2.2: Arduino Pro Micro..... | 43 |
| Figura 2.2.3: Diagrama de pines del Arduino Pro Micro..... | 44 |
| Figura 2.2.3: Módulo SIM800L..... | 46 |
| Figura 2.2.4: Configuración de pines del Módulo SIM800L..... | 47 |
| Figura 2.2.5: Servidores web más utilizados Diciembre 2017..... | 49 |
| Figura 2.7.1: Diagrama de casos de uso..... | 52 |
| Figura 2.8.1: Diagrama de clases..... | 60 |
| Figura 2.9.1: Diagrama de bloques del sistema..... | 69 |
| Figura 2.10.2: Modelo conceptual de la base de datos..... | 75 |
| Figura 2.10.2: Diagrama de la base de datos..... | 76 |
| Figura 2.11.1: Modelo del sistema..... | 77 |
| Figura 3.1.1: Esquema de conexión del dispositivo físico..... | 81 |
| Figura 3.1.2: Proceso de instalación de Fritzing..... | 82 |
| Figura 3.1.3: Ventana principal de Fritzing..... | 83 |
| Figura 3.1.4: Instalación del IDE Arduino..... | 84 |
| Figura 3.1.5: Ventana principal del IDE Arduino..... | 85 |
| Figura 3.1.6: Circuito Final..... | 86 |
| Figura 3.2.1: Apache Web Server – Proceso de instalación..... | 88 |
| Figura 3.2.2: Apache Web Server – Pagina de bienvenida..... | 89 |
| Figura 3.2.3: PHP – Proceso de instalación..... | 90 |
| Figura 3.2.4: MySQL server – Proceso de instalación..... | 91 |
| Figura 3.2.5: MySQL (MariaDB) – Sesión activa..... | 91 |
| Figura 3.2.6: phpMyAdmin – Proceso de instalación..... | 92 |
| Figura 3.2.7: phpMyAdmin – Interfaz principal..... | 93 |
| Figura 3.2.8: Sublime-text – Proceso de instalación..... | 94 |
| Figura 3.2.9: Sublime-text – Interfaz principal..... | 94 |
| Figura 3.2.10: Web service – index..... | 95 |

| | |
|--|-----|
| Figura 3.2.11: Web service – registro..... | 96 |
| Figura 3.2.12: Web service – login de usuario..... | 96 |
| Figura 3.2.13: Web service – Administración de contactos..... | 97 |
| Figura 3.2.14: Web service – Administración de dispositivos..... | 97 |
| Figura 3.2.15: Web service – Administración de alertas..... | 98 |
| Figura 3.2.16: Web service – Página de localización GPS..... | 98 |
| Figura 3.3.1: Esquema físico de la base de datos..... | 99 |
| Figura 3.4.1: Diagrama del algoritmo que captura los clicks del usuario..... | 101 |
| Figura 3.4.2: Diagrama de flujo de la petición HTTP..... | 103 |
| Figura 3.4.3: Diagrama de flujo del envío individual de un comando AT..... | 103 |
| Figura 3.4.4: Diagrama de secuencia de la notificación por MCU..... | 105 |
| Figura 3.4.5: Diagrama de secuencia de la notificación por Web Service..... | 105 |
| Figura 3.4.6: Diagrama de Localización por GPS..... | 106 |

INTRODUCCIÓN:

RESUMEN

En este trabajo se diseña y se desarrolla un botón de pánico inteligente utilizando una placa Arduino y un módulo GSM del lado del hardware y una página web que servirá para cargar la configuración y realizar las acciones especificadas por el usuario.

ANTECEDENTES DEL PROYECTO.

El presente trabajo presenta un diseño y una metodología para el desarrollo de un botón inteligente configurable vía web, usando materiales de bajo costo y de fácil adquisición, esto en respuesta a la creciente ola de inseguridad que azota a la ciudadanía en México. Anteriormente como requisito de la materia Sistemas programables, en el Instituto Tecnológico de Pachuca, se diseñó una Alarma con sensor de movimiento con notificaciones en redes sociales denominada ASIS (Alarma Simple vía Internet y SMS). Donde se utilizó una placa Arduino Yun y sensores PIR.

OBJETIVOS GENERALES Y ESPECÍFICOS DEL PROYECTO.

OBJETIVO GENERAL

Desarrollar un sistema de alertas mediante un botón de pánico basado en el Internet de las cosas, configurable por medio de una plataforma web.

OBJETIVOS ESPECÍFICOS

- Diseñar y desarrollar un dispositivo físico que integre un botón de pánico el cual haga uso de las redes GSM y GPS para comunicarse con un web server.
- Diseñar y desarrollar un web service que se comunique con el dispositivo físico.
- Integrar el dispositivo físico y el web server.
- implementar en el web server las alertas que se tienen contempladas.

METODOLOGÍA

El desarrollo de este trabajo está dividido en bloques de un bimestre cada uno siguiendo la secuencia descrita a continuación:

1. En el primer bloque se hizo una revisión bibliográfica con el fin de contar con las fuentes de información necesarias para tener información de calidad para desarrollar el proyecto de forma correcta y se realizará un análisis de requerimientos de software y de hardware.
2. En el segundo bloque se hizo una comparativa del hardware y software disponible buscando la combinación exacta de estos, que satisfaga las exigencias del proyecto.
3. En el tercer bloque se dedicó a la programación del dispositivo “botón”.
4. En el cuarto bloque se creó el sitio web que servirá para configurar las funciones principales.
5. En el quinto bloque se hicieron las pruebas necesarias, con el fin de pulir detalles y encontrar fallas en el diseño y funcionamiento.
6. Finalmente en el sexto bloque se redactó el documento que describirá a detalle el diseño y desarrollo del trabajo propuesto.

CAPITULO 1:

MARCO TEÓRICO

En este capítulo se presentan de manera breve, una explicación sobre Internet de las cosas, el uso de microcontroladores o dispositivos con recursos reducidos, la comunicación inalámbrica entre los principales actores del Internet de las cosas así como las tecnologías y los protocolos de red involucrados en el desarrollo de sistemas con uso de esta tecnología, finalmente se aborda el estado del arte.

1.1 INTERNET DE LAS COSAS

1.1.1 DEFINICIÓN

El término Internet de las Cosas (IoT, por sus siglas en inglés) no está bien definido, se ha utilizado y mal utilizado como palabra de moda en la investigación científica, así como estrategias de marketing y ventas. Hasta hoy sigue siendo difícil llegar a una definición clara del IoT. Una definición ha sido formulada recientemente en the Strategic Research Agenda of the Cluster of European Research Projects on the Internet of Things (CERP-IoT 2009):

IoT es una parte integrada de Future Internet y podría definirse como una infraestructura de red global dinámica con capacidades de auto-configuración basadas en protocolos de comunicación estándar e interoperables donde las "cosas" físicas y virtuales tienen identidades, atributos físicos, comportamientos virtuales, utilizan interfaces inteligentes, y se integran perfectamente en la red de información. En el IoT, se espera que las "cosas" se conviertan en participantes activos en los procesos empresariales, informativos y sociales, donde se les permite interactuar y comunicarse entre sí y con el medio ambiente intercambiando datos e información "percibidos" sobre el medio ambiente, reaccionando autónomamente a los acontecimientos del "mundo real/físico" e influenciándolo mediante la ejecución de procesos que activan acciones y crean servicios con o sin intervención humana directa. Interfaces en forma de servicios facilitan las interacciones con estas "cosas

inteligentes" a través de Internet, consultar y cambiar su estado y cualquier información asociada con ellos, teniendo en cuenta las cuestiones de seguridad y privacidad" [1].

El IoT tiene como objetivo cerrar la brecha entre el mundo físico y su representación dentro del mundo digital. La idea es integrar el estado de las "cosas" que forman el mundo en aplicaciones de software, haciendo que se beneficien de la información de contexto del mundo. La tecnología IoT se está desarrollando rápidamente y muchas aplicaciones se han construido recientemente. Aunque todavía hay muchos desafíos en los campos de redes inalámbricas de sensores (WSN), identificación por radiofrecuencia (RFID), procesamiento de datos, seguridad, etc.

El IoT es un concepto en el que el mundo virtual de la tecnología de la información se integra perfectamente con el mundo real de las cosas. El mundo real se vuelve más accesible a través de computadoras y dispositivos en red tanto en los negocios como escenarios cotidianos. Con acceso a información de grano fino, la gerencia puede moverse libremente de macro a micro niveles y podrá medir, planear y actuar de acuerdo al escenario planteado. Sin embargo, el IoT es más que una herramienta de negocio para gestionar los procesos de negocio de manera más eficiente, también permitirá una forma de vida más conveniente [1].

El IoT se compone de dispositivos físicos de red, junto con dispositivos integrados, sensores, software y conectividad de red de los componentes del sistema para el intercambio de datos. Se trata de conectar "cosas" como teléfonos inteligentes, televisiones, refrigeradores o sensores a Internet para vincular los dispositivos de forma inteligente y así permitir nuevas formas de comunicación entre ellos, sistemas computacionales y personas. El IoT integra dominios tradicionales incluyendo sistemas de control y automatización, sistemas embebidos, redes de sensores inalámbricos para la comunicación de dispositivo a dispositivo (D2D) a través de Internet [2].

1.1.2 CARACTERÍSTICAS

Desde un punto de vista técnico, IoT no es el resultado de una sola tecnología nueva; en cambio, varios desarrollos técnicos complementarios proporcionan capacidades que, juntas ayudan a integrar el mundo virtual y el físico [3]. Estas capacidades incluyen:

- Comunicación y cooperación

Los objetos tienen la capacidad de conectarse con los recursos de Internet o incluso entre sí, para hacer uso de datos y servicios además de actualizar su estado. Las tecnologías inalámbricas como GSM y UMTS, Wi-Fi, Bluetooth, ZigBee y otros estándares de redes inalámbricas actualmente en desarrollo, particularmente aquellos relacionados con las Redes de Área Personal Inalámbrica (WPANs), son de importancia primordial aquí.

- Direccionamiento

Dentro de una IoT, los objetos pueden ser localizados y dirigidos a través de descubrimiento, búsqueda o servicios de nombre, y por lo tanto, de forma remota interrogado o configurado.

- Identificación

Los objetos son identificables de forma única. RFID(Radio Frequency Identification), NFC (Near Field Communication) y códigos de barras ópticamente legibles son ejemplos de tecnologías con las que incluso se pueden identificar objetos pasivos que no tienen recursos energéticos integrados (con la ayuda de un "mediador" como un lector RFID o móvil teléfono). La identificación permite que los objetos se vinculen a la información asociada con el objeto particular y que puedan ser recuperados desde un servidor, siempre que el mediador esté conectado a la red

- Detección

Los objetos recogen información sobre su entorno con sensores, lo registran, lo reenvían o reaccionan directamente a él.

- Actuación

Los objetos contienen actuadores para manipular su entorno (por ejemplo, mediante la conversión de señales eléctricas en movimiento mecánico). Tales actuadores pueden usarse para controlar remotamente procesos del mundo real a través de Internet.

- Procesamiento de información incorporado

Los objetos inteligentes cuentan con un procesador o micro controlador, además de capacidad de almacenamiento. Estos recursos pueden usarse, por ejemplo, para procesar e interpretar la información del sensor, o para dar a los productos una "memoria" de cómo se han utilizado.

- Localización

Las cosas inteligentes son conscientes de su ubicación física, o se pueden localizar. GPS o la red de telefonía móvil son tecnologías adecuadas para lograr esto, así como medidas de tiempo de vuelo por ultrasonido, UWB (banda ultra ancha), estaciones base WLAN vecinas, lectores RFID con coordenadas conocidas y tecnologías ópticas.

- Interfaces de usuario

Los objetos inteligentes pueden comunicarse con las personas de manera apropiada (directa o indirectamente, por ejemplo, a través de un teléfono inteligente). Los paradigmas de interacción innovadores son relevantes aquí, tales como interfaces de usuario tangibles, pantallas flexibles basadas en polímeros y métodos de reconocimiento de voz, imagen o gestos.

1.1.3 FASES

Fase 1: Recopilación de datos, adquisición, percepción

Como en las computadoras tradicionales existen los dispositivos de entrada como lo es el mouse y el teclado, de igual forma en la tecnología del IoT existe una etapa donde se deben ingresar los datos, ya sea en una aplicación de medicina o de seguimiento vehicular, el primer paso es recopilar o adquirir los datos de los dispositivos. En base de las características de éstos, se utilizan diferentes tipos de recopiladores de datos. Pueden ser estáticos como los sensores corporales o etiquetas RFID, y dinámicos, sensores y chips.

Fase 2: Almacenamiento

Después de la fase 1 es necesario almacenar la información de forma temporal para posteriormente ser procesada en la nube. Los dispositivos o “cosas” normalmente traen incorporada una memoria de almacenamiento para esta tarea.

Fase 3: Tratamiento inteligente de los datos

En el IoT, los datos almacenados en la nube son utilizados para proveer servicios inteligentes en tiempo real, analizar y responder a las consultas además de brindar la posibilidad de controlar el dispositivo remotamente. Gracias a este manejo de los datos, en esta tecnología no importa si lo que se conecta son los autos, los refrigeradores, las luces en nuestro hogar, e incluso elementos insospechados, tales como nuestros libros, la ropa de cama o las plantas ornamentales

Fase 4: Transmisión de datos

La transmisión de información ocurre en todas las fases:

- De los sensores, etiquetas RFID, o chips hacia otros dispositivos o la nube
- De la nube a los procesadores
- De los procesadores a los controladores, dispositivos y usuarios finales.

Fase 5: Entrega

La entrega de los datos procesados a los dispositivos debe ser correcta, íntegra y en tiempo, Este resultado siempre debe alcanzarse en el IoT.

1.1.4 SEGURIDAD

La seguridad es una propiedad del sistema; el sistema puede ser tan seguro como su componente más vulnerable. Aún no ha surgido una visión unificada de las arquitecturas de seguridad en el Internet de las cosas. Las características de seguridad son provistas por los componentes de las diferentes capas de operación del IoT:

- dispositivos
- redes físicas
- redes Inalámbricas
- middleware

En los dispositivos, algunos pero no todos los procesadores de bajo consumo de energía implementan funciones como arranque seguro (secure boot), cifrado AES-NI acelerado por la propia CPU, HDCP para protección de contenidos de audio y vídeo y ECC (control de corrección de errores). En cuanto a las redes, una comunicación segura requiere encriptación y mecanismos de autorización, así como un método de enrutamiento seguro en una red. Los esquemas de cifrado tradicionales, como AES [21], RSA [22], etc., proporcionan un alto nivel de seguridad, como se ha demostrado en los sistemas informáticos de uso general, pero son bastante exigentes en recursos computacionales y de memoria [23].

En el caso de los dispositivos que se conectan de forma inalámbrica a la nube o a un web server, la seguridad también se implementa en los protocolos de comunicación y en la programación del web server.

En el IoT Developer Survey elaborado y publicado por Eclipse IoT Working Group, AGILE IoT, IEEE, y Open Mobile Alliance [32] se listan las tecnologías de seguridad más utilizadas en el último año, las cuales se muestran a continuación:

- Seguridad en la comunicación
- Cifrado de información
- Json Web token
- Infraestructura de llave publica
- Secure Boot
- Blockchain
- Seguridad en Hardware

En la seguridad de la comunicación existen diferentes protocolos que cubren este aspecto, y que han ido evolucionando de acuerdo a las exigencias de seguridad de la comunicación en Internet. Estos protocolos se utilizan tanto en el Internet de uso tradicional como en el IoT, en [32] muestra que los protocolos más utilizados debido a su confiabilidad en los últimos años son los siguientes:

- Secure Sockets Layer (SSL)
- Transport Layer Security (TLS)
- Datagram Transport Layer Security (DTLS)
- Hypertext Transfer Protocol Secure (HTTPS)

La comunicación GSM / GPRS, también proporciona controles de seguridad. El operador del sistema quiere asegurarse de que el suscriptor que solicita el servicio sea válido (autenticación). El suscriptor, por otro lado, quiere tener acceso a los servicios sin comprometer la privacidad (confidencialidad de los datos del suscriptor). Los controles de seguridad del GSM logrados usando los cuatro mecanismos principales [24], es decir, cada abonado se identifica usando un mecanismo de seguridad criptográfico; la información de seguridad de los suscriptores se almacena en una plataforma informática segura llamada módulo SIM (módulo de identidad del suscriptor) o tarjeta inteligente; el operador GSM mantiene el secreto de los algoritmos criptográficos y las claves para autenticar al suscriptor

y proporcionar privacidad de voz; los algoritmos se almacenan en el SIM y el Centro de autenticación (AUC); las claves criptográficas no se comparten con otras administraciones GSM [25].

1.1.5 EVOLUCIÓN

El IoT está integrado por diferentes categorías, las denominadas "cosas" pueden etiquetarse, detectarse e incrustarse. Etiquetar Cosas proporciona una identificación perfecta y rentable de artículos, permitiendo un seguimiento en tiempo real, alta trazabilidad e interconexión con bases de datos. Los sensores nos permiten medir y detectar cambios en nuestro entorno, y finalmente las cosas embebidas proporcionan información sobre un objeto en particular.

En la última década, estas tecnologías se han extendido bastante, su uso va desde dispositivos de uso común en la vida cotidiana, hasta en los más complejos como lo es ámbito médico, militar y espacial, su desarrollo abarca varios dominios, como en la identificación por radiofrecuencia (RFID), comunicación máquina a máquina (M2M) y comunicación tipo máquina (MTC), Wireless sensor and actuator networks (WSAN) y Web-of-things(WoT). El campo del IoT es relativamente joven y sigue dominando entre las tecnologías más prometedoras. Se espera que la adopción de las diversas tecnologías de IoT se expanda rápidamente en los próximos años, y se reflejará en el número de cosas conectadas, ingresos esperados y tasas de crecimiento anual [4].

El IoT marca la evolución actual de la tecnología en el sector de las telecomunicaciones que está permitiendo una nueva ola de servicios que mejoran la vida en toda la economía. El IOT tiene el potencial de aportar beneficios sociales y económicos sustanciales a los gobiernos, los ciudadanos, los usuarios finales y las empresas a través de la creación de nuevos puestos de trabajo, mejoras en la prestación de servicios y personalización y un uso más eficiente de los escasos recursos. El IOT ofrece una oportunidad para que las industrias se conecten en un grado que nunca antes había sido posible [5].

1.2 SISTEMAS EMBEBIDOS

En el IOT existen una gran cantidad de componentes diferentes, como sensores, actuadores y dispositivos. Los que reciben la mayor carga en procesamiento de información trabajan con procesadores de 32 y 64 bits, entre estos se encuentran las computadoras y los Smartphones. Mientras que en el punto de partida de esta tecnología, las "cosas", que son

encargadas de interactuar con el mundo físico se encuentran dispositivos equipados solo con microcontroladores de 8 bits. Tomando en cuenta lo anterior, Para que un sistema IOT se ejecute debidamente el software utilizado debe ser compatible con todos los rangos de dispositivos, incluidos los de baja potencia, adaptándose tanto a sus funcionalidades como a sus limitaciones. Estos microcontroladores generalmente forman sistemas embebidos, en esta sección se presenta su definición y sus características principales.

1.2.1 DEFINICIÓN

Se conoce como sistema embebido a un circuito electrónico que está diseñado para cumplir una labor específica en un producto. La inteligencia artificial, secuencias y algoritmos de un sistema embebido, están residentes en la memoria de una pequeña computadora denominada microcontrolador. A diferencia de los sistemas computacionales de oficina y Laptops, estos sistemas solucionan un problema específico y están dispersos en todos los ambientes posibles de la vida cotidiana [33].

El término "sistema embebido" hace referencia a todo circuito electrónico digital capaz de realizar operaciones de computación, generalmente en tiempo real, que sirven para cumplir una tarea específica de un producto.

Los sistemas embebidos no son equivalentes a los sistemas de cómputo usados en las Laptops o en computadoras de escritorio que se venden en las tiendas tecnológicas, ya que los sistemas embebidos suelen tener recursos limitados y aplicaciones específicas que los hacen sumamente útiles en múltiples ambientes, como en el campo automotriz (sistemas de inyección de gasolina, alarmas contra robos, control de climatización, sistema de frenado ABS) o en teléfonos móviles, ipad, reproductores, Blu-ray, refrigeradores, alarmas de casas, lavadoras, cámaras fotográficas, instrumentación industrial, equipos médicos, Set Top Boxes, entre otros [34].

1.2.2 CARACTERÍSTICAS

La arquitectura de un sistema embebido contiene un microprocesador o microcontrolador dedicado capaz de ejecutar instrucciones a una determinada velocidad, controlada por una señal de reloj. De acuerdo con la arquitectura del microprocesador del sistema embebido, los recursos internos (periféricos) y la máxima frecuencia de operación, se define la potencia de procesamiento. Normalmente, este parámetro se mide en unidades de MIPS (millones de instrucciones por segundo).

El programa que ejecuta un sistema embebido es por lo general elaborado en lenguajes como el ensamblador, ANSI C, C++ o Basic. Algunos sistemas embebidos tienen la capacidad de ejecutar sistemas operativos limitados, conocidos como RTOS (Real Time Operating Systems), que permiten que el procesador ejecute diversas tareas, asignándoles una prioridad y un orden de ejecución. Algunos de estos sistemas operativos son el MQX de Freescale, μ C/OS-II, FreeRTOS y μ Linux, entre los más conocidos.

