



Universidad Nacional Autónoma de México

Facultad de Ingeniería

**Construcción de un localizador GPS en Ciudad
Universitaria para dispositivos móviles Android.**

**TESIS
QUE PARA OBTENER EL TÍTULO DE
INGENIERA EN COMPUTACIÓN**

**PRESENTA:
Dan Karla Cerón Jiménez**

**DIRECTOR DE TESIS:
Ing. Manuel Castañeda Castañeda**



Dedico esta tesis a la Sra. Elia Figueroa

El viaje no termina jamás. Solo los viajeros terminan. Y también ellos pueden subsistir en memoria, en recuerdo, en narración... El objetivo de un viaje es solo el inicio de otro viaje.

José Saramago

Agradecimientos

A mis papas, Delia y Martin, por el amor, la educación, el apoyo y el esfuerzo que realizaron para que mis hermanos y yo seamos mejores personas. Me motivaron a forjar un cimiento y poder terminar este gran logro en mi vida. Los amo

A mi compañero de vida Jorge, por no permitir que hiciera a un lado mis sueños por más difíciles que parecieran. Gracias por tu comprensión, orientación, apoyo y sobre todo gracias por ser un gran hombre y estar hombro con hombro conmigo estos últimos cuatro años. Te amo amor

A mis hermanos, Michael, Giovanni y Dulce, por sus consejos y porque siempre están a mi lado cuándo más los necesito, estoy muy orgullosa de ustedes. Los quiero mucho.

A mis abuelitas, Antonia y Guadalupe, por enseñarme a ser una mujer fuerte y luchar por mis objetivos, abuelita Toña ya no llegaste a ver este trabajo terminado pero sé que estarás conmigo siempre. Las quiero mucho

A mi jefa, profesora, compañera y amiga, Marian, por tu gran apoyo y orientación incondicional en mi formación profesional y en la realización de este proyecto.

A Gaby Camacho, Gaby Ramírez, Ale, Adolfo y Anduin por brindarme su amistad y compañerismo en UNICA, gracias por su gran apoyo en mi formación profesional.

A Verito, Jericó y Wendy, por sus consejos, por su ayuda y por brindarme su amistad en tan corto tiempo.

A mis cuñadas y amigas, Ivette, Lupe, Silvia y Roció, por su gran apoyo y por permitir que formara parte de su familia.

A todos y cada una de las personas por darme la sabiduría y el tiempo para realizarme profesionalmente.

ÍNDICE

PRIMERA PARTE	2
1. INTRODUCCIÓN.....	2
1.1 DEFINICIÓN DEL PROBLEMA	2
1.2 OBJETIVO GENERAL.....	3
1.3 OBJETIVOS ESPECÍFICOS.....	3
1.4 LIMITANTES	3
1.5 RESULTADOS ESPERADOS	4
1.6 METODOLOGÍA DEL TRABAJO	4
2. MARCO TEÓRICO.....	7
2.1 INGENIERÍA DE SOFTWARE	7
2.1.1 Modelado de sistemas	7
2.1.2 Ciclos de Vida del Software	8
2.1.3 Verificación y validación de software.....	8
2.1.4 Administración de proyectos de software	9
2.2 BASES DE DATOS RELACIONALES	10
2.3 REDES CELULARES	11
2.4 SISTEMA SATELITAL DE POSICIONAMIENTO (GPS)	14
2.5 DISPOSITIVOS MÓVILES INTELIGENTES.....	16
2.6 TECNOLOGÍAS DE INFORMACIÓN EN DISPOSITIVOS MÓVILES.....	19
2.7 SISTEMA OPERATIVO ANDROID	20
2.7.1 APLICACIONES DE LOCALIZACIÓN	24
3. ANÁLISIS	27
3.1 LEVANTAMIENTO Y ANÁLISIS DE REQUERIMIENTOS.....	27
3.1.1 TIPOS DE REQUERIMIENTO	29
3.2 MODELADO DEL SISTEMA	31
3.2.1 DIAGRAMAS DE FLUJO DE DATOS.....	32
3.2.2 CASOS DE USO.....	33
3.2.3 DICCIONARIO DE DATOS	34
3.3 DISEÑO DE LA ARQUITECTURA DE DESARROLLO	34
SEGUNDA PARTE	37

4. DISEÑO DE LA ARQUITECTURA.....	37
4.1 PARADIGMAS DE PROGRAMACIÓN	37
4.2 PLATAFORMAS DE DESARROLLO MÓVIL.....	40
4.3 MATRIZ DE PRUEBA Y TESTING	42
5. IMPLEMENTACIÓN	48
5.1 PLAN DE IMPLEMENTACIÓN	48
5.1.1 MANUAL TÉCNICO.....	49
5.2 PAQUETE DE DISTRIBUCIÓN.....	54
5.3 REQUERIMIENTOS DE INSTALACIÓN.....	55
6. RESULTADOS	57
7. CONCLUSIONES	66
8. REFERENCIAS.....	69
ANEXOS	i
I. MANUAL DE USUARIO.....	i
II. SOPORTE TÉCNICO	xiv
III. INTERFAZ INTEGRADA DE DESARROLLO ANDROID	xvii
IV. CODIFICACIÓN.....	xxiv

IMÁGENES.

<i>Ilustración 1 Modelo Incremental</i>	5
<i>Ilustración 2 Tipos de requerimientos</i>	29
<i>Ilustración 3 Componentes del Diagrama de Flujo</i>	32
<i>Ilustración 4 Componentes del diagrama casos de uso</i>	33
<i>Ilustración 5 Paradigma orientado a objetos</i>	39
<i>Ilustración 6 Logotipo de Java</i>	39
<i>Ilustración 7 Recuperación de contraseña</i>	43
<i>Ilustración 8 Modulo de Testing</i>	45
<i>Ilustración 9 Modulos de Pruebas</i>	46
<i>Ilustración 10 Tabla de la base de datos del sistema</i>	49
<i>Ilustración 11 Página principal del sistema</i>	50
<i>Ilustración 12 Diagrama de flujo de la aplicación</i>	52
<i>Ilustración 13 Casos de uso del sistema</i>	53
<i>Ilustración 14 Pruebas realizadas en el dispositivo Samsung S3 mini</i>	62
<i>Ilustración 15 Pruebas realizadas en el dispositivo Inco Storm</i>	62
<i>Ilustración 16 Pruebas realizadas en el dispositivo LG L9</i>	63
<i>Ilustración 17 Pruebas realizadas en el dispositivo Samsung S4</i>	64
<i>Ilustración 18 Icono de la App</i>	i
<i>Ilustración 19 Splash de la App</i>	ii
<i>Ilustración 20 Pantalla principal de la App</i>	iii
<i>Ilustración 21 Autenticando</i>	iv
<i>Ilustración 22 Datos incorrectos</i>	v
<i>Ilustración 23 Página para registrar usuarios</i>	vi
<i>Ilustración 24 Preguntas de seguridad</i>	vii
<i>Ilustración 25 Confirmación de nuevo usuario</i>	viii
<i>Ilustración 26 Página para recuperar contraseña</i>	ix
<i>Ilustración 27 Pa'gina para responder pregunta secreta</i>	x
<i>Ilustración 28 Confirmación del cambio de contraseña</i>	xí
<i>Ilustración 29 Mensaje de GPS desactivado</i>	xi
<i>Ilustración 30 Mapa con ubicaciones</i>	xii
<i>Ilustración 31 Mapa con ubicaciones 2</i>	xiii
<i>Ilustración 32 Configuraciones del teléfono</i>	xiv
<i>Ilustración 33 Error al ingresar el usuario</i>	xv
<i>Ilustración 34 Icono de una señal de datos móviles</i>	xv
<i>Ilustración 35 Icono de una señal WiFi</i>	xv
<i>Ilustración 36 GPS Desactivado</i>	xvi
<i>Ilustración 37 Mapa sin ubicaciones de los contactos</i>	xvi
<i>Ilustración 38 Página para descargar Eclipse y SDK Android</i>	xvii
<i>Ilustración 39 Requerimientos para descargar el paquete de Eclipse y SDK Android</i>	xviii
<i>Ilustración 40 Descarga del paquete Eclipse y SDK Android</i>	xviii
<i>Ilustración 41 Ruta para guardar los proyectos</i>	xix
<i>Ilustración 42 SDK Manager para descargar más herramientas</i>	xix
<i>Ilustración 43 Creación de un nuevo proyecto</i>	xx
<i>Ilustración 44 Ubicación del archivo activity_main.xml</i>	xx

<i>Ilustración 45</i> <i>activity_main.xml</i> _____	xx
<i>Ilustración 46</i> <i>AndroidManifest.xml</i> _____	xxi
<i>Ilustración 47</i> <i>Android Virtual Device Manager</i> _____	xxi
<i>Ilustración 48</i> <i>Configuración de la máquina virtual</i> _____	xxii
<i>Ilustración 49</i> <i>Botón para ejecutar la aplicación</i> _____	xxii
<i>Ilustración 50</i> <i>Iniciando la máquina virtual de Android</i> _____	xxii
<i>Ilustración 51</i> <i>Ejecutando el proyecto</i> _____	xxiii
<i>Ilustración 52</i> <i>Código clase SplashActivity</i> _____	xxiv
<i>Ilustración 53</i> <i>Código clase ContactosActivity</i> _____	xxv
<i>Ilustración 54</i> <i>Código clase Login</i> _____	xxv
<i>Ilustración 55</i> <i>Código clase Mapa</i> _____	xxvi

TABLAS.

<i>Tabla 1</i> <i>Redes de corto alcance</i> _____	11
<i>Tabla2</i> <i>Redes de largo alcance</i> _____	14
<i>Tabla3</i> <i>Generación de celulares</i> _____	17
<i>Tabla4</i> <i>Clasificación de dispositivos móviles</i> _____	18
<i>Tabla 5</i> <i>Versiones del sistema operativo Andrid</i> _____	21
<i>Tabla 6</i> <i>Aplicaciones de localización</i> _____	24
<i>Tabla 7</i> <i>Fuentes de Información de requerimientos</i> _____	27
<i>Tabla 8</i> <i>Tipos de requerimiento</i> _____	30
<i>Tabla 9</i> <i>Matriz de prueba</i> _____	42
<i>Tabla 10</i> <i>Resultados de la pruebas</i> _____	57

CAPÍTULO 1

INTRODUCCIÓN

PRIMERA PARTE

1. INTRODUCCIÓN

Los dispositivos móviles son cada vez más utilizados por nuestra sociedad y permiten que las personas tengan un mayor acceso a la información, por ende los servicios para los dispositivos móviles deben satisfacer los requerimientos de los usuarios.

Hace más de treinta años apareció la primera computadora móvil, a partir de entonces la tecnología móvil ha evolucionado muy rápido, dando lugar a nuevos dispositivos como celulares, Palm, Low-end Smartphones, High-end Smartphones, Netbooks y Tablet. Con el paso del tiempo estos dispositivos han sido de gran utilidad, no solo como una herramienta de trabajo sino también como un modo de comunicación por lo que se ha vuelto indispensable para un gran sector de la población.

Actualmente los sistemas operativos que existen para dispositivos móviles son Palm SO, Symbian, Windows Mobile, iOS y Android. Este último a diferencia de los demás es un sistema operativo abierto que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo por ello es que la aplicación desarrollada en esta tesis estará basada en Android.

La aplicación que se desarrolló junto con este trabajo permitirá, mediante la localización GPS, ubicar dispositivos móviles con sistema Android; así mismo, tener la localización aproximada de nuestros amigos dentro de Ciudad Universitaria que también dispongan de dispositivos compatibles. Además de evitar el uso de mapas cartográficos con licencia, se desarrollará un mapeo geográfico limitado a Ciudad Universitaria. Al ser una aplicación Web no será necesario realizar una instalación previa de un programa dado que la aplicación será alojada en un servidor Web gratuito para disminuir los costos de desarrollo.

1.1 DEFINICIÓN DEL PROBLEMA

Los estudiantes de Ciudad Universitaria de la ciudad de México tienen la necesidad de mantener comunicación constante con sus compañeros de carrera, amigos o con profesores, para resolver una tarea o realizar un trabajo. Actualmente existe una gran variedad de formas de comunicación, pero involucran algunos inconvenientes como el costo o la demora de una respuesta y es por ello que se pensó en desarrollar una aplicación en la plataforma Android para que el usuario pueda localizar a sus contactos de forma que no le tenga un costo.

1.2 OBJETIVO GENERAL

Desarrollar una aplicación web para dispositivos móviles con sistema operativo Android que permita localizar de una forma rápida dispositivos mediante GPS dentro de Ciudad Universitaria. Esto para conocer la ubicación, lo más aproximada posible, de nuestros amigos o conocidos y a su vez poderlos clasificar en grupos específicos. Así mismo la aplicación solo es una herramienta adicional a las diferentes formas de comunicación que ya existen y por ello no tendrá costo alguno, ya que la aplicación a desarrollar solo pedirá que se conecte a internet.

1.3 OBJETIVOS ESPECÍFICOS

- Mapeo cartográfico de Ciudad Universitaria.
- Instalación y configuración de un servidor de base de datos para alojamiento de mapas cartográficos.
- Implementación de aplicaciones en dispositivos móviles.
- Difusión de la aplicación para lograr el mayor número de usuarios.

1.4 LIMITANTES

- Al ser una aplicación móvil gratuita, se catalogará como Software libre¹ por ello no se tendrá un control en cuanto a la cantidad de dispositivos que instalarán la aplicación.
- Por viabilidad técnica solo se hará el mapeo cartográfico² de Ciudad Universitaria de la Ciudad de México, ya que no se cuenta con recursos para realizar un mapeo extenso.
- Los dispositivos móviles deberán contar con acceso a datos y GPS³ por lo tanto no todos los dispositivos móviles tendrán acceso a la aplicación.

¹ Los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

² Son mapas que muestran la altitud, escalas, latitudes y contexto geográfico.

³ Sistema de Posicionamiento Global es un sistema de navegación basado en satélites

1.5 RESULTADOS ESPERADOS

Optimizar la comunicación entre los estudiantes de Ciudad Universitaria, sin suplantar otros medios de comunicación, para ello, los usuarios con Android instalarán la aplicación que previamente descargarán de un sitio web habilitado para tales efectos. Cada usuario obtendrá la ubicación aproximada de sus amigos dentro de Ciudad Universitaria y a su vez podrá clasificarlos en grupos específicos, esto para tener una organización y poder agilizar la búsqueda de los contactos.

1.6 METODOLOGÍA DEL TRABAJO

Una metodología para el desarrollo de software es un conjunto de procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado⁴.

Un modelo de desarrollo son aquellas estrategias que ayudan a organizar las diferentes etapas y actividades del ciclo de vida del software; estos modelos ayudan al control y a la coordinación del proyecto, y dependerá del tipo de proyecto el modelo a elegir.

Las principales diferencias entre distintos modelos de ciclo de vida están divididas en tres grandes grupos:

- **El alcance de ciclo de vida:** saber si es viable el desarrollo de un producto, el desarrollo completo o el desarrollo completo más las actualizaciones y el mantenimiento.
- **La cualidad y cantidad de etapas:** las etapas en las que se dividirá el ciclo de vida, según el que se adopte y el proyecto para el cual se adopte.
- **La estructura y sucesión de las etapas:** si existe retroalimentación entre ellas, y si existe la libertad de repetirlas (iterar).

Una vez analizadas las ventajas y desventajas de los modelos de programación existentes se propone a utilizar el modelo Incremental en el desarrollo del presente trabajo

⁴ Dante Cantone. (Septiembre 30, 2006). Implementación y Debugging. Creative Andina Corp.

El modelo incremental se realiza construyendo módulos que cumplen las diferentes funciones del sistema. Esto permite ir aumentando gradualmente las capacidades del software. Facilita la tarea del desarrollo permitiendo a cada miembro del equipo desarrollar un módulo particular en el caso de que el proyecto sea realizado por un equipo de programadores.

Es una repetición del modelo en cascada cuyo enfoque es metodológico de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior aplicándose este ciclo en cada funcionalidad del programa a construir. Al final de cada ciclo se entrega una versión al cliente que contiene una nueva funcionalidad.

El modelo no está diseñado para cierto tipo de aplicaciones, sino que está orientado a cierto tipo de usuario o cliente. Es verdaderamente útil cuando el usuario necesite entregas rápidas, aunque sean parciales.

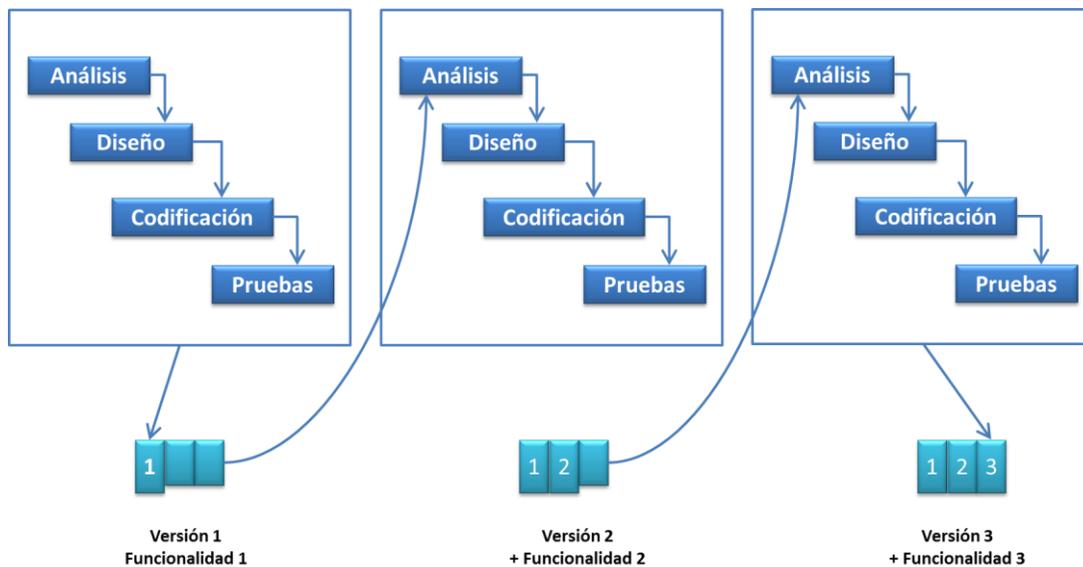


Ilustración 1 Modelo Incremental

CAPÍTULO 2

MARCO TEÓRICO

2. MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

La ingeniería de software marca los procedimientos teóricos y prácticos que se deben de seguir para desarrollar software útil y costeable. Esta disciplina nació en 1968 como propuesta a la llamada *Crisis del software*; desde entonces se lograron grandes avances para mejorar el análisis, la calidad, el tiempo de entrega y el costo del software.

