

**“Proyecto Heimdall”**

**PRESENTA:**

**Aguilar Álvarez Arturo (1502371)**

**Puc Pech Andro Israel (1502392)**

**Puerto Che Jesús Alberto (1502105)**

**Sáenz Domínguez Andrés (1502273)**

**TI41 - TSU en Tecnologías de la Información y la Comunicación**

**Asesor de la materia: Liu Manzanilla Javier Adrián**

**Agradecimientos**

**Sáenz Domínguez Andrés:**

Agradezco a mis padres porque gracias a ellos y por los valores que ellos me han inculcado, hoy soy quien soy, también porque me han apoyado psicológicamente en los momentos difíciles para seguir adelante y luchar por mis sueños. Estoy muy agradecidos con ustedes papa, mama por no dejar que me desviara por un mal camino.

Agradezco a mis hermanos Juan Román y José Alberto Sáenz Domínguez por brindarme el apoyo económico para poder solventar mis gastos desde el inicio de la carrera. Gracias hermanos por comprender esos momentos difíciles por los que ustedes pasaron y que hoy estoy viviendo.

Agradezco a mi equipo de trabajo: Andro Israel Puc Pech, Arturo Aguilar Álvarez, Jesús Alberto Puerto Che por apoyarme en el desarrollo de este proyecto y por qué han resuelto en muchas ocasiones mis dudas y han compartido sus conocimientos conmigo.

Agradezco a todos los maestros porque han dedicado tiempo, esfuerzo y mucho conocimiento no solo conmigo, sino que también con mis demás compañeros en el desarrollo de nuestros proyectos.

Agradezco a todas las personas que creyeron en este proyecto. Gracias por y para ustedes este proyecto se está desarrollando.

Gracias a todos por apoyarme.

**Aguilar Álvarez Arturo**

Agradezco a mi familia en general por permitirme dar estudio, gracias a ellos y al esfuerzo que han hecho para que yo pude continuar con mis estudios, a la universidad tecnológica de la Rivera Maya por permitirme ser parte de ella, a mis profesores que gracias a ellos tengo el conocimiento que me permitió realizar este proyecto, ellos fueron los que me marcaron metas para avanzar en mi paso por esta universidad, a mi equipo de trabajo que con su ayuda se pudo realizar este trabajo.

Manifiesto gratitud, por la atención prestada.

**Puerto Che Jesús Alberto**

Primeramente, quiero agradecer a mi familia que me ha apoyado, con ánimos para seguir adelante por el camino con el estudio y con este proyecto, gracias a su esfuerzo dedicación y apoyo.

Seguidamente agradezco a mi equipo de trabajo de Heimdall por dejarme formar parte de este proyecto, con el fin de seguir desarrollando mis conocimientos en Android.

Agradezco a mi tutor académico Julián Esteban Villegas Alonzo, quien estuvo al seguimiento del proyecto. Cumpliendo con su apoyo y experiencia sobre la rama de programación y base de datos.

**Dedicatorias**

**Sáenz Domínguez Andrés:**

Este proyecto va dedicado a mis familiares

A mis padres Andrés Sáenz Domínguez, María Elena Domínguez Estrada porque me han brindado su apoyo incondicionalmente, a mi hermano Juan Román Sáenz Domínguez y a José Alberto Sáenz Domínguez porque me ha demostrado esfuerzo, coraje y valentía para salir adelante y a todos mis demás hermanos que de manera indirecta han participado en la construcción de este proyecto.

A mis maestros.

Javier Adrián Liu Manzanilla, Geovani Pacheco Galaz por brindarnos el apoyo de orientarnos en el desarrollo del proyecto y a todos los maestros que nos motivan día a día a seguir adelante y que nos ayudaron en asesorías y dudas presentadas.

A mis compañeros de equipo.

Arturo Aguilar Álvarez, Andro Israel Puc Pech, Jesús Alberto Puerto Che que en conjunto trabajamos y nos apoyamos mutuamente para sacar adelante este proyecto.

Finalmente, a todos aquellos que marcaron cada etapa en nuestro camino universitario, y que de alguna u otra manera han participado en el desarrollo de nuestra formación profesional.

**Aguilar Álvarez Arturo**

Dedico este trabajo principalmente a Dios por permitirme vivir con salud y por ayudarme a llegar hasta este punto de mi vida. A mi padre Arturo Aguilar Vázquez y a mi madre Reyna Albares Escandón por ser los cimientos de la familia a la que pertenezco, dedico de corazón este trabajo a mi hermana Gabriela Yamileth Aguilar Alvarez para que veas en mi un ejemplo a seguir, a mis amigos ya que nos apoyamos mutuamente en nuestra formación profesional y por compartir buenos y malos momentos.

**Puerto che Jesús Alberto**

Este trabajo está dedicado a todo aquel que me ayudo con ánimos para seguir adelante en seguir estudiando a mi familia, amigos y grupo de la escuela.