Una de las características actuales de todo sistema embebido, y muy importante, es el bajo consumo de energía. Como muchas de sus aplicaciones involucran dispositivos que utilizan baterías, los modos de bajo consumo de corriente (modo sleep) son sumamente importantes para que puedan ser utilizados en este tipo de aplicaciones [31].

A continuación, se presentan en resumen las características más importantes:

- **Completo:** Todos disponen de los bloques esenciales: Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.
- **Función específica:** Un sistema embebido ejecuta un programa de forma repetida al contrario de una computadora o un Smartphone donde se ejecuta una amplia variedad de programas.
- **Limitado:** Deben ser poco costosos, de tamaño reducido y un consumo mínimo de energía.

- **Rapidez:** A veces deben realizar cálculos en tiempo real sin ningún retraso, En caso de no ser necesario trabajar en tiempo real, estos cálculos deben realizarse de forma precisa y acelerada para garantizar la entrega de los resultados dentro de un tiempo límite. La mayoría de los microcontroladores pueden ejecutar entre 1 y 5 millones de instrucciones por segundo.
- **Programación:** Se suele programar en lenguajes de bajo nivel tipo ensamblador o C. donde el usuario debe desarrollar controladores para los diferentes periféricos.
- **Aprendizaje:** Dominar la programación de microcontroladores requiere un periodo de aprendizaje relativamente largo.
- **Flexible:** Todas las entradas de un microcontrolador son de propósito general. Esta flexibilidad obliga al programador a desarrollar su propia electrónica.
- **Bajo consumo energético:** Su diseño es simple, por lo que su consumo es inferior a 50 mA. se pueden lograr aplicaciones con consumos por debajo de los 5 mA.

1.3 WEB SERVICES

1.3.1 DEFINICIÓN

Un servicio web es una tecnología integrada por protocolos y estándares que permite intercomunicar aplicaciones o sistemas independientemente de la plataforma en cual se ejecuten o el lenguaje de programación en el cual estén hechos ofreciendo un conjunto de operaciones disponibles en red. Un servicio web permite que los dispositivos se comuniquen entre sí, compartiendo información de una forma simple y en un formato estándar facilitando agregar más dispositivos a la red de operación.

Hoy en día Internet ofrece una amplia variedad de servicios con diversas funciones. Estos servicios se clasifican como basados en: WSDL, REST, XML-RPC, Atom o JSON-RPC. Siendo las dos primeras opciones las más populares para ofrecer plataformas, donde la reutilización y composición de servicios reducen costos y dan valor agregado [9].

1.3.2 RESTFUL

El término Representational State Transfer (REST) identifica un estilo de arquitectura orientada al diseño de software basado en la red. REST es una técnica o familia de arquitecturas software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding [28], uno de los principales autores de la especificación del protocolo HTTP y luego el término se extendió para describir un estilo de construcción de servicios web, basado en los principios establecidos por Fielding [9], éste se ha denominado RESTful [27].

Los servicios web basados en la arquitectura REST se conocen como RESTful Web Services. Estos servicios web usan métodos HTTP para implementar el concepto de arquitectura REST. Un servicio web RESTful normalmente define un URI (identificador uniforme de recursos), que es un servicio que proporciona una representación de recursos como JSON y un conjunto de métodos HTTP. El estado del recurso se representa en texto plano, XML o en HTML que contienen enlaces y formularios, que permiten saber y modificar el estado del recurso. Un servicio web RESTful contiene lo siguiente:

- **URI del recurso:** Por ejemplo: <http://ejemplo.com/producto/libro/1>
- **El tipo de la representación de dicho recurso:** Por ejemplo, JSON, XML y TXT.
- **Operaciones soportadas:** GET, PUT, POST, DELETE, PURGE, entre otros.
- **Hipervínculos:** la respuesta puede incluir enlaces hacia otras acciones sobre los recursos.

Una de las características clave de un servicio web RESTful es el uso explícito de los métodos HTTP de una manera que sigue el protocolo definido por RFC 2616. HTTP GET, por ejemplo, se define como un método de producción de datos que está destinado a ser utilizado por una aplicación cliente para recuperar un recurso, recuperar datos de un servidor web o ejecutar una consulta con la expectativa de que el servidor web buscará y responderá con un conjunto de recursos coincidentes [29].

En la arquitectura REST los métodos HTTP se usan explícitamente y de una manera consistente con la definición del protocolo. Este principio básico de diseño REST establece un mapeo uno a uno entre operaciones de creación, lectura, actualización y eliminación (CRUD) y métodos HTTP. De acuerdo con este mapeo:

- **POST**: para crear un recurso en el servidor.
- **GET**: para recuperar un recurso, use GET.
- **PUT**: para cambiar el estado de un recurso o actualizarlo.
- **DELETE**: para eliminar o eliminar un recurso.

El comportamiento de POST varía. Se puede usar para crear un subordinado del recurso de destino identificado en la solicitud del cliente, o para "comportamientos no estándar" donde otros métodos no son adecuados. Por ejemplo, un servicio de fondos mutuos podría aceptar solicitudes para ejecutar funciones como "Fondos de intercambio", "Vender acciones X" y "Vender monto en dólares" a través de POST. Este método también se puede usar como una solución cuando PUT y DELETE están desactivados en el servidor web, o bloqueados en el firewall. En esta situación, los comportamientos que normalmente se llevarían a cabo con otros métodos se ajustan a través de POST. Muchos defensores de REST argumentan en contra de tunelización a través de POST ya que tiende a ofuscar el propósito de la solicitud. Las solicitudes POST tampoco pueden ser almacenadas en caché por intermediarios.

El estilo arquitectónico REST generalmente describe las siguientes seis condiciones al tiempo que deja la implementación de los componentes individuales a los desarrolladores.

1. Modelo cliente servidor
2. Sin estados
3. En caché
4. Interfaz uniforme
5. Sistema de capas
6. Código bajo demanda

1.4 TECNOLOGÍAS Y PROTOCOLOS DE RED

1.4.1 IPV6

IPv6 es una nueva versión del Protocolo de Internet, diseñado como el sucesor de IP versión 4 (IPv4) [26]. Los cambios de IPv4 a IPv6 caen principalmente en las siguientes categorías:

1) Capacidades de direccionamiento expandidas

IPv6 aumenta el tamaño de la dirección IP de 32 bits a 128 bits, para soportar más niveles de direccionamiento jerárquico, un número mucho mayor de nodos direccionables y una configuración automática más sencilla de las direcciones. La escalabilidad del enrutamiento de multi-difusión se mejora añadiendo un campo de "ámbito" a las direcciones de multi-difusión. Y se define un nuevo tipo de dirección llamada "anycast address", utilizada para enviar un paquete a cualquiera de un grupo de nodos.

2) Simplificación de formato de encabezado

Algunos campos de encabezado IPv4 se han eliminado o se han hecho opcionales, para reducir el costo de procesamiento de casos comunes del manejo de paquetes y para limitar el costo de ancho de banda del encabezado IPv6.

3) Soporte mejorado para extensiones y opciones

Los cambios en la forma en que se codifican las opciones de encabezado IP permiten un reenvío más eficiente, unos límites menos estrictos en la longitud de las opciones y una mayor flexibilidad para introducir nuevas opciones en el futuro.

4) Capacidad de etiquetado de flujo

Se agrega una nueva capacidad para permitir el etiquetado de paquetes Pertenecientes a determinados flujos de tráfico para los cuales el remitente Solicita un tratamiento especial, como la calidad de servicio no predeterminada o el servicio "en tiempo real".

5) Autenticación y privacidad

El protocolo IPv6 ha resuelto algunos de los problemas de seguridad que se encuentran en las redes IPv4 mediante la adición obligatoria del IPsec (seguridad IP). Como resultado, IPv6 es más eficiente. IPsec mejora el protocolo IP original al proporcionar la autenticidad, integridad, confidencialidad y control de acceso a cada paquete IP a través de la utilización de dos protocolos: AH (encabezamiento de autenticación) y ESP (carga útil de seguridad de encapsulación). Por otra parte, la expansión del número de bits en el campo de dirección de 128 bits que ofrece IPv6 crea una barrera significativa para los atacantes que desean realizar el escaneo de puertos completo. Además, es posible vincular una clave pública de firma a una dirección IPv6: CGA (Dirección generada criptográficamente).

IPv6 ofrece también mejoras en la seguridad de la movilidad. A pesar de que el protocolo de Internet MobileIP está disponible en IPv4 e IPv6, en IPv6 fue construido en el protocolo en lugar de ser añadido como una nueva función en IPv4.

Esto significa que cualquier nodo IPv6 puede utilizar una IP móvil tanto como sea necesario. MobileIPv6 utiliza dos extensiones titulares: Una cabecera de enrutamiento para el registro y un objetivo principal para la entrega de datos entre nodos móviles y sus nodos fijos correspondientes.

1.4.2 REDES DE TELEFONÍA MÓVIL

GPRS significa General Packet Radio System, es una tecnología que provee acceso de radio paquetes sobre la red GSM existente, en este sentido, constituye una extensión de conmutación de paquetes sobre dicha red. Fue introducida para proporcionar un acceso más eficiente de las redes celulares sobre las redes públicas de datos en comparación con aquellos proporcionados por los servicios tradicionales, basados en conmutación de circuitos, que eran ofrecidos por la tecnología GSM tradicional. Esta tecnología permite acomodar, de una forma más eficiente, fuentes de datos que tienen, por lo general, una naturaleza a ráfagas. Entre sus premisas de diseño vale la pena mencionar las siguientes: Se diseñó como una arquitectura abierta sobre la cual pudieran ser ofrecidos servicios IP, la misma infraestructura debería soportar diferentes interfaces de aire, debería permitir la integración de la infraestructura de telefonía y la infraestructura de Internet. Entre los beneficios derivados de su implantación se mencionan: la transmisión de información sobre la red GSM existente para proveer un servicio de datos de alta velocidad que permanezca «siempre activo» (always on), reduciendo de esta forma el tiempo empleado en la configuración y liberación de las conexiones [6][7].

Origen

Los orígenes de la telefonía móvil se remontan a los inicios de la Segunda Guerra Mundial (1940), durante esos años, Motorola creó un equipo llamado Handie Talkie H12-16, que permitía el contacto con las tropas vía ondas de radio cuya banda de frecuencias en ese tiempo no superaban los 60 MHz. La 0G representa a la telefonía móvil previa a la era celular. Los teléfonos móviles de esta generación eran unos radio teléfonos disponibles como un servicio comercial conectado a la red de telefonía fija, con sus propios números, eran una especie de red como la radio policíaca o el servicio de despacho de taxis. Esos teléfonos móviles usualmente eran montados en carros o camionetas, aunque también se fabricaron modelos portátiles (portafolio). Por lo general, el transmisor (Transmisor-Receptor) era montado en la parte posterior del vehículo (su peso era de unos 50 kg) y unido al resto del equipo (auricular, pantalla y teclado) colocado cerca del asiento del conductor. Los teléfonos

celulares aparecieron debido a la invención de las células hexagonales en 1947, los ingenieros de Bell Labs en AT&T desarrollaron más a fondo esta tecnología durante los años 60. Durante una llamada, la frecuencia del canal no se podía cambiar automáticamente a partir de una célula (área de cobertura de la celda) a otra célula cuando la persona (con su equipo móvil) viajaba del área de una célula al área de otra célula

1) Primera Generación (1G)

Los sistemas de comunicaciones móviles de primera generación o 1G representan al conjunto de estándares celulares que emplean tecnologías analógicas, por lo que comúnmente se habla indistintamente de sistemas analógicos o 1G. Se trataba de sistemas pioneros que introducían por primera vez una característica revolucionaria para los servicios de comunicación comerciales de los años 80, como era la movilidad. Por ello a pesar de que sus prestaciones fueron ampliamente superadas por sistemas más modernos (de 2G), significaron un punto de partida de éxito para el posterior desarrollo de las comunicaciones móviles. La principal característica de estos sistemas era su capacidad para ofrecer servicios de comunicación de voz sobre conmutación de circuitos. Además de la voz, permitían la transmisión de datos empleando módems analógicos convencionales, aunque con una capacidad muy limitada (dificilmente superaban los 4800 bps). Una de las limitaciones de esta tecnología es que la señalización se realizaba "en banda", por lo que, además de ser perceptible por el usuario, no permitía el uso de telefax y módems. La Primera Generación de comunicaciones móviles está basada en FDMA (Frequency Division Multiple Access) y se caracteriza por ser analógica y estrictamente para voz. La tecnología predominante de esta generación es AMPS (Advanced Mobile Phone System) usada principalmente en los Estados Unidos.

2) Segunda Generación (2G)

Se conoce como 2G a la segunda generación de sistemas de comunicación móvil. 2G no es un estándar o un protocolo en sí mismo, sino más bien una forma de marcar el cambio de la comunicación móvil analógica a digital con la incorporación de TDMA (Time Division Múltiple

Access). 2G arribó alrededor de 1990 y su desarrollo derivó de la necesidad de poder manejar un mayor número de llamadas en prácticamente los mismos espectros de frecuencia.

Las tecnologías predominantes son: GSM (Global System por Mobile Communications), IS-136 (conocido también como TIA/EIA136 o ANSI-136), CDMA (Code Division Multiple Access) y PDC (Personal Digital Communications), éste último utilizado en Japón. En Estados Unidos y otros países a 2G se la conoce también como PCS (Personal Communication Services).

La tecnología 2G, a fin de satisfacer un mayor requerimiento en las tasas de transmisión de datos, evolucionó a partir de 3 actualizaciones de TDMA: High Speed Circuit Switched Data (HSCSD), General Packet Radio Service (GPRS) y Enhanced Data Rate for GSM Evolution (EDGE), estos últimos categorizados comercialmente como 2.5 y 2.75G respectivamente y que se caracterizan por incorporar nodos de conmutación de paquetes (Packet Switch, PS) a los ya existentes nodos de conmutación de circuitos (Circuit Switch, CS), la razón es que se trata de tipos de tráfico absolutamente distintos. Por un lado, en Circuit Switch se deben garantizar los recursos de ancho de banda, mientras que en Packet Switch la red realiza lo que se conoce como "best effort" para asignar el ancho de banda.

3) Tercera Generación (3G).

Los sistemas de comunicaciones móviles 3G se caracterizan por la convergencia de voz y datos con acceso a Internet; en otras palabras, es apta para aplicaciones multimedia y altas tasas de transmisión de datos.

Los sistemas de este estándar son básicamente una mejora lineal de los sistemas 2G y al igual que sus evoluciones intermedias, está basado en una infraestructura de backbone paralela, consistente por un lado en nodos de conmutación de circuitos y por otro lado en nodos de conmutación de paquetes (Circuit-Switched and Packet-Switched Domains).

La International Telecommunication Union (ITU) definió las demandas de las redes 3G bajo el estándar IMT-2000. Este sistema se desarrolló principalmente bajo las tecnologías UMTS (Universal Mobile Telecommunications System) usando WCDMA como sucesora 3G de

GSM. Desde 1985 se establecieron en la UIT grupos de trabajo para buscar soluciones a las limitantes de GSM y GRPS. La UIT impulso un proyecto de amplia cooperación internacional conocido como ITM-2000 o Telecomunicaciones Móviles Internacionales 2000, más adelante también conocidos como sistemas móviles de tercera generación o 3G.

Aspectos claves:

- Alto grado de uniformidad de diseño a escala mundial
- Itinerancia (roaming) mundial
- Capacidad para aplicaciones multimedia y una amplia gama de servicios y terminales (ej. video y teleconferencia, Internet de alta velocidad, voz y datos de alta velocidad, datos hasta 2 Mbps).
- Permite a los operadores ofrecer servicios como telefonía inalámbrica, video llamadas y datos inalámbricos de banda ancha en un entorno móvil.
- Provee velocidades de datos de canal de hasta 14.4 Mbps de bajada y 5.8 Mbps de subida. Basada en la familia de estándares IMT-2000 de UIT 3GPP (3rd Generation Partnership Project) continuó el trabajo definiendo un sistema que cumple el estándar: UMTS (Universal Mobile Telecommunications System).

4) Cuarta Generación (4G)

Conocida también como LTE o Long Term Evolution, 4G es el estándar de comunicaciones móviles más moderno que existe, tanto que todavía no ha sido implementado en muchos países. Sin duda alguna, la característica más importante de este nuevo protocolo es la alta tasa de transmisión que puede llegar a alcanzar, en teoría unos 300 Mb/s. 4G fue diseñada con el propósito de satisfacer la demanda de los usuarios que requerían un mayor ancho de banda y capacidad para poder utilizar con comodidad servicios como la televisión móvil, web 2.0, videoconferencias y demás. Si bien la primera implementación comercial de LTE se realizó en Estocolmo en el año 2009, no ha sido implementada totalmente, por lo que muchos usuarios de telefonía móvil del mundo todavía no pueden beneficiarse de esta fantástica tecnología.

Es la sucesora de las tecnologías 2G y 3G, y precede a la próxima generación, la 5G. Al igual que en otras generaciones, la Unión Internacional de Telecomunicaciones (UIT) creó un comité para definir las especificaciones. Este comité es el IMT-Advanced y en él se definen los requisitos necesarios para que un estándar sea considerado de la generación 4G. Entre los requisitos técnicos que se incluyen hay uno muy claro: las velocidades máximas de transmisión de datos deben estar entre 100 Mbit/s para una movilidad alta y 1 Gbit/s para movilidad baja. De aquí se empezó a estudiar qué tecnologías eran las candidatas para llevar la “etiqueta 4G”. Hay que resaltar que los grupos de trabajo de la UIT no son puramente teóricos, sino que la industria forma parte de ellos y estudian tecnologías reales existentes en el momento. Por esto, el estándar LTE (Long Term Evolution) de la norma 3GPP no es 4G porque no cumple los requisitos definidos por la IMT-Advanced en características de velocidades pico de transmisión y eficiencia espectral. Aun así, la UIT declaró en 2010 que los candidatos a 4G, como era aquel, podían publicitarse como 4G. La 4G está basada completamente en el protocolo IP, siendo un sistema y una red, que se alcanza gracias a la convergencia entre las redes de cable e inalámbricas [8].

1.5 ESTADO DEL ARTE

En lo que se refiere al estado del arte de los dispositivos de seguridad personal en el IoT se tiene que los botones de pánico convencionales hacen uso de las redes de telefonía celular GSM para enviar SMS, además de realizar llamadas, y GPS para rastrear el dispositivo. Punait et al. [10] propone un dispositivo de seguridad personal que puede ser utilizado por cualquier persona en forma de dispositivo portátil compatible con filtro Kalman, utiliza una Raspberry Pi 3 Model B y un módulo GSM/GPS. Reddy et al. [11] presenta un botón de emergencia utilizando Arduino UNO y un Módulo GSM/GPS, su dispositivo utiliza las redes de telefonía celular para enviar y recibir SMS en una situación de peligro. La empresa Laipac Technology ofrece varias alternativas en su serie S911 [12], por otro lado, la startup Nimb presentó en 2016 un anillo [13] que cuenta con un botón de pánico y que se conecta a una app instalada en el smartphone del usuario, este anillo se comunica con la app enviándole las instrucciones que realizar, posteriormente la app es la encargada de llamar a los

contactos, grabar audio, y enviar mensajes de texto. Tanto con los productos de Laipac como con Nimb, el usuario no es capaz de definir el comportamiento del botón, el uso del dispositivo está limitado a la respuesta de pulsar una vez el botón. Ashwini et al. [14] presenta un dispositivo que garantiza la seguridad de las mujeres, el dispositivo consiste en un microcontrolador PIC, un módulo GSM y módulos GPS. El sistema se asemeja a un reloj normal que cuando se activa, rastrea el lugar de las mujeres usando GPS y envía mensajes emergentes usando GSM, a contactos sos y la sala de control policial. La principal ventaja de este sistema es que el usuario no requiere un teléfono inteligente a diferencia de otras aplicaciones que se han desarrollado anteriormente. El uso de componentes sofisticados asegura la precisión y lo hace confiable, finalmente Harikiran et al. [15] presenta un completo sistema que está compuesto por un dispositivo que integra múltiples dispositivos, su hardware es una "banda inteligente" portátil que se comunica continuamente con el teléfono inteligente que tiene acceso a Internet. La aplicación del smartphone está programada y precargada con todos los datos necesarios, que incluyen el comportamiento humano y las reacciones a diferentes situaciones como la ira, el miedo y la ansiedad. El software o la aplicación tiene acceso a los servicios de GPS y mensajería que están preprogramados de tal manera que cada vez que recibe una señal de emergencia, puede enviar una solicitud de ayuda junto con las coordenadas de ubicación a la estación de policía más cercana, familiares y personas en el radio cercano que tiene aplicación. Esta acción permite la ayuda instantánea de la policía y del público en el radio cercano que puede alcanzar a la víctima con gran precisión.

CAPITULO 2

ANÁLISIS Y DISEÑO

En este capítulo se detallan los elementos que integran el sistema propuesto, se mencionan las características principales de los componentes del dispositivo físico, posteriormente se presenta como está conformado el web service, la base de datos y finalmente los diagramas de casos de uso, clases y bloques.

2.1 ANÁLISIS DE LOS REQUERIMIENTOS

En esta sección se presenta el conjunto de tareas relacionadas con la determinación de las necesidades del Sistema propuesto; el objetivo principal es detallar de una manera clara, medible y comprobable todos y cada uno de los requerimientos antes de avanzar a la fase de diseño.