Se denominó *Crisis del software* al conjunto de problemas asociados al cómo desarrollar el software. Los principales problemas eran:

- Planificación deficiente. No se estimaban los tiempos correctamente y los proyectos llegaban a tener años de retraso.
- Costo elevado. Derivado de una mala planificación, los costos se disparaban inimaginablemente por lo tanto los proyectos se quedaban trancos al no ser financiados.
- Mala calidad por una incorrecta definición, se provocaba la insatisfacción y falta de confianza de los clientes.

El software se trata de un elemento lógico, por ello “*el éxito de este se mide por calidad de una única entidad en vez de por muchas entidades fabricadas*”⁵.

2.1.1 Modelado de sistemas

El modelado de sistemas consiste en dividir el sistema en subsistemas por medio de un diagrama de bloques indicando con flechas las relaciones que pueden ser de flujo de datos.

El modelado de sistemas ha sido utilizado para identificar si los subsistemas se pueden desarrollar en paralelo y a su vez identificar los tiempos disponibles de estos subsistemas. Una de las ventajas que se puede obtener al realizar el sistema en paralelo es ahorrar tiempo en entrega del producto.

⁵ Roger S. Pressman. (Marzo 4, 2010). Ingeniería del Software. McGraw-Hill

2.1.2 Ciclos de Vida del Software

El ciclo de vida es un conjunto de fases sucesivas por las cuales pasa el software al ser diseñado, desarrollado y hasta la finalización de su uso. Cada fase se construye agrupando determinadas tareas involucradas en el sistema. Los entregables son los productos intermedios que permiten evaluar el proceso del software, estos pueden ser documentos o software.

Las fases del ciclo de vida son:

- Inicio
- Planificación
- Ejecución
- Cierre del proyecto

En cada fase se debe de definir las siguientes cuestiones

- El trabajo técnico que se debe de realizar
- Cuando se deben generar los productos entregables
- Quienes están involucrados
- Cómo controlar y aprobar

2.1.3 Verificación y validación de software

La verificación y validación de software es una disciplina técnica de la Ingeniería de Software la cual tiene como propósito ayudar a las organizaciones de desarrollo a construir software de calidad durante el ciclo de vida del software.

El término de verificación se refiere a si se está construyendo bien, por ello se debe de realizar la pregunta “¿Se está construyendo bien?”. Se deben de realizar pruebas para verificar que el sistema va por buen camino y tenga el mínimo de errores. Estas pruebas pueden ser en cada fase o al finalizar el producto esto se hace dependiendo del ciclo de vida del mismo.

La validación hace referencia a si el producto satisface los requerimientos establecidos por el cliente. En cada fase del ciclo de vida debe de haber entregables, ya sea en documentación o software, se deben de revisar junto con el cliente, para saber si cumple con sus expectativas.

2.1.4 Administración de proyectos de software

La administración de proyectos de software consiste en gestionar el desarrollo del producto con el plazo y fondos establecidos. Dentro de la administración se deben de considerar los siguientes elementos.

- Estructura
- Proceso Administrativo
- Proceso de Desarrollo
- Organización de los tiempos

Así mismo debe de controlar los siguientes factores

- El costo
- Las capacidades
- La calidad
- La duración

2.2 BASES DE DATOS RELACIONALES

La evolución de la tecnología nos ha llevado a pasar la información de una hoja de papel, a archivos electrónicos, ya que recibimos y manejamos volúmenes enormes de información y nos es necesario obtenerla de manera rápida y segura, sin que se pierdan datos. Una base de datos es un conjunto de datos interrelacionados almacenados en conjunto.

También existen bases de datos multimedia o geográficas, en las cuales no solo se almacena texto plano sino que se almacenan dibujos, videos o mapas, datos del tiempo e imágenes satelitales.

La BD se almacena en un sistema denominado Sistema Gestor de Bases de Datos este sistema debe de ser seguro, concurrente e integral. La seguridad hace referencia a la protección de la base de datos frente a los accesos no autorizados. El gestor debe de ser concurrente para permitir la ejecución de los procesos en paralelo, accediendo a la información compartida y la integridad hace referencia a la coherencia de los datos almacenados.

Las ventajas de las bases de datos son:

- Control de redundancia de datos.
- Coherencia de datos.
- Mejor accesibilidad de los datos y mayor capacidad de respuesta.
- Mayor productividad.
- Mantenimiento simplificado.
- Mayor nivel de concurrencia
- Simplifica la comunicación.
- Permite la seguridad.
- Garantiza la integridad.

Las desventajas que presentan son:

- Complejidad.
- Tamaño.
- Costo del DBGS.
- Costo del hardware adicional.
- Costos de conversión.
- Mayor impacto de los fallos.

2.3 REDES CELULARES

Una comunicación se considera como móvil cuando al menos de uno de los receptores se desplaza, requiriendo y manteniendo la comunicación durante su desplazamiento. Todas las aplicaciones móviles requieren conectividad inalámbrica.

Existen dos tipos de redes inalámbricas, las de corto y largo alcance, las primeras operan sobre un espectro sin licencia (2.4, 5 o 40 GHz), conectan edificios, casas, escuelas, etc. A su vez las redes de corto alcance se dividen en WPAN y WLAN están son redes inalámbricas, sus características de encuentran en la tabla 1.

Los espectros sin licencia son bandas de radiofrecuencia que se pueden utilizar sin licencia.

Tabla 1 Redes de corto alcance

<p>WPAN (Wireless Personal Area Networks)</p> <p>Área de cobertura: Espacio personal 10 metros. Función: Tecnología de remplazo de cables. Costo asociado: Muy bajo Velocidad típica: 0.1 a 4 Mbps Estándares: IrDA, Bluetooth, 802.15, Zigbee.</p>	<p>WLAN (Wireless Local Area Networks)</p> <p>Área de cobertura: Edificios, campus 100 metros. Función: Extensión o alternativa de LAN's con cable. Costo asociado: Bajo-medio Velocidad típica: 1-100 Mbps Estándares: 802.11a/ b/g/n, HIPERLAN/2</p>
<p>IrDA (Infrarrojo)</p> <ul style="list-style-type: none"> • Tecnología infrarroja desarrollada por Infrared Data Association. • Dos Protocolos: IrDA DATA (para la entrega de datos), IrDA CONTROL (mecanismo de control) • Rango de comunicación: 1 a 2 metros • Opción de baja potencia • Comunicación bidireccional • 9600 bps a 4 Mbps 	<p>WiFi (WirelessFidelity) 802.11b</p> <ul style="list-style-type: none"> • Especificación aprobada por el IEEE⁶ en 1999 junto con 802.11^a • Apoyado por compañías • Funciona en 2.4 GHz • Congestión de espectro (hornos, teléfonos inalámbricos, transmisores A/V, etc) • Difícil de hacer hand-off con AP's de varios fabricantes • Primer esquema de seguridad WEP (Wireless Equivalent Privacy) muy vulnerable • Actualmente se tienen esquemas más seguros como WPA y WPA2 (Wireless Protected Access)

⁶ Instituto de Ingeniería Eléctrica y Electrónica es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

<ul style="list-style-type: none"> • Bajo costo • Conexión en línea de vista 	<ul style="list-style-type: none"> • Velocidad máxima: 11 Mbps • Gran éxito comercial • Rango típico: 100 metros
<p>Bluetooth</p> <ul style="list-style-type: none"> • Creada por Ericsson en 1994. • Actualmente lo maneja el SIG (Bluetooth Special Interest Group) • No requiere línea de vista • Rango típico: 1 a 10 metros, con amplificadores puede llegar a más. • Modo de autodescubrimiento • Problema de sobrecarga de espectro • Voz y datos • Gran soporte de las marcas • Velocidades típicas de 720 kbps a 2 Mbps 	<p>WiFi (WirelessFidelity) 802.11a</p> <ul style="list-style-type: none"> • Alternativa de alta velocidad para WLAN's aprobada en 1999 • Velocidad de 54 Mbps • Utiliza el espectro de 5GHz (no disponible en todos los países) • No compatible con el estándar 802.11b por diferencias en modulación • Espectro no tan congestionado • Menor rango de 25 a 50 metros • Equipos más costosos • Mayor consumo de energía • Sin éxito comercial
<p>802.15</p> <ul style="list-style-type: none"> • Grupo de trabajo del IEEE 802 especializado en WPAN's • IEEE 802.15.1-2002 derivó en Bluetooth V1.1 • Posteriormente se deslindó del Bluetooth SIG • Coexiste con tecnologías 802.11 • Actualmente continúan trabajando • Las tecnologías desarrolladas pueden servir de base para otras WPAN's. 	<p>WiFi (WirelessFidelity) 802.11g</p> <ul style="list-style-type: none"> • Compatible con 802.11b • Frecuencia de 2.4 GHz • Hasta 54 Mbps sólo usando AP's y tarjetas 802.11g • AP 802.11g => Tarjeta 802.11b • AP 802.11b => Tarjeta 802.11g
<p>ZigBee</p> <ul style="list-style-type: none"> • Basada en el estándar IEEE802.15.4-2003 • Bajo consumo de energía • Se está enfocando a la domótica (automatizar viviendas) • 868 MHz en Europa, 915 MHz en EUA y 2.4 GHz en casi todo el mundo • 250 kbps • Rango típico: 10 a 75 metros 	<p>WiFi (WirelessFidelity) 802.11e, 802.11f, 802.11h, 802.11i</p> <ul style="list-style-type: none"> • Se usan para cumplir o mejorar propósitos en específico para la familia 802.11x • Algunos aún están en espera de certificación por el IEEE

	<p>WiFi (WirelessFidelity) 802.11n</p> <ul style="list-style-type: none"> • Compatible con 802.11b/g • Frecuencia de 2.4 o 5 GHz • Velocidad cuatro veces aproximadamente de 802.11b/g (600Mbps teórico) • Velocidades típicas: 80 a 100 Mbps • Rango aproximado: 400 metros • No pasó la primera votación en Mayo de 2006 • Su punto crítico fue la interferencia • Por su tardanza en ser ratificado salió al mercado el borrador conocido como Draft-n • Ratificado finalmente en Septiembre de 2009
	<p>WMAN: WiMax⁷</p> <ul style="list-style-type: none"> • Worldwide Interoperability for Microwave Access • También conocido como la familia 802.16 • Existen varios con el 802.16a (2 a 11 GHz), 802.16b (10 a 66 GHz), 802.16d (2 a 11 GHz) • El 802.16d se usa para conexiones fijas • El 802.16e se usa para conexiones móviles. Tiene mejoras cobertura, consumo de energía y rehusó de frecuencias. • 802.16m: Extensión para alcanzar hasta 1Gbps en comunicación fija. • Rango de más de 100 km • No hay espectro uniforme globalmente aceptado

Y finalmente están las redes de largo alcance, estas operan sobre un espectro con licencia, utiliza proveedores de servicio (carriers u operadores) y conectan áreas metropolitanas, estados y países, sus características se encuentran en la tabla 2.

⁷ Es una norma de transmisión de datos que utiliza las ondas de radio en las frecuencias de 2,3 a 3,5 GHz y puede tener una cobertura de hasta 50 km.

Tabla2 Redes de largo alcance

<p>WWAN Wireless Wide Area Networks</p> <ul style="list-style-type: none"> • Área de cobertura: Nacional, La proporcionan operadores • Función: Extensión de LAN's • Costo asociado: Medio-alto • Velocidad típica: 8 kbps a 10 Mbps • Estándares: GSM, CDMA IX EV-DO, UMTS, etc 	<p>Redes Satelitales⁸</p> <ul style="list-style-type: none"> • Área de cobertura: Cobertura continental o global • Función: Extensión de LAN's • Costo asociado: Muy alto • Velocidad típica: 2 kbps a 2 Mbps • Estándares: TDMA, CDMA, FDMA, Iridium
---	--

2.4 SISTEMA SATELITAL DE POSICIONAMIENTO (GPS)

El GPS fue una implementación del programa NAVSTAR, GPS (Navigation System Timing And Ranging, Global Positioning System), concebido para uso militar, hoy en día ya es un servicio público de gran importancia. Este es el sistema satelital que brinda mayores y mejores prestaciones y tiene como objetivo la determinación de coordenadas de puntos estáticos o en movimiento de un sistema de referencia mundial.

Para la obtención de las coordenadas del receptor se necesita como mínimo cuatro señales de satélites, conocer la posición orbital de los satélites con respecto a la Tierra, el tiempo empleado para que la señal llegue al receptor, y con estos datos se realizan cálculos de triangulación.

El sistema GPS está constituido por tres segmentos

- Espacial
- De control
- Del usuario

El segmento espacial está conformado por 24 satélites Navstar que se encuentran en seis planos orbitales alrededor de la Tierra, con una inclinación de 55° respecto al plano del Ecuador y una distribución aproximadamente uniforme, tienen un promedio de 15 años de vida, desde 1978 se han lanzado cuatro generaciones de satélites, cada generación sustituye al anterior, mejorando su funcionamiento y características.

Los satélites GPS emiten de forma permanente señales con datos de su posición orbital, hora exacta de la emisión de las señales y la posición de los otros satélites GPS, los datos son transmitidos en forma de ondas electromagnéticas con frecuencia de microondas entre 1.6 y 1.2 GHz.

⁸ Es una conexión a internet utilizando como medio de enlace un satélite.

El segmento de control tiene las siguientes funciones:

- Monitoreo y control permanente de los satélites
- Sincronización de los relojes de los satélites con el tiempo GPS
- Transmisión, a cada satélite de la información procesada

Este segmento está integrado por una *Estación de Control Maestra (MCS)*, varias *Estaciones de monitoreo (MS)* y *Antenas Terrestres (GA)*.

Su función de las estaciones de monitoreo es determinar las distancias de los satélites visibles y transmitirlos a la estación de control maestra. Esta última se encuentra ubicada en la Base de la Fuerza Aérea Schriever en el estado de Colorado y su función es calcular los parámetros orbitales y los de los relojes, con los datos recibidos de las estaciones de monitoreo, posteriormente los transmite a las antenas terrestres que los transfieren a los satélites.

El segmento de control está integrado por diez estaciones ubicadas en:

- Colorado Springs (EUA)
- Isla Ascensión (Atlántico Sur)
- Diego García (Índico)
- Kwajalein (Pacífico Occidental)
- Hawaii (Pacífico Oriental)
- Quito (Ecuador)
- Buenos Aires (Argentina)
- Hermitage (Inglaterra)
- Bahrein (Golfo Pérsico)
- Smithfield (Australia)

El segmento de usuario está constituido por instrumentos que pueden captar y procesar la señal emitida por los satélites, estos instrumentos están constituidos por una antena y un receptor.

El receptor varía entre diferentes modelos, pero suele disponer de la siguiente información:

- Satélites localizados
- Satélites en seguimiento
- Intensidad de cada señal recibida
- Condición de cada satélite en seguimiento
- Posición : longitud, latitud, altitud
- Calidad de la geometría de observación.

Dependiendo la precisión con que se obtengan los resultados, los receptores se clasifican en:

- Geodésicos.- Es una señal informativa permanente que podemos encontrar en el campo, que nos indica la altura exacta de ese punto sobre el nivel del mar, y que forma parte de una red de triángulos cuyas coordenadas se han calculado con la mayor precisión posible.
- Topográficos.- Utilizan el sistema de representación de planos acotados, mostrando la elevación del terreno utilizando líneas que conectan los puntos con la misma cota respecto de un plano de referencia, denominadas curvas de nivel.

2.5 DISPOSITIVOS MÓVILES INTELIGENTES

En 1973 se realizó la primera llamada por celular, quién la realizó fue el ingeniero eléctrico del Instituto de Tecnología Illinois ejecutivo de Motorola Martin Cooper, tomó su DynaTAC 8000X y llamó a Engel, jefe de Investigación y Desarrollo de la compañía rival Bell Laboratories, para informarle desde donde le estaba marcando.

En 1947 ya se había establecido el concepto de comunicaciones móviles por los Bell Laboratories, pero por dificultades con los permisos de espectro de radio, la idea estaba truncada. Hasta el año de 1983 se obtuvo el permiso de poner en el mercado el primer celular DynaTAC, pesaba 780 gramos y medía 33 x 9 x 4.5. Este mismo año fue implementado el primer sistema celular, donde las antenas cubrían 15 cuadras a la redonda, para que la llamada tuviera continuidad cuando el usuario se desplazaba físicamente, la central transportaba la señal de antena en antena.

El ingeniero Martin Cooper no se iba a imaginar que 40 años después de que realizó la primera llamada por celular, el teléfono móvil que inventó sería hoy en día un dispositivo móvil inteligente, es decir un teléfono móvil con una mini computadora integrada, o viceversa.

En 1993 IBM lanzó al mercado a Simon, un teléfono que no sólo realizaba llamadas también contaba con un asistente digital personal, ya que contaba con calendario, libreta de direcciones, reloj mundial, calculadora, bloc de notas y juegos. También se caracterizó por su pantalla táctil para marcar los números y el texto se ingresaba con un teclado QWERTY, pero su precio era muy elevado y solamente los altos ejecutivos podrían obtenerlo.

Posteriormente en 1996 salió a producción la Palm Pilot, no era un celular pero sí un dispositivo inteligente, que por ser portátil acostumbró a los usuarios a llevar sus datos de un lado a otro, era tecnología de última generación y popularizó las siglas PDA (Asistente Digital Personal).

En el 2002 RIM (Research In Motion) entró al mercado de los teléfonos móviles y lanzó su BlackBerry 5810 este móvil tenía la capacidad de navegar por internet y revisar el correo electrónico, su desventaja fue que para realizar llamadas se utilizaban auriculares.

Apple se introdujo al mercado de los móviles en el 2007, lanzando al mercado el iPhone, tuvo y ha tenido un éxito rotundo, caracterizado por su pantalla táctil y la mejor experiencia para navegar en internet, ese mismo año Google presentó su sistema operativo Android, que también ha tenido un gran éxito; haciendo que dispositivos de diferentes marcas optaran por tener este sistema operativo.