Este trabajo refleja las horas de trabajo en equipo, las horas sin dormir y el empeño que se aplicó en este proyecto.

Este Proyecto se dedica a todos los jóvenes con sueños de poder programar, en cualquier lenguaje de programación, ya que, con el sentimiento de querer aprender, se puede lograr cosas increíbles como lo este proyecto, que inicio desde la nada, y hoy es una aplicación de que sirve a la comunidad.

Se dedica a los pequeños proyectos innovadores, y sustentables, que inician siendo un sueño y pueden culminar en un beneficio para la sociedad.

Índice

[Ficha del documento 1](#_Toc468561179)

[Introducción 2](#_Toc468561180)

[Justificación 3](#_Toc468561181)

[Alcance 3](#_Toc468561182)

[Descripción del proyecto 4](#_Toc468561183)

[Objetivos 5](#_Toc468561184)

[Objetivo General 5](#_Toc468561185)

[Objetivos Específicos 5](#_Toc468561186)

[CAPÍTULO I. MARCO TEÓRICO 6](#_Toc468561187)

[API 6](#_Toc468561188)

[Variables 6](#_Toc468561189)

[Método 6](#_Toc468561190)

[NDK 7](#_Toc468561191)

[Gradle 7](#_Toc468561192)

[Android Studio 8](#_Toc468561193)

[JAVA 8](#_Toc468561194)

[Teléfono inteligente. 8](#_Toc468561195)

[Definición de aplicaciones, con sistema Android. 9](#_Toc468561196)

[Activity en Android Studio 9](#_Toc468561197)

[Android 9](#_Toc468561198)

[Angular JS 10](#_Toc468561199)

[Framework 10](#_Toc468561200)

[Framework Web 10](#_Toc468561201)

[Patrón de diseño de software 11](#_Toc468561202)

[Modelo MVC 11](#_Toc468561203)

[Materialize 12](#_Toc468561204)

[Apache 12](#_Toc468561205)

[Arquitectura cliente-servidor 12](#_Toc468561206)

[CAPÍTULO II. MARCO CONTEXTUAL 13](#_Toc468561207)

[Antecedentes 14](#_Toc468561208)

[Descripción del proyecto 14](#_Toc468561209)

[Alcance y restricciones del Proyecto 15](#_Toc468561210)

[Alcances: 15](#_Toc468561211)

[Restricciones: 15](#_Toc468561212)

[CAPÍTULO III.Desarrollo del Proyecto 16](#_Toc468561213)

[Conclusión 17](#_Toc468561214)

[Anexos 18](#_Toc468561215)

[Fuentes de Información (Formato APA6) 31](#_Toc468561216)

[Glosario 33](#_Toc468561217)

# 

# Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Verificado dep. Calidad.** |
| 03/12/2016 |  | Aguilar Álvarez Arturo  Puc Pech Andro Israel  Puerto Che Jesús Alberto  Sáenz Domínguez Andrés |  |

Documento validado por las partes en fecha:

|  |  |
| --- | --- |
| Por la comunidad | Por la universidad |
| -- | -- |

# Introducción

En el presente documento se presentarán las especificaciones, requerimientos, requisitos, funcionalidades y el desarrollo del proyecto Heimdall (Aplicación para seguridad y rastreo de dispositivos móviles) basándose en los estándares de calidad, así como dar a conocer el funcionamiento de la aplicación en distintos dispositivos.

De igual forma se describirá el proceso de desarrollo de la aplicación Heimdall, quienes conformamos Heimdall y se presentara el código que se realizó durante el desarrollo del proyecto cabe mencionar que dentro del código no estarán disponibles todas las funciones que nosotros proponemos ya que la aplicación que se está presentando en estos momentos sigue en desarrollo.

# Justificación

Elegimos este proyecto, ya que somos usuarios de teléfonos celulares y es muy probable perder o que nos roben nuestros dispositivos móviles, por lo tanto, nos propusimos a crear una alternativa que permita saber y tener mayor seguridad en nuestros teléfonos celulares. Por lo tanto, la idea de Heimdall surge dentro de esta problemática tan común dentro de la población de México.

Este proyecto va dirigido a cualquier usuario de teléfono celular con sistema operativo Android.

## Alcance

Los alcances que posee este proyecto es el poder ayudar a usuarios con dispositivos Android en la búsqueda, control de sus dispositivos Android. Esta especificación de requisitos está dirigida tanto a usuarios del sistema como a desarrolladores, a los usuarios para dar a conocer todos los aspectos funcionales y no funcionales del sistema y a los desarrolladores para dar un nuevo panorama del desarrollo de nuevos proyectos.

# Descripción del proyecto

Esta aplicación puedes ser usada por cualquier usuario que posea un Smartphone con el sistema operativo Android, la principal funcionalidad de esta aplicación es rastrear dispositivos en caso de pérdida.