Con base en los objetivos de esta tesis se han definido requerimientos funcionales y no funcionales que se desarrollan en los diferentes elementos de este trabajo.

2.1.1 REQUERIMIENTOS FUNCIONALES

Un requerimiento funcional es una descripción de los comportamientos o servicios internos que desempeñará tanto el dispositivo físico como el web service, estas descripciones mencionan claramente la manera en la que el software debe reaccionar ante entradas particulares, y en algunos casos también presentar lo que el sistema no debe hacer.

A continuación, se citarán los requerimientos funcionales analizados, empleando para ello un esquema de tabla, en la cual se especifican los siguientes identificadores:

- Número de identificación del requerimiento
- Nombre del requerimiento
- Descripción general
- Prioridad con la que cuenta el requisito dentro de la aplicación

Estos requerimientos se muestran a continuación de la tabla 2.1.1 a la 2.1.18:

| | |
|-------------|---|
| Número | RF01 |
| Nombre | Botón funcional |
| Descripción | El dispositivo físico debe identificar el tipo de comando introducido por el usuario y enviarlo al web service, en caso que no sea posible la conexión debe esperar un tiempo aleatorio e intentar nuevamente |
| Prioridad | Alta |

Tabla 2.1.1: RF01 – Botón funcional

| | |
|-------------|---|
| Número | RF02 |
| Nombre | Interpretar respuestas |
| Descripción | El dispositivo físico debe ser capaz de recibir la respuesta del web server y procesar la información recibida. |
| Prioridad | Alta |

Tabla 2.1.2: RF02 – Interpretar respuestas

| | |
|-------------|--|
| Número | RF03 |
| Nombre | Notificaciones locales |
| Descripción | El dispositivo físico puede enviar mensajes de texto y realizar llamadas de voz si es que la alerta esté configurada de esa forma. |
| Prioridad | Alta |

Tabla 2.1.3: RF03 – Notificaciones locales

| | |
|-------------|---|
| Número | RF04 |
| Nombre | Inicio de sesión |
| Descripción | Se permite el acceso únicamente a usuarios registrados. |
| Prioridad | Alta |

Tabla 2.1.4: RF04 – Inicio de sesión

| | |
|-------------|---|
| Número | RF05 |
| Nombre | Permitir al usuario crear, modificar y eliminar sus contactos |
| Descripción | El web service debe permitir al usuario administrar libremente sus contactos. |
| Prioridad | Alta |

Tabla 2.1.5: RF05 – Permitir al usuario crear, modificar y eliminar sus contactos

| | |
|-------------|---|
| Número | RF06 |
| Nombre | Permitir al usuario administrar, activar y desactivar alertas |
| Descripción | El web service debe permitir al usuario crear, modificar, activar, desactivar y eliminar sus alertas. |
| Prioridad | Alta |

Tabla 2.1.6: RF06 – Permitir al usuario administrar, activar y desactivar alertas.

| | |
|-------------|--|
| Número | RF07 |
| Nombre | Permitir al usuario administrar sus dispositivos |
| Descripción | El web service debe permitir al usuario administrar libremente sus dispositivos. |
| Prioridad | Alta |

Tabla 2.1.7: RF07 – Permitir al usuario administrar sus dispositivos

| | |
|-------------|--|
| Número | RF08 |
| Nombre | Permitir a un administrador crear, editar y eliminar usuarios |
| Descripción | El web service debe permitir al administrador crear, editar y eliminar usuarios. |
| Prioridad | Alta |

Tabla 2.1.8: RF08 – Permitir a un administrador crear, editar y eliminar usuarios.

| | |
|-------------|--|
| Número | RF09 |
| Nombre | Presentar lista de usuarios |
| Descripción | El web service debe permitir al administrador la visualización de una lista de los usuarios registrados en el sistema, con la finalidad de agilizar su administración. |
| Prioridad | Alta |

Tabla 2.1.9: RF09 – Presentar lista de usuarios

| | |
|-------------|--|
| Número | RF10 |
| Nombre | Presentar lista de contactos |
| Descripción | El web service debe permitir al usuario registrado la visualización de una lista de sus contactos, con la finalidad de agilizar su administración. |
| Prioridad | Alta |

Tabla 2.1.10: RF10 – Presentar lista de contactos

| | |
|-------------|---|
| Número | RF11 |
| Nombre | Presentar lista de dispositivos |
| Descripción | El web service debe permitir al usuario registrado la visualización de una lista de sus dispositivos, con la finalidad de agilizar su administración. |
| Prioridad | Alta |

Tabla 2.1.11: RF11 – Presentar lista de dispositivos

| | |
|-------------|--|
| Número | RF12 |
| Nombre | Presentar lista de alertas |
| Descripción | El web service debe permitir al usuario registrado la visualización de una lista de sus alertas, con la finalidad de agilizar su administración. |
| Prioridad | Alta |

Tabla 2.1.12: RF12 – Presentar lista de alertas

| | |
|-------------|--|
| Número | RF13 |
| Nombre | Mapa de la ubicación |
| Descripción | El web service debe permitirle al usuario y al contacto que elija, la visualización del mapa con la ubicación del dispositivo físico en caso de haber sido activado. |
| Prioridad | Alta |

Tabla 2.1.13: RF13 – Mapa de ubicación

| | |
|-------------|---|
| Número | RF14 |
| Nombre | El dispositivo puede enviar mensajes de texto |
| Descripción | El dispositivo físico debe ser capaz de enviar mensajes de texto al contacto seleccionado por el usuario. |
| Prioridad | Alta |

Tabla 2.1.14: RF14 – El dispositivo físico puede enviar mensajes de texto

| | |
|-------------|---|
| Número | RF15 |
| Nombre | El web server puede enviar mensajes de texto |
| Descripción | El web server debe ser capaz de enviar mensajes de texto al contacto seleccionado por el usuario. |
| Prioridad | Alta |

Tabla 2.1.15: RF15 – El web server puede enviar mensajes de texto

| | |
|-------------|---|
| Número | RF16 |
| Nombre | El dispositivo puede realizar llamadas de voz |
| Descripción | El dispositivo físico debe ser capaz de realizar llamadas de voz al contacto seleccionado por el usuario. |
| Prioridad | Alta |

Tabla 2.1.16: RF16 – El dispositivo físico puede realizar llamadas de voz

| | |
|-------------|---|
| Número | RF17 |
| Nombre | El web server puede realizar llamadas de voz |
| Descripción | El web server debe ser capaz de realizar llamadas de voz al contacto seleccionado por el usuario. |
| Prioridad | Alta |

Tabla 2.1.17: RF17 – El web server puede realizar llamadas de voz.

| | |
|-------------|--|
| Número | RF18 |
| Nombre | El web server notificar por internet |
| Descripción | El web server debe ser capaz de lanzar la alerta usando los servicios de internet configurados (Twitter, Telegram, Facebook, Google Maps o E-mail). A un contacto seleccionado por el usuario. |
| Prioridad | Alta |

Tabla 2.1.18: RF18 – El web server notificar por internet.

2.1.2 REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales o también conocidos como atributos de calidad, son requisitos empleados para opinar o valorar la operación de una aplicación, en lugar de juzgar sus comportamientos específicos.

| | |
|-------------|--|
| Número | RNF01 |
| Nombre | Dispositivo sencillo |
| Descripción | El dispositivo físico debe presentar únicamente el botón para activar la alarma. |
| Prioridad | Alta |

Tabla 2.1.19: RNF01 – Dispositivo sencillo

| | |
|-------------|--|
| Número | RNF02 |
| Nombre | Interfaz sencilla y amigable |
| Descripción | El web service debe presentar únicamente las herramientas necesarias para cada interfaz, además debe mostrar de manera precisa la información requerida. |
| Prioridad | Alta |

Tabla 2.1.20: RNF02 – Interfaz sencilla y amigable

| | |
|-------------|---|
| Número | RNF03 |
| Nombre | Diseño responsivo |
| Descripción | La interfaz web debe mostrarse correctamente en cualquier dispositivo no importando que navegador use o que resolución tenga. |
| Prioridad | Alta |

Tabla 2.1.21: RNF03 – Diseño responsivo

2.2 ELEMENTOS DEL SISTEMA

El Hardware del sistema es un dispositivo de tamaño compacto, integra una placa Arduino Pro Micro y un Módulo SIM800L; y un Web server desarrollado en PHP, con MySQL como gestor de base de datos funcionando en un servidor Web Apache en una distribución Linux Debian 9. Estos elementos se comunican por medio de la red de telefonía celular GSM, utilizando HTTP como protocolo de comunicación.

A continuación, en la figura 2.2.1 se muestra el diagrama general del sistema propuesto, donde el usuario interactúa directamente con el dispositivo físico y por medio de una GUI web o aplicación móvil accede al Web server para configurar el comportamiento del botón, así como el tipo de alertas y contactos que desea activar.

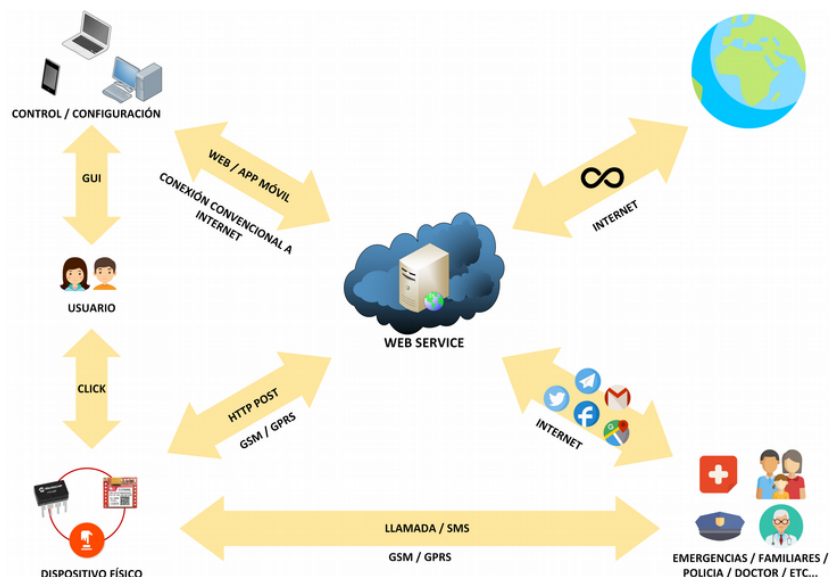


Figura 2.2.1: Diagrama del sistema.

2.3 ARDUINO PRO MICRO

Arduino es una placa electrónica programable de código abierto con puertos de entrada/salida (E/S) y un entorno de desarrollo (IDE) basado en Processing. Arduino puede utilizarse para desarrollar objetos interactivos independientes o puede conectarse a un software como Flash, Processing, VVVV o Max/MSP. Las placas se pueden montar a mano o se pueden adquirir preensamblados; el IDE de código abierto se puede descargar gratuitamente desde www.arduino.cc [16].

El origen de Arduino se remonta al año 2005, en el instituto IVRAE en Italia, gracias al estudiante de la Universidad de Colombia Hernando Barragán que en su proyecto de tesis desarrollo Wiring [17], el precursor de Arduino, el objetivo principal del proyecto fue simplificar el proceso de trabajar con micro controladores.

2.3.1 CARACTERÍSTICAS

Es similar al Pro Mini con la diferencia que posee un ATmega32U4 integrado, este chip microcontrolador hace una gran diferencia ya que también puede ser utilizado como dispositivo de interfaz humana. El transceptor USB dentro del 32U4 es el que permite acoplar una conectividad USB. Realiza todas las funciones Arduino con: 4 canales de 10 - bit ADC, 5 pines PWM, 12 DIO, conexiones serie Rx y Tx y un regulador de voltaje que acepta hasta 12V. [18]. Ver figura 2.2.2.

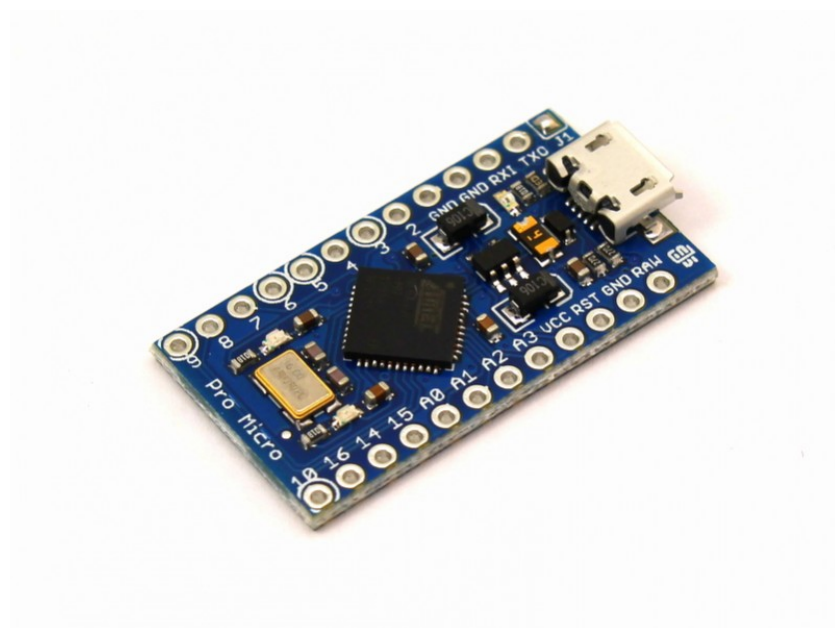


Figura 2.2.2: Arduino Pro Micro

El Arduino Pro Micro posee una gran variedad de pines de propósito general que se adaptan a cualquier proyecto, en la figura 2.2.3 se presenta el diagrama y funcionamiento de estos pines:

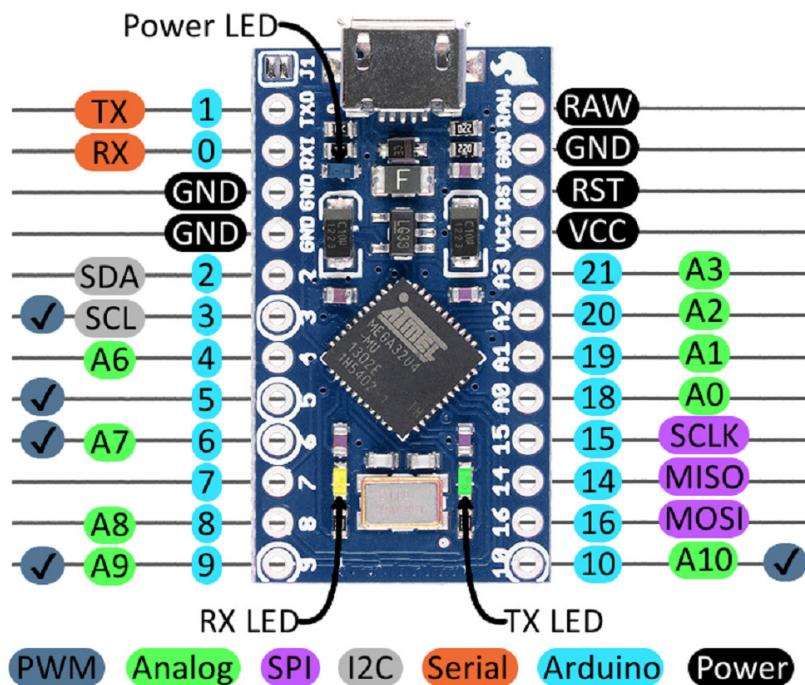


Figura 2.2.3: Diagrama de pines del Arduino Pro Micro

A continuación, en la tabla 2.2.1 se presentan las principales características del Arduino Pro Micro:

| ELEMENTO | INFORMACIÓN |
|-----------------------------------|------------------|
| Microcontrolador | ATmega32U4 |
| Voltaje de operación | 5v – 12v |
| Voltaje de entrada (Recomendado) | 7v |
| Voltaje de entrada (Limite) | 12v |
| Tensión de salida | 5v |
| Corriente máxima de salida | 150mA |
| Pines totales | 21 |
| Pines para entrada/salida digital | 12 |
| Pines de entrada Analógica | 4 |
| Velocidad | 16mhz |
| Corriente de entrada por pin IO | 20mA |
| Memoria FLASH | 32KB |
| SRAM | 2.5KB |
| EEPROM | 1KB |
| Dimensiones | 3.31cm X 1.78 cm |

Tabla 2.2.1: Características del Arduino Pro Micro

2.4 SIM800L

El módulo SIM800L es un módulo GSM/GPRS Quad - Band que trabaja con frecuencias de 850/900/1800/1900 MHz. En este sistema se configura el modulo para enviar los mensajes HTTP al web server alojado en un Servidor funcionando con el sistema operativo GNU/Linux, utilizando la tecnología GSM/GPRS. Ver figura 2.2.3.



Figura 2.2.3: Módulo SIM800L

2.4.1 CARACTERÍSTICAS

En la figura 2.2.4 se presenta la configuración de los pines del Módulo SIM800L:

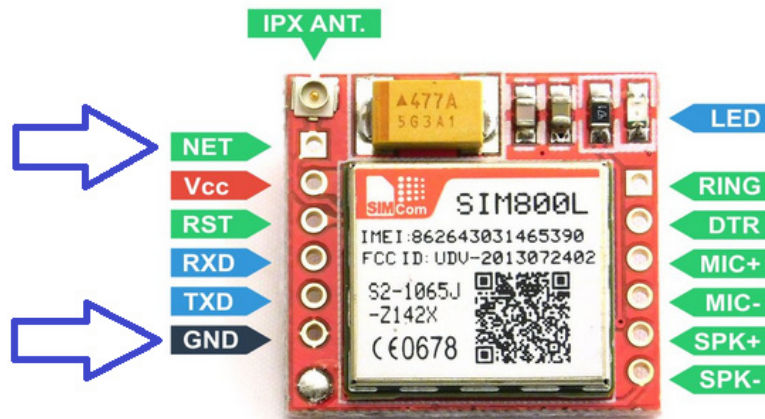


Figura 2.2.4: Configuración de pines del Módulo SIM800L

- **IPX ANT y NET:** Conexión de la antena.
- **RST:** Reset, Reinicia el módulo.
- **RXD y TXD:** Comunicación serial.
- **LED:** Indicador de funcionamiento.
- **RING:** Se activa al recibir una llamada o un mensaje, se conecta a un buzzer.
- **DTR:**
- **MiC+ y MiC-:** Micrófono para las llamadas de voz.
- **SPK+ y SPK- :** Bocina para las llamadas de voz.

A continuación, en la tabla 2.2.3 se presentan las principales características del Módulo SIM800L:

| ELEMENTO | INFORMACIÓN |
|-----------------------------------|--|
| Voltaje de operación | 3.4v – 4.4v |
| Voltaje recomendado | 4v |
| Consumo en modo de suspensión | 2mA |
| Consumo en modo inactivo | 7mA |
| Consumo normal en transmisión GSM | 350mA |
| Consumo Pico en Transmisión GSM | 2000mA |
| Interfaz | UART y comandos AT |
| Ranura SIM | MicroSIM |
| Frecuencias | Cuatribanda: 850, 900, 1800 y 1900 MHz |
| Temperatura de operación | -45 – 85° C |
| Dimensiones | 1.5cm x 1.8cm |

Tabla 2.2.3: Características del Módulo SIM800L

2.5 WEB SERVICE

El segundo elemento que conforma el servicio propuesto es el web service, programado en su totalidad en PHP, con MySQL como gestor de base de datos y hospedado en un web server con sistema operativo Linux, configurado como servidor web. A continuación, se detallan los elementos de software necesarios para el funcionamiento del web service.

Para hospedar el web server se necesita un servidor web que soporte el lenguaje de programación PHP, se optó por utilizar Apache, es el servidor web más utilizado desde hace más de 10 años como lo muestra la figura 2.2.4, publicada en el sitio Netcraft [19], el servidor Apache es líder mundial debido a sus múltiples ventajas las cuales mencionaremos brevemente a continuación:

- Es modular.
- Es una tecnología gratuita.
- Es de código fuente abierto.
- Trabaja con PYTHON, PERL, PHP, RUBY, REXX, MONO y otros lenguajes de programación.
- Trabaja en Windows, Linux y Mac.
- Incluye soporte “SSL” y “TLS”.
- Gran comunidad de soporte.

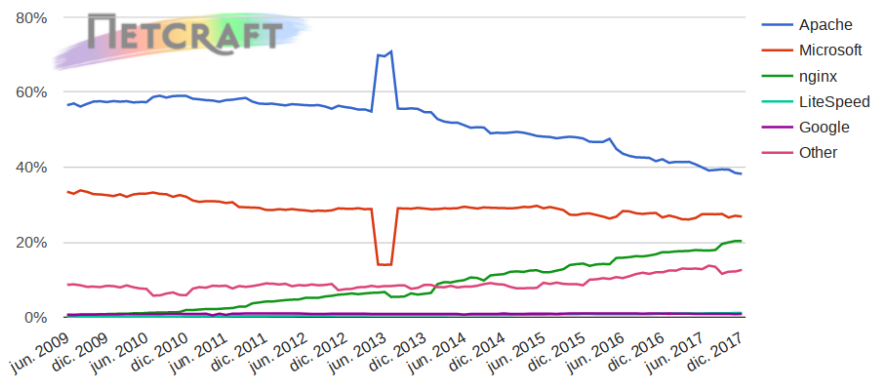


Figura 2.2.5: Servidores web más utilizados Diciembre 2017

2.6 MYSQL SERVER

Una parte importante en el desarrollo del sistema propuesto es la base de datos, tomando en cuenta los resultados de la encuesta elaborada por Eclipse IoT Working Group, AGILE IoT, IEEE, y Open Mobile Alliance en su edición 2018 [32], de entre las Bases de datos más utilizadas en el desarrollo de IoT se optó por utilizar MySQL debido a la experiencia de uso en trabajos previos, MySQL funciona en Sistemas Operativos Windows o Linux, es totalmente compatible con PHP; su conectividad, velocidad y seguridad hace que sea el servidor apropiado para almacenar la información de usuarios, contactos, alertas y otras configuraciones.