Los llamados Smartphone son teléfonos móviles inteligentes, el término inteligente hace referencia a la capacidad del dispositivo de poder ser usado como una computadora de bolsillo, cabe referirse que no está reemplazando a la computadora, sólo realiza funciones básicas para el usuario; por ejemplo:

- Llamadas telefónicas
- Envío de mensajes SMS
- Conectarse a una red inalámbrica
- Navegar en la internet
- Editar documentos
- Tomar fotos y videos
- GPS
- Y el uso de aplicaciones con diferentes funcionalidades

En la tabla 3 se muestran las generaciones de los celulares

Tabla3 Generación de celulares

Primera Generación (1G)	<ul style="list-style-type: none"> • AMPS(Advanced Mobile Phone Service) • Analógico • Sólo servicios de voz • Ineficiente • Poca capacidad (usuarios simultáneos) • Técnica de acceso al medio
Segunda Generación (2G y 2.5G)	<ul style="list-style-type: none"> • PCS (Personal Communications Services) 1,900 MHz • Digital • Mayor capacidad (usuarios simultáneos) en el mismo espectro • Mayor duración de la batería • Mejor calidad de comunicación • Incremento de la seguridad • Tecnologías: CDMA One (IS 95), TDMA, GSM • Velocidades 2G: 9.6 a 64 kbps

<p>Tercera Generación (3G)</p>	<ul style="list-style-type: none"> • Mayor ancho de banda • Mayor eficiencia • Velocidades superiores a 144 kbps • WCDMA(UMTS: Universal Mobile Telecommunications System) Canal de 5 MHz (en 2G CDMA tenía canales de 1.25 Mhz y TDMA 30 kHz) • CDMA 2000 1x, EV-DO, EV-DV
--------------------------------	--

La tabla 4 muestra una clasificación de los dispositivos móviles

Tabla4 Clasificación de dispositivos móviles

<p>Web enabled phones</p> 	<ul style="list-style-type: none"> • Uso principal: voz • Display limitado (color o monocromático) • Profundidad de colores limitada • Gran cantidad de usuarios • Buena duración de la batería • Poder de procesamiento muy limitado • Descarga de datos lenta • Aplicaciones de mensajería
<p>Low-end Smartphone</p> 	<ul style="list-style-type: none"> • Uso principal: voz • Similares en forma y tamaño a los Web-enabled phones • Proven la capacidad de ejecutar aplicaciones de terceros • Simplicidad en su configuración • Descarga de aplicaciones vía OTA (over the air) • Buena duración de la batería
<p>Palm-sized PDA's</p> 	<ul style="list-style-type: none"> • Pantallas sensibles al tacto • Displays más grandes (desde 320 x 240) • Módems integrados o con la posibilidad de agregarlos • Procesadores ARM o X Scale a 416, 520, 624 MHz • Mayor consumo de batería

<p>High-end Smartphone</p> 	<ul style="list-style-type: none"> • Dirigidos en un inicio al mercado corporativo • Enfocados a comunicación por voz y a un uso intenso de datos • Aplicaciones con arquitectura de cliente inteligente • Browsers para WML y HTML • Duración de la batería muy limitada • Form-factor muy similar a un PDA
<p>Hand held PC´s</p> 	<ul style="list-style-type: none"> • Diseño similar a un Laptop • También conocidas como Clam Shell Computers por su forma • Apareció primero que los PDA´s • Teclado completo tipo Qwerty • Uso principal: captura y edición de datos • Prácticamente desplazadas por los PDA´s • Sin éxito comercial
<p>Tablet PC´s, Netbooks, Laptops</p> 	<ul style="list-style-type: none"> • Poder de procesamiento similar a una computadora de escritorio • Displays grandes • Baterías limitadas • Actualmente con gran éxito comercial • Se predice que sus ventas sobrepasarán en breve a las computadoras de escritorio

2.6 TECNOLOGÍAS DE INFORMACIÓN EN DISPOSITIVOS MÓVILES

La tecnología ha evolucionado exponencialmente y dado esto, la información cada vez está más al alcance de nuestras manos; por ejemplo, la podemos encontrar en folletos, libros, revistas y periódicos, gracias al invento de Johannes Gutenberg, la imprenta. Así también el radio creado por Guillermo Marconi, uno de los medios más accesibles a la información y sin menospreciar a la televisión ideada John Bair. Pero el mayor impacto que ha tenido la tecnología hasta nuestros días es la creación del internet.

Con el internet expandiéndose a varios países y evolucionando rápidamente, las TIC's deben de adaptarse e ir ampliando horizontes para optar por nuevas formas de educación e información confiable, ya que no todo lo que encontramos en el internet es verídico pero actualmente es más fácil buscar información en la web que en un libro.

Ahora con la conexión inalámbrica a la web ya sea con WI-FI o 3G, la información se descarga en dispositivos móviles, se puede por ejemplo revisar el periódico desde tu celular o Tablet en cualquier momento o lugar donde tengas acceso a la red. De igual manera puedes observar la TV y escuchar la radio desde tu dispositivo móvil, prácticamente la información y comunicación se encuentra al alcance de nuestras manos. Cuando aparecieron los primeros celulares se creía que el tener uno de esos dispositivos era un lujo no una necesidad, ahora los tiempos han cambiado y para las personas se ha vuelto una necesidad ya que el usuario prefiere la movilidad.

Hasta Marzo del 2013 en México había un 101.7 millones de usuarios en telefonías móviles; es decir, la penetración es del 86.2 % de la población y 12 millones de conexiones a internet contratadas. Según la firma Competitive Intelligence, pronostica que terminando este año llegara a los 34.7 millones de conexiones.

El primer trimestre según la firma IDC los envíos de teléfonos convencionales cayeron un 30 %, pronto habrá un 50 % de estos teléfonos contra un 50 % de Smartphone.

Podemos destacar que al evolucionar la tecnología y la accesibilidad a ésta, también se debe de evolucionar la forma de pensar, así como los métodos de aprendizaje, deben apoyarse a los nuevos dispositivos móviles llamados smartphone.

2.7 SISTEMA OPERATIVO ANDROID

Un sistema operativo es un software que actúa como intermediario entre el usuario de la computadora y el hardware de la misma.

Android es un sistema operativo para móviles además de una plataforma de software basada en el núcleo de Linux, es una plataforma de código abierto que permite el desarrollo de las aplicaciones por terceros a través del SDK⁹, proporcionada por Google, y mediante el lenguaje de programación java.

Android fue presentado por Google el 5 de Noviembre del 2007 y en el 2008 fue lanzada la primera versión comercial. Desde ese año Android ha sufrido muchos cambios, novedades y actualizaciones.

⁹ Software Development Kit.- es un conjunto de herramientas de desarrollo de software.

A partir de la versión 1.5 las versiones de este sistema operativo son llamadas por orden alfabético y con nombre de postres en inglés las cuáles son Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, JellyBean y la última versión que fue lanzada en Octubre del 2013 es KitKat. Las características de las versiones de Android se muestran en la tabla 5.

Tabla 5 Versiones del sistema operativo Andrid

Versión	Fecha de Lanzamiento	Principales Características
1.0	23 de Septiembre del 2008	<ul style="list-style-type: none"> • Android Market • Navegador Web • Google Maps • Google Calendar • Google Contacs • Youtube
1.1	9 de Febrero del 2009	<ul style="list-style-type: none"> • Mejoras al uso de los mapas • Capacidad de almacenar el contenido adjunto en mensajes
1.5 Cupcake	30 de Abril del 2009	<ul style="list-style-type: none"> • Soporte para Widgets¹⁰ • Capacidad de subir videos a Youtube • Capacidad de subir videos a Picassa • Características de copiar y pegar en el navegador web • Transiciones de pantalla animadas
1.6 Donut	15 de Septiembre del 2009	<ul style="list-style-type: none"> • Capacidad de funcionar en distintas resoluciones de pantalla • Búsqueda universal • Buscador por voz • Mejoras a la interfaz de usuario.

¹⁰ Son aplicaciones o programas que dan fácil acceso a funciones frecuentemente usadas.

Versión	Fecha de Lanzamiento	Principales Características
2.0 Eclair	26 de Octubre del 2009	<ul style="list-style-type: none"> • Permite añadir distintas cuentas de correo electrónico • Sincronización automática de contactos • Integración de Google maps a la información de los contactos • Fondos de pantalla animados
2.2.x Froyo	20 de Mayo del 2010	<ul style="list-style-type: none"> • Rediseño de la pantalla de inicio • Mejoras en la velocidad y desempeño • Soporte de Android Cloud to Device Messaging • Permite el uso del dispositivo como un punto de acceso portátil • Capacidad de almacenar Apps en memorias SD • Mejoras al Market
2.3.x Gingerbread	6 de Diciembre del 2010	<ul style="list-style-type: none"> • Incremento a la velocidad y simplicidad de la interfaz de usuario • Soporte nativo para los sensores del dispositivo • Recogedor de basura concurrente para aumentar el desempeño • Mejoras al manejo de la energía • Escáner NFC

Versión	Fecha de Lanzamiento	Principales Características
3.x Honeycomb	22 de Febrero del 2011	<ul style="list-style-type: none"> • Grandes cambios a la interfaz de usuario • Inclusión de la Systembar y la ActionBar • Rediseño del teclado en pantalla • Soporte de video Chat usando Google Talk • Conectividad para accesorios USB • Botones virtuales en lugar de botones físicos.
4.0.x Ice CreamSandwich	19 de Octubre del 2011	<ul style="list-style-type: none"> • Mejoras al rendimiento y la velocidad • Botones virtuales en lugar de botones físicos • FaceUnlock¹¹ • Mejoras a la cámara • AndroidBeam¹²
4.xJellyBean	Lanzado en Octubre del 2012	<ul style="list-style-type: none"> • Mejora la fluidez y rapidez • Dictado por voz mejorado y sin la necesidad de estar conectado a Internet (Sólo para el idioma Ingles) • Transferencia de videos por medio de la conexión NTF¹³ • Mayor seguridad con el cifrado de aplicaciones

¹¹ Es una aplicación el cuál su función es desbloquear el dispositivo por medio del reconocimiento facial del usuario.

¹² Permite que los datos se transfieran a través de la comunicación de campo cercano (NFC).

¹³ Near Field Communication.- Es una tecnología inalámbrica que funciona en la banda de los 13.56 MHz

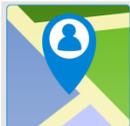
Versión	Fecha de Lanzamiento	Principales Características
4.4.x KitKat	Lanzado en Octubre del 2013	<ul style="list-style-type: none"> • Más conectividad • Mejoras en el NFC • Impresión de documentos por Wi-Fi • Gestión de almacenamiento • Detector y contador de pasos • Grabación de pantalla

Android es uno de los sistemas operativos móviles más usados, en competitividad se encuentran IOS, Black Berry, Windows Mobile y Symbian.

2.7.1 APLICACIONES DE LOCALIZACIÓN

Hoy en día existen infinidad de aplicaciones dedicadas a la localización de dispositivos móviles por medio del GPS, algunas de éstas con gran popularidad en Google Play¹⁴, estas se describen en la tabla 6.

Tabla 6 Aplicaciones de localización

Aplicación	Descripción	Propietario
Rastrear Celular 	Con más de 10,000 descargas Esta aplicación envía las posiciones GPS del dispositivo móvil a una cuenta de correo.	MAPME.NET
Alert. US 	Con más de 1,000 descargas Se envía alertas a otros dispositivos con el posicionamiento.	ALERT.US
Mi mapa de localización gps 	Con más de 100,000 descargas Inserta en un mapa, la localización actual del dispositivo.	ADCOMS

¹⁴ Tienda en línea de aplicaciones móviles para dispositivos con sistema operativo Android, desarrollada y operada por Google.

Aplicación	Descripción	Propietario
<p>Localización por GPS</p> 	<p>Con más de 10,000 descargas Comparte la ubicación del dispositivo con amigos, socios y colegas.</p>	<p>APPS TIME</p>
<p>Brújula con localización GPS</p> 	<p>Con más de 100,00 descargas Brújula con datos de posición y localización por GPS por coordenadas y Google Maps</p>	<p>ALVARO CANTEIRO</p>

CAPÍTULO 3

ANÁLISIS

3. ANÁLISIS

3.1 LEVANTAMIENTO Y ANÁLISIS DE REQUERIMIENTOS

En términos de Ingeniería de Software, según la IEEE un requerimiento se define:

- a) Condición o capacidad que necesita el usuario para lograr un objetivo o solucionar un problema.
- b) Una condición o capacidad que debe tener el sistema para satisfacer un contrato, estándar, especificación de Software u otro documento formal.

En el análisis del sistema existen fuentes de información de requerimientos para el desarrollo del sistema, éstas se describen en la tabla 7.

Tabla 7 Fuentes de Información de requerimientos

<i>Fuentes de Información</i>	<i>Descripción</i>
<i>Marketing</i>	Especificación de requerimientos de alto nivel, se revisa si es necesario hacer cambios a los requerimientos que actualmente maneja la empresa.
<i>Administración</i>	Proporciona requerimientos del negocio y parámetros importantes dentro de la lógica del negocio que deben ser tenidos en cuenta para el desarrollo de la solución.
<i>Aspectos legales</i>	Son las implicaciones legales que afectarían el desarrollo de la solución, así como las licencias de las herramientas de desarrollo y componentes adicionales.
<i>Analistas del negocio</i>	Dan a conocer las necesidades del negocio, las necesidades funcionales y de rendimiento. También evalúan si es necesario hacer cambios a los requerimientos que actualmente se plantearon en la empresa

<i>Fuentes de Información</i>	<i>Descripción</i>
<i>Software</i>	Hacen énfasis en los requerimientos de Software. También evalúan si es necesario hacer cambios a los requerimientos actualmente proyectados en la empresa.
<i>Hardware</i>	Especifican requerimientos a nivel de infraestructura y recursos físicos en los que se basará el desarrollo del sistema.
<i>Usuarios</i>	Describen requerimientos de usuarios y atributos a evaluar para los mismos.
<i>Soporte Técnico</i>	Dan asistencia a los usuarios en la descripción de los requerimientos del usuario. Buscan orientar lo que dice el usuario de una forma más técnica, facilitando así el entendimiento del equipo de desarrollo de la solución.

3.1.1 TIPOS DE REQUERIMIENTO

Según el área los requerimientos se clasifican para un mejor entendimiento del analista de sistemas. En la ilustración 2, se muestran los tipos de requerimientos, y en la tabla 8 se describe cada uno de ellos.

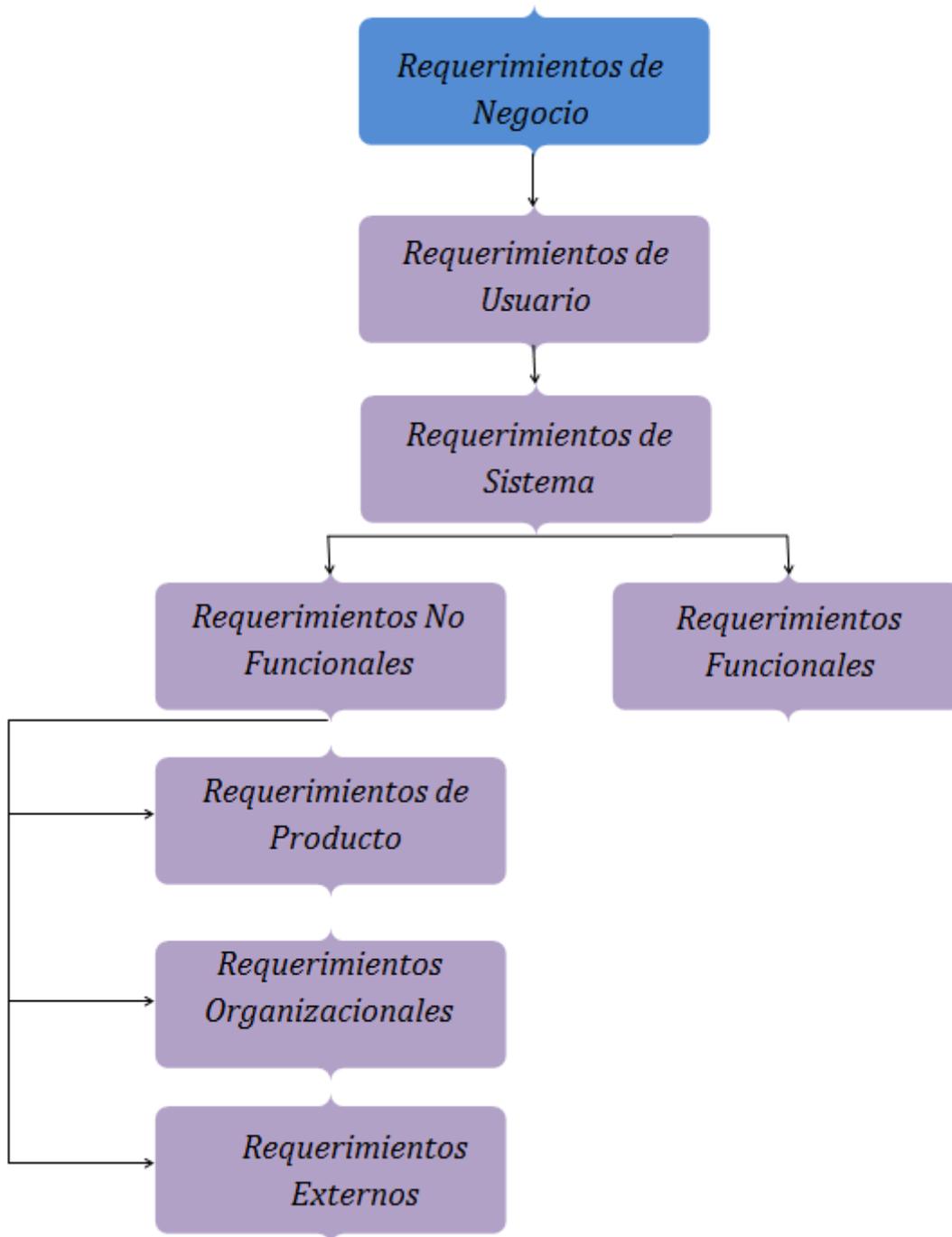


Ilustración 2 Tipos de requerimientos

Tabla 8 Tipos de requerimiento

NOMBRE	DESCRIPCIÓN
Requerimientos de Negocio	Describen las necesidades que el Software cubrirá sin descuidar las reglas de negocio de la organización.
Requerimientos de Usuario	Estos requerimientos son las funcionales específicas que los usuarios esperan ver en la solución.
Requerimientos de Sistema	Estos requerimientos acoplan los requerimientos de usuario al modelo de la solución, con estos requerimientos ya se tiene un cierto nivel avanzado y son la base para empezar el diseño del sistema. Estos hacen referencia hacia los requerimientos funcionales y no funcionales.
Requerimientos Funcionales	Describen lo que el sistema o software debe hacer. Estos siempre se refieren a funciones específicas que deben hacer la solución y su respectivo análisis en las definiciones de requerimientos de negocio y requerimientos del sistema.
Requerimientos No Funcionales	Los requerimientos no funcionales especifican propiedades del sistema como pueden ser la seguridad y la fiabilidad del sistema.
Requerimientos de Producto	Estos requerimientos especifican el comportamiento del producto, en este caso de todo el sistema. Como son la funcionalidad, fiabilidad, usabilidad, eficiencia y portabilidad del producto.