La aplicación Heimdall constará de 2 aplicaciones, una web y otra móvil. La aplicación web, será una sola aplicación de una página en modo de panel de administración para poder acceder fácilmente a todos los controles del dispositivo al que esté vinculado previamente con la aplicación en Android.

**Restricciones**

Se deben conocer las siguientes restricciones para poder utilizar y/o desarrollar la aplicación

**Uso:**

1. La aplicación se podrá utilizar únicamente con conexión a Internet.
2. El dispositivo a utilizar deberá contar con una versión de Android igual o mayor a 4.4 (KitKat)
3. Se deberá contar con la aprobación del usuario para poder hacer uso de los datos de su dispositivo.

**Desarrollo:**

1. Lenguajes en uso: Java, HTML, PHP y SQL
2. Framework en uso: AngularJS, Materialize
3. El sistema hace uso del modelo MVC
4. Los servidores deben estar optimizados para realizar consultas frecuentemente.
5. Tanto la aplicación web como la móvil deberá contar con el diseño “Material Design” desarrollado por Google Inc.

# Objetivos

## Objetivo General

* Localizar el teléfono celular

## Objetivos Específicos

* Utilizar la tecnología GPS
* Hacer sonar el teléfono celular
* General el código de la aplicación Heimdall y de la aplicación web
* Crear el servidor de Heimdall

# CAPÍTULO I. MARCO TEÓRICO

## API

Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés *Application Programming Interface*. En español sería Interfaz de Programación de Aplicaciones.

Una API nos permite implementar las funciones y procedimientos que engloba en nuestro proyecto sin la necesidad de programarlas de nuevo. En términos de programación, es una capa de abstracción.

## Variables

Una variable es un espacio de la memoria del ordenador a la que asignamos un contenido que puede ser un valor numérico (sólo números, con su valor de cálculo) o alfanumérico (sólo texto o texto con números). Cada variable tiene un único nombre el cual no puede ser cambiado. Dos o más variables pueden tener el mismo contenido, pero no el mismo nombre. El nombre de una variable comenzará siempre por una letra, pudiendo contener a continuación tanto letras como números.

## Método

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, y/o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

## NDK

El Android NDK (Native Development Kit) permite a los desarrolladores reutilizar código escrito en C/C++ introduciéndolo en las aplicaciones a través de JNI (Java Native Interface). El NDK hace que la ejecución de la aplicación sea en cierto modo más rápida, ya que pasará a ejecutarse directamente en el procesador y no es interpretado por una máquina virtual.

El NDK puede ser útil y beneficioso para ciertos tipos de aplicaciones, pero no para otros. El uso de código nativo en Android no se traduce en una notable mejora de rendimiento, pero de lo que sí estoy seguro, es de que aumenta la complejidad de la aplicación hasta límites insospechados.

## Gradle

Gradle es una herramienta de automatización de la construcción de nuestro código que bebe de las aportaciones que han realizado herramientas como ant y maven, pero intenta llevarlo todo un paso más allá. Para empezar, se apoya en Groovy y en un DSL (Domain Specific Language) para trabajar con un lenguaje sencillo y claro a la hora de construir el build comparado con Maven. Por otro lado, dispone de una gran flexibilidad que permite trabajar con ella utilizando otros lenguajes y no solo Java. Dispone por otro lado de un sistema de gestión de dependencias sólido.

## Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en [IntelliJ IDEA](https://www.jetbrains.com/idea/). Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android.

## JAVA

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

## Teléfono inteligente

Teléfono inteligente (*Smartphone* en inglés) es un término comercial para denominar a un teléfono que ofrece la posibilidad de instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. El término «inteligente» hace referencia a cualquier interfaz, como un teclado QWERTY en miniatura, una pantalla táctil (lo más habitual, denominándose en este caso «teléfono móvil táctil»), o simplemente el sistema operativo móvilque posee, diferenciando su uso mediante una exclusiva disposición de los menús, teclas, atajos, etc.

## Aplicaciones en Android

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varias cruces aplicaciones de la plataforma web móvil marcos.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

## Activity en Android Studio

El MainActivity.java es la actividad principal de nuestra aplicación, es cada una de las pantallas o vistas que forman una aplicación. en esa clase definiremos los métodos y llamadas a otras clases, así como el funcionamiento del layout “nuevo\_activity”. No es necesario que tenga ese nombre en concreto, podemos llamarle como queramos, configurando siempre antes com actividad principal la clase que seleccionemos.

## Android

Sistema Operativo y plataforma de desarrollo Open Source para dispositivos móviles. Se compone de aplicaciones que se ejecutan en un framework Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL. Todo montado sobre S.O. Linux Kernel 2.6

## Angular JS

AngularJS es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications. Es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente.

## Framework

Un Framework es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

## Framework Web

(Web application framework) Un framework para aplicaciones web es un framework que sirve para el desarrollo web: aplicaciones web, sitios web dinámicos y servicios web.