Características:

- Escrito en C y en C++
- Multiplataforma
- Sometido constantemente a pruebas de rendimiento
- Diseñado en varias capas con módulos independientes
- Multiproceso usando hilos del kernel
- Implementa tablas HASH en memoria
- Probado con un amplio rango de compiladores diferentes
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas HASH en memoria, que son usadas como tablas temporales.

Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.

El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL (<http://developer.kde.org/~sewardj/>).

El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible. [20].

2.7 MODELADO DE CASOS DE USO

El siguiente diagrama de casos de uso muestra algunas funciones que el servicio podría proporcionar. Dos actores que usan interfaces administrativas son el root y el webmaster. El webmaster utiliza un subconjunto de funciones disponibles para el root. Todos los casos de uso mostrados que son abstractos, lo son porque cada uno representa algún grupo o "paquete" de funcionalidad administrativa.

2.7.1 DIAGRAMA DE CASOS DE USO

La figura 2.7.1 muestra el diagrama de casos de uso global del sistema, esto permite tener una visión general del funcionamiento del mismo, sus actores y la relación que existe entre estos.

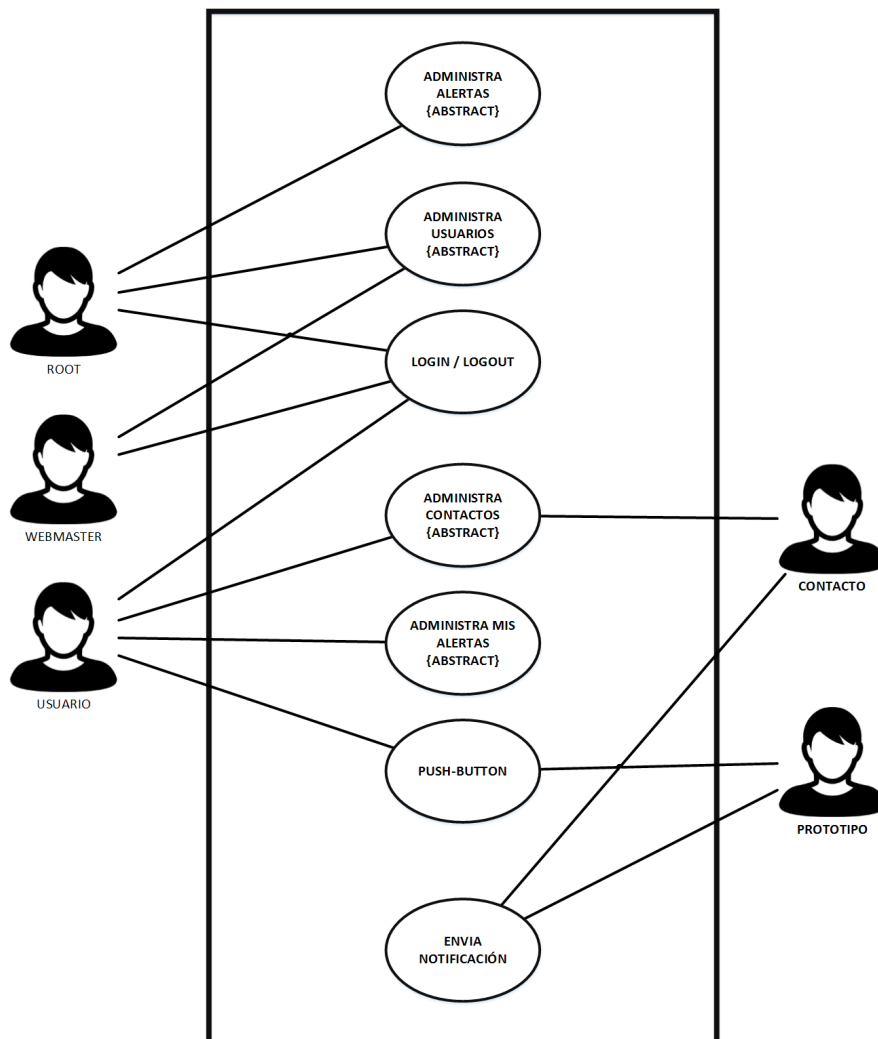


Figura 2.7.1: Diagrama de casos de uso.

2.7.2 ESPECIFICACIONES DE CASOS DE USO

Más adelante se muestra la descripción de cada uno de esos casos pertenecientes al sistema.

| | |
|------------------|--|
| Caso de uso: | Administrar alertas |
| Descripción: | El usuario root o administrador general puede crear, modificar y eliminar alertas disponibles en el sistema. |
| Actores: | Usuario Root |
| Pre-condiciones: | El usuario root debe tener sesión iniciada. |
| Flujo normal: | |

Tabla 2.7.1: Descripción del caso de uso Administrar alertas.

| | |
|------------------|---|
| Caso de uso: | Administrar usuarios |
| Descripción: | El usuario root o administrador general puede crear, modificar y eliminar usuarios del sistema. |
| Actores: | Usuario Root, webmaster |
| Pre-condiciones: | El usuario debe tener sesión iniciada. |
| Flujo normal: | <p>Crear usuario:</p> <p>El actor pulsa el botón nuevo usuario Ingresa el nombre del nuevo usuario ingresa el password del nuevo usuario ingresa el rango del nuevo usuario</p> <p>Modificar usuario:</p> <p>El actor pulsa el botón ver usuarios El actor selecciona el usuario El actor actualiza la información del usuario</p> <p>Eliminar Usuario:</p> <p>El actor pulsa el botón ver usuarios El actor selecciona el usuario El actor pulsa el botón eliminar usuario</p> |

Tabla 2.7.2: Descripción del caso de uso Administrar usuarios.

| | |
|------------------|---|
| Caso de uso: | Login / logout |
| Descripción: | El actor de este caso puede iniciar sesión en el sistema solo si esta registrado en la base de datos, las opciones de la interfaz de usuario dependerán del rango del actor. |
| Actores: | Usuario Root, webmaster, usuario |
| Pre-condiciones: | El web server debe estar disponible |
| Flujo normal: | Login: El actor pulsa el botón login El actor ingresa su nombre de usuario El actor ingresa su password inicia sesión logout: El actor pulsa el botón de login Termina su sesión |

Tabla 2.7.3: Descripción del caso de uso Login / Logout.

| | |
|------------------|--|
| Caso de uso: | Administrar contactos |
| Descripción: | El usuario puede agregar, editar y eliminar contactos, un contacto es a la persona a la cual le llegara la alerta en caso de una emergencia. |
| Actores: | Usuario |
| pre-condiciones: | El usuario debe tener sesión iniciada. |
| Flujo normal: | <p>Crear contacto:</p> <p>El actor pulsa el botón nuevo contacto</p> <p>El actor Ingresa el nombre del nuevo contacto</p> <p>El actor ingresa el número telefónico del nuevo contacto</p> <p>El actor ingresa el email del nuevo contacto</p> <p>modificar contacto:</p> <p>El actor pulsa el botón ver contacto</p> <p>El actor selecciona el contacto</p> <p>El actor actualiza la información del contacto</p> <p>eliminar contacto:</p> <p>El actor pulsa el botón ver contacto</p> <p>El actor selecciona el contacto</p> <p>El actor pulsa el botón eliminar contacto</p> |

Tabla 2.7.4: Descripción del caso de uso Administrar contactos.

| | |
|------------------|---|
| Caso de uso: | Administrar alertas |
| Descripción: | El usuario puede agregar, editar y eliminar alertas, una alertas es la notificación que el sistema enviara al contacto seleccionado |
| Actores: | Usuario |
| pre-condiciones: | El usuario debe tener sesión iniciada. |
| Flujo normal: | <p>Crear Alerta:</p> <p>El actor pulsa el botón nuevo alerta</p> <p>El actor selecciona el tipo de alerta</p> <p>El actor selecciona el tipo de notificación</p> <p>El actor selecciona el contacto a notificar</p> <p>El actor pulsa el botón Guardar Alerta</p> <p>Modificar Alerta:</p> <p>El actor pulsa el botón ver mis alertas</p> <p>El actor selecciona la alerta</p> <p>El actor pulsa el botón editar alerta</p> <p>El actor modifica la información de la alerta</p> <p>El actor pulsa el botón Actualizar Alerta</p> <p>Eliminar Alerta:</p> <p>El actor pulsa el botón ver mis alertas</p> <p>El actor selecciona la alerta</p> <p>El actor pulsa el botón eliminar alerta</p> |

Tabla 2.7.5: Descripción del caso de uso Administrar alertas.

| | |
|------------------|--|
| Caso de uso: | Push button |
| Descripción: | El usuario interactúa directamente con el dispositivo físico, pulsando el botón para enviar la notificación en caso de una emergencia. |
| Actores: | Usuario. Prototipo |
| Pre-condiciones: | El dispositivo físico debe estar en funcionamiento. El dispositivo físico debe tener conectividad a Internet. |
| Flujo normal: | El usuario pulsa el botón de pánico El dispositivo físico interpreta el tipo de comando El dispositivo físico se comunica por medio de Internet con el web Service El web Service interpreta el comando El web Service lanza la notificación al contacto previamente seleccionado. |

Tabla 2.7.6: Descripción del caso de uso Push button.

2.8 DIAGRAMA DE CLASES

En la Figura 2.8.1 vemos que el sistema está integrado por 14 clases, 10 en la parte del web server y 4 en el dispositivo físico con el botón.

Las 10 clases que componen el web server brindan una interfaz visual de acceso a la configuración de las alertas, contactos y dispositivos.

- index
- login
- logout
- usuario
- contacto
- alerta
- dispositivo
- facebook
- email
- gps

Las 4 restantes están del lado del dispositivo físico:

- comando
- sms
- llamadaCelular
- llamadaCasa

DESARROLLO DE UN BOTÓN DE PÁNICO INTELIGENTE
CON LA TECNOLOGÍA DE INTERNET DE LAS COSAS

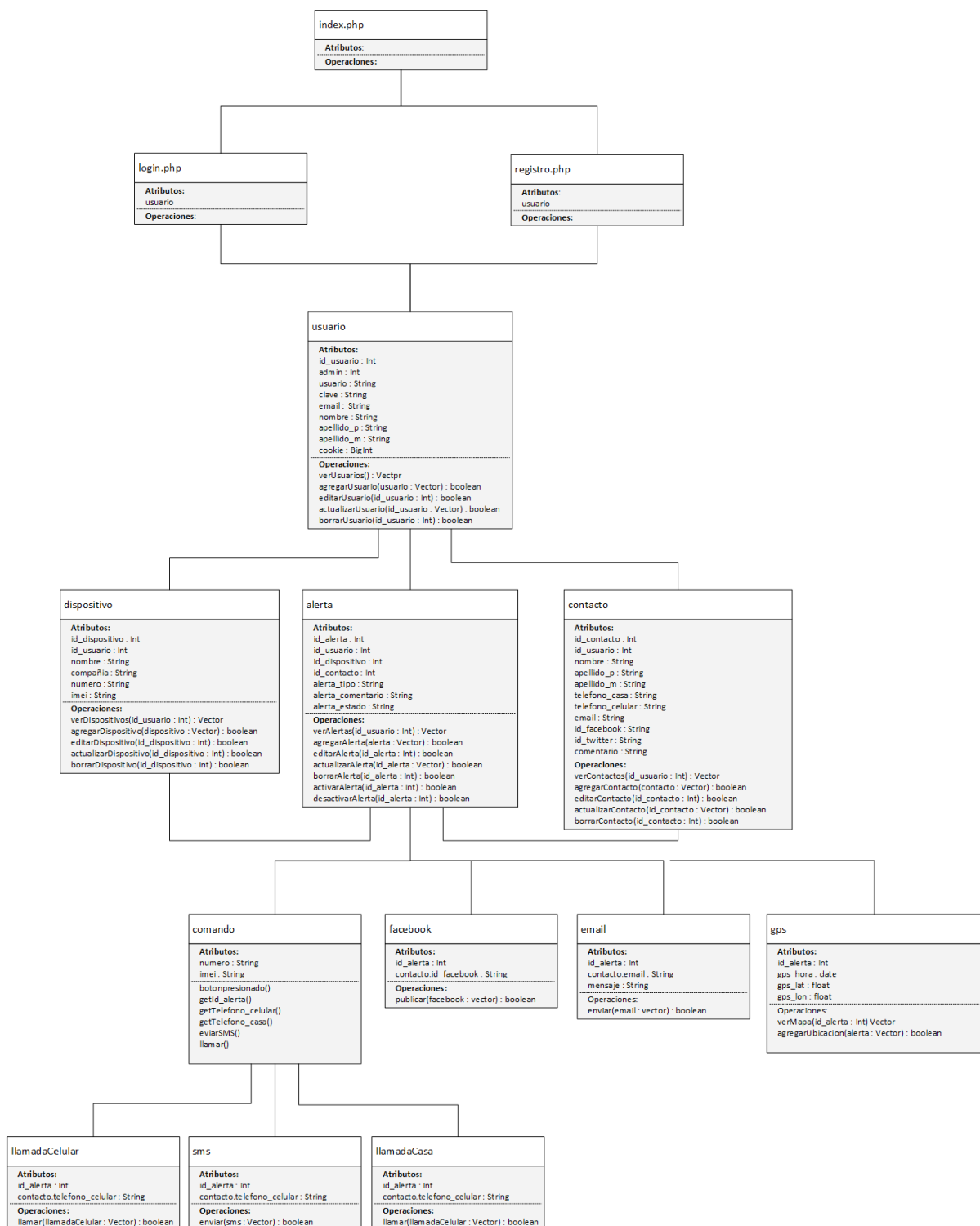


Figura 2.8.1: Diagrama de clases

Enseguida, se muestra la descripción de cada una de las clases que componen al sistema, Los atributos y operaciones demasiado obvios no se describen a detalle.

En la tabla 2.8.1 se realiza la descripción de la clase index, que es la interfaz principal del web service.

| | |
|-------------|--|
| Clase | Index.php |
| Descripción | Interfaz principal del sistema, muestra las opciones de registro y login, además del menú principal para los administradores y usuarios registrados. |
| Atributos | |
| Operaciones | |

Tabla 2.8.1: Descripción de la tabla index.php.

En la tabla 2.8.2 Se describe la clase registro, que es la que nos permite registrarnos como un nuevo usuario en el sistema.

| | |
|-------------|---|
| Clase | registro.php |
| Descripción | Permite el registro de nuevos usuarios, se conecta con la base de datos insertando un nuevo registro. |
| Atributos | |
| Operaciones | |

Tabla 2.8.2: Descripción de la tabla registro.php.

En la tabla 2.8.3 se se describe la clase login, encargada del inicio de sesión de usuarios y administradores.

| | |
|-------------|---|
| Clase | login.php |
| Descripción | Una vez que el usuario está registrado puede iniciar sesión gracias a esta clase. |
| Atributos | |
| Operaciones | |

Tabla 2.8.3: Descripción de la tabla login.php.

En la tabla 2.8.4 se describe la clase usuario, sus atributos y operaciones.

| Clase | Usuario |
|-------------|--|
| Descripción | Clase que nos permite acceder a la base de datos para manejar la información de los usuarios y administradores del sistema. |
| • Atributos | <ul style="list-style-type: none">• id_usuario: identificador único de usuario.• admin: Indica si el usuario es Administrador del sistema.• usuario: nombre de usuario.• clave: contraseña de autenticación.• email: E-mail del usuario.• nombre: nombre(s) del usuario.• apellido_p: apellido paterno.• apellido_m: apellido materno.• cookie: almacena la cookie que permite iniciar sesión. |
| Operaciones | <ul style="list-style-type: none">• verUsuarios• agregaUsuario• editaUsuario• actualizaUsuario• borraUsuario |

Tabla 2.8.4: Descripción de la tabla usuario.

En la tabla 2.8.5 se describe la clase dispositivo, sus atributos y operaciones.

| | |
|-------------|---|
| Clase | dispositivo |
| Descripción | <ul style="list-style-type: none">• Esta clase al igual que la anterior nos permite manejar los registros concernientes a la información de los dispositivos de cada usuario. |
| Atributos | <ul style="list-style-type: none">• id_usuario: identificador único del usuario propietario• id_dispositivo: identificador único del dispositivo• nombre: nombre del dispositivo• compañía: compañía telefónica del SIM del dispositivo• numero: número telefónico del SIM• imei: número de 16 caracteres que identifica universalmente a cada SIM, permite autenticar el dispositivo en el sistema. |
| Operaciones | <ul style="list-style-type: none">• verDispositivos• agregaDispositivo• editaDispositivo• actualizaDispositivo• eliminaDispositivo |

Tabla 2.8.5: Descripción de la tabla dispositivo.

En la tabla 2.8.6 se describe la clase alerta, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | alerta |
| Descripción | Esta clase relaciona la información de las clases contacto, dispositivo y comando para lanzar las notificaciones configuradas por el usuario. |
| Atributos | <ul style="list-style-type: none">• id_usuario: identificador único del usuario propietario.• id_alerta: identificador único de la alerta.• id_dispositivo: identificador del dispositivo en uso.• id_contacto: contacto al cual se alertara.• alerta_tipo: puede ser SMS, llamada, email, etc...• alerta_comentario: comentario o nota.• alerta_estado: activada o desactivada. |
| Operaciones | <ul style="list-style-type: none">• verAlertas• agregaAlerta• editaAlerta• actualizaAlerta• eliminaAlerta |

Tabla 2.8.6: Descripción de la tabla alerta.

En la tabla 2.8.7 se describe la clase contacto, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | contacto |
| Descripción | La clase contacto le permite al usuario gestionar la información de sus contactos. |
| Atributos | <ul style="list-style-type: none">• id_usuario: identificador único del usuario• id_contacto: identificador único del contacto• nombre: nombre del contacto.• apellido_p: apellido paterno.• apellido_m: apellido materno.• telefono_casa: teléfono de casa.• telefono_celular: telefonocelular.• email: email del contacto.• id_facebook: id de facebook.• twitter: usuario de twitter.• comentario: comentario o nota. |
| Operaciones | <ul style="list-style-type: none">• verContactos• agregaContacto• editaContacto• actualizaContacto• eliminaContacto |

Tabla 2.8.7: Descripción de la tabla contacto.

En la tabla 2.8.8 se describe la clase comando que está en el lado del dispositivo físico, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | comando |
| Descripción | Esta clase permite la comunicación del prototipo con el web server, permite realizar 2 tipos de notificaciones, y enviar la localización gps del usuario. |
| Atributos | <ul style="list-style-type: none">• Número: número telefónico del SIM del dispositivo.• imei: IMEI del SIM. |
| Operaciones | <ul style="list-style-type: none">• botonpresionado• getId_alerta• getTelefono_casa• getTelefono_celular• enviaSMS• llama |

Tabla 2.8.8: Descripción de la tabla comando.

En la tabla 2.8.9 se describe la clase sms, sus atributos y operaciones.

| | |
|-------------|---|
| Clase | sms |
| Descripción | Permite enviarle un mensaje de texto (SMS) al contacto previamente seleccionado. |
| Atributos | <ul style="list-style-type: none">• id_alerta: identificador único de la alerta• contacto.numero_celular: número del contacto a notificar. |
| Operaciones | <ul style="list-style-type: none">• enviaSMS |

Tabla 2.8.9: Descripción de la tabla sms.

En la tabla 2.8.10 se describe la clase llamadaCasa, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | LlamadaCasa |
| Descripción | Permite hacer una llamada al número de casa del contacto seleccionado. |
| Atributos | <ul style="list-style-type: none">• id_alerta• contacto.numero_casa |
| Operaciones | <ul style="list-style-type: none">• llamar |

Tabla 2.8.10: Descripción de la tabla llamadacasa.

En la tabla 2.8.11 se describe la clase llamadaCelular, sus atributos y operaciones.

| | |
|-------------|---|
| Clase | LlamadaCelular |
| Descripción | Permite hacer una llamada al número celular del contacto seleccionado. |
| Atributos | <ul style="list-style-type: none">• id_alerta• contacto.numero_celular |
| Operaciones | <ul style="list-style-type: none">• llamar |

Tabla 2.8.11: Descripción de la tabla llamadacelular.

En la tabla 2.8.12 se describe la clase facebook, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | facebook |
| Descripción | Permite publicar la alerta en el perfil de facebook del contacto seleccionado. |
| Atributos | <ul style="list-style-type: none">• id_alerta• contacto.id_facebook |
| Operaciones | <ul style="list-style-type: none">• notificar |

Tabla 2.8.12: Descripción de la tabla facebook.