NOMBRE	DESCRIPCIÓN
Requerimientos Organizacionales	Son aquellos requerimientos que surgen de políticas y procedimientos de la organización, algunos de estos requerimientos son requerimientos de implementación y requerimientos de entrega.
Requerimientos Externos	Son aquellos requerimientos que se derivan de la interacción del sistema con el entorno que lo rodea, algunos son requerimientos de compatibilidad, requerimientos éticos y requerimientos legislativos.

El levantamiento de requerimientos, por ser una actividad enteramente humana, debe ser realizado por un ingeniero de software que disponga de una gran capacidad comunicativa con el cliente para poder definir lo que realmente se necesita y lo que se desea, esto es el alcance real para determinar hasta qué punto se puede realizar el trabajo deseado.

3.2 MODELADO DEL SISTEMA

Los modelos del sistema son representaciones graficas que describen el problema a resolver del sistema, los procesos del negocio y el sistema que se va a desarrollar, son una abstracción del sistema que hace que se resalte y simplifiquen las características más relevantes del sistema, por ello el modelo omite los detalles.

Estos modelos sirven para una mejor comprensión de los requerimientos solicitados por los usuarios que no son técnicos expertos. Se centran en lo que tiene que hacer el sistema y no en cómo lo debe de hacer, también son un puente entre el análisis del sistema y el diseño.

El modelo ayuda a que el analista entienda la función, la información y el comportamiento del sistema, se convierte en la clave para la determinación de la función y la eficacia de la especificación, así mismo el modelado es la base del diseño, proporcionando una representación esencial del software.

Dependiendo el modelo a utilizar es el tipo de abstracción que se le realiza al sistema, no hay un modelado específico para cada sistema, el modelo a utilizar depende de las preferencias personales o de la empresa.

3.2.1 DIAGRAMAS DE FLUJO DE DATOS

Los diagramas de flujo de datos son un modelo que gráficamente muestra el flujo de datos, es decir el proceso de los datos, así como las transformaciones y modificaciones que se realizan en los datos.

La sencillez de la notación de los diagramas de flujo hace que sean más utilizados para un análisis de sistemas estructurados. En la siguiente figura se presentan algunas de las notaciones de los diagramas de flujo de datos.

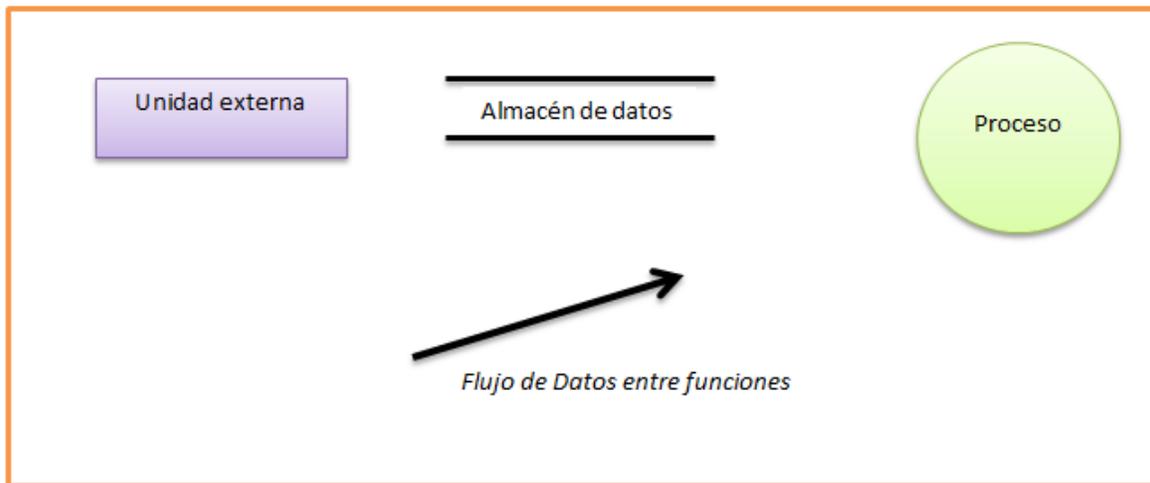


Ilustración 3 Componentes del Diagrama de Flujo

A los diagramas de flujo de datos también se les conoce como diagramas de burbujas, los DFD (diagramas de flujo de datos) puede tener varios niveles, el nivel cero es una burbuja el cual muestra el sistema completo con sus conectores a otras funciones o sistemas.

El DFD puede ser una herramienta valiosa para el análisis de requerimientos del software, no se debe confundir los diagramas de flujo con los DFD ya que los primeros muestran una lógica de como estructurar el sistema y los DFD muestra la representación del flujo de los datos sin entrar a la parte lógica.

3.2.2 CASOS DE USO

Los casos de uso son una técnica basada en escenarios se utilizan para describir modelos de sistemas orientados a objetos. Un conjunto de casos de uso representan todas las posibles interacciones a representar en los requerimientos del sistema.

Un diagrama de casos de uso sirve para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas y consta de los siguientes elementos representados en la figura.

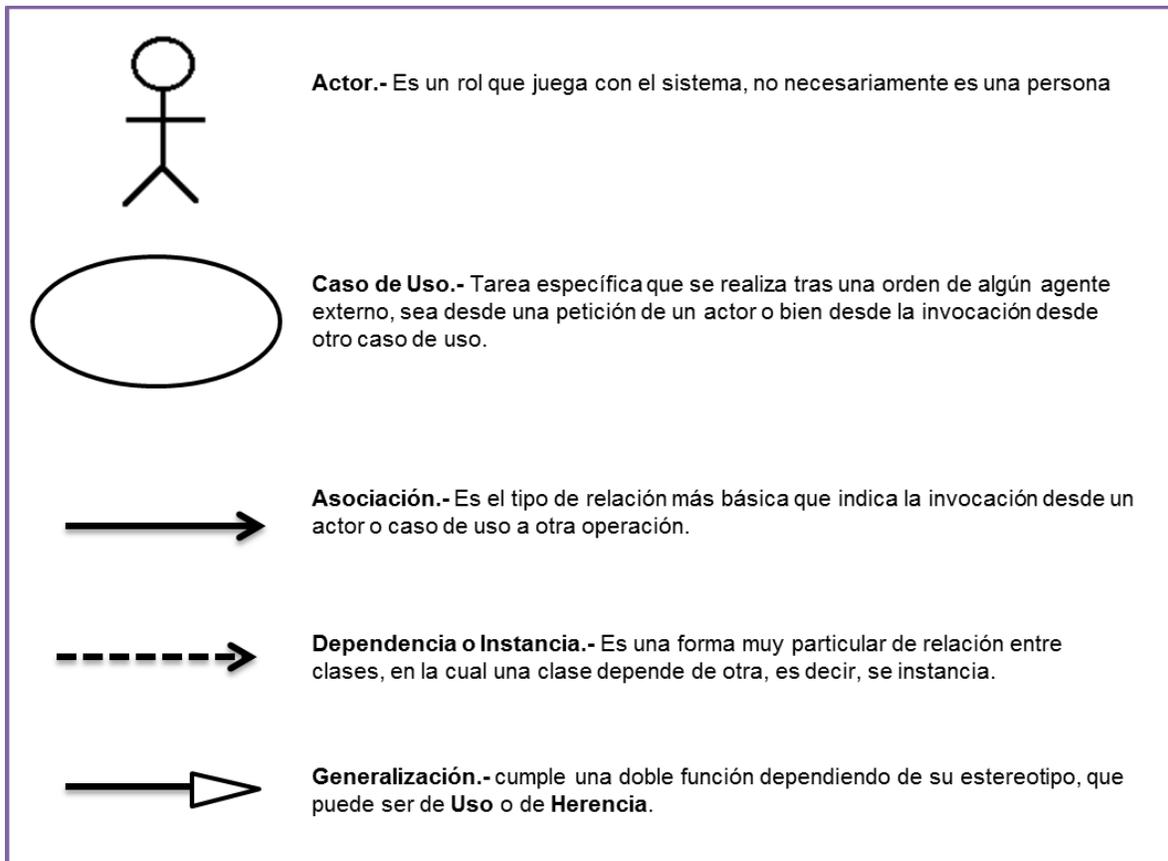


Ilustración 4 Componentes del diagrama casos de uso

UML (Lenguaje de Modelamiento Unificado) utiliza los casos de uso para la obtención de los requerimientos del sistema, permitiendo que el analista se centre en las necesidades del usuario y a su vez describe los casos de uso con mayor importancia para el negocio.

Una limitación de utilizar los casos de uso es que no se establecen por completo los requerimientos funcionales y no funcionales, pero se pueden utilizar conjuntamente con puntos de vista indirectos cuando éstos reciben resultados del sistema (como un informe de gestión).

3.2.3 DICCIONARIO DE DATOS

El diccionario de datos es una lista de nombres ordenados alfabéticamente que se incluye en los modelos del sistema, contiene las definiciones precisas y rigurosas que permiten que el analista y el usuario tengan una misma comprensión de las palabras utilizadas.

Actualmente casi siempre se implementa el diccionario de datos como parte de una herramienta de análisis y diseño estructurado.

Las ventajas de utilizar el diccionario de datos son las siguientes:

1. Comprueba la unicidad de los nombres y alerta a los analistas si hay duplicidad.
2. Sirve como almacén de información de la organización.

Por comodidad, utilidad y mejor organización todos los nombres de entidades, atributos, relaciones o servicios deberían de introducirse en el diccionario.

Existen herramientas para la automatización del diccionario de datos, solo por mencionar algunas, por ejemplo Erwin Data Modeler, Data Dictionary Creator y MySQLWorkbench.

3.3 DISEÑO DE LA ARQUITECTURA DE DESARROLLO

La toma de decisiones de la planeación lógica del software se basa en la esencia del diseño del software, el diseño se empieza a realizar desde que el sistema aún se encuentra en planeación, se puede basar en borradores, bocetos, modelos o métodos estructurados. No existe fórmula para diseñar, ni hay formas buenas o malas, se va aprendiendo a diseñar conforme se observe diseños ya existentes y discutiendo el diseño con otros involucrados en la organización lógica del software, también existen las buenas prácticas para el diseño del sistema que son aquellas técnicas o métodos que permiten incrementar la satisfacción del cliente al incorporar su uso en nuestro proceso.

Una buena práctica para realizar el diseño es que debe de ser de bajo acoplamiento y alta cohesión. Cohesión se refiere a la medida en la que un componente se dedica a realizar solo la tarea para la cual fue creado, delegando las tareas complementarias a otros componentes y el acoplamiento es la medida en que los cambios de un componente tiende a necesitar cambios de otro componente.

Un diseño arquitectónico es un marco para el control y comunicación de los subsistemas que proporcionan algún conjunto de servicio relacionados con el sistema, es decir el sistema se divide en subsistemas.

El diseño arquitectónico es la primera fase del diseño del sistema, está relacionado con el establecimiento de un marco estructural básico que identifica los principales componentes de un sistema y las comunicaciones entre estos componentes.

Las ventajas que tiene el diseño arquitectónico son:

1. Comunicación entre los involucrados usando como punto de discusión la arquitectura del sistema
2. Realizar un análisis del sistema, así se conoce si el sistema puede cumplir con los requerimientos tales como el rendimiento, la fiabilidad y mantenibilidad.
3. Reutilización del software a gran escala, se puede utilizar la misma arquitectura en sistemas similares.

Un diseño de un subsistema es una descomposición abstracta de un sistema en componentes de grano grueso, cada uno de los cuales puede ser un sistema importante por sí mismo.

CAPÍTULO 4

DISEÑO DE LA ARQUITECTURA

SEGUNDA PARTE

4. DISEÑO DE LA ARQUITECTURA

4.1 PARADIGMAS DE PROGRAMACIÓN

Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso del diseño, orientar la forma de pensar y de solucionar los problemas y, por lo tanto, determinan la estructura final de un programa.

La clasificación de los paradigmas de programación parte de los principios fundamentales de cada paradigma en cuanto a las orientaciones sobre la forma para construir las soluciones, ésta clasificación es la siguiente:

- Paradigmas procedimentales. Detallan paso a paso el mecanismo para obtener la solución
- Paradigmas declarativos. Especifican que se desea obtener; pero no requieren indicar el cómo se va a obtener.

Los principales paradigmas son:

Paradigmas procedimentales

- Paradigma orientado a Objetos
- Paradigma Imperativo

Paradigmas declarativos

- Paradigma Funcional
- Paradigma Lógico

Paradigma orientado a Objetos.- Concibe un sistema como un conjunto de entidades (objetos) que representan al mundo real. Este paradigma pretende desarrollar los sistemas con modelos más cercanos a la realidad que a las especificaciones computacionales; a su vez construye componentes del software que sean reutilizables, así mismo diseña una implementación de manera que puedan ser extendidos y modificados con el mínimo impacto en el resto de su estructura. Sus principales características son:

- Definición de clases y herencia.
- Objetos como abstracción de datos y procedimientos.
- Polimorfismo y revisión de tipos en tiempo de ejecución.

Paradigma Imperativo.- Son una serie de sentencias, ejecutadas según un control de flujo explícito, que modifican el estado del programa, es decir, sentencias que indican realizar determinada operación que modifican los datos guardados en memoria. Los conceptos básicos del paradigma son basados en la arquitectura de la máquina de Von Neumann. Las variables son celdas de memoria que contienen datos o referencias, pueden ser modificadas y representan el estado del programa, la sentencia principal es la asignación. Sus principales características son:

- Definición de Procedimientos
- Definición de tipo de datos
- Chequeo de tipos en tiempo de compilación
- Cambio de estado de variables
- Pasos de ejecución de un proceso

Paradigma Funcional.- Basado en el modelo matemático de composición funcional, la operación fundamental es la aplicación de una función a una serie de argumentos. La evaluación se guía por el concepto de sustitución, es decir el resultado de un cálculo es la entrada del siguiente hasta que una composición produce el resultado deseado. Sus características son:

- La computación se realiza mediante la evaluación de expresiones
- Definición de funciones
- Funciones como datos primitivos
- Valores sin efectos laterales, no existe la asignación
- Programación declarativa

Paradigma de Lógico.- Resuelve problemas como las relaciones de un conjunto de datos en las que aplica reglas de deducción, generando conclusiones aceptadas como válidas. Usa la lógica de predicados, o aserciones lógicas que representa el conocimiento sobre un ámbito. Sus principales características son:

- Definición de reglas
- Unificación como elemento de computación

Al finalizar el análisis de los diferentes paradigmas de programación definidos anteriormente, se concluyó que el más óptimo para el desarrollo de la aplicación es el paradigma orientado a objetos por las siguientes razones:

- Permite la reutilización y extensión del código
- La abstracción maximiza la calidad de los diseños

- Se agiliza el desarrollo del software
- Facilita la creación de programas visuales a través de sus plataformas de desarrollo
- Permite generar una interfaz de pantalla atractiva para el usuario.

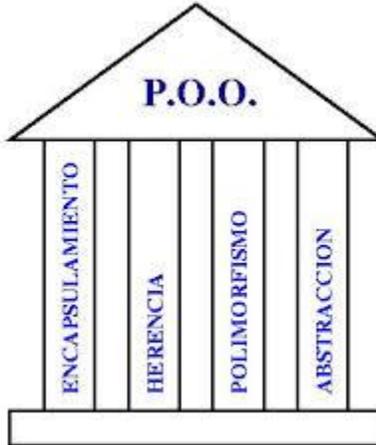


Ilustración 5 Paradigma orientado a objetos

Dentro de este paradigma se eligió el lenguaje de programación Java por las siguientes ventajas:

- Es el lenguaje oficial para el desarrollo de aplicaciones Android, considerado así por Google.
- Contiene las mejores y más completas herramientas como Eclipse ADT y Android Studio.
- Java implementa una API con clases predefinidas que se pueden integrar fácilmente al desarrollo.
- Disponibilidad de miles de librerías como por ejemplo JSON, XML, HTML, conexión a bases de datos, etc.
- Existen gran cantidad de documentación oficial y en blogs soportada por desarrolladores expertos.



Ilustración 6 Logotipo de Java

4.2 PLATAFORMAS DE DESARROLLO MÓVIL

Un IDE (Entorno de Desarrollo Integrado) es un entorno de programación empaquetado, éste es conformado por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDE´s pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. En el mercado existen una gran variedad de herramientas y emuladores gratuitos que pueden configurarse como IDE´s esto permite que los programadores de software cuenten con diversas opciones para desarrollar sus aplicaciones.

Los IDE´s más utilizados para programar en Android son:

- Eclipse
- NetBeans
- IntelliJIdea

El IDE que se utiliza para desarrollar en IOS es el

- XCODE

Para desarrollar aplicaciones en Windows Phone existe el IDE

- Microsoft Visual Studio 2010 Express for Windows Phone

En el desarrollo de aplicaciones en Symbian existen los IDE´s

- Code Warrior
- VisualStudio.NET
- Carbide C++

Y finalmente para las aplicaciones de BlackBerry los IDE´s más utilizados son:

- BlackBerry JDE
- Eclipse con BlackBerry JDE Plug-in
- NetBeans

Se utilizó el IDE Eclipse para el desarrollo de la app este fue elegido basándose en la siguiente tabla comparativa.

	Eclipse	IntelliJ IDEA	Netbeans
Licencia	Gratuita	De prueba	Gratuita
Idioma	Ingles	Ingles	Español
Compatible con el sistema operativo Windows 7	Sí	No	Sí
Soporta Java	Sí	Sí	Sí
Comentarios Generales	Más rápido Más flexible Más plugins Mejor soporte para desarrollo Android	En su entorno de pago es considerado el mejor IDE de Java	Más pesado sobre todo si se tiene varios proyectos. Herramientas Swing estándar. GUI más intuitiva. Mejor soporte para PHP Permite importar proyectos de Eclipse y otros IDE's
Logo			

4.3 MATRIZ DE PRUEBA Y TESTING

La Matriz de prueba también conocida como matriz de trazabilidad de requerimientos es un documento, usualmente en forma de tabla que relaciona los requerimientos contra casos de prueba o escenario, su principal función de la matriz de pruebas es verificar que todos y cada uno de los requerimientos sea cubierto por caso de prueba.