Los frameworks proporcionan herramientas, bibliotecas, plantillas, códigos y aplicaciones de ejemplos, etc., que facilitan el desarrollo web.

## Patrón de diseño de software

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

## Modelo MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

* El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
* La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
* El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

## Materialize

Materialize CSS se define en su web como “Un framework web frontend moderno y responsivo basado en Google Material Design”.

## Apache

El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

## Arquitectura cliente-servidor

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor.

## 

# 

# CAPÍTULO II. MARCO CONTEXTUAL

## Antecedentes

Nuestro mercado se ubica en el sector de Tecnología y seguridad, ya que nuestra aplicación “Heimdall” es una app para celulares con sistema operativo Android que permite el monitoreo del mismo en caso de pérdida, robo, etc.

Uno puede identificar el mercado meta si separa a los clientes en grupos con necesidades similares, esto se conoce como la segmentación del mercado; en nuestro mercado se distinguen varias formas de segmentación como lo son la geográfica, demográfica y de beneficios y por frecuencia de uso.

Al cierre del 2015, se reportó el robo de 609 mil 547 teléfonos celulares en México, lo que representó el hurto de mil 670 dispositivos al día y un aumento de 37.8 por ciento comparado con un año antes.

Viendo las siguientes cifras, nos podemos dar cuenta que contamos con un mercado bastante amplio el cual se puede aprovechar, ya que muchas de estos clientes afectados, verían nuestra aplicación como una solución al problema de la pérdida de su información personal.

Al tener esta problemática la mayoría de los usuarios afectados, menciona que, desearía poder realizar las siguientes acciones:

* **Localizar** y rastrear su dispositivo
* **Iniciar una alarma** incluso si el dispositivo está en modo silencioso
* **Borrar los datos** del dispositivo

Estas acciones son las que Heimdall se enfoca en desarrollar y mejorar para que el usuario tenga un control total de su dispositivo.

## Descripción del proyecto

La aplicación será un producto diseñado para trabajar tanto en entornos WEB y dispositivos móviles, lo que permitirá ser usado desde cualquier lugar siempre y cuando el dispositivo esté conectado a la Red, lo que permitirá una mayor portabilidad. La conexión es necesaria ya que los datos se almacenarán en una base de datos y para poder usar las funcionalidades tanto la Aplicación Web como el móvil se conectarán a ella y deberán actualizar los datos cada que se realice una operación.

## Alcance y restricciones del Proyecto

### Alcance

Los alcances que posee este proyecto es el poder ayudar a usuarios con dispositivos Android en la búsqueda, control de sus dispositivos. Esta especificación de requisitos está dirigido tanto a usuarios del sistema como a desarrolladores, a los usuarios para dar a conocer todos los aspectos funcionales

y no funcionales del sistema y a los desarrolladores para dar un nuevo panorama del desarrollo de nuevos proyectos.

### Restricciones

* Interfaz para ser usada con internet.
* Lenguajes y tecnologías en uso: HTML, JAVA.
* El servidor debe ser capaz de atender consultas concurrentes.
* El sistema se diseñará según un modelo cliente/servidor.
* El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

# 

# CAPÍTULO III.Desarrollo del Proyecto

# Conclusión

Viendo las siguientes cifras de delincuencia y que estos están enfocados en robo de celulares, nos podemos dar cuenta que contamos con un mercado bastante amplio el cual se puede aprovechar, ya que muchas de estos clientes afectados, ven nuestra aplicación como una solución al problema de la pérdida de su información personal.

Al tener esta problemática la mayoría de los usuarios afectados, menciona que, desearía poder realizar las siguientes acciones:

* Localizar y rastrear su dispositivo
* Iniciar una alarma incluso si el dispositivo está en modo silencioso
* Borrar los datos del dispositivo

Por lo cual desarrollamos la aplicación que contiene esas funcionalidades, y muchas más como es la vibración y el sonido del dispositivo, así como respaldar los datos del celular en nuestra base de datos.

Aprendimos como utilizar los diversos sensores de los dispositivos móviles, y como mandar datos desde los móviles a la base de datos, aprendimos como interactuar, con la interfaz de Android.

Llevar a cabo este proyecto ha sido difícil ya que el trabajar en equipo no es una tarea fácil ya que es la primera vez que hemos tenido que desarrollar una aplicación como un equipo de trabajo de igual forma esto nos ayudó a conocernos y aprender de las experiencias que hemos tenido como miembros de este quipo de Heimdall.