En la tabla 2.8.13 se describe la clase gps, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | gps |
| Descripción | Permite visualizar la localización GPS del usuario. |
| Atributos | <ul style="list-style-type: none">• id_alerta• gps_lat• gps_lon• hora |
| Operaciones | <ul style="list-style-type: none">• EnviaUbicación• verMapa |

Tabla 2.8.13: Descripción de la tabla gps.

Finalmente, en la tabla 2.8.14 se describe la clase email, sus atributos y operaciones.

| | |
|-------------|--|
| Clase | email |
| Descripción | Esta clase permite enviar un email al contacto seleccionado. |
| Atributos | <ul style="list-style-type: none">• id_alerta• contacto.email• mensaje |
| Operaciones | <ul style="list-style-type: none">• enviaEmail |

Tabla 2.8.14: Descripción de la tabla email

2.9 DIAGRAMA DE BLOQUES

En la Figura 2.9.1 se muestra el diagrama de bloques del sistema, donde se aprecian los diferentes componentes que lo integran. El dispositivo físico está integrado por un microcontrolador con un botón y un módulo SIM800L que se comunica por medio de la red celular GSM / GPRS y utilizando el protocolo HTTP con el Web service, que al recibir la señal del dispositivo físico realiza las notificaciones previamente configuradas por el usuario.

Estas notificaciones pueden ser un simple correo electrónico, contacto por las redes sociales Facebook y Twitter, o censando la ubicación por medio de las coordenadas GPS enviadas por el dispositivo.

Además de “activar” el envío de notificaciones del Web service, el dispositivo físico puede preguntarle al web Service a que números notificar por medio de una llamada o un mensaje de texto. Todas estas notificaciones pueden ser configuradas por el usuario por medio de la interfaz web o una aplicación móvil.

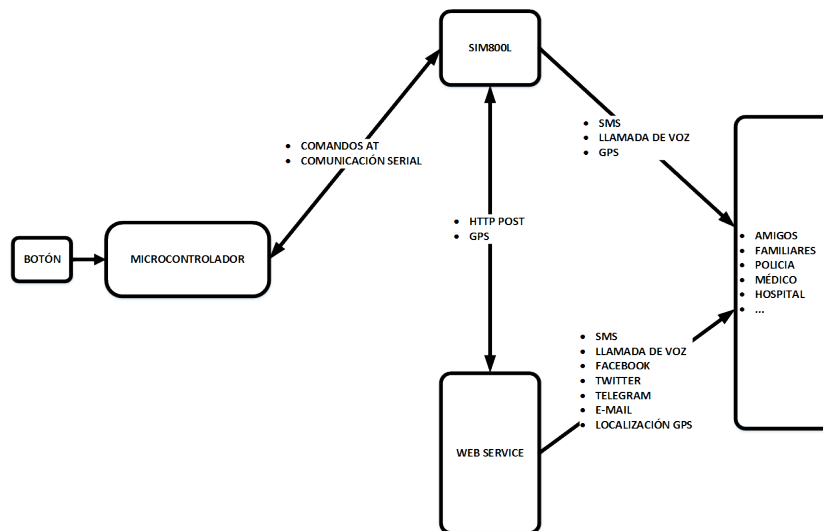


Figura 2.9.1: Diagrama de bloques del sistema.

2.10 DISEÑO DE LA BASE DE DATOS

A continuación, se presenta el diccionario de datos, en el cual se presenta la forma de almacenamiento de los datos utilizados en el desarrollo de la base de datos para el web server.

2.10.1 DICCIONARIO DE DATOS

A continuación, se presenta el conjunto de meta-datos el cual posee las características lógicas de los datos almacenados en las tablas de la base de datos.

La información se muestra en forma de tabla siguiendo con el siguiente estándar:

- Clave de la tabla
- Nombre del campo
- Tipo de campo
- Longitud del campo
- Descripción

La especificación de cada tabla se muestra a continuación de la tabla 2.10.1 a la 2.10.5:

| Tabla usuario | | | | |
|---------------|------------|---------|----------|--|
| Clave | Campo | Tipo | Longitud | Descripción |
| Primaria | id_usuario | Int | 3 | Identificador único de usuario |
| --- | admin | TinyInt | 1 | Indica si el usuario es admin (1) o usuario normal (0) |
| --- | usuario | VARCHAR | 59 | Almacena el nombre del usuario registrado |
| --- | clave | VARCHAR | 32 | Almacena la clave del usuario registrado |
| --- | nombre | VARCHAR | 50 | Almacena el nombre personal del usuario |
| --- | apellido_p | VARCHAR | 50 | Almacena el apellido paterno |
| --- | apellido_m | VARCHAR | 50 | Almacena el apellido materno |
| --- | cookie | VARCHAR | 50 | Cookie de inicio de sesión. |

Tabla 2.10.1: Tabla usuario

| Tabla contacto | | | | |
|----------------|-------------|---------|----------|---|
| Clave | Campo | Tipo | Longitud | Descripción |
| Primaria | id_contacto | Int | 3 | Identificador único de contacto |
| Secundaria | id_usuario | Int | 3 | Indica a que usuario pertenece el contacto |
| --- | nombre | VARCHAR | 50 | Almacena el nombre del contacto |
| --- | apellido_p | VARCHAR | 50 | Almacena el apellido paterno del contacto |
| --- | apellido_m | VARCHAR | 50 | Almacena el apellido materno del contacto |
| --- | tel_casa | VARCHAR | 10 | Almacena el número del domicilio |
| --- | tel_cel | VARCHAR | 10 | Almacena el número de celular |
| --- | email | VARCHAR | 50 | Almacena el E-mail |
| --- | id_twitter | VARCHAR | 20 | Almacena el usuario de Twitter asociado al contacto |
| --- | id_facebook | VARCHAR | 20 | Almacena el id de Facebook asociado al contacto |
| --- | id_telegram | VARCHAR | 20 | Almacena el id de Telegram asociado al contacto |
| --- | Nota | VARCHAR | 300 | Almacena un comentario o nota |

Tabla 2.10.2: Tabla contacto

| Tabla dispositivo | | | | |
|-------------------|----------------|---------|----------|--|
| Clave | Campo | Tipo | Longitud | Descripción |
| Primaria | id_dispositivo | Int | 3 | Identificador único del dispositivo |
| Secundaria | id_usuario | Int | 3 | Indica a que usuario pertenece |
| --- | nombre | VARCHAR | 50 | Almacena el nombre del dispositivo |
| --- | número | VARCHAR | 10 | Almacena el número del SIM del dispositivo |
| --- | IMEI | VARCHAR | 15 | Almacena el IMEI del dispositivo |

Tabla 2.10.3: Tabla dispositivo

| Tabla alerta | | | | |
|--------------|----------------|---------|----------|--|
| Clave | Campo | Tipo | Longitud | Descripción |
| Primaria | id_alerta | Int | 3 | Identificador único de la alerta |
| Secundaria | id_usuario | Int | 3 | Indica a que usuario pertenece la alerta |
| Secundaria | id_dispositivo | Int | 3 | Indica el dispositivo asociado a la alerta |
| Secundaria | id_contacto | Int | 3 | Indica el contacto al cual se notificará en caso de activarse |
| --- | tipo | Int | 2 | Indica el tipo de comando asociado a la alerta |
| --- | notificación | Int | 2 | Indica que tipo de notificación esta asociada a la alerta |
| --- | mensaje | VARCHAR | | Almacena el mensaje personalizado que se utilizará en la notificación. |
| --- | estado | TinyInt | 1 | Indica el estado de la alerta. Activa (1) o inactiva (0). |

Tabla 2.10.4: Tabla alerta

| Tabla gps | | | | |
|-----------|-------|---------|----------|--|
| Clave | Campo | Tipo | Longitud | Descripción |
| --- | IMEI | VARCHAR | 15 | Indica el dispositivo asociado a las coordenadas |
| --- | lat | DOUBLE | 15 | Indica la latitud de la localización |
| --- | lon | DOUBLE | 15 | Indica la longitud de la localización |
| --- | hora | DATE | --- | Indica la hora en la cual fue capturada la localización. |

Tabla 2.10.5: tabla gps

2.10.2 MODELO CONCEPTUAL

En la Figura 2.10.2 se presenta el modelo conceptual de la base de datos en donde se definen las entidades, las relaciones existentes entre esas entidades y las restricciones de integridad que se aplican sobre ellas.

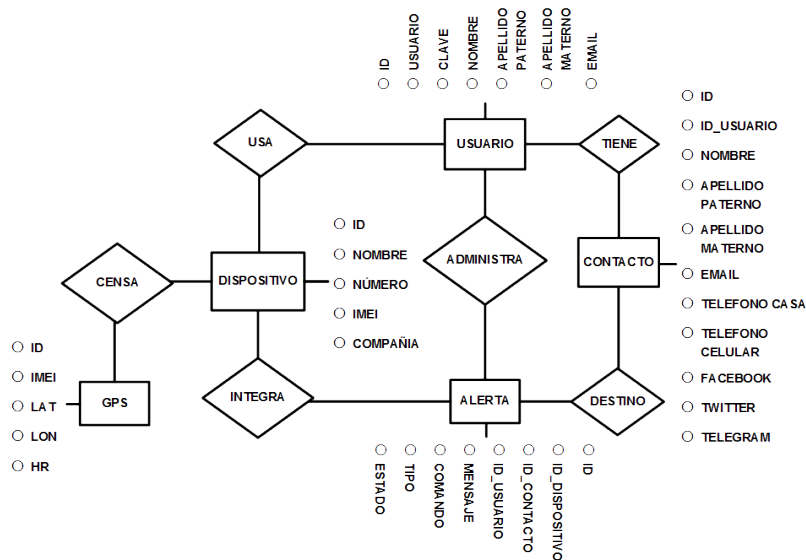


Figura 2.10.2: Modelo conceptual de la base de datos.

2.10.3 MODELO RELACIONAL

Con base en el diccionario de datos anterior, los requerimientos funcionales y el modelo conceptual se presenta el siguiente diagrama entidad relación el cual muestra la forma en la cual se almacenan los datos.

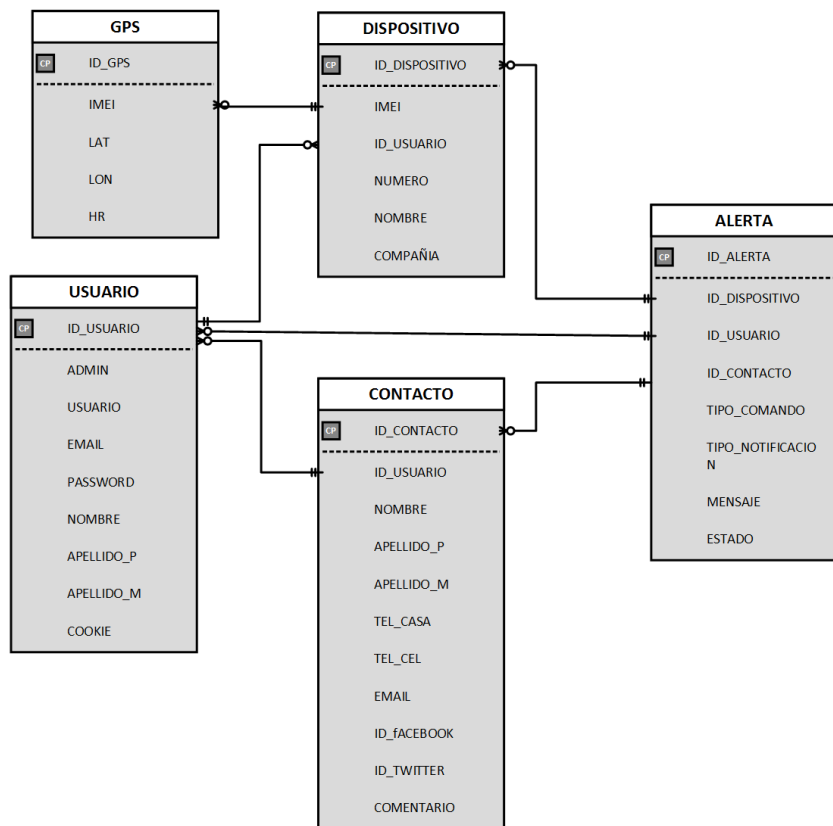


Figura 2.10.2: Diagrama de la base de datos.

2.11 MODELO DEL SISTEMA

La figura 2.11.1 se presenta el modelo del sistema, el cual muestra el funcionamiento general y el flujo de información del sistema:

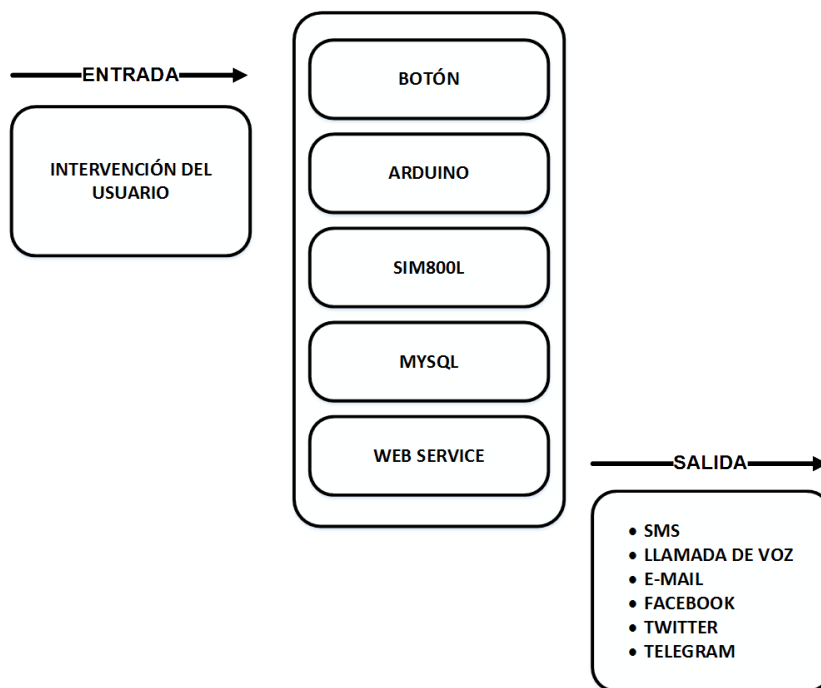


Figura 2.11.1: Modelo del sistema.

2.11.1 ENTRADA

El evento que se tomará como dato de entrada es cuando el usuario presiona el botón del dispositivo físico.

Este evento es capturado y procesado por el dispositivo físico, el cual identifica el tipo de acción a realizar para posteriormente mandarla al web service.

2.11.2 PROCESO

Esta etapa comprende desde la identificación del comando ingresado por parte del dispositivo, este comando es capturado y procesado por el web service, el cual verifica la

autenticidad del dispositivo físico por medio de su IMEI, y realiza una consulta a la base de datos para encontrar las alertas y contactos asignados al comando recibido.

2.11.3 SALIDA

De acuerdo a lo descrito anteriormente en esta etapa se lanza una notificación al contacto seleccionado. Estas notificaciones pueden ser lanzadas desde el dispositivo físico o desde el web server.

Cuando la alerta es por dispositivo el web service responde la petición HTTP con los datos del contacto, posteriormente el dispositivo físico identifica si la alerta se trata de un SMS o una llamada de voz y finalmente la realiza. Las alertas por dispositivo pueden ser 3:

- SMS
- Llamada de voz a domicilio
- Llamada de voz a celular

En caso de que la alerta sea por web service, éste busca la información del contacto y llama a la API del servicio de mensajería asociado, para finalmente lanzarla.

- SMS
- Llamada de voz a domicilio
- Llamada de voz a celular
- Mensaje por Facebook
- Mensaje por Bot de Telegram
- Mensaje por Twitter
- E-mail

CAPITULO 3: DESARROLLO E IMPLEMENTACIÓN

En este capítulo se presenta a detalle el desarrollo del sistema propuesto, comenzando con el dispositivo físico, los aspectos más importantes en su programación así como la comunicación de este con el web server, en la segunda sección se aborda el Web server, el lenguaje de programación y el manejo de la base de datos. Para finalizar se muestra como es la comunicación entre los componentes del sistema.

3.1 DISPOSITIVO FÍSICO

El hardware principal es el Arduino Pro Micro el cual fue descrito en la sección 2.1 del presente trabajo, su microcontrolador atmega32u es el encargado de capturar el comando ingresado por el usuario y posteriormente enviarlo al web service con ayuda del módulo sim800L. A continuación, se describe el hardware y software necesario para la construcción del dispositivo físico.

3.1.1 MATERIALES

La tabla 3.1.1 presenta la lista de los materiales necesarios para la construcción del dispositivo físico, el principal componente es el Módulo SIM800L que permite la comunicación con el web service.

| No. | Elemento | Cantidad |
|-----|-------------------------------|----------|
| 1 | Arduino Pro Micro | 1 |
| 2 | Módulo SIM800L | 1 |
| 3 | Protoboard | 1 |
| 4 | Cables MH | 20 |
| 5 | Baterías Li-ion 18630 de 3.6v | 2 |
| 6 | Resistencia de 10k | 2 |
| 7 | Resistencia de 20k | 2 |
| 8 | Botón | 1 |

Tabla 3.1.1: Materiales necesarios para la construcción del dispositivo.

3.1.2 CONSTRUCCIÓN

En la figura 3.1.2 se muestra el esquema de conexión para el módulo SIM800L y el Arduino Pro Micro, éste circuito permite la configuración del módulo SIM800L a través de los comandos AT para poder conectarse a la red de telefonía celular.

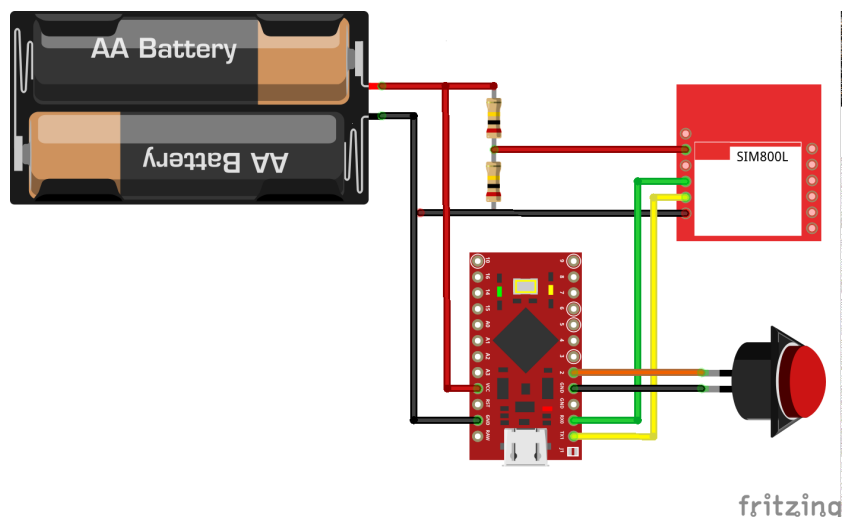


Figura 3.1.1: Esquema de conexión del dispositivo físico.

3.1.3 SOFTWARE

A continuación, en la tabla 3.1.1 se presenta el software utilizado para el diseño y desarrollo del dispositivo físico.

| Tarea | Software |
|--------------------------|-------------|
| Diseño y esquematización | Fritzing |
| Programación | Arduino IDE |

Tabla 3.1.2: Software necesario para el desarrollo del dispositivo físico.

3.1.4 CONFIGURACIÓN DE LAS HERRAMIENTAS

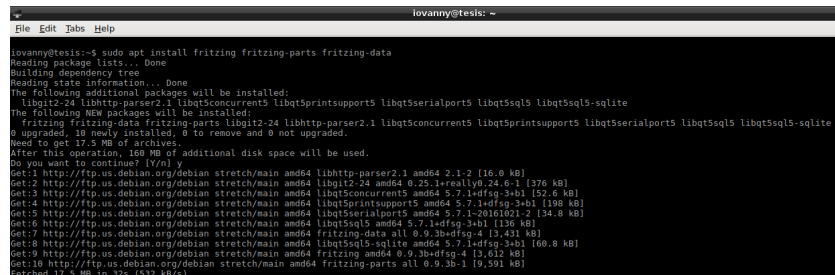
El software necesario para el desarrollo del dispositivo físico se encuentra en los repositorios de la distribución Debian, la instalación de estos paquetes es llevada a cabo mediante el

gestor de paquetes APT. Este gestor funciona por línea de comandos, por lo que se utilizó LXTerminal para esta tarea.

a) Fritzing

Fritzing es un proyecto de código abierto diseñado para ayudar a una transición de un prototipo a un proyecto terminado. Dirigido a usuarios que desean producir o documentar circuitos y experimentos, uno comienza construyendo un prototipo físico y luego recreándolo con el editor gráfico de Fritzing. Desde allí se puede generar un esquema, ilustraciones de PCB y archivos de producción de PCB [35]. A continuación, se presenta la instalación utilizando el gestor de paquetes APT, después en la figura 3.1.2 una captura de pantalla del proceso de instalación, la figura 3.1.3 la captura de la interfaz de usuario y finalmente en la tabla 3.1.3 la descripción de los paquetes necesarios para utilizar Fritzing.

```
$ sudo apt install fritzing fritzing-data fritzing-parts
```



```
lovanny@tesis: ~
File Edit Tabs Help

lovanny@tesis:~$ sudo apt install fritzing fritzing-parts fritzing-data
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgit2-24 libhttp-parser2.1 libqt5concurrent5 libqt5sprintsupport5 libqt5serialport5 libqt5sql5 libqt5sql5-sqlite
The following NEW packages will be installed:
  fritzing fritzing-data fritzing-parts libgit2-24 libhttp-parser2.1 libqt5concurrent5 libqt5sprintsupport5 libqt5serialport5 libqt5sql5 libqt5sql5-sqlite
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 17.5 MB of archives.
After this operation, 160 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 libhttp-parser2.1 amd64 2.1.2 [16.0 kB]
Get:2 http://ftp.us.debian.org/debian stretch/main amd64 libgit2-24 amd64 0.25.1-1really0.24.0-1 [376 kB]
Get:3 http://ftp.us.debian.org/debian stretch/main amd64 libqt5concurrent5 amd64 5.7.1+dfsg-3+b1 [52.6 kB]
Get:4 http://ftp.us.debian.org/debian stretch/main amd64 libqt5sprintsupport5 amd64 5.7.1+dfsg-3+b1 [198 kB]
Get:5 http://ftp.us.debian.org/debian stretch/main amd64 libqt5serialport5 amd64 5.2.1-20161021-2 [34.0 kB]
Get:6 http://ftp.us.debian.org/debian stretch/main amd64 libqt5sql5 amd64 5.7.1+dfsg-3+b1 [136 kB]
Get:7 http://ftp.us.debian.org/debian stretch/main amd64 fritzing-data all 0.9.3b+dfsg-4 [3,431 kB]
Get:8 http://ftp.us.debian.org/debian stretch/main amd64 libqt5sql5-sqlite amd64 5.7.1+dfsg-3+b1 [68.0 kB]
Get:9 http://ftp.us.debian.org/debian stretch/main amd64 fritzing amd64 0.9.3b+dfsg-4 [3,612 kB]
Get:10 http://ftp.us.debian.org/debian stretch/main amd64 fritzing-parts all 0.9.3b-1 [9,591 kB]
Fetched 17.5 MB in 32s (632 kB/s)
```

Figura 3.1.2: Proceso de instalación de Fritzing.