No existe un formato único de Matriz de Trazabilidad de Requerimientos y varían en complejidad. Pero deben tener como mínimo 2 elementos: lista completa de requerimientos y uno o más Casos de Prueba que cubran dichos requerimientos.

La tabla 9 muestra la matriz de prueba que se realizó a la aplicación para probar la autenticación de los usuarios.

Tabla 9 Matriz de prueba

Área	Función/ Tarea	Prueba	Resultados Esperados	Resultados de la Prueba	Comentarios
Seguridad	Autenticación	Usuario no existe	Nombre de usuario o contraseña incorrectos	Nombre de usuario o contraseña incorrectos	OK
		Contraseña incorrecta	Nombre de usuario o contraseña incorrectos	Nombre de usuario o contraseña incorrectos	OK
		Usuario y contraseña validos	Conectarse a la aplicación, mostrar el mapa	Se conectó a la aplicación, apareció el mapa	OK
		Presionar botón regresar	Salir de la aplicación	Termino del programa	OK
	Nuevo usuario	Campos incompletos	Denegar el alta del usuario	Se denegó el alta del usuario	OK
		Diferentes caracteres a los numéricos en el campo del nombre de usuario	Pedir solo caracteres numéricos	Solicita ingresar solo caracteres numéricos	OK
		Longitud de caracteres en el nombre de usuario igual a diez dígitos	Indicar el número exacto de caracteres permitidos	Solicita ingresar solo diez caracteres numéricos	OK

Área	Función/ Tarea	Prueba	Resultados Esperados	Resultados de la Prueba	Comentarios
	Nuevo usuario	Longitud de caracteres en el campo de la contraseña de cinco a diez dígitos	Indicar el número exacto de caracteres permitidos	Solicita ingresar de cinco a diez caracteres	OK
	Recuperar contraseña	Verificar que el usuario existe	Mostrar la pregunta secreta del usuario	Mostro la pregunta secreta del usuario	OK
		Verificar que la respuesta de la pregunta secreta sea correcta	Permitir el ingreso de una nueva contraseña	Permite el ingreso de una nueva contraseña	OK
Ejecución	Acceso a la aplicación	Si el dispositivo no tiene conexión a internet	Denegar el acceso	Indica que el usuario o contraseña son incorrectos	OK
		Si el GPS del dispositivo se encuentra apagado	Mostrar el mapa con un aviso de que el GPS se encuentra inactivo	Muestra el mapa junto con un mensaje de "GPS desactivado"	OK
		Si no se tienen contactos que utilicen la aplicación	Mostrar el mapa solo con la ubicación del dispositivo	Muestra la ubicación del dispositivo en uso	OK

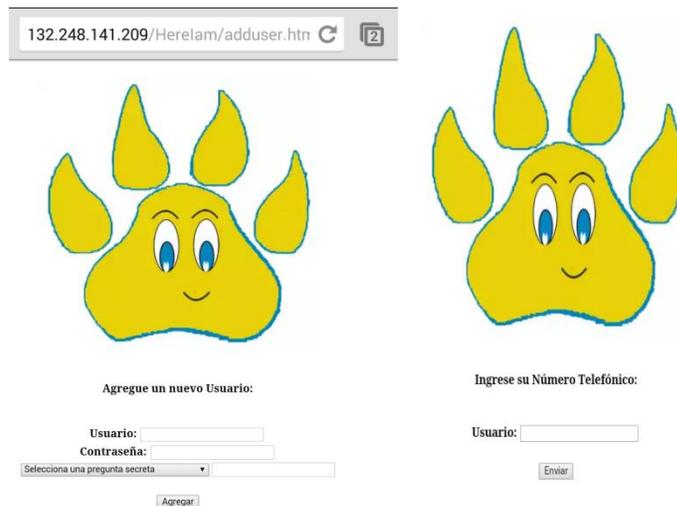


Ilustración 7 Recuperación de contraseña



Ilustración 9 Autenticación de usuario



Ilustración 10 Verificando datos de usuario

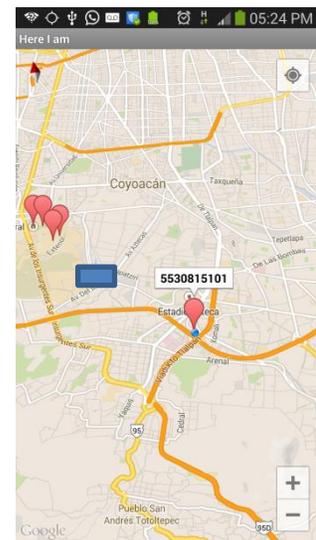


Ilustración 11 Mapa con ubicaciones

El testing consiste en realizar una verificación dinámica del comportamiento de un software usando un conjunto finito de casos de prueba, un testing funcional valida si el comportamiento observado del software probado cumple o no con sus especificaciones.

El testing nos es útil para evaluar la calidad del producto, y para mejorarlo al identificar los defectos y problemas.

Si el software es muy pequeño si se podrá testear como una sola unidad, de lo contrario es casi imposible, en estos casos de que el software es complejo se sugiere realizar el testing guiado por la siguiente estructura.

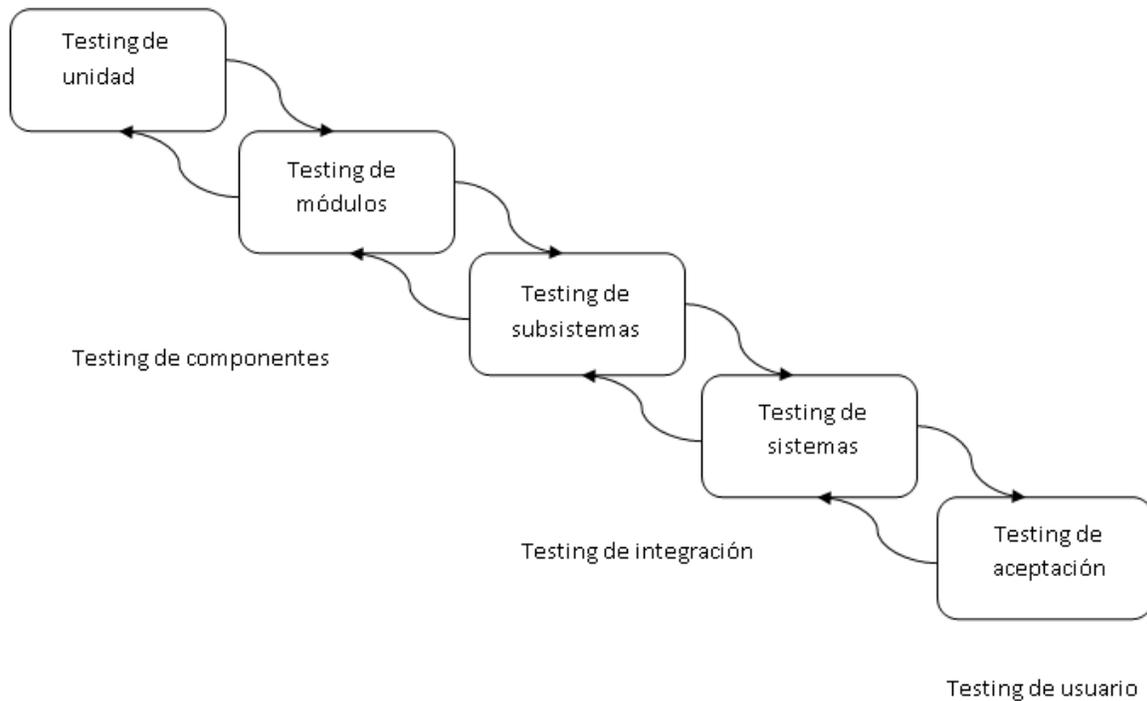


Ilustración 8 Modulo de Testing

Cuando se realiza la prueba y en un módulo se encuentra un error, después de corregir el error se deberá empezar el testeo en un módulo anterior al que se encontró el error, es decir se sugiere seguir el siguiente procedimiento al encontrar un error en el sistema.

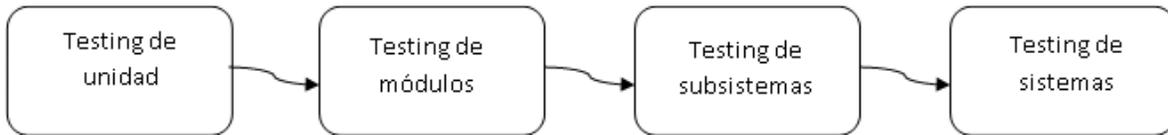


Ilustración 9 Modulos de Pruebas

Existen dos metodologías clásicas

- **Testing estructural o de caja blanca.-** Son pruebas que utilizan las estructuras de control para derivar los casos de prueba, son pruebas a nivel componente, por ejemplo: Probar los caminos posibles en estructuras de bifurcación, tanto verdaderas como falsas, que garanticen por lo menos una vez las rutas que se pueden seguir en una estructura, que se ejecuten todos los bucles en sus límites y dentro de sus límites operacionales, que ejerciten estructuras de datos internos para asegurar su validez, etc.
- **Testing basado en modelos o de caja negra.-** Estas pruebas son a nivel de entorno, se presupone que los módulos ya están bien probados y se sugieren pruebas como: Verificar el correcto funcionamiento de acuerdo a ciertas clases de datos o valores de entrada, cuales tasas de datos y cual volumen tolera el sistema, pruebas de comportamiento y el desempeño del sistema, etc.

Las pruebas de la aplicación fueron basadas en el testing de caja negra para verificar que el ítem que se está probando, cuando se dan las entradas apropiadas produce los resultados esperados. Los datos de prueba se eligieron atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa funcione bien.

CAPÍTULO 5

IMPLEMENTACIÓN

5. IMPLEMENTACIÓN

5.1 PLAN DE IMPLEMENTACIÓN

Las primeras pruebas se realizaron con las siguientes versiones del sistema operativo Android, estas versiones se eligieron por la demanda de dispositivos que las contienen, así como para tener más variedad de versiones.

- Android 4.4.2 (kitKat)
- Android 4.2.2 (JellyBean)
- Android 4.1.2 8 (JellyBean)
- Android 4.0.4(Ice CreamSandwich)
- Android 2.3.4 (Gingerbread)
- Android 2.2 (Froyo)

Los modelos de los dispositivos móviles en los cuales se instalaron la aplicación son los siguientes, estos por la diferencia del tamaño de la pantalla.

1. CyberPad 754TPC
2. Inco Storm
3. LG L9
4. Motorola Moto G
5. Motorola XT316
6. Samsung Galaxy Ace
7. Samsung Galaxy S3 mini
8. Samsung Galaxy S4
9. Sony Xperia
10. Sony XperiaZL

Se realizarán las siguientes pruebas

- ✓ Vista en los diferentes tipos de display
- ✓ Tiempo de inicio de la aplicación
- ✓ Validación de usuarios
- ✓ Tiempo de respuesta del servidor
- ✓ Porcentaje de error de ubicación
- ✓ Vista de ubicación de los contactos

Posteriormente de obtener pruebas exitosas se dará de alta una cuenta en Google Play para la descarga gratuita de esta aplicación, especificando que fue creada para la comunidad universitaria.

En el capítulo 4.3 se encuentra la matriz de pruebas que se realizó a la aplicación y en el capítulo 6 se muestran los resultados de las pruebas que se le realizaron a los diferentes dispositivos.

5.1.1 MANUAL TÉCNICO

Esta aplicación se realizó con la finalidad de poder localizar a compañeros y profesores dentro de ciudad universitaria sin realizar una llamada telefónica.

La base de datos en donde se alojan los usuarios se llama “hereiam” contiene una tabla llamada “usuario” y los campos que conforman esta tabla son los siguientes:

- Username.- Es de tipo double, ya que solo acepta como usuario números telefónicos de diez dígitos. Este campo es la llave primaria de la base de datos.
- Passw.- De tipo varchar con una longitud de diez caracteres.
- Pregunta.- Tipo varchar de longitud cuarenta y cinco, se tomó esta longitud ya que las preguntas ya se encuentran predeterminadas.
- Respuesta.- Tipo varchar con longitud de cincuenta caracteres.
- Latitud.- Tipo double, por el tipo de valor que nos da el sistema satelital de posicionamiento.
- Longitud.- De igual manera que el campo de latitud es de tipo double.

	Campo	Tipo
<input type="checkbox"/>	username	double
<input type="checkbox"/>	passw	varchar(10)
<input type="checkbox"/>	pregunta	varchar(45)
<input type="checkbox"/>	respuesta	varchar(50)
<input type="checkbox"/>	latitud	double
<input type="checkbox"/>	longitud	double

Ilustración 10 Tabla de la base de datos del sistema

Se utilizó el manejo de bases de datos mysql por las siguientes razones:

- Es un código abierto
- Facilidad de configuración e instalación.
- Seguridad en cuanto a los permisos y privilegios
- Puede ser ejecutado en una máquina con bajos recursos

El sistema se encuentra en la siguiente dirección <http://132.248.141.209/HereIam/> es la página principal, solamente muestra el logo del sistema.

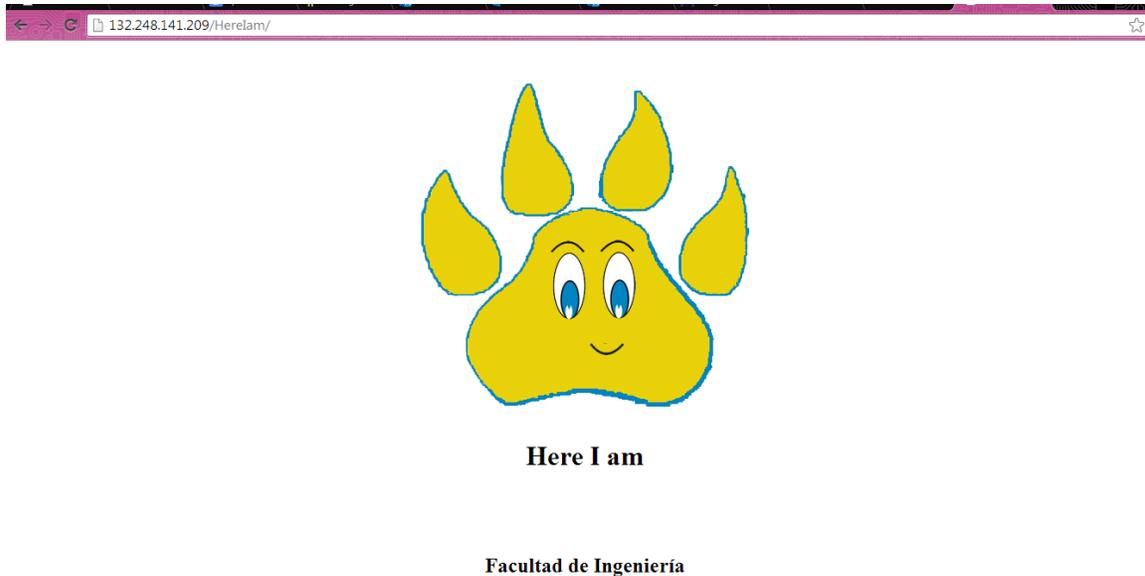


Ilustración 11 Página principal del sistema

Los siguientes nombres de archivos que se describen conforman la parte web del sistema.

access.php

Recibe los campos usuario y contraseña e invoca al archivo funciones_bd.php para realizar la consulta en la base de datos. Devuelve un uno si los datos son correctos y devuelve un cero si son incorrectos. Este archivo fue creado para que desde java haga la petición al php utilizando el objeto json.

funciones_bd.php

Requiere el archivo conconnectbd.php para realizar la conexión a la base de datos. Este archivo realiza las consultas, inserciones y update a la base de datos.

conex.php

Realiza la conexión a la base de datos, este archivo contiene el usuario y contraseña de la base.

adduser.html

Pide los datos de un nuevo usuario. Valida en las cajas de texto que el usuario sea un número de diez dígitos y la contraseña sea de un mínimo de cinco caracteres y un máximo de diez caracteres.

adduser.php

Solicita el archivo funciones_bd.php para realizar una inserción a la base de datos, como el usuario es la llave primaria no permite agregar un usuario ya existente y manda un mensaje para que utilice otro usuario. Si se insertaron los datos correctamente, manda un mensaje que fue satisfactorio los datos que inserto.

pass.php

Para que el usuario pueda recuperar la contraseña y pueda acceder nuevamente a la aplicación primero debe proporcionar su usuario, este archivo realiza una consulta a la base de datos para verificar si el usuario existe, si esto es correcto nos manda a la siguiente página que es el archivo obt_pass.php pero si no existe el usuario en la base de datos envía un mensaje para que verifique su usuario.

obt_pass.php

Al realizar la consulta se obtiene la pregunta secreta que selecciona cuando el usuario se registró. Este archivo muestra al usuario su pregunta secreta y deberá responderla correctamente, realiza la consulta para verificar la respuesta si es incorrecta envía un mensaje de datos equivocados, pero si es correcta pasa al siguiente archivo rest_pass.php.

local.php

Solicita el archivo conex.php para realiza un update en la base de datos modificando los campos latitud y longitud.

consulta.php

Realiza consultas para obtener las coordenadas de los usuarios, este archivo devuelve los campos usuario, latitud y longitud por medio del objeto json.

En la parte anterior fue explicada la base de datos y la parte web del sistema, a continuación se explicará brevemente los archivos que contiene la aplicación. La siguiente imagen muestra el diseño del diagrama de flujo utilizado para el desarrollo de la aplicación.