# Anexos

Código fuente: AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.arturo.finemyandy"**>  
  
 <**uses-permission android:name="android.permission.INTERNET"** />  
 <**uses-permission android:name="android.permission.VIBRATE"** />  
 <**uses-permission android:name="android.permission.CAMERA"** />  
 <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />  
 <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".LoginActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".MainActivity"**></**activity**>  
 </**application**>  
  
</**manifest**>

Código fuente: activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.arturo.finemyandy.MainActivity"**>  
  
 <**Button  
 android:text="LogOut"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btLogOut"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="17dp"  
 android:layout\_marginTop="19dp"** />  
  
 <**Button  
 android:text="Linterna"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btVibrar"  
 android:layout\_toStartOf="@+id/btLogOut"  
 android:layout\_marginTop="26dp"  
 android:id="@+id/btLinterna"** />  
  
 <**Button  
 android:text="Vibrar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="33dp"  
 android:id="@+id/btVibrar"  
 android:layout\_below="@+id/btLogOut"  
 android:layout\_alignStart="@+id/btLinterna"** />  
  
 <**Button  
 android:text="sonar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btLinterna"  
 android:layout\_alignEnd="@+id/btVibrar"  
 android:layout\_marginTop="29dp"  
 android:id="@+id/btSonar"** />  
  
 <**Button  
 android:text="gps"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btSonar"  
 android:layout\_alignEnd="@+id/btSonar"  
 android:layout\_marginTop="33dp"  
 android:id="@+id/btGPS"** />  
  
</**RelativeLayout**>

Código Fuente: MainActivity.java

**package** com.example.arturo.finemyandy;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.media.MediaPlayer;  
**import** android.os.Vibrator;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** EditText **email**,**contraseña**;  
 **private** Button **btLogout**,**btLinterna**,**btVibar**,**btGPS**;  
  
 MediaPlayer **sonido1**;  
  
 Button **btSonido**;  
  
 **int a**=0;  
 **int b**=0;  
  
 android.hardware.Camera **camara**;  
  
 GPSTracker **gps**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
 **sonido1**= MediaPlayer.*create*(**this**, R.raw.***sonido1***);  
  
 **btVibar**=(Button) findViewById(R.id.***btVibrar***);  
 **btLinterna**=(Button) findViewById(R.id.***btLinterna***);  
 **btSonido**= (Button)findViewById(R.id.***btSonar***);  
 **btGPS**=(Button)findViewById(R.id.***btGPS***);  
  
  
  
 **final** Vibrator vibradorr=(Vibrator) **this**.getSystemService(Context.***VIBRATOR\_SERVICE***);  
  
 **btVibar**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 vibradorr.vibrate(800);  
 }  
 });  
 **btGPS**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
**gps** = **new** GPSTracker(MainActivity.**this**);  
  
 **if**(**gps**.canGetLocation()){  
  
 **double** latitude = **gps**.getLatitude();  
 **double** longitude = **gps**.getLongitude();  
  
 Toast.*makeText*(getApplicationContext(), **"tu localizacion es - \nLat: "** + latitude + **"\nLong: "** + longitude, Toast.***LENGTH\_LONG***).show();  
 }**else**{  
 **gps**.showSettingsAlert();  
 }  
  
 }  
  
  
 });  
  
 **btLinterna**.setOnClickListener(**new** View.OnClickListener(){  
 @Override  
 **public void** onClick(View view){  
 **if** (**a**==0){  
 **camara**= android.hardware.Camera.*open*();  
 android.hardware.Camera.Parameters parameters=**camara**.getParameters();  
 parameters.setFlashMode(android.hardware.Camera.Parameters.***FLASH\_MODE\_TORCH***);  
 **camara**.setParameters(parameters);  
 **camara**.startPreview();  
 **a**=1;  
 }**else** {  
 android.hardware.Camera.Parameters parameters=**camara**.getParameters();  
 parameters.setFlashMode(android.hardware.Camera.Parameters.***FLASH\_MODE\_OFF***);  
 **camara**.setParameters(parameters);  
 **camara**.startPreview();  
 **camara**.release();  
 **a**=0;  
 }  
 }  
 });  
 **btSonido**.setOnClickListener(**new** View.OnClickListener(){  
 @Override  
 **public void** onClick(View view) {  
 **sonido1**.start();  
 }  
 });  
 }  
}

Código fuente:aActivity\_login.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.arturo.finemyandy.LoginActivity"**>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textEmailAddress"  
 android:ems="10"  
 android:id="@+id/email"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:hint="Email"  
 android:layout\_marginTop="10dp"**/>  
  
 <**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textPassword"  
 android:ems="10"  
 android:id="@+id/password"  
 android:layout\_below="@+id/email"  
 android:hint="Password"  
 android:layout\_marginTop="10dp"**/>  
  
 <**Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="LOGIN"  
 android:id="@+id/button"  
 android:layout\_below="@+id/password"  
 android:layout\_marginTop="10dp"  
 android:onClick="checkLogin"**/>  
  
 <**Button  
 android:text="next"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/btNextMain"  
 android:layout\_below="@+id/button"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="123dp"** />  
</**RelativeLayout**>

Código fuente:LoginActivity.java

**package** com.example.arturo.finemyandy;  
  
**import** android.app.ProgressDialog;  
**import** android.content.Intent;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.io.OutputStreamWriter;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
  