DESARROLLO DE UN BOTÓN DE PÁNICO INTELIGENTE
 CON LA TECNOLOGÍA DE INTERNET DE LAS COSAS

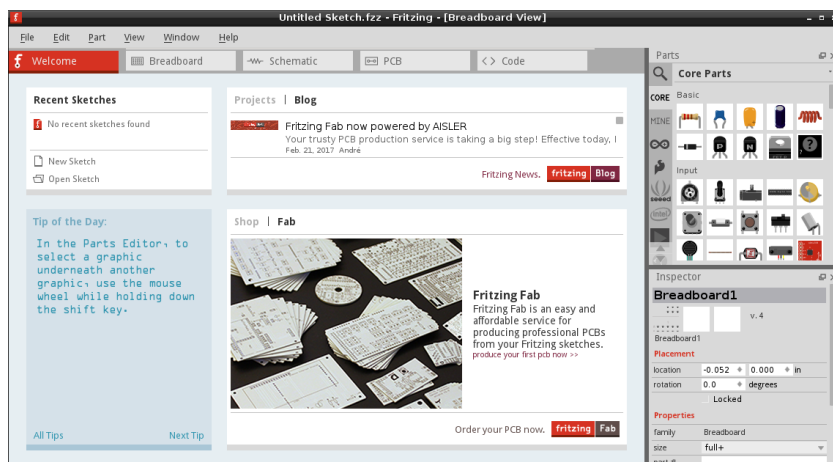


Figura 3.1.3: Ventana principal de Fritzing.

| Paquete | Descripción |
|----------------|--|
| fritzing | Paquete principal fritzing. |
| fritzing-data | Este paquete contiene los archivos de datos independientes de arquitectura para Fritzing. |
| fritzing-parts | Este paquete contiene los archivos de partes independientes de arquitectura para Fritzing. |

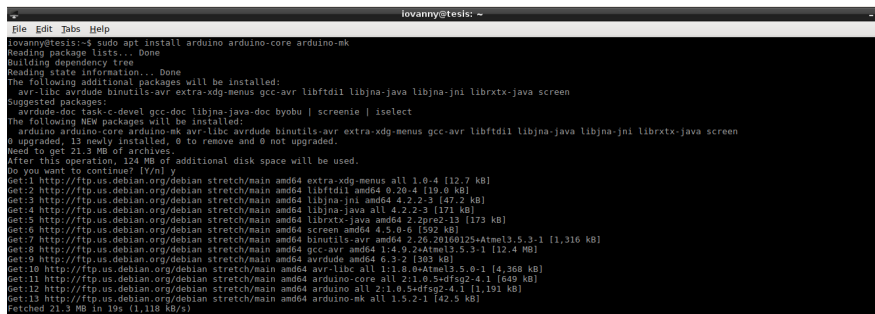
Tabla 3.1.3: Fritzing – Descripción de los paquetes instalados.

b) Arduino IDE

Arduino es una plataforma de creación de prototipos de electrónica de código abierto basada en hardware y software flexible y fácil de usar. Está dirigido a artistas, diseñadores, aficionados y cualquier persona interesada en crear objetos o entornos interactivos [35].

Para utilizarlo se deben instalar los paquetes arduino, arduino-core y arduino-mk utilizando apt, en la tabla 3.1.4 se muestra el proceso de instalación.

```
$ sudo apt install arduino arduino-core arduino-mk
```



```
lovanny@tesis: ~  
File Edit Tabs Help  
lovanny@tesis:~$ sudo apt install arduino arduino-core arduino-mk  
Loading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  avr-libc avrdude binutils-avr extra-xdg-menus gcc-avr libftdi1 libjna-java libjna-jni librx-java screen  
Suggested packages:  
  avrdude-doc task-c-devel gcc-doc libjna-java-doc byobu | screenie | iselect  
The following NEW packages will be installed:  
  arduino arduino-core arduino-mk avr-libc avrdude binutils-avr extra-xdg-menus gcc-avr libftdi1 libjna-java libjna-jni librx-java screen  
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.  
Need to get 22.3 MB of archives.  
After this operation, 124 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 extra-xdg-menus all 1.0-4 [12.7 kB]  
Get:2 http://ftp.us.debian.org/debian stretch/main amd64 libftdi1 amd64 0.20-4 [19.0 kB]  
Get:3 http://ftp.us.debian.org/debian stretch/main amd64 libjna-jni amd64 4.2.2-2 [47.2 kB]  
Get:4 http://ftp.us.debian.org/debian stretch/main amd64 libjna-java all 4.2.2-3 [171 kB]  
Get:5 http://ftp.us.debian.org/debian stretch/main amd64 librx-java amd64 2.2pp62-13 [173 kB]  
Get:6 http://ftp.us.debian.org/debian stretch/main amd64 screen amd64 4.5.0-6 [592 kB]  
Get:7 http://ftp.us.debian.org/debian stretch/main amd64 binutils-avr amd64 2.26.20160125-Atmel3.5.3-1 [1,316 kB]  
Get:8 http://ftp.us.debian.org/debian stretch/main amd64 gcc-avr amd64 1:4.9.2-Atmel3.5.3-1 [12.4 kB]  
Get:9 http://ftp.us.debian.org/debian stretch/main amd64 avrdude amd64 4.3-2 [103 kB]  
Get:10 http://ftp.us.debian.org/debian stretch/main amd64 avr-libc all 1:1.8.0-Atmel3.5.0-1 [4,368 kB]  
Get:11 http://ftp.us.debian.org/debian stretch/main amd64 arduino-core all 2:1.0.5-dfsg2-4.1 [649 kB]  
Get:12 http://ftp.us.debian.org/debian stretch/main amd64 arduino all 2:1.0.5-dfsg2-4.1 [1,191 kB]  
Get:13 http://ftp.us.debian.org/debian stretch/main amd64 arduino-mk all 1.5.2-1 [42.5 kB]  
Fetched 22.3 MB in 19s (1,116 kB/s)
```

Figura 3.1.4: Instalación del IDE Arduino

Una vez que APT termina de instalar y configurar Arduino y todos los paquetes necesarios, el IDE ya está disponible en el menú de Programación de Debian, la interfaz principal del IDE Arduino se muestra en la figura 3.1.5.

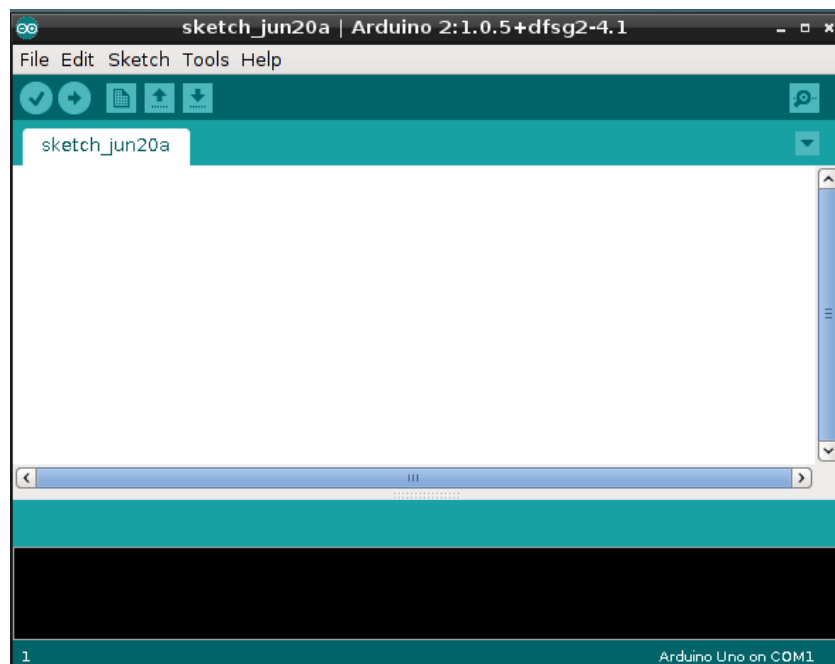


Figura 3.1.5: Ventana principal del IDE Arduino.

| Paquete | Descripción |
|--------------|---|
| arduino | Este paquete instala el entorno de desarrollo integrado que permite la escritura del programa, la verificación del código, la compilación y la carga en la placa de desarrollo Arduino. |
| arduino-core | Este paquete contiene un conjunto mínimo de herramientas que permiten programar un Arduino. También contiene ejemplos y bibliotecas. |
| arduino-mk | Este paquete instalará un Makefile para permitir la programación CLI de la plataforma Arduino. |

Tabla 3.1.4: Arduino – Descripción de los paquetes instalados.

3.1.6 CIRCUITO FINAL

A continuación, en la Figura 3.1.6 se presenta el circuito final del prototipo.

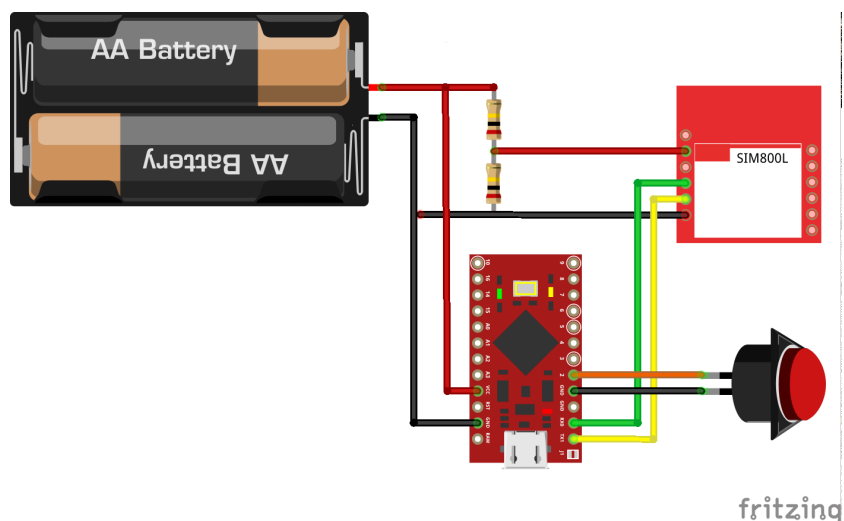


Figura 3.1.6: Circuito Final

3.2 WEB SERVICE

Continuando con el desarrollo del proyecto, se construyó un web service que también es la interfaz principal del usuario, en él, el usuario puede administrar contactos, dispositivos y alarmas, todo esto en una interfaz sencilla y amigable. En esta sección se describe a detalle el proceso de desarrollo.

3.2.1 SOFTWARE

A continuación, en la tabla 3.2.1 se presentan las herramientas utilizadas en el desarrollo del web service:

| Tareas | Herramientas |
|--------------------------------|--|
| Sistema Operativo del servidor | GNU Linux Debian 9 Stretch |
| Servidor Web | Apache Web Server |
| Gestor de Base de Datos | MySQL |
| Front-end | PHP |
| Back-end | Javascript Bootstrap CSS3.0 HTML5 |
| IDE | Sublime-text phpMyAdmin |

Tabla 3.2.1: Herramientas utilizadas en el desarrollo del web service.

3.2.2 CONFIGURACIÓN DE LAS HERRAMIENTAS

El web server funciona en un servidor con sistema operativo Linux, por lo tanto, la instalación de los paquetes necesarios se hizo mediante la terminal de comandos LXTerminal. A continuación, se describe el proceso de instalación y configuración de cada paquete utilizado.

a) Apache Web Server

Para implementar el web service se necesita un servidor web con conexión a internet que esté a la espera de las peticiones que haga el dispositivo físico. Como se mencionó en el capítulo 2 el servidor web Apache es la mejor opción por defecto para implementar un servidor web para contenido estático y dinámico. Apache está disponible en el sistema de gestión de paquetes APT que utiliza Debian lo que simplifica bastante su instalación, ya que APT se encarga de instalar Apache y todas sus dependencias, además de configurar el firewall del servidor para que el servidor web esté disponible al instante. El comando para instalar Apache junto con el módulo para el soporte de PHP 7.0 es el siguiente:

```
$ sudo apt install apache2 libapache2-mod-php7.0
```

El proceso de instalación de Apache y el módulo de soporte de PHP 7.0 se presenta en la figura 3.2.1:

```
lovanny@otis1:~$ sudo apt install apache2 libapache2-mod-php7.0
[sudo] password for lovanny:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap php-common php7.0-cli php7.0-common php7.0-json php7.0-opcache
  php7.0-readline ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.0 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap php-common php7.0-cli
  php7.0-common php7.0-json php7.0-opcache php7.0-readline ssl-cert
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 5.542 kB of archives.
After this operation, 29.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://security.debian.org/debian-security stretch/updates/main amd64 apache2-bin amd64 2.4.25-3+deb9u4 [1,183 kB]
Get:2 http://security.debian.org/debian-security stretch/updates/main amd64 apache2-utils amd64 2.4.25-3+deb9u4 [1217 kB]
Get:3 http://security.debian.org/debian-security stretch/updates/main amd64 apache2-data all 2.4.25-3+deb9u4 [1602 kB]
Get:4 http://security.debian.org/debian-security stretch/updates/main amd64 apache2 amd64 2.4.25-3+deb9u4 [236 kB]
Get:5 http://ftp.us.debian.org/debian stretch/main amd64 libapr1 amd64 1.5.2-5 [96.6 kB]
Get:6 http://ftp.us.debian.org/debian stretch/main amd64 libaprutil1 amd64 1.5.4-3 [165.8 kB]
Get:7 http://ftp.us.debian.org/debian stretch/main amd64 libaprutil1-dbd-sqlite3 amd64 1.5.4-3 [19.3 kB]
Get:8 http://ftp.us.debian.org/debian stretch/main amd64 libaprutil1-ldap amd64 1.5.4-3 [17.4 kB]
Get:9 http://ftp.us.debian.org/debian stretch/main amd64 php-common all 1:49 [14.1 kB]
Get:10 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-common amd64 7.0.27-0+deb9u1 [877 kB]
Get:11 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-json amd64 7.0.27-0+deb9u1 [17.6 kB]
Get:12 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-opcache amd64 7.0.27-0+deb9u1 [77.5 kB]
Get:13 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-readline amd64 7.0.27-0+deb9u1 [132.7 kB]
Get:14 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-cli amd64 7.0.27-0+deb9u1 [1,284 kB]
Get:15 http://ftp.us.debian.org/debian stretch/main amd64 libapache2-mod-php7.0 amd64 7.0.27-0+deb9u1 [1,224 kB]
Get:16 http://ftp.us.debian.org/debian stretch/main amd64 ssl-cert all 1.0.39 [20.8 kB]
Fetched 5.542 kB in 13s (369 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1:amd64.
Reading database ... 18279 files and directories currently installed.
Preparing to unpack .../00-libapr1_1.5.2-5_amd64.deb ...
Unpacking libapr1:amd64 (1.5.2-5) ...
Selecting previously unselected package libaprutil1:amd64.
```

Figura 3.2.1: Apache Web Server – Proceso de instalación.

En la tabla 3.2.1 se hace una descripción de los paquetes necesarios para poder habilitar Apache web server el servidor:

| Paquete | Descripción |
|-----------------------|----------------------------------|
| apache2 | Servidor web Apache |
| libapache2-mod-php7.0 | Soporte para el lenguaje PHP 7.0 |

Tabla 3.2.1: Servidor web Apache - Descripción de los paquetes instalados.

Si la instalación se realizó correctamente al ingresar en el navegador la dirección <http://localhost> se visualizará la página de bienvenida del servidor web Apache, como se muestra en la figura 3.2.2:

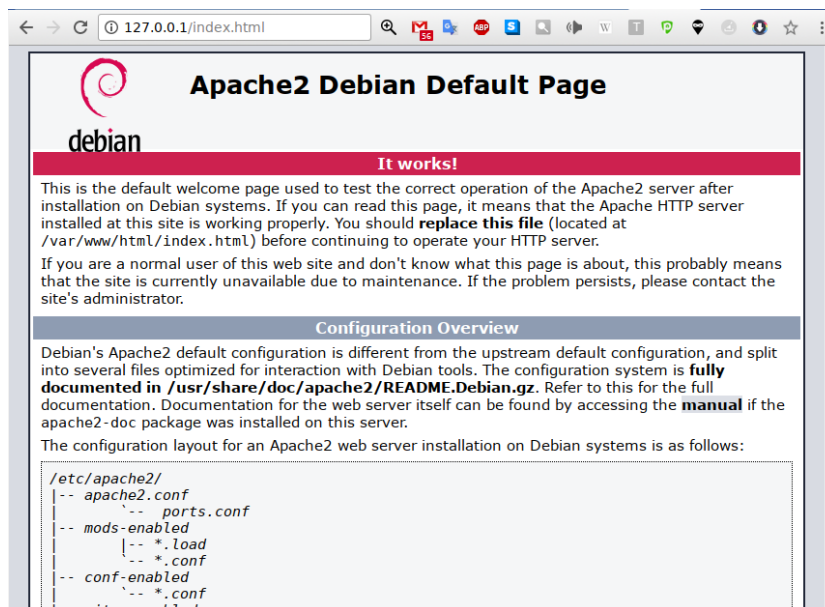


Figura 3.2.2: Apache Web Server – Pagina de bienvenida.

b) PHP

La instalación del intérprete del lenguaje de programación PHP en su versión 7.0 y su conector para MySQL se realiza instalando los siguientes paquetes con APT:

```
$ sudo apt install php7.0 php7.0-mysql
```

El proceso de instalación se muestra en la figura 3.2.3:

```
iovanny@tesis:~$ sudo apt install php7.0 php7.0-mysql
[sudo] password for iovanny:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  php7.0 php7.0-mysql
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 177 kB of archives.
After this operation, 568 kB of additional disk space will be used.
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 php7.0 all 7.0.27-0+deb9u1 [52.9 kB]
Get:2 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-mysql amd64 7.0.27-0+deb9u1 [124 kB]
Fetched 177 kB in 0s (277 kB/s)
Selecting previously unselected package php7.0.
(Reading database ... 139613 files and directories currently installed.)
Preparing to unpack ../php7.0_7.0.27-0+deb9u1_all.deb ...
Unpacking php7.0 (7.0.27-0+deb9u1) ...
Selecting previously unselected package php7.0-mysql.
Preparing to unpack ../php7.0-mysql_7.0.27-0+deb9u1_amd64.deb ...
Unpacking php7.0-mysql (7.0.27-0+deb9u1) ...
Setting up php7.0-mysql (7.0.27-0+deb9u1) ...

Creating config file /etc/php/7.0/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/7.0/mods-available/mysqli.ini with new version
Creating config file /etc/php/7.0/mods-available/pdo_mysql.ini with new version
Setting up php7.0 (7.0.27-0+deb9u1) ...
Processing triggers for libapache2-mod-php7.0 (7.0.27-0+deb9u1) ...
```

Figura 3.2.3: PHP – Proceso de instalación.