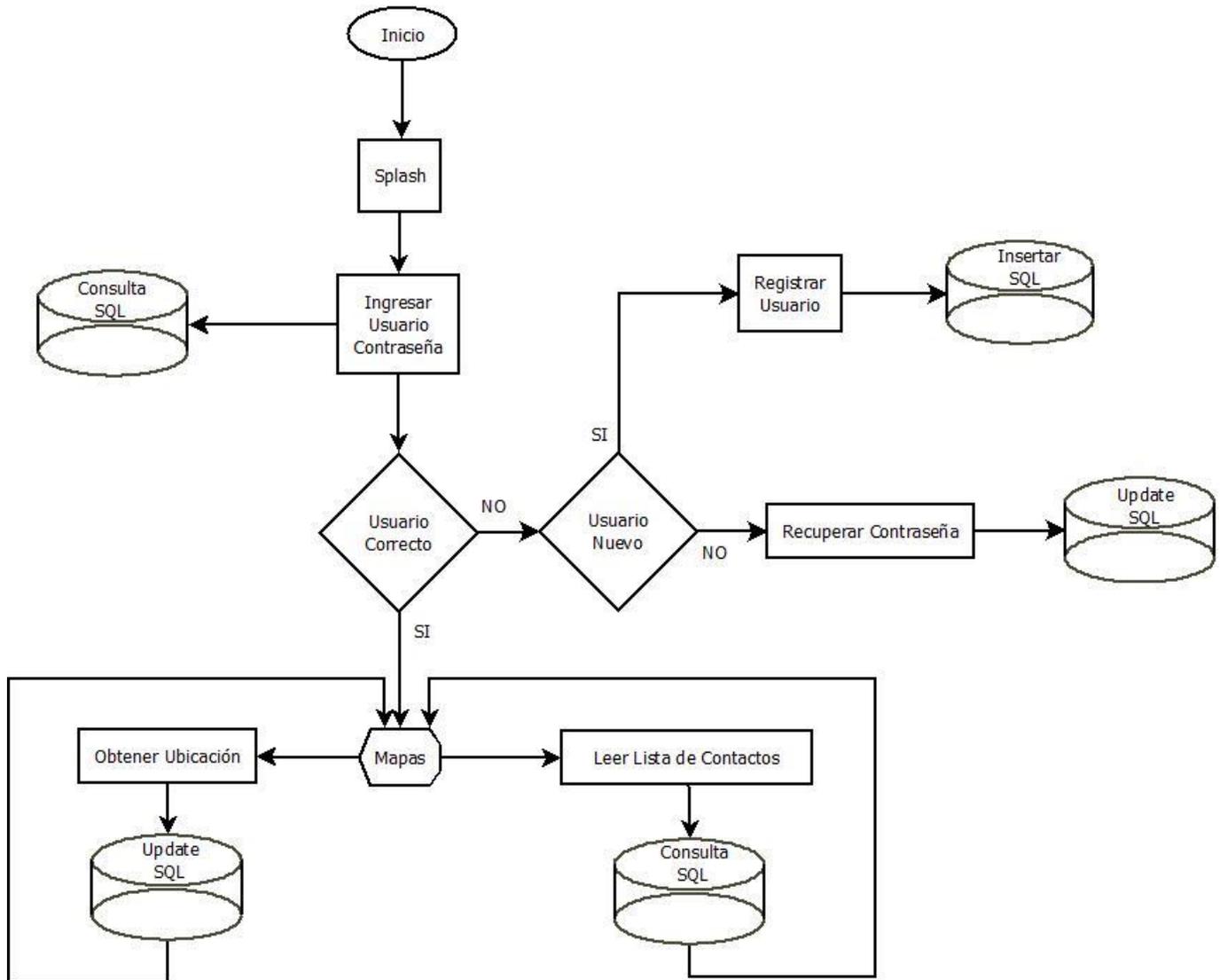


Ilustración 12 Diagrama de flujo de la aplicación

La clase *QuizSplashActivity* contiene el splash screen de la aplicación, ésta es una pantalla de presentación que aparece al inicio de la aplicación solo por unos segundos. Dentro de esta clase se encuentra el método *startAnimating* que manda a llamar los recursos de animación que se encuentran en la carpeta de /res. Después de que se terminan las animaciones se realiza una pausa de dos segundos con el método *onPause*, al término de este tiempo se pasa a la siguiente actividad que es la clase *Login*.

La siguiente actividad es la clase *Login*, esta contiene los métodos ingresar, recuperar contraseña y registrar usuario. Los tres métodos envían peticiones al servidor web y éste a su vez a la base de datos para realizar las consultas o los update en la base. Los métodos *olvpasword* y registrar son ligas que re direccionan al servidor web, por ello el usuario realiza estos pasos en un navegador de internet. El método ingresar recibe el login y password ingresado por el usuario, realiza una petición al servidor web y base de datos sin tener que salir de la aplicación, si los datos son incorrectos envía un mensaje en pantalla para que el usuario ingrese nuevamente sus datos pero si los datos son correctos se manda a la siguiente actividad que es la clase mapa.

La clase Mapa tiene el método *onCreate* este método obtiene los mapas de google y los carga en el dispositivo, a su vez también carga el botón de localización. El método *MiLocationListener* obtiene las coordenadas de nuestro dispositivo móvil, guarda estas variables y las envía al método *asyncupd* también verifica que el GPS este habilitado en caso contrario manda un mensaje en pantalla al usuario. *ReadContacts* realiza la lectura de los contactos almacenados en el dispositivo móvil y los almacena en una variable para enviarlos al siguiente método que es *asynconsul* este método realiza la petición al servidor web para realizar una consulta de las ubicaciones de los contactos registrados en la base de datos. Al igual que el método *asynconsul* el método *asyncupd* realiza una petición al servidor web para modificar las coordenadas del dispositivo móvil cada diez segundos.

En la siguiente ilustración se muestra el diagrama de casos de uso, dentro de este diagrama se dan a conocer las principales clases de la aplicación así como la función de cada una de ellas.

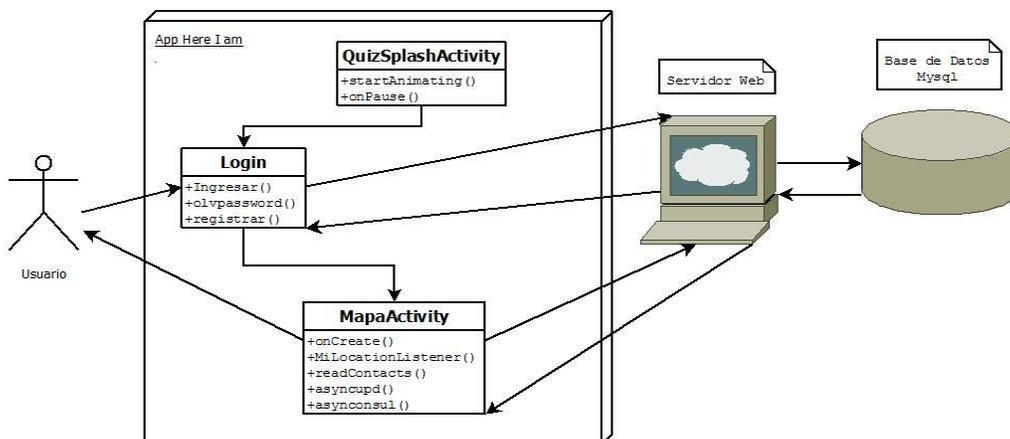


Ilustración 13 Casos de uso del sistema

5.2 PAQUETE DE DISTRIBUCIÓN

La distribución de la aplicación sólo admite el tipo de paquete apk¹⁵, éste se usa para distribuir e instalar componentes empaquetados para la plataforma Android.

Este tipo de paquete contiene lo siguiente

- **AndroidManifest.xml.-** Dentro de este archivo se declara todo lo que se encuentre dentro de la aplicación las actividades, los servicios, etc. y como todas estas piezas interactúan entre sí y con el sistema.
- **Classes.dex.-** En este archivo se encuentra el programa (conjunto de instrucciones) que ejecuta la aplicación para desarrollar su función.
- **Resources.arsc.-** Este archivo contiene los recursos previamente compilados, por ejemplo XML binario.
- **Res (carpeta).-** Contiene los recursos no compiladas en resources.arsc.
- **Lib (carpeta).-** Contiene el código compilado que es específico de una capa de software de un procesador, el directorio se divide en más directorios dentro de ella:
 - armeabi
 - armeabi-v7a
 - x86
 - mips

Para instalar un archivo .apk es necesario abrir el archivo dentro del equipo móvil que tenga sistema operativo Android y el mismo sistema operativo reconoce el archivo como un paquete para instalar alguna aplicación

¹⁵ .apk es un paquete para el sistema operativo Android

5.3 REQUERIMIENTOS DE INSTALACIÓN

Para la instalación de la aplicación se requiere lo siguiente:

- ✓ Sistema Operativo Android 4.x.- Se pide este mínimo de versión ya que es el que soporta el nuevo api versión 2 de google maps.
- ✓ Espacio libre mínimo de 6 MB.- Este es el tamaño de la aplicación para poderla instalar y almacenar en el dispositivo móvil.
- ✓ Conexión inalámbrica Wireless Fidelity o Datos Móviles. Se requiere acceso a internet ya que se manda peticiones a un servidor en donde se encuentra almacenada la base de datos y el sistema para el registro de los nuevos usuarios y para la recuperación de la contraseña. También así para tener acceso a los mapas de google maps.
- ✓ Conexión al sistema satelital de posicionamiento (GPS). Con el GPS obtendremos las coordenadas de nuestra ubicación, así podemos interpretarlo en el mapa y dejar que nuestros contactos que tengan instalada la aplicación observen nuestra ubicación.
- ✓ Dispositivo móvil con display touch.- Para mejor interacción con los mapas se pide una pantalla touch, ya que es más flexible para el usuario el manejo de los botones del zoom y el botón de nuestra ubicación.

CAPÍTULO 6

RESULTADOS

6. RESULTADOS

Se realizaron pruebas en diez dispositivos móviles los cuales fueron elegidos por las características diferentes que cada uno contenía, sin perder de vista los requerimientos que la aplicación solicita para su funcionamiento. Las pruebas fueron realizadas en Ciudad Universitaria en diferentes campus, los resultados de las pruebas se muestran en la siguiente tabla.

Tabla 10 Resultados de la pruebas

Dispositivo	Características	Observaciones	¿Su funcionamiento fue satisfactorio?
CyberPad 754TPC	<p>Sistema Operativo: 4.0.4 Ice CreamSandwich</p> <p>Pantalla: Touch</p> <p>Tamaño: 7 pulgadas</p> <p>Resolución: 800 x 480</p> <p>Conectividad: WiFi 802.11</p>	<p>La aplicación fue instalada satisfactoriamente, su funcionalidad fue buena. El inconveniente que tuvo fue que como es una tablet no guarda contactos por lo tanto sólo te muestra el mapa con la ubicación del dispositivo pero no muestra la ubicación de otros contactos ya que no se tienen guardados en la tablet.</p>	NO
Inco Storm	<p>Sistema Operativo: 4.2 Jelly</p> <p>Pantalla: Touch</p> <p>Tamaño: 5 pulgadas</p> <p>Resolución: 480 x 854</p> <p>Conectividad: WiFi802.11, 3G</p>	<p>En este dispositivo el funcionamiento de la aplicación fue satisfactorio, se utilizó conexión a WiFi y la velocidad en la cual se mostraron los contactos fue aceptable.</p>	SI

Dispositivo	Características	Observaciones	¿Su funcionamiento fue satisfactorio?
LG L9	<p>Sistema Operativo: 4.0.4 Ice CreamSandwich</p> <p>Pantalla: Touch</p> <p>Tamaño: 4.7 pulgadas</p> <p>Conectividad: Wi-Fi 802.11 b/g/n; DLNA; Wi-Fi Direct</p>	<p>Con este dispositivo se realizaron las pruebas con conexión WiFi y los resultados fueron aprobatorios. Por el tamaño de la pantalla las imágenes se observan perfectamente.</p>	SI
Motorola Moto G	<p>Sistema Operativo: 4.4 KitKat</p> <p>Pantalla: Touch</p> <p>Tamaño: 4.5 pulgadas</p> <p>Resolución: 720 x 1280</p> <p>Conectividad: Wi-Fi 802.11 b/g/n; banda dual, DLNA</p>	<p>En este dispositivo se utilizaron los datos móviles para la conexión a internet, al utilizar este tipo de conexión se vio un poco lento en mostrar las ubicaciones de los contactos almacenados en el dispositivo. Las imágenes y el diseño de la aplicación se observaron perfectamente.</p>	SI

Dispositivo	Características	Observaciones	¿Su funcionamiento fue satisfactorio?
Motorola XT316	<p>Sistema Operativo: 2.3 Gingerbread</p> <p>Pantalla: Touch</p> <p>Tamaño: 2.8 pulgadas</p> <p>Resolución: 240 x 320</p> <p>Conectividad: Wi-Fi 802.11 b/g/n</p>	<p>Se instaló la aplicación perfectamente, pero al querer ingresar con el usuario y contraseña para que cargaran los mapas, se producía un error y la aplicación se cerraba forzosamente, este error se provocó ya que la versión del sistema operativo no es la adecuada.</p>	NO
Samsung Galaxy Ace	<p>Sistema Operativo: 2.2 Froyo</p> <p>Pantalla: Touch</p> <p>Tamaño: 3.5 pulgadas</p> <p>Resolución: 320 x 480</p> <p>Conectividad: Wi-Fi 802.11 b/g/n; DLNA</p>	<p>Al igual que el dispositivo Motorola XT316 la instalación se realizó satisfactoriamente pero al cargar los mapas la aplicación se cierra, por ello no se puede utilizar la aplicación con este dispositivo. Las imágenes y el diseño de la app se observan perfectamente.</p>	NO

Dispositivo	Características	Observaciones	¿Su funcionamiento fue satisfactorio?
Samsung Galaxy S3 mini	<p>Sistema Operativo: 4.1 JellyBean</p> <p>Pantalla: Touch</p> <p>Tamaño: 4 pulgadas</p> <p>Resolución: 480 x 800</p> <p>Conectividad: Wi-Fi 802.11 a/b/g/n; DLNA; Wi-Fi Direct</p>	<p>Aunque la pantalla de este dispositivo es un poco pequeña en comparación de los otros dispositivos, la aplicación se observa perfectamente. Con respecto a visualizar la ubicación de los contactos fue algo tardado ya que se utilizaron datos móviles y no se tenía una buena señal.</p>	SI
Samsung Galaxy S4	<p>Sistema Operativo: 4.2.2 JellyBean</p> <p>Pantalla: Touch</p> <p>Tamaño: 4.99 pulgadas</p> <p>Resolución: 1080 x 1920</p> <p>Conectividad: Wi-Fi 802.11 a/b/g/n/ac; DLNA; Wi-Fi Direct; banda dual</p>	<p>Este dispositivo fue el que se utilizó para elaborar la aplicación desde sus inicios, en este dispositivo se basaron el tamaño de las imágenes y la versión del sistema operativo a utilizar.</p>	SI
Sony Xperia	<p>Sistema Operativo: 2.3 Gingerbread</p> <p>Pantalla: Touch</p> <p>Tamaño: 4.3 pulgadas</p> <p>Resolución: 720 x 1280</p> <p>Conectividad: Wi-Fi 802.11 b/g/n; DNLA</p>	<p>Por la versión del sistema operativo en este dispositivo ya no se instaló la aplicación ya que por pruebas anteriores sabemos que la app no iba a funcionar correctamente.</p>	NO

Dispositivo	Características	Observaciones	¿Su funcionamiento fue satisfactorio?
XperiaZL	<p>Sistema Operativo: 4.1.2 JellyBean</p> <p>Pantalla: Touch</p> <p>Tamaño: 5 pulgadas</p> <p>Resolución: 1080 x 1920</p> <p>Conectividad: Wi-Fi 802.11 a/b/g/n; DNLA; Wi-Fi Direct; banda dual</p>	<p>En este dispositivo se utilizó la conexión WiFi pero el modem al que se realizó la conexión a internet es privada por lo tanto se necesita contraseña para acceder al modem. La aplicación funciono satisfactoriamente, al mostrar el mapa y los contactos ubicados la visualización fueron casi inmediatos por el tipo de velocidad de la conexión a internet.</p>	SI

Las siguientes imágenes muestran las pruebas que se hicieron con algunos dispositivos.