**public class** LoginActivity **extends** AppCompatActivity {  
  
 **private** Button **btNextMain**;  
 **public static final int *CONNECTION\_TIMEOUT***=10000;  
 **public static final int *READ\_TIMEOUT***=15000;  
 **private** EditText **etEmail**;  
 **private** EditText **etPassword**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_login***);  
  
 **btNextMain**=(Button) findViewById(R.id.***btNextMain***);  
 **etEmail** = (EditText) findViewById(R.id.***email***);  
 **etPassword** = (EditText) findViewById(R.id.***password***);  
  
  
 **btNextMain**.setOnClickListener(**new** View.OnClickListener(){  
 @Override  
 **public void** onClick(View view){  
 Intent intent =**new** Intent(LoginActivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 }  
 });  
 }  
  
 **public void** checkLogin(View arg0) {  
  
 **final** String email = **etEmail**.getText().toString();  
 **final** String password = **etPassword**.getText().toString();  
  
 **new** AsyncLogin().execute(email,password);  
  
 }  
  
 **private class** AsyncLogin **extends** AsyncTask<String, String, String>  
 {  
 ProgressDialog **pdLoading** = **new** ProgressDialog(LoginActivity.**this**);  
 HttpURLConnection **conn**;  
 URL **url** = **null**;  
  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
  
 **pdLoading**.setMessage(**"\tLoading..."**);  
 **pdLoading**.setCancelable(**false**);  
 **pdLoading**.show();  
  
 }  
 @Override  
 **protected** String doInBackground(String... params) {  
 **try** {  
  
 *// poner URL DEL PHP* **url** = **new** URL(**"http://192.168.43.5/FindMyAndy/login.inc.php"**);  
  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 **return "exception"**;  
 }  
 **try** {  
 **conn** = (HttpURLConnection)**url**.openConnection();  
 **conn**.setReadTimeout(***READ\_TIMEOUT***);  
 **conn**.setConnectTimeout(***CONNECTION\_TIMEOUT***);  
 **conn**.setRequestMethod(**"POST"**);  
  
 */* **conn**.setDoInput(**true**);  
 **conn**.setDoOutput(**true**);  
  
 *// Append parameters to URL* Uri.Builder builder = **new** Uri.Builder()  
 .appendQueryParameter(**"username"**, params[0])  
 .appendQueryParameter(**"password"**, params[1]);  
 String query = builder.build().getEncodedQuery();  
  
 *// Open connection for sending data* OutputStream os = **conn**.getOutputStream();  
 BufferedWriter writer = **new** BufferedWriter(  
 **new** OutputStreamWriter(os, **"UTF-8"**));  
 writer.write(query);  
 writer.flush();  
 writer.close();  
 os.close();  
 **conn**.connect();  
  
 } **catch** (IOException e1) {  
 *//* ***TODO Auto-generated catch block*** e1.printStackTrace();  
 **return "exception"**;  
 }  
  
 **try** {  
  
 **int** response\_code = **conn**.getResponseCode();  
  
 *// Check if successful connection made* **if** (response\_code == HttpURLConnection.***HTTP\_OK***) {  
  
 *// Read data sent from server* InputStream input = **conn**.getInputStream();  
 BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(input));  
 StringBuilder result = **new** StringBuilder();  
 String line;  
  
 **while** ((line = reader.readLine()) != **null**) {  
 result.append(line);  
 }  
  
 *// Pass data to onPostExecute method* **return**(result.toString());  
  
 }**else**{  
  
 **return**(**"unsuccessful"**);  
 }  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 **return "exception"**;  
 } **finally** {  
 **conn**.disconnect();  
 }  
  
  
 }  
  
 @Override  
 **protected void** onPostExecute(String result) {  
  
 *//this method will be running on UI thread* **pdLoading**.dismiss();  
  
 **if**(result.equalsIgnoreCase(**"true"**))  
 {  
 Intent intent = **new** Intent(LoginActivity.**this**,MainActivity.**class**);  
 startActivity(intent);  
 LoginActivity.**this**.finish();  
  
 }**else if** (result.equalsIgnoreCase(**"false"**)){  
  
Toast.*makeText*(LoginActivity.**this**, **"invalid email o password"**, Toast.***LENGTH\_LONG***).show();  
  
 } **else if** (result.equalsIgnoreCase(**"exception"**) || result.equalsIgnoreCase(**"unsuccessful"**)) {  
  
 Toast.*makeText*(LoginActivity.**this**, **"OOPs! Algo salio mal."**, Toast.***LENGTH\_LONG***).show();  
  
 }  
 }  
  
 }  
  
  
}

Código fuente: GPSTracker.java

**package** com.example.arturo.finemyandy;  
  
**import** android.Manifest;  
**import** android.app.AlertDialog;  
**import** android.app.Service;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.location.Location;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.os.Bundle;  
**import** android.os.IBinder;  
**import** android.provider.Settings;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.util.Log;  
  