En la tabla 3.2.2 se describen los paquetes necesarios para poder ejecutar PHP en el servidor.

| Paquete | Descripción |
|--------------|--|
| php7.0 | Interprete del lenguaje PHP del lado del servidor, Instala todas las dependencias necesarias para la ejecución de PHP. |
| php7.0-mysql | Provee el módulo MySQL para PHP. Permite que PHP se pueda conectar de forma local o remota a una base de datos MySQL |

Tabla 3.2.2: PHP - Descripción de los paquetes instalados.

c) MySQL

Para poder gestionar la información de los usuarios en el web server se utilizó MySQL Server, su instalación, de la misma forma que el paquete anterior se realizó por medio del gestor de paquetes APT, los comandos para su instalación y configuración se describen a continuación:

```
$ sudo apt install mysql-client mysql-server
```

El proceso de instalación se presenta en la figura 3.2.4 y posteriormente en la figura 3.2.5 se muestra la sesión iniciada por medio de mysql-client.

```
lovanny@tesis:~$ sudo apt install mysql-client mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  mysql-client mysql-server
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 3,484 B of archives.
After this operation, 16.4 kB of additional disk space will be used.
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 mysql-client amd64 5.5.9999+default [1,698 B]
Get:2 http://ftp.us.debian.org/debian stretch/main amd64 mysql-server amd64 5.5.9999+default [1,786 B]
Fetched 3,484 B in 0s (9,729 B/s)
Selecting previously unselected package mysql-client.
(Reading database ... 140461 files and directories currently installed.)
Preparing to unpack .../mysql-client_5.5.9999+default_amd64.deb ...
Unpacking mysql-client (5.5.9999+default) ...
Selecting previously unselected package mysql-server.
Preparing to unpack .../mysql-server_5.5.9999+default_amd64.deb ...
Unpacking mysql-server (5.5.9999+default) ...
Setting up mysql-server (5.5.9999+default) ...
Setting up mysql-client (5.5.9999+default) ...
lovanny@tesis:~$
```

Figura 3.2.4: MySQL server – Proceso de instalación.

```
root@tesis:/home/lovanny# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.1.26-MariaDB-0+deb9u1 Debian 9.1

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

MariaDB [(none)]>
```

Figura 3.2.5: MySQL (MariaDB) – Sesión activa.

En la tabla 3.2.3 se describen los paquetes necesarios para habilitar el servidor de MySQL y configurar las bases de datos.

| Paquete | Descripción |
|--------------|---|
| mysql-client | Este paquete incluye los binarios del cliente y las herramientas adicionales innotop y mysqlreport. |
| mysql-server | Este paquete contiene toda la infraestructura necesaria para configurar las bases de datos del sistema. |

Tabla 3.2.3: MySQL - Descripción de los paquetes instalados.

d) phpMyAdmin

phpMyAdmin es una herramienta escrita en PHP que permite gestionar MySQL desde una interfaz web, permite agregar, modificar y eliminar registros, tablas y bases de datos en un entorno visual sin necesidad de ingresar instrucciones por línea de comandos.

```
$ sudo apt install phpmyadmin
```

A continuación, en la figura 3.2.6 se presenta una captura del proceso de instalación y posteriormente en la figura 3.2.7 una captura de pantalla de la interfaz de phpMyAdmin.

```
lovanny@tesis16:~$ sudo apt install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libcurl3 libjs-jquery libjs-sphinxdoc libjs-underscore libzip4 php-bz2 php-c
  php-pear php-php-gettext php-phpseclib php-tcpdf php-xml php-zip php7.0-bz2 php7.0-curl php7.0-gd php7.0-mbstring php7.0-xml
Suggested packages:
  php-libzstd php-mcrypt php-gmp php5-imagick
Recommended packages:
  php5-gd php5-mcrypt
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql javascript-common libcurl3 libjs-jquery libjs-sphinxdoc libjs-underscore libzip4 php-bz2 php-c
  php-pear php-php-gettext php-phpseclib php-tcpdf php-xml php-zip php7.0-bz2 php7.0-curl php7.0-gd php7.0-mbstring php7.0-xml
0 upgraded, 26 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.2 MB of archives.
After this operation, 54.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 dbconfig-common all 2.0.8 [617 kB]
Get:2 http://security.debian.org/debian-security stretch/updates/main amd64 libcurl3 amd64 7.52.1-5+deb9u6 [291 kB]
Get:3 http://ftp.us.debian.org/debian stretch/main amd64 dbconfig-mysql all 2.0.8 [998 B]
Get:4 http://ftp.us.debian.org/debian stretch/main amd64 javascript-common all 11 [6,120 B]
Get:5 http://ftp.us.debian.org/debian stretch/main amd64 libjs-jquery all 3.1.1-2 [154 kB]
Get:6 http://ftp.us.debian.org/debian stretch/main amd64 libjs-underscore all 1.8.3-dfsg-1 [63.8 kB]
Get:7 http://ftp.us.debian.org/debian stretch/main amd64 libjs-sphinxdoc all 1.4.9-2 [69.5 kB]
Get:8 http://ftp.us.debian.org/debian stretch/main amd64 libzip4 amd64 1.1.2-1.1+b1 [40.6 kB]
Get:9 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-bz2 amd64 7.0.27-0+deb9u1 [9,982 B]
Get:10 http://ftp.us.debian.org/debian stretch/main amd64 php-bz2 all 1:7.0+49 [5,048 B]
Get:11 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-curl amd64 7.0.27-0+deb9u1 [27.4 kB]
Get:12 http://ftp.us.debian.org/debian stretch/main amd64 php-curl all 1:7.0+49 [5,048 B]
Get:13 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-gd amd64 7.0.27-0+deb9u1 [27.3 kB]
Get:14 http://ftp.us.debian.org/debian stretch/main amd64 php-gd all 1:7.0+49 [5,042 B]
Get:15 http://ftp.us.debian.org/debian stretch/main amd64 php7.0-mbstring amd64 7.0.27-0+deb9u1 [465 kB]
Get:16 http://ftp.us.debian.org/debian stretch/main amd64 php-mbstring all 1:7.0+49 [5,058 B]
Get:17 http://ftp.us.debian.org/debian stretch/main amd64 php-xml all 1:7.0+49 [5,052 B]
```

Figura 3.2.6: phpMyAdmin – Proceso de instalación.

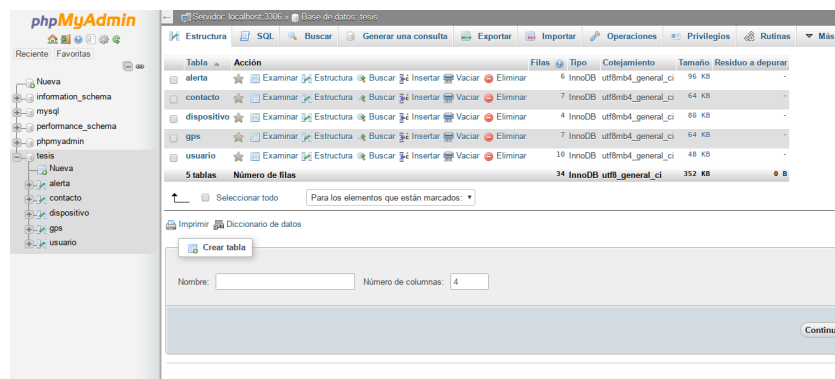


Figura 3.2.7: phpMyAdmin – Interfaz principal.

En la tabla 3.2.4 se describe el paquete phpmyadmin.

| Paquete | Descripción |
|------------|---|
| phpmyadmin | Este paquete permite la administración de MySQL o MariaDB con una interfaz web. |

Tabla 3.2.4: phpMyAdmin - Descripción de los paquetes instalados.

e) Sublime-text

Sublime-text a diferencia de los paquetes anteriores no está disponible en el gestor de paquetes APT, para instalarlo es necesario agregar los repositorios del sitio oficial, la secuencia de este proceso se presenta a continuación:

1.- Se instala la llave GPG:

```
$ wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add -
```

2.- Se verifica APT pueda trabajar con el protocolo HTTPS:

```
$ sudo apt install apt-transport-https
```

3.- Se agrega el canal Stable:

```
$ echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list
```

4.- Se actualizan los paquetes y se instala Sublime-text:

```
$ sudo apt update
```

```
$ sudo apt install sublime-text
```

En la figura 3.2.8 se muestra el proceso de instalación y en la figura 3.2.9 la interfaz principal:

```
lovanny@tesis:~$ wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add -
[sudo] password for iovanny:
OK
lovanny@tesis:~$ sudo apt install apt-transport-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 171 kB of archives.
After this operation, 243 kB of additional disk space will be used.
Get:1 http://ftp.us.debian.org/debian stretch/main amd64 apt-transport-https amd64 1.4.8 [171 kB]
Fetched 171 kB in 0s (228 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 145119 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.4.8_amd64.deb ...
Unpacking apt-transport-https (1.4.8) ...
Setting up apt-transport-https (1.4.8) ...
lovanny@tesis:~$ echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list
deb https://download.sublimetext.com/ apt/stable/
lovanny@tesis:~$ sudo apt install sublime-text
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sublime-text
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 8,494 kB of archives.
After this operation, 25.1 MB of additional disk space will be used.
Get:1 https://download.sublimetext.com apt/stable/ sublime-text 3176 [8,494 kB]
Fetched 8,494 kB in 15s (535 kB/s)
Selecting previously unselected package sublime-text.
(Reading database ... 143125 files and directories currently installed.)
Preparing to unpack .../sublime-text_3176_amd64.deb ...
Unpacking sublime-text (3176) ...
Processing triggers for mime-support (3.60) ...
Processing triggers for desktop-file-utils (0.23-1) ...
Processing triggers for hicolor-icon-theme (0.15-1) ...
Setting up sublime-text (3176) ...
lovanny@tesis:~$
```

Figura 3.2.8: Sublime-text – Proceso de instalación.

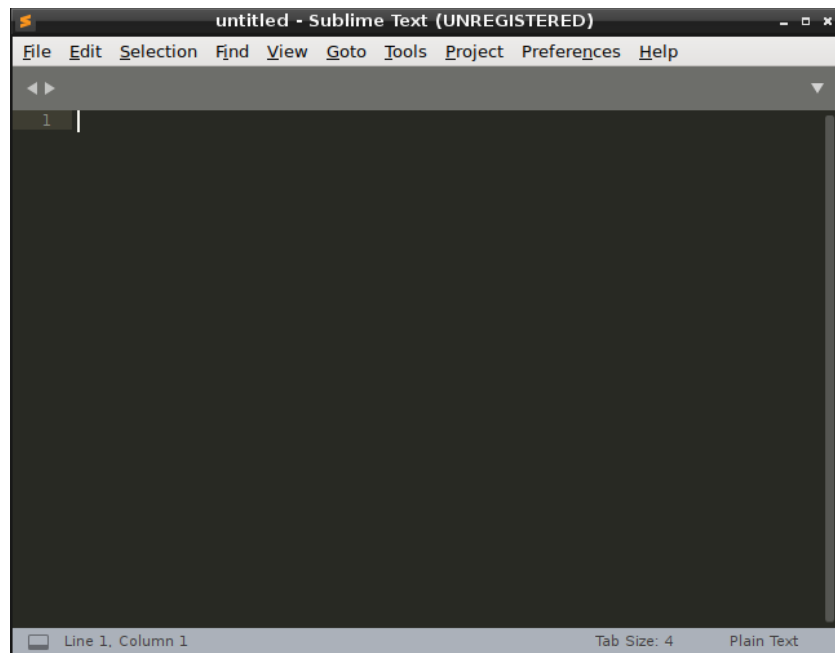


Figura 3.2.9: Sublime-text – Interfaz principal.

En la tabla 3.2.5 se describen los paquetes necesarios para instalar y utilizar Sublime-text.

| Paquete | Descripción |
|---------------------|---|
| apt-transport-https | Este paquete permite el uso de las líneas 'deb https: // foo distro main' en la /etc/apt/sources.list para que todos los gestores de paquetes que utilizan la biblioteca libapt-pkg puedan acceder a metadatos y paquetes disponibles en fuentes accesibles a través de https (Protocolo de transferencia de hipertexto seguro). |
| Sublime-text | Editor de texto y código, escrito en C++. |

Tabla 3.2.5: Sublime-text - Descripción de los paquetes instalados.

3.2.3 FRONT-END

Un aspecto importante del web service es la interfaz de usuario, para esta tarea se usaron las herramientas descritas en la tabla 3.2.1, cada interfaz se realizó para una función en específico. A continuación se presentan cada una de ellas.

En la figura 3.2.10 se muestra la página principal de la interfaz de usuario, en donde el visitante puede registrarse o iniciar sesión.



Figura 3.2.10: Web service – index.

En la figura 3.2.11 se muestra la página de registro de usuario.

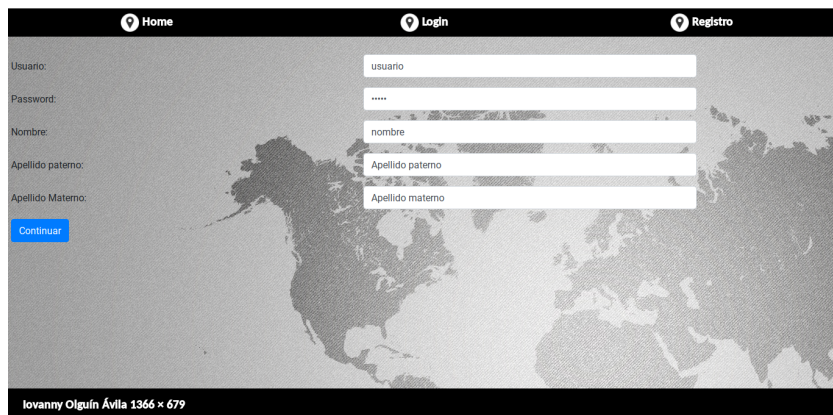


Figura 3.2.11: Web service – registro.

En la figura 3.2.12 se presenta la página de inicio de sesión o login, en donde usuarios y administradores pueden acceder al sistema ingresando su usuario y password.

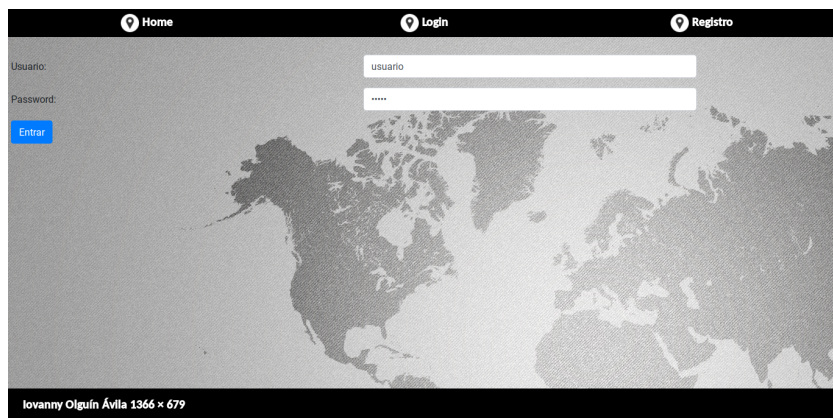


Figura 3.2.12: Web service – login de usuario.

En la figura 3.2.13 se presenta la página de administración de contactos, en donde por medio de las opciones del menú “contactos” en el menú superior, el usuario puede crear, editar y eliminar sus contactos.

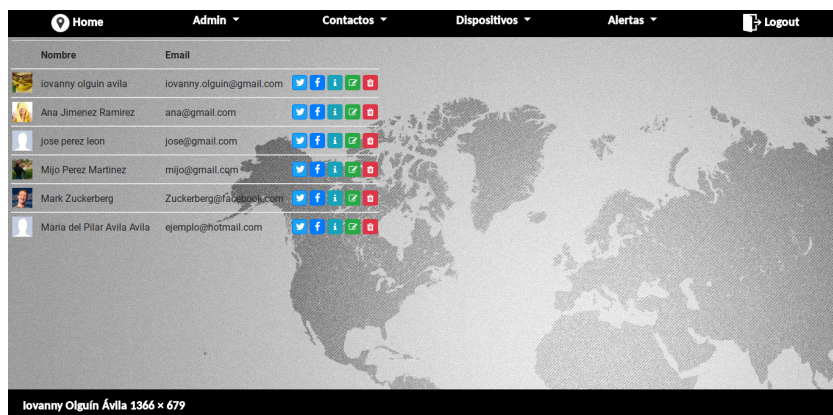


Figura 3.2.13: Web service – Administración de contactos.

En la figura 3.2.14 se presenta la página de administración de dispositivos, en donde por medio de las opciones del menú “dispositivos” situado en la parte superior, el usuario puede crear, editar y eliminar la información de sus dispositivos.

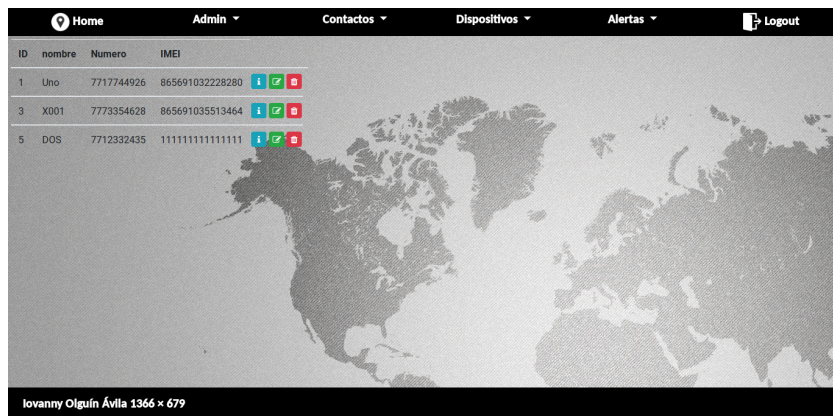


Figura 3.2.14: Web service – Administración de dispositivos.

En la figura 3.2.15 se presenta la página de administración de alertas, donde mediante las opciones del menú “alertas”, el usuario puede crear, editar, probar y eliminar sus alertas.

| ID | Dispositivo | Número | Contacto | Notificación | Mensaje | Tipo | Estado |
|----|-------------|------------|-----------------|--------------|--|------|----------|
| 1 | X001 | 7773354628 | iovanny | LLAMADA_CEL | Ayuda, tuve un problema. Marcame a mi telefono | 1 | Activo |
| 2 | X001 | 7773354628 | iovanny | EMAIL | Llama al Doctor, tuve un problema | 2 | Inactivo |
| 9 | X001 | 7773354628 | Mark | Twitter | Me estoy sintiendo mal, ayúdame! | 1 | Activo |
| 10 | DOS | 7712332435 | Mark | Telegram | Tengo problemas, ayúdame! | 2 | Inactivo |
| 11 | X001 | 7773354628 | Maria del Pilar | Twitter | Hello my name is iovanny | 1 | Activo |

Figura 3.2.15: Web service – Administración de alertas.

En la figura 3.2.16 se presenta la página de administración de alertas, en donde, mediante las opciones del menú superior, el usuario puede crear, editar, probar y eliminar sus alertas.

| ID | Dispositivo | Número | Contacto | Notificación | Mensaje | Tipo | Estado |
|----|-------------|------------|-----------------|--------------|--|------|----------|
| 1 | X001 | 7773354628 | iovanny | LLAMADA_CEL | Ayuda, tuve un problema. Marcame a mi telefono | 1 | Activo |
| 2 | X001 | 7773354628 | iovanny | EMAIL | Llama al Doctor, tuve un problema | 2 | Inactivo |
| 9 | X001 | 7773354628 | Mark | Twitter | Me estoy sintiendo mal, ayúdame! | 1 | Activo |
| 10 | DOS | 7712332435 | Mark | Telegram | Tengo problemas, ayúdame! | 2 | Inactivo |
| 11 | X001 | 7773354628 | Maria del Pilar | Twitter | Hello my name is iovanny | 1 | Activo |

Figura 3.2.16: Web service – Página de localización GPS.

3.3 BASE DE DATOS

3.3.1 ESQUEMA FÍSICO

Como se muestra en la figura 3.3.1 se diseñó el esquema físico de la base de datos a partir de la lista descriptiva de objetos y asociaciones identificadas usando para ello MySQL como motor de base de datos

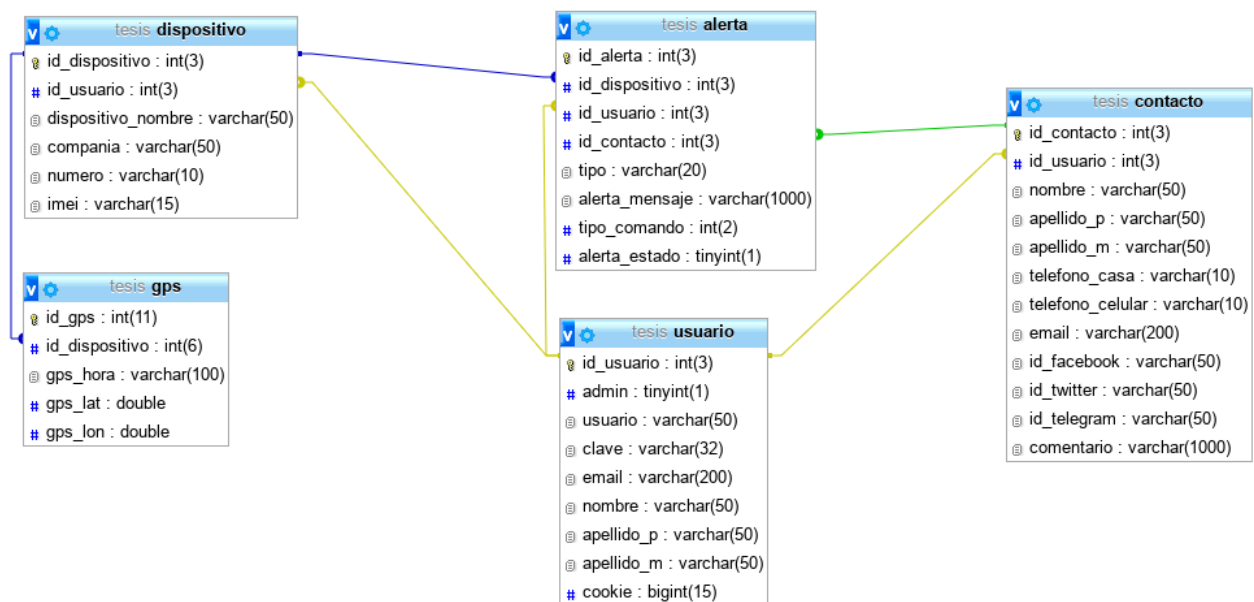


Figura 3.3.1: Esquema físico de la base de datos.