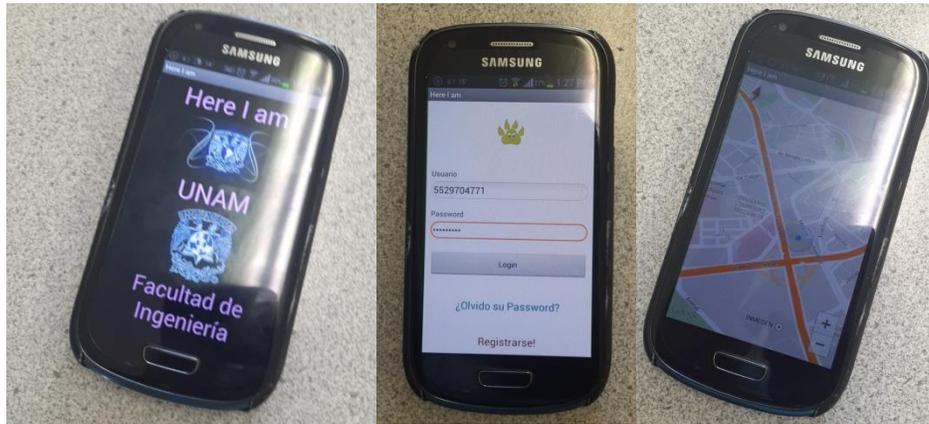


Ilustración 14 Pruebas realizadas en el dispositivo Samsung S3 mini



Ilustración 15 Pruebas realizadas en el dispositivo Inco Storm

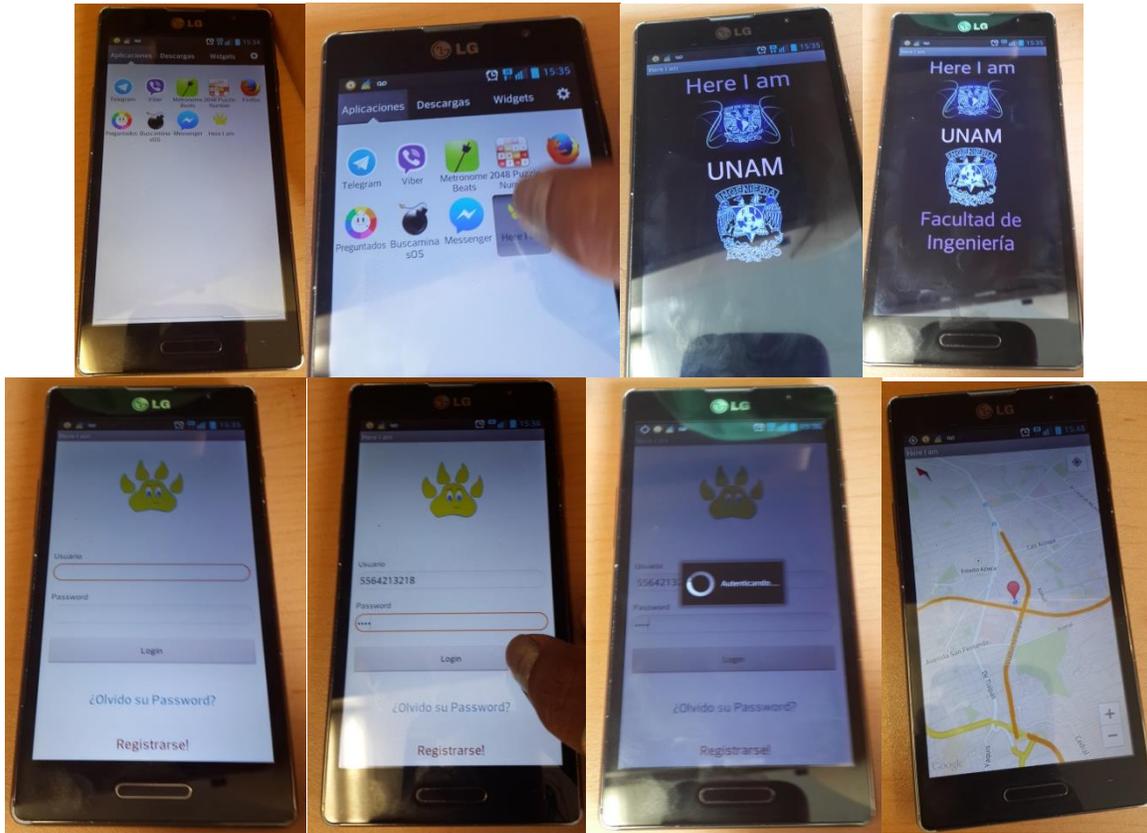


Ilustración 16 Pruebas realizadas en el dispositivo LG L9

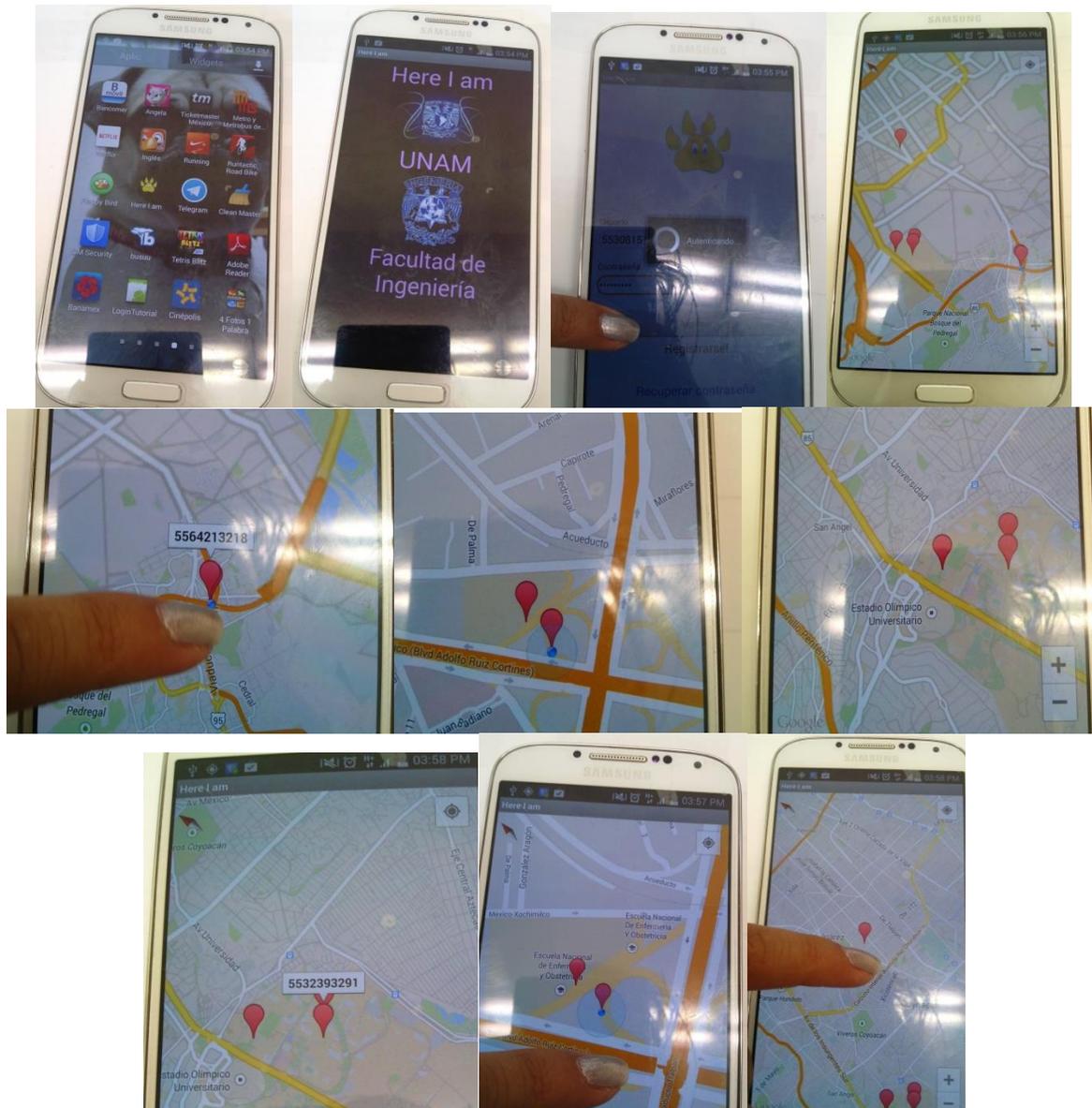


Ilustración 17 Pruebas realizadas en el dispositivo Samsung S4

CAPÍTULO 7

CONCLUSIONES

7. CONCLUSIONES

De acuerdo a los objetivos particulares y presentados en un inicio:

1. Mapeo cartográfico de Ciudad Universitaria.
2. Instalación y configuración de un servidor de base de datos para alojamiento de mapas cartográficos.
3. Implementación de aplicaciones en dispositivos móviles.
4. Difusión de la aplicación para lograr el mayor número de usuarios.

Se puede concluir, de acuerdo a cada uno respectivamente, lo siguiente:

Objetivo 1.- Se utilizaron los mapas cartográficos de Google ya que es un servicio gratuito a menos que la aplicación sobrepase veinte cinco mil solicitudes al día, la matrícula escolar de la Facultad de Ingeniería es de 11746 en el periodo semestral 2014-2, considerado que el sesenta por ciento de los alumnos poseen un dispositivo móvil con las características para el buen funcionamiento de la aplicación se toman en cuenta siete mil alumnos de la facultad de ingeniería que podría utilizar la aplicación, así que mientras no se sobrepasen las solicitudes puestas por Google la aplicación será gratuita.

Objetivo 2.- Ya que se utilizaron los mapas de Google, no se almacenaron mapas en el servidor si no que se realiza una petición por medio del api versión dos de Google. En el servidor se encuentra la base de datos y el sistema web para la autenticación de los usuarios. No es necesario almacenar en la base de datos los contactos que se encuentran en cada dispositivo móvil ya que solo se realizan consultas en la base del dispositivo móvil y en la base de la aplicación.

Objetivo 3.- Al comienzo del desarrollo de aplicación se tenía en cuenta que se pudiera implementar la app en dispositivos móviles que tuvieran sistema operativo Android, pero conforme se fue avanzando en el desarrollo, se tomaron en cuenta otros aspectos que no coincidían con la idea principal de la aplicación. En primera instancia el api de Google maps cambio a su versión dos y está en el desarrollo ya no es compatible con versiones anteriores a 4.0 de Android. Y en segunda instancia la aplicación solo se puede utilizar en Smartphones ya que la app solicita ingresar a la base de datos del celular para poder realizar una consulta de los contactos.

Objetivo 4.- Se empieza a realizar la difusión de la aplicación, por el momento solo se ha difundido por medio de compañeros de la Facultad de Ingeniería, así que se cuenta con veinte usuarios utilizando la aplicación en su Smartphone. Hasta el momento los usuarios consideran que es una buena aplicación, aunque hay que realizar una mejora como por ejemplo hacer que la visualización de la ubicación de los contactos sea más rápida cuando el celular esté conectado por medio de datos móviles y no por WiFi.

Al desarrollar esta aplicación se consolidaron los conceptos de especificación de requerimientos, diseño y desarrollo, verificación y validación e integración de productos en ambiente productivo; todo lo anterior definido dentro del ciclo de vida del software.

Se amplió la base de conocimiento del lenguaje de programación orientado a objetos Java, los lenguajes para el desarrollo web como PHP y HTML así como la administración de bases de datos.

Se logró una mejor definición de los requerimientos para el desarrollo de sistemas dando a conocer al usuario el producto final.

Se amplió un control del proyecto de desarrollo mediante documentación que se genera en cada uno de los procesos del ciclo de vida; capacitar al usuario para el uso de la aplicación y mantener un soporte de segundo nivel.

Adicional al desarrollo de soluciones, se procuró que la aplicación cumpliera con los objetivos establecidos, dado a circunstancias de tiempo y herramientas al alcance para el término del desarrollo de la aplicación se tuvieron que cambiar algunos objetivos.



REFERENCIAS

8. REFERENCIAS

- Roger S. Pressman. (Marzo 4, 2010). Ingeniería del Software. McGraw-Hill
- Dante Cantone. (Septiembre 30, 2006). Implementación y Debugging. Creative Andina Corp.
- Ian Sommerville. (Madrid 2005). Ingeniería del Software. Pearson.
- Gustavo Noguera. (Argentina, 2005). GPS Posicionamiento Satelital. Universidad Nacional de Rosario.
- Laboratorio Nacional de Calidad del Software. (Marzo 2009) Ingeniería del Software: Metodologías y ciclos de vida. Inteco
- Jesús M. Salinas. Noviembre 1997. EDUTEC Revista Electrónica de Tecnología Educativa. Grupo de Tecnología Educativa
- Computer Science Curricula 2013. Strawman Draft (February 2012)
- Wieggers, Karl. Software Requirements. Segunda edición. Microsoft Press. 2003.
- Aurum, Aybüke; Wohlin, Claes. Engineering and Managing Software Requirements. Springer. 2005.
- Smartphone statistics and market share". Email Marketing Reports. www.emailmarketing-reports.com/wirelessmobile/cmartphone-statistics.htm. Feb 2012.
- http://profesores.fi-b.unam.mx/carlos/aydoo/conceptos_oo.html, Información general sobre Metodología orientada a objetos, Consultada: Enero 2014.
- <http://developer.android.com/sdk/>, Información sobre instalación del SDK Android, Consultada: Noviembre 2013.
- <https://developers.google.com/maps/documentation/javascript/v2/reference?csw=1>, Documentación del API V2 maps Google, Consultada: Diciembre 2013



ANEXOS

ANEXOS

I. MANUAL DE USUARIO

Con esta aplicación se podrán ubicar a los contactos almacenados en el dispositivo móvil, sólo debe de registrar su número otorgando una contraseña y una pregunta de seguridad para recuperar esta última, está app solo puede ser utilizada por usuarios que estén registrados y que tengan un dispositivo móvil con el sistema operativo Android.

Los objetivos del sistema son los siguientes:

- ✓ Ubicar contactos que se encuentren registrados en la base de datos, sin tener la necesidad de realizar una llamada o enviar un mensaje para preguntar en donde se encuentran.
- ✓ Poder trasladarnos a la ubicación de nuestro contacto en menor tiempo, ya que al realizar la llamada deberás de esperar que tu contacto responda sobre su ubicación.
- ✓ Que el usuario pueda ahorrar con los servicios de pago del dispositivo móvil ya que en gran parte de ciudad universitaria se cuenta con conexión wifi.

Este es el icono de la aplicación nombrada “Here I am”.

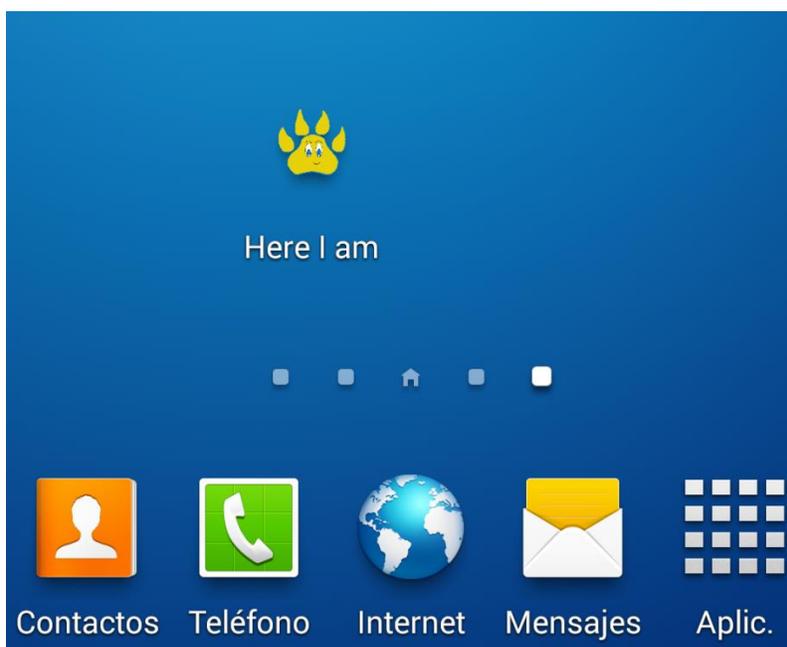


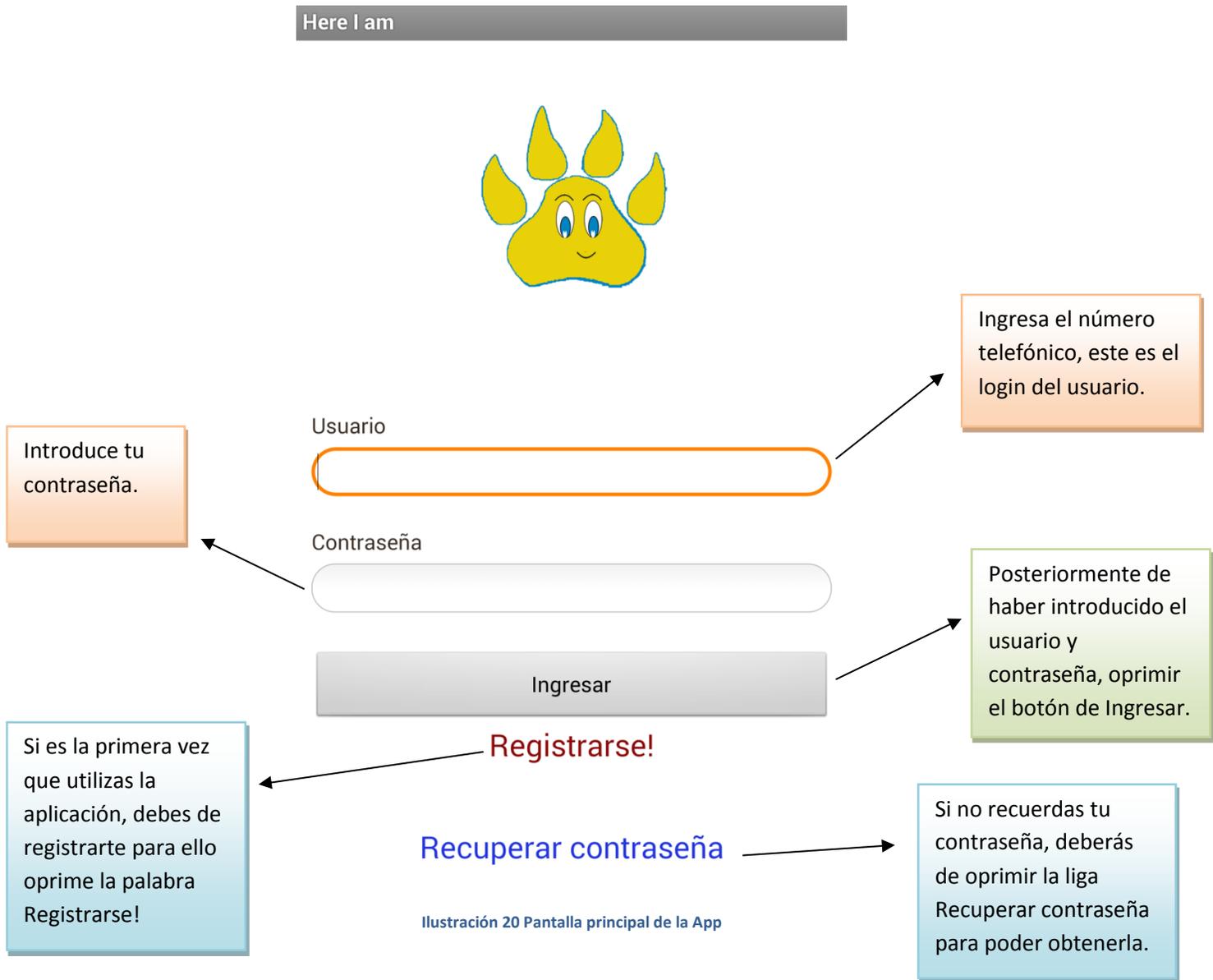
Ilustración 18 Icono de la App

Al iniciar la aplicación se despliega la pantalla que contiene el Splash está contiene los escudos de la Universidad Nacional Autónoma de México y el de la Facultad de Ingeniería.



Ilustración 19 Splash de la App

Esta es la pantalla que le muestra al usuario para que se autentique, registre o recupere su contraseña.



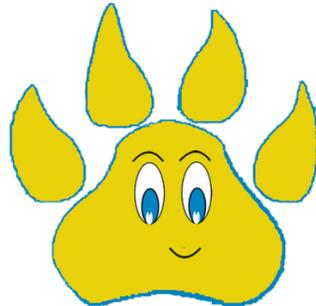
Al oprimir el boton de ingresar, aparece el siguiente mensaje.



Ilustración 21 Autenticando

Si los datos son incorrectos aparece el mensaje de datos incorrectos y se deben de introducir nuevamente los datos.

Here I am



Usuario

55????????

Contraseña

.....