*/\*\*  
 \* Created by ARTUR on 30/10/2016.  
 \*/***public class** GPSTracker **extends** Service **implements** LocationListener {  
  
 **private final** Context **mContext**;  
  
 *// flag for GPS status* **boolean isGPSEnabled** = **false**;  
  
 *// flag for network status* **boolean isNetworkEnabled** = **false**;  
  
 *// flag for GPS status* **boolean canGetLocation** = **false**;  
  
 Location **location**; *// location* **double latitude**; *// latitude* **double longitude**; *// longitude  
  
 // The minimum distance to change Updates in meters* **private static final long *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES*** = 10; *// 10 meters  
  
 // The minimum time between updates in milliseconds* **private static final long *MIN\_TIME\_BW\_UPDATES*** = 1000 \* 60 \* 1; *// 1 minute  
  
 // Declaring a Location Manager* **protected** LocationManager **locationManager**;  
  
 **public** GPSTracker(MainActivity context) {  
 **this**.**mContext** = context;  
 getLocation();  
 }  
  
 **public** Location getLocation() {  
 **try** {  
 **locationManager** = (LocationManager) **mContext** .getSystemService(Context.***LOCATION\_SERVICE***);  
  
 *// getting GPS status* **isGPSEnabled** = **locationManager** .isProviderEnabled(LocationManager.***GPS\_PROVIDER***);  
  
 *// getting network status* **isNetworkEnabled** = **locationManager** .isProviderEnabled(LocationManager.***NETWORK\_PROVIDER***);  
  
 **if** (!**isGPSEnabled** && !**isNetworkEnabled**) {  
 *// no network provider is enabled* } **else** {  
 **this**.**canGetLocation** = **true**;  
 *// First get location from Network Provider* **if** (**isNetworkEnabled**) {  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
  
 }  
 **locationManager**.requestLocationUpdates(  
 LocationManager.***NETWORK\_PROVIDER***,  
 ***MIN\_TIME\_BW\_UPDATES***,  
 ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, (LocationListener) **this**);  
 Log.*d*(**"Network"**, **"Network"**);  
 **if** (**locationManager** != **null**) {  
 **location** = **locationManager** .getLastKnownLocation(LocationManager.***NETWORK\_PROVIDER***);  
 **if** (**location** != **null**) {  
 **latitude** = **location**.getLatitude();  
 **longitude** = **location**.getLongitude();  
 }  
 }  
 }  
 *// if GPS Enabled get lat/long using GPS Services* **if** (**isGPSEnabled**) {  
 **if** (**location** == **null**) {  
 **locationManager**.requestLocationUpdates(  
 LocationManager.***GPS\_PROVIDER***,  
 ***MIN\_TIME\_BW\_UPDATES***,  
 ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, (LocationListener) **this**);  
 Log.*d*(**"GPS Enabled"**, **"GPS Enabled"**);  
 **if** (**locationManager** != **null**) {  
 **location** = **locationManager** .getLastKnownLocation(LocationManager.***GPS\_PROVIDER***);  
 **if** (**location** != **null**) {  
 **latitude** = **location**.getLatitude();  
 **longitude** = **location**.getLongitude();  
 }  
 }  
 }  
 }  
 }  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return location**;  
 }  
  
 */\*\*  
 \* Stop using GPS listener  
 \* Calling this function will stop using GPS in your app  
 \* \*/* **public void** stopUsingGPS(){  
 **if**(**locationManager** != **null**){  
 **locationManager**.removeUpdates(GPSTracker.**this**);  
 }  
 }  
  
 */\*\*  
 \* Function to get latitude  
 \* \*/* **public double** getLatitude(){  
 **if**(**location** != **null**){  
 **latitude** = **location**.getLatitude();  
 }  
  
 *// return latitude* **return latitude**;  
 }  
  
 */\*\*  
 \* Function to get longitude  
 \* \*/* **public double** getLongitude(){  
 **if**(**location** != **null**){  
 **longitude** = **location**.getLongitude();  
 }  
  
 *// return longitude* **return longitude**;  
 }  
  
 */\*\*  
 \* Function to check GPS/wifi enabled  
 \** ***@return*** *boolean  
 \* \*/* **public boolean** canGetLocation() {  
 **return this**.**canGetLocation**;  
 }  
  
 */\*\*  
 \* Function to show settings alert dialog  
 \* On pressing Settings button will lauch Settings Options  
 \* \*/* **public void** showSettingsAlert(){  
 AlertDialog.Builder alertDialog = **new** AlertDialog.Builder(**mContext**);  
  
 *// Setting Dialog Title* alertDialog.setTitle(**"GPS is settings"**);  
  
 *// Setting Dialog Message* alertDialog.setMessage(**"GPS is not enabled. Do you want to go to settings menu?"**);  
  
 *// On pressing Settings button* alertDialog.setPositiveButton(**"Settings"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog,**int** which) {  
 Intent intent = **new** Intent(Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  
 **mContext**.startActivity(intent);  
 }  
 });  
  