3.4 COMUNICACIÓN

3.4.1 INTRODUCCIÓN

En esta sección se explica cómo se realizó la comunicación entre los componentes del sistema, se detalla cómo es que el dispositivo físico logra conectarse a internet y hacer la petición HTTP al web server, que a su vez incorpora una API RESTFUL para procesar las peticiones.

3.4.2 HTTP POST

En esta sección se describe cada paso involucrado en el envío de información por parte del dispositivo físico hacia el web server, el proceso completo consta de 5 etapas.

- Captura de clicks de usuario
- Petición HTTP POST.
- Recepción de información
- Envío de notificaciones.
- Actualización de ubicación por GPS.

El funcionamiento básico del dispositivo físico radica en un pequeño programa que se ejecuta infinitamente, el cual espera que el usuario presione el botón de pánico, cuando llega este evento se pone en marcha un algoritmo que captura el número de veces que el usuario presiona el botón, cada que el botón está presionado se toma el tiempo y se suma a un total, que al llegar al límite de 2 segundos guarda el número de clicks y lo envía al web service. A continuación, en la figura 3.4.1 se presenta el funcionamiento del algoritmo que captura los clicks del usuario.

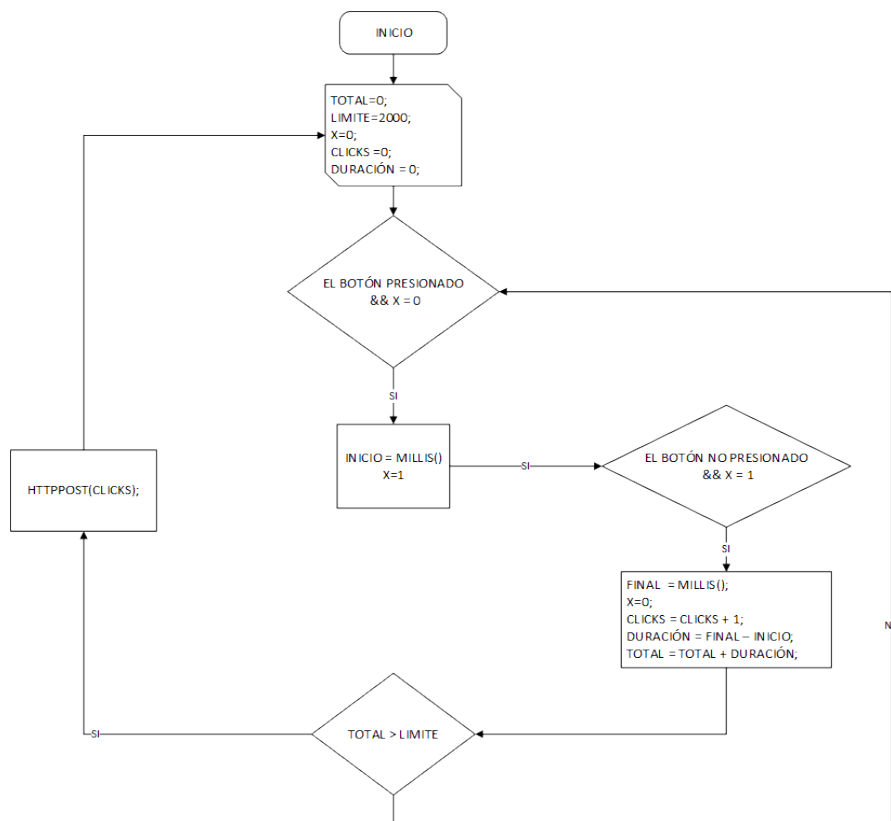


Figura 3.4.1: Diagrama del algoritmo que captura los clicks del usuario.

Cuando es capturado el número de clicks se tienen listos los 2 valores necesarios para hacer la petición HTTP al web server:

- IMEI
- Número de clicks

El paso siguiente es realizar la petición, esto se consigue ingresando una serie de comandos AT para que el módulo SIM800L realice la conexión, envío y recepción de información, a continuación, en la tabla 3.4.1 se presenta la serie de comandos necesarios.

| COMANDO | FUNCIÓN |
|--|--|
| At+Cgatt? | Consulta el estado de servicio. |
| At+Cstt="Apn","User","Pass" | Configura el APN. |
| At+Ciicr | Abre la conexión inalámbrica. |
| At+Cifsr | Obtiene la dirección IP. |
| At+Cipstatus | Obtiene el estado de la conexión. |
| At+Cipstart="Tcp\","ejemplo.com\ ,80 | Inicia una conexión TCP a la dirección remota, en el puerto 80. |
| At+Cipsend=55 | Prepara el envío de 55 bytes. |
| Post Http://ejemplo.com/C/X/Imei Http/1.0 | Realiza la conexión utilizando el método post. C : indica al web service que se trata de un comando. x : número de clicks (opción de comando). imei : número único que identifica al módulo, necesario para autenticarlo con el web service. |
| At+Cipshut | Termina la conexión. |

Tabla 3.4.1: HTTP POST – Secuencia de comandos AT.

En la figura 3.4.2 se presenta el diagrama de flujo correspondiente al envío de todas las instrucciones mencionadas en la Tabla 3.4.1, y en la figura 3.4.3 se presenta el diagrama de flujo de el algoritmo que ingresa las instrucciones individualmente.

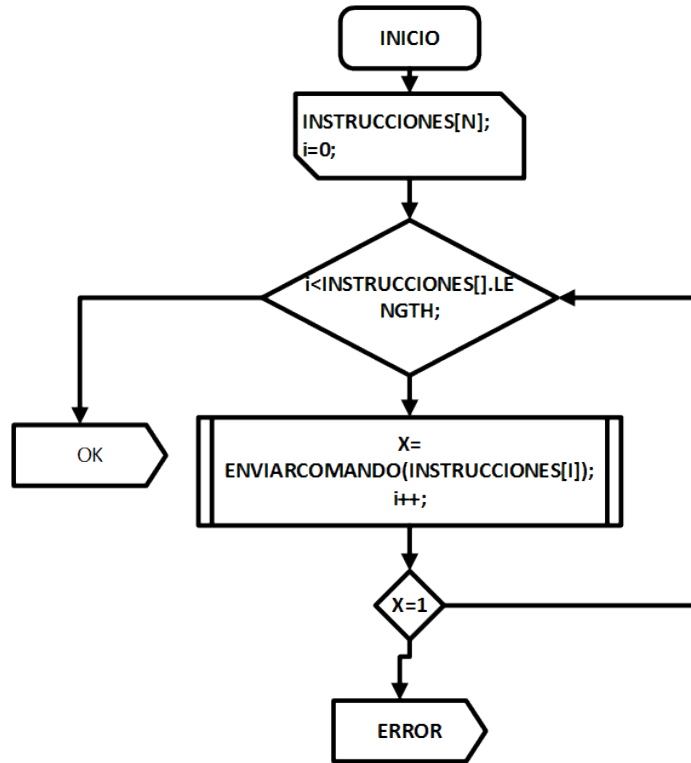


Figura 3.4.2: Diagrama de flujo de la petición HTTP.

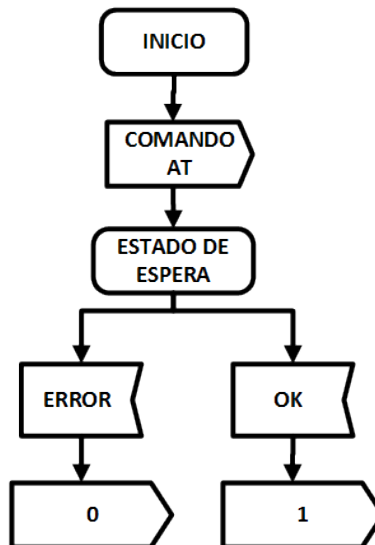


Figura 3.4.3: Diagrama de flujo del envío individual de un comando AT.

Para finalizar esta sección se incluye el código correspondiente a la petición HTTP POST, Este método realiza una petición HTTP y posteriormente guarda la respuesta recibida en una variable, se han eliminado algunos fragmentos para facilitar su lectura.

```
void httpPost(String &respuesta){
    SIM800L.println("ATE1Q0");
    SIM800L.println("AT+CIPSHUT");
    delay(2000);
    SIM800L.println("AT+CSTT=\"internet.movistar.com.mx\", \"movistar\", \"movistar\"");
    delay(2000);
    SIM800L.println("AT+CIICR");
    delay(5000);
    SIM800L.println("AT+CIFSR");
    delay(2000);
    SIM800L.println("AT+CIPSTART=\"TCP\", \"iovanny.tk\", 80");
    delay(3000);
    SIM800L.println("AT+CIPSEND=64\r\n");
    delay(3000);
    unsigned long previous;
    while( Serial.available() > 0) Serial.read();
    previous = millis();
    SIM800L.println("POST http://iovanny.tk/c/7717744926/" + imei + " HTTP/1.0\r\n");
    do{ valor += SIM800L.readString();
    }
    while(((millis() - previous) < 30000));
    delay(500);
    respuesta = valor;
    SIM800L.println("AT+CIPSHUT");
    delay(1000); }
```

3.4.3 DIAGRAMAS DE FUNCIONAMIENTO

Como se explicó en la sección 2.11 el sistema es capaz de enviar dos tipos de notificaciones, por MCU y por Web service, además de actualizar una página con la localización del usuario por medio de GPS, En esta sección se presentan los diagramas de secuencia correspondientes.

a) Diagrama de secuencia de la notificación por MCU.

En la figura 3.4.4 se presenta el diagrama de secuencia correspondiente a las notificaciones enviadas por el Microcontrolador Arduino Pro Micro, en este caso el web server responde con los datos del contacto y el mensaje que debe enviarse.

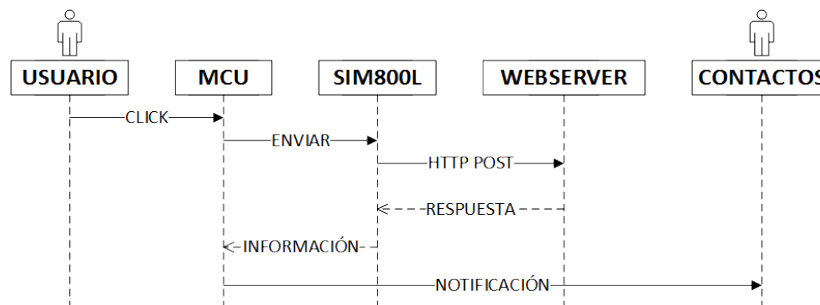


Figura 3.4.4: Diagrama de secuencia de la notificación por MCU.

b) Diagrama de secuencia de la notificación por Web Service

En la Figura 3.4.5 se presenta el diagrama de secuencia de las notificaciones por web service, en este caso el web service identifica que tipo de notificación se debe realizar y la lanza. Las opciones de notificación por web service están descritas en la sección 2.11.

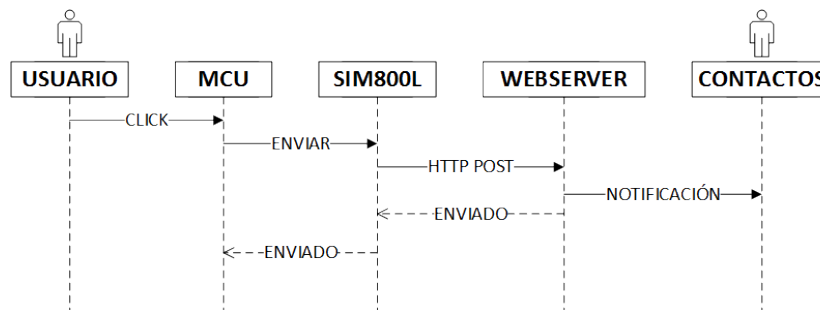


Figura 3.4.5: Diagrama de secuencia de la notificación por Web Service.

c) Diagrama de Localización por GPS.

Finalmente, en la figura 3.4.6 se presenta el diagrama de secuencia de la actualización de la localización GPS del usuario, este ciclo se ejecuta hasta que el usuario desactive la notificación por medio de la interfaz web o si se termina la energía de la batería del dispositivo.

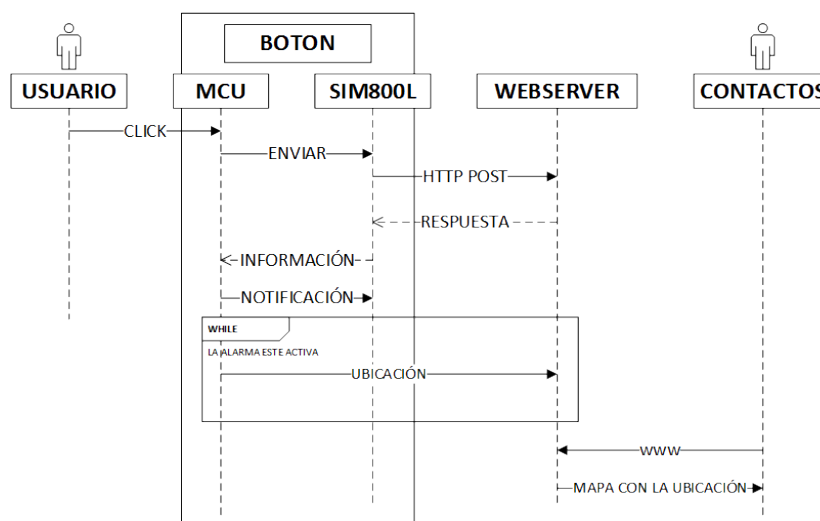


Figura 3.4.6: Diagrama de Localización por GPS.

CAPITULO 4

CONCLUSIONES

El internet de las cosas ofrece una infinidad de posibilidades para acceder y administrar cualquier tipo de dispositivo de forma remota, diariamente nacen nuevas soluciones que incorporan esta tecnología gracias a que su uso es gratuito y a la gran cantidad de información que existe en Internet.

En esta tesis se diseñó y desarrolló un sistema que integra un microcontrolador, un módulo GSM y un web service para ofrecer un servicio de botón de pánico inteligente, configurable mediante una interfaz web. Su uso es de gran utilidad ante situaciones de riesgo, como lo son robos, asaltos, secuestros y emergencias médicas.

En cuanto al proceso de diseño y desarrollo se concluyen los siguientes puntos:

- Se puede hacer mucho con los recursos reducidos que ofrece un microcontrolador como un Attiny85 o un Arduino Pro Micro, teniendo el conocimiento del software, hardware y programación requerida para lograr conectarlo con entidades más grandes, como un Webservice en la nube.
- Con la estructura construida en este trabajo, se pueden incorporar una gran cantidad de sensores en el dispositivo físico,.
- De la experiencia en el desarrollo del web service, adquirí conocimiento para poder implementar la arquitectura REST en proyectos futuros, ya que el objetivo de esta arquitectura es la simplicidad y facilidad de uso, sin duda se convertirá en uno de los grandes estandares para la comunicación de elementos en el Internet de las cosas.
- El servicio se puede ampliar incorporando en el web service las API que ofrecen los servicios de Internet, sin necesidad de modificar o rediseñar el dispositivo físico.
- Gracias a la conectividad del dispositivo físico se puede integrar perfectamente al celular, lo que permitiría manipular de forma remota cualquier smartphone.

REFERENCIAS

- [1] Uckelmann, D., Harrison, M., & Michahelles, F. (Eds.). (2011). *Architecting the Internet of Things*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- [2] Bhatt, C., Dey, N., & Ashour, A. S. (Eds.). (2017). *Internet of Things and Big Data Technologies for Next Generation Healthcare (Vol. 23)*. Cham: Springer International
- [3] Mattern, F., & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. In *From active data management to event-based systems and more* (pp. 242-259). Springer Berlin Heidelberg.
- [4] Vishwanath, S., Vaidya, K., Nawal, R., Kumar, A., Parthasarathy, S., and Verma, S. (2012), Touching lives through mobile health: Assessment of the global market opportunity, GSMA / PwC Report, available online at <http://www.gsma.com/documents/gsma-pwcreport-touching-lives-through-mobile-health-assessment-of-the-global-marketopportunity/22215>, last retrieved April 26, 2012.
- [5] GSMA, (2015), Realising the benefits of mobileenabled IoT solutions,GSMA / PwC Report, available online at <https://www.gsma.com/iot/pwc-report-realising-the-benefits-of-mobile-iot-solutions/>
- [6] de la Cruz, Á. P., Villalba, R. L., Castellanos, S., Guerrero, A., Pinilla, L. P., Castellanos, L., & Ariza, A. P. G. (2004). Evolución de los sistemas móviles celulares GSM 13.
- [7] Tripathi, N., & Reed, J. H. (2014). *Cellular communications: a comprehensive and practical guide*. John Wiley & Sons.
- [8] Barrera Vargas, E., & Ortega Almeida, R. D. (2017). *Análisis de cambios en características fundamentales de la telefonía celular desde 1G hasta 5G*. Chicago
- [9] Rodríguez-López, M. L. (2017). *Metodología para describir servicios RESTful consumidos automáticamente por clientes máquina (Master's thesis)*.
- [10] Punait, S., & Askhedkar, A. R. (2017). *Improved Wearable Personal Security System*.

- [11] Reddy, A. U. K., Sushmitha, P., Gayathri, I., Sandhya, K., & Suresh, N. (2017). Self Defense System for Women Safety with Location Tracking and SMS Alerting.
- [12] "S911 Series – Laypac Technology." [Online]. Available: <http://www.laipac.com/personal-locators.html>[Accessed: 01 - Dic – 2017].
- [13] "Nimb: Smart ring with a panic button. - Nimb" [Online]. Available: <https://nimb.com/>[Accessed: 01 - Dic – 2017].
- [14] AshwiniNikam.ImprovedSecurityfortheGirlsSafety,"InternationalJournalof Innovative Research in Advanced Engineering,Volume 1, Issue 9, October 2014
- [15] Harikiran, G. C., Menasinkai, K., & Shirol, S. (2016, March). Smart security solution for women based on Internet of Things (IOT). In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on (pp. 3551-3554). IEEE.
- [16] Banzi, M., & Shiloh, M. (2014). Getting started with Arduino: the open source electronics prototyping platform. Maker Media, Inc..
- [17] Barragn, H. (2016). The Untold History of Arduino. Wiring,[Online]. Available: <http://arduinhistory.github.io/>.[Accessed 20 May 2016].
- [18] "ProMicro - 3.3V/8MHz - DEV - 12587 - SparkFunElectronics." [Online]. Available: <https://www.sparkfun.com/products/12587>. [Accessed: 29 - Aug – 2017].
- [19] Netcraft. Web server survey. (2017). Retrieved December 2017, [Online]. Available: <https://news.netcraft.com/archives/2017/12/26/december-2017-web-server-survey.html>. [Accessed: 01 - Jan – 2018].
- [20] Widenius, M., & Axmark, D. (2002). MySQL reference manual: documentation from the source. " O'Reilly Media, Inc."
- [21] NIST. (2001). Advanced Encryption Standard. FIPS Publication 197, November 26, 2001.
- [22] Rivest, R. L., Shamir, A., & Adleman, L. (Feb. 1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120–126.
- [23] Serpanos, D., & Wolf, M. (2018). Security and Safety. In Internet-of-Things (IoT) Systems (pp. 55-76). Springer, Cham.

- [24] Jörg Eberspächer and Hans-Jörg Vögel. GSM switching, services and Protocols. John Wiley and Sons, 1999.
- [25] Peng, C. (2000). GSM and GPRS security. HUT TML.
- [26] Postel, J. (1981). RFC 791: Internet protocol.
- [27] Richardson, L., & Ruby, S. (2008). RESTful web services. " O'Reilly Media, Inc."
- [28] Fielding, R. T. (2000). REST: architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California.
- [29] Rodriguez, A. (2008). Restful web services: The basics. IBM developerWorks, 33.
- [30] Daigneau, R. (2011). Service Design Patterns: fundamental design solutions for SOAP/WSDL and restful Web Services. Addison-Wesley.
- [31] Arriarán, S. S. (2015). Todo sobre sistemas embebidos. Universidad Peruana de Ciencias Aplicadas (UPC).
- [32] Eclipse IoT Working Group. IEEE, Agile-IoT EU, and IoT Council. 2018. IoT Developer Survey 2018.(2018).
- [33] Galeano, G. (2009). Programación de sistemas embebidos en C (Vol. 1). Alfaomega.
- [34] Arriarán, S. S. (2015). Todo sobre sistemas embebidos. Universidad Peruana de Ciencias Aplicadas (UPC).
- [35] Debian – Packages. Retrieved December 2017, [Online]. Available: <https://www.debian.org/distrib/packages.en.html>. [Accessed: 06 - Jun – 2018].