 *// on pressing cancel button* alertDialog.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.cancel();  
 }  
 });  
  
 *// Showing Alert Message* alertDialog.show();  
 }  
  
 @Override  
 **public void** onLocationChanged(Location location) {  
 }  
  
 @Override  
 **public void** onProviderDisabled(String provider) {  
 }  
  
 @Override  
 **public void** onProviderEnabled(String provider) {  
 }  
  
 @Override  
 **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  
 }  
  
 @Override  
 **public** IBinder onBind(Intent arg0) {  
 **return null**;  
 }

# Fuentes de Información (Formato APA6)

Castañares I. (03/05/2016).A diario se roban en el país mil 670 celulares(http://www.elfinanciero.com.mx/empresas/robo-de-celulares-crece-37-en-mexico-en-2015.html).

Codigo Facilito (29/06/2015).Curso de Android Studio (https://codigofacilito.com/videos/curso\_introduccion\_a\_android\_instalacion\_de\_android\_studio).

Revelo J. (12/06/2014).Crear Procedimientos Almacenados en MySQL(http://www.hermosaprogramacion.com/2014/06/mysql-procedure-show-create/).

(11/12/2014).Android Studio v1.0: características y comparativa con Eclipse(http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/).

pedrini210.(s/f).Carácteristicas y cualidades de Android Studio(http://blog.desdelinux.net/caracteristicas-y-cualidades-de-android-studio/#).

Azaustre C.(13/01/2014).Tutorial de AngularJS. Ejemplo de aplicación web conectada a una API REST con Node(https://carlosazaustre.es/blog/tutorial-ejemplo-de-aplicacion-web-con-angular-js-y-api-rest-con-node/).

(13/05/2016).Android Studio 2.0(https://desarrolladores.googleblog.com/2016/05/android-studio-20.html).

Revelo J.(20/10/2014).Tutorial De Bases De Datos SQLite En Aplicaciones Android(http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/).

ExpoCodeTECH(09/12/2012).Expo-Tips: Como obtener la Ubicación actual con el GPS de Android.(http://expocodetech.com/expo-tips-como-obtener-la-ubicacion-actual-con-el-gps-de-android/).

El androide libre(1/08/2010).Aprendiendo Android V: Inicialización a la API del GPS(http://www.elandroidelibre.com/2010/08/aprendiendo-android-v-inicializacion-a-la-api-del-gps.html.

Básalo Alberto y Álvarez Miguel Ángel. (28 de agosto de 2014). Que es Angular JS. Recuperado el

19 de noviembre del 2016. http://www.desarrolloweb.com/articulos/que-es-angularjs- descripcion-framework-javascript-conceptos.html.

M. Cristina. 19 de noviembre del 2016. ¿Que son los frameworks?. [Definición de frameworks Web]. http://blog.nubelo.com/que-son-los-frameworks

Alegsa Leandro. (16 de noviembre del 2010). Diccionario de informática y tecnología. Recuperado el 19 de noviembre del 2016. http://www.alegsa.com.ar/Dic/framework\_para\_aplicaciones\_web.php.

Tedeschi Nicolás. (s. f.). ¿Qué es un patrón de diseño? Recuperado el 19 de noviembre del 2016.https://msdn.microsoft.com/es-es/library/bb972240.aspx

Universidad de Alicante. (s.f.). Servicio de informática ASP.NET MVC 3 Framewor. Recuperado el 19 de noviembre del 2016. https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

Álvarez Sara. (30 de Agosto del 2007). Arquitectura Cliente-Servidor. Recuperado el 19 de   
Noviembre del 2016. http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html

# Glosario

**API**. Application Programming Interface.

**APK**: Es la extensión de las aplicaciones Android, variante del Java. Así pues, un **APK** es cualquier aplicación que te puedes instalar en tu móvil.

**JDK**. Java Developement Kit.

**PHP**

**Android** **Manifest** : Es el archivo en el que se definen los rasgos principales de una aplicación: nombre, componentes o permisos entre otros.

**Backup**: Copia de seguridad de nuestros datos.

**BUG**: Defecto en el software o un hardware que no ha sido descubierto por los creadores o diseñadores de los mismos durante su fabricacion o desarrollo.

**View**: El objeto más importante cuando estamos montando una interfaz gráfica. De él extenderán la mayoría de componentes que utilicemos. Básicamente consiste en un área de la pantalla que admitirá determinadas funcionalidades

**UI**: Interfaz de usuario

**SU**: "Super user" o super usuario. Lo que es lo mismo: acceso root

**SDK:** Un kit de desarrollo de software o SDK (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto

**Rootear**: Modificar el sistema para poder disfrutar de permisos de root(=administrador, superusuario).

**Layout**: Fichero XML que describe la composición de una pantalla

**GPS**: Sistema americano de navegación y localización mediante satélites.