Ingresar

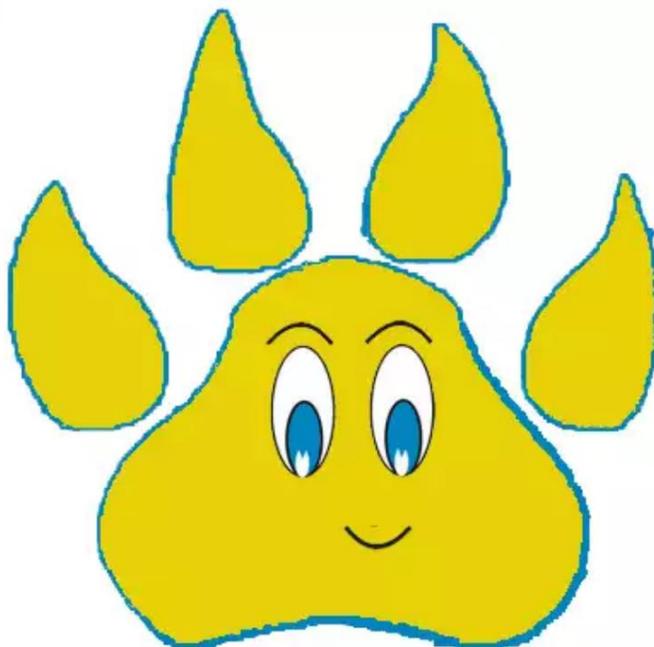
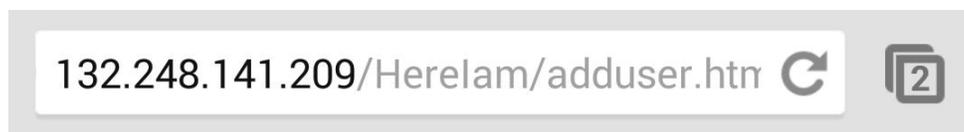
Registrarse!

Error:Nombre de usuario o password incorrectos

[Recuperar contraseña](#)

Ilustración 22 Datos incorrectos

Para registrar un nuevo usuario se abrirá una página en un navegador de internet, se tiene que llenar todos los campos requeridos.



Agregue un nuevo Usuario:

Usuario:

Contraseña:

Selecciona una pregunta secreta

Ilustración 23 Página para registrar usuarios

Existe tres tipos de preguntas diferentes, este campo ayuda a recuperar la contraseña de una forma segura.

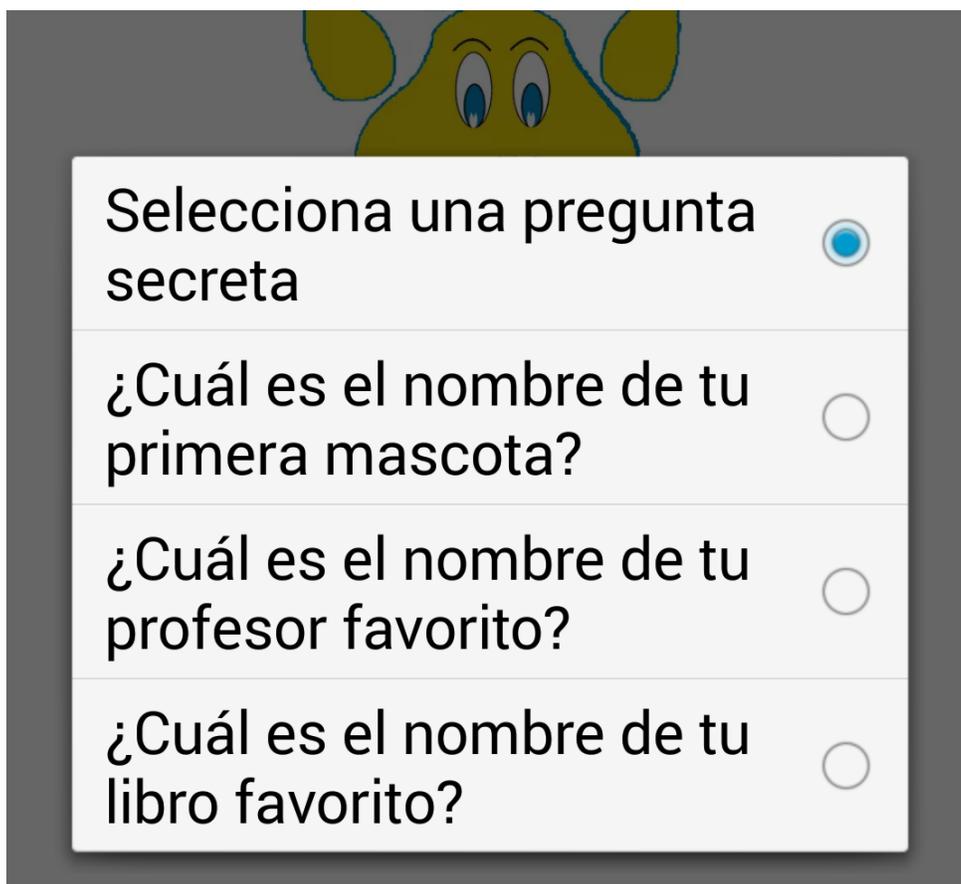
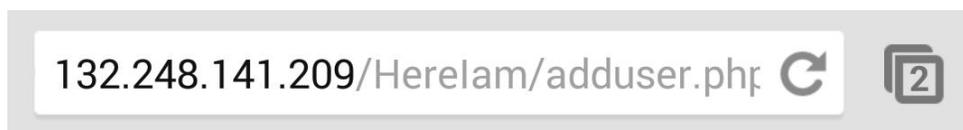


Ilustración 24 Preguntas de seguridad

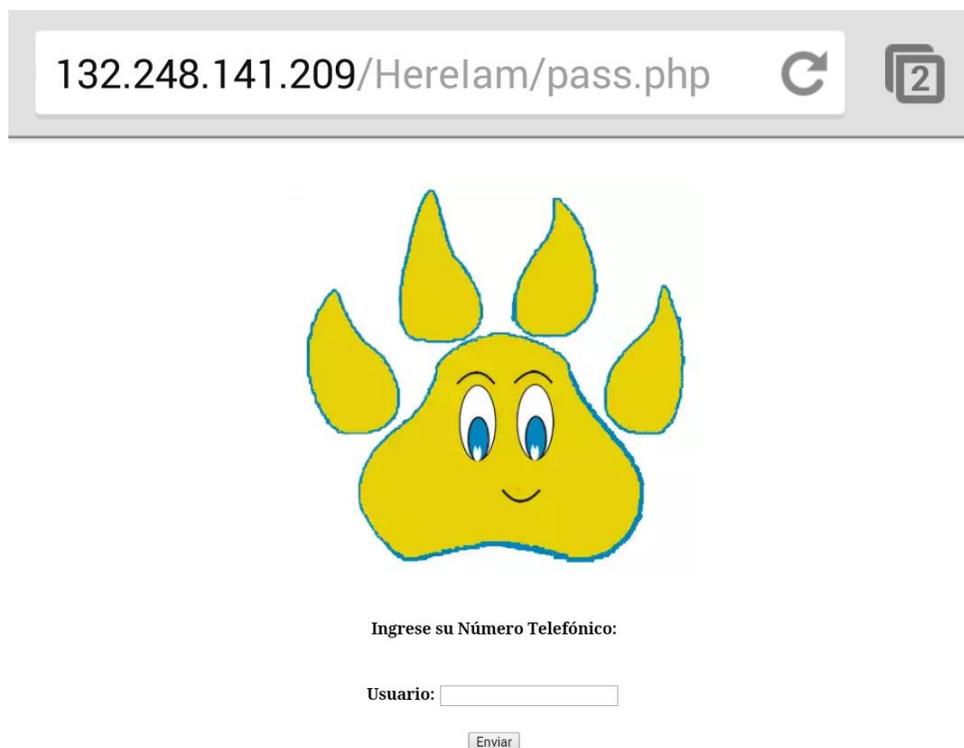
Después de introducir los datos y oprimir el botón de agregar, aparece la siguiente pantalla confirmando que el usuario ya fue dado de alta.



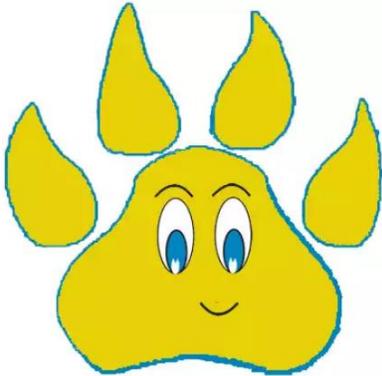
Tu usuario fue agregado satisfactoriamente, Ahora ya podrás utilizar nuestra App

Ilustración 25 Confirmación de nuevo usuario

Para recuperar la contraseña se debe de oprimir la liga de “Recuperar contraseña” y abre la siguiente página, aquí pide el usuario que es el número telefónico.



132.248.141.209/Herelam/pass.php



Ingrese su Número Telefónico:

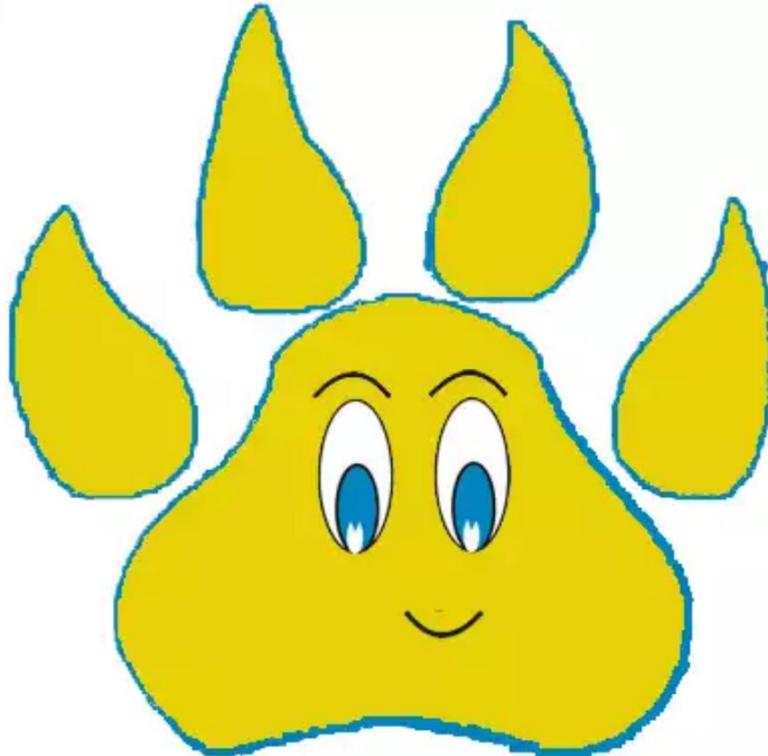
Usuario:

Enviar

Ilustración 26 Página para recuperar contraseña

Al ingresar el usuario y oprimir enviar, nos muestra la pregunta secreta que se eligió cuando se registró el usuario.

132.248.141.209/Herelam/obt_pass.ph



¿Cuál es el nombre de tu primera mascota?

Enviar

Ilustración 27 Página para responder pregunta secreta

Se debe llenar el campo de la respuesta, si los datos son equivocados se envía un mensaje en pantalla y vuelve a pedir el usuario, pero si el campo es correcto nos da la opción de ingresar una nueva contraseña, posteriormente confirma que la contraseña ha sido cambiada satisfactoriamente.

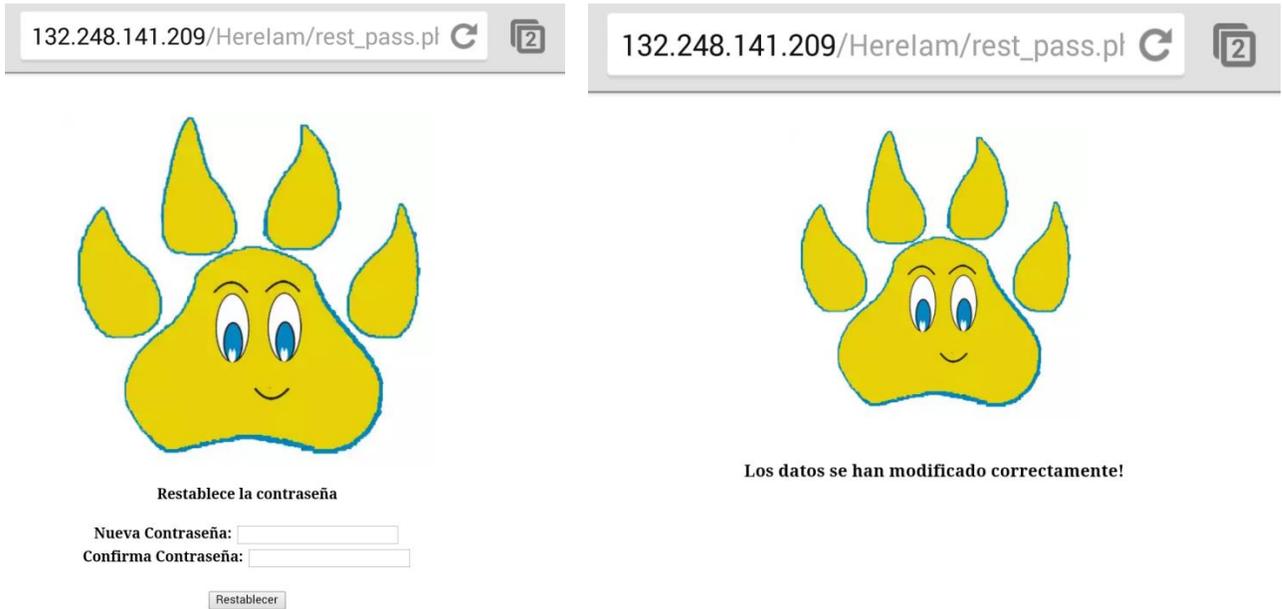


Ilustración 28 Confirmación del cambio de contraseña

Al ingresar a la aplicación con tu usuario y contraseña se cargara en la pantalla los mapas de google, si el GPS del dispositivo está desactivado aparecerá un mensaje en pantalla y solo te mostrara el mapa pero no las ubicaciones de los contactos.

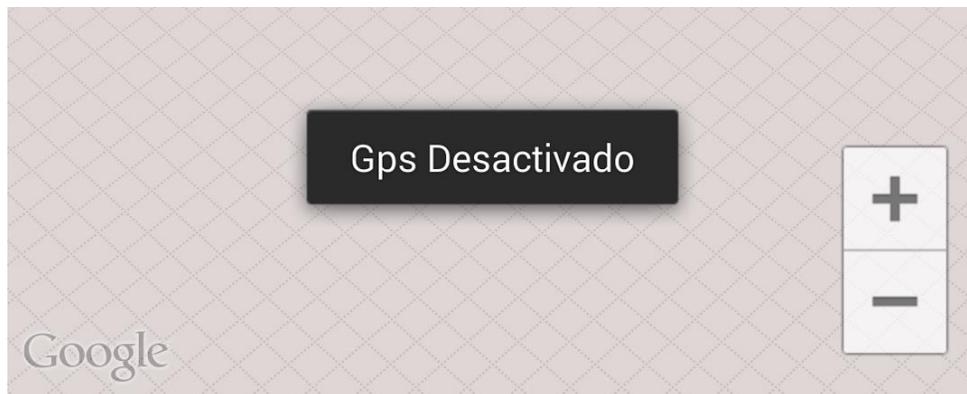


Ilustración 29 Mensaje de GPS desactivado

La siguiente imagen es como funciona la aplicación satisfactoriamente. El punto azul es la ubicación de nuestro dispositivo que estamos utilizando con la aplicación, al seleccionar el globo aparece nuestro usuario que es el número telefónico.

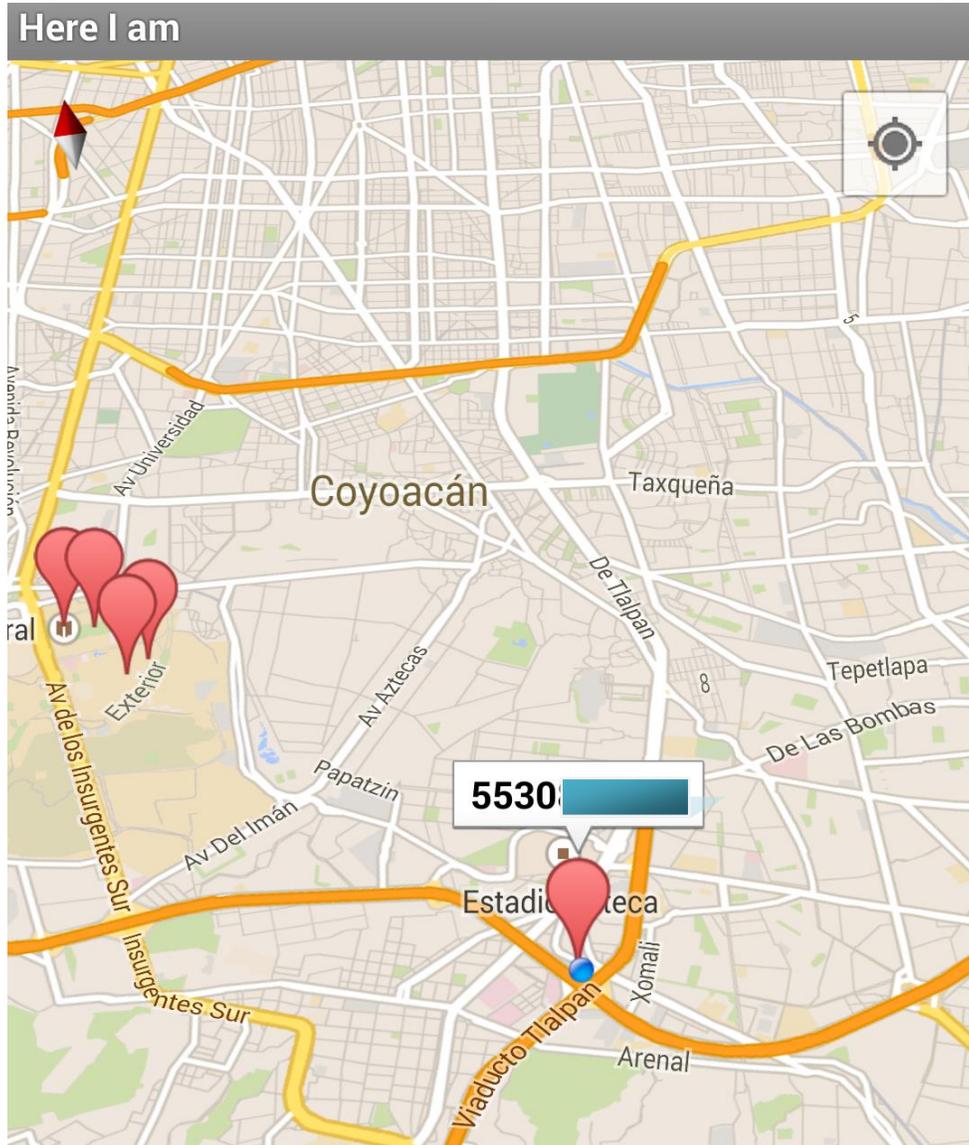


Ilustración 30 Mapa con ubicaciones

Si seleccionamos otro globo rojo que aparecen en el mapa, aparecerá el número telefónico de nuestro contacto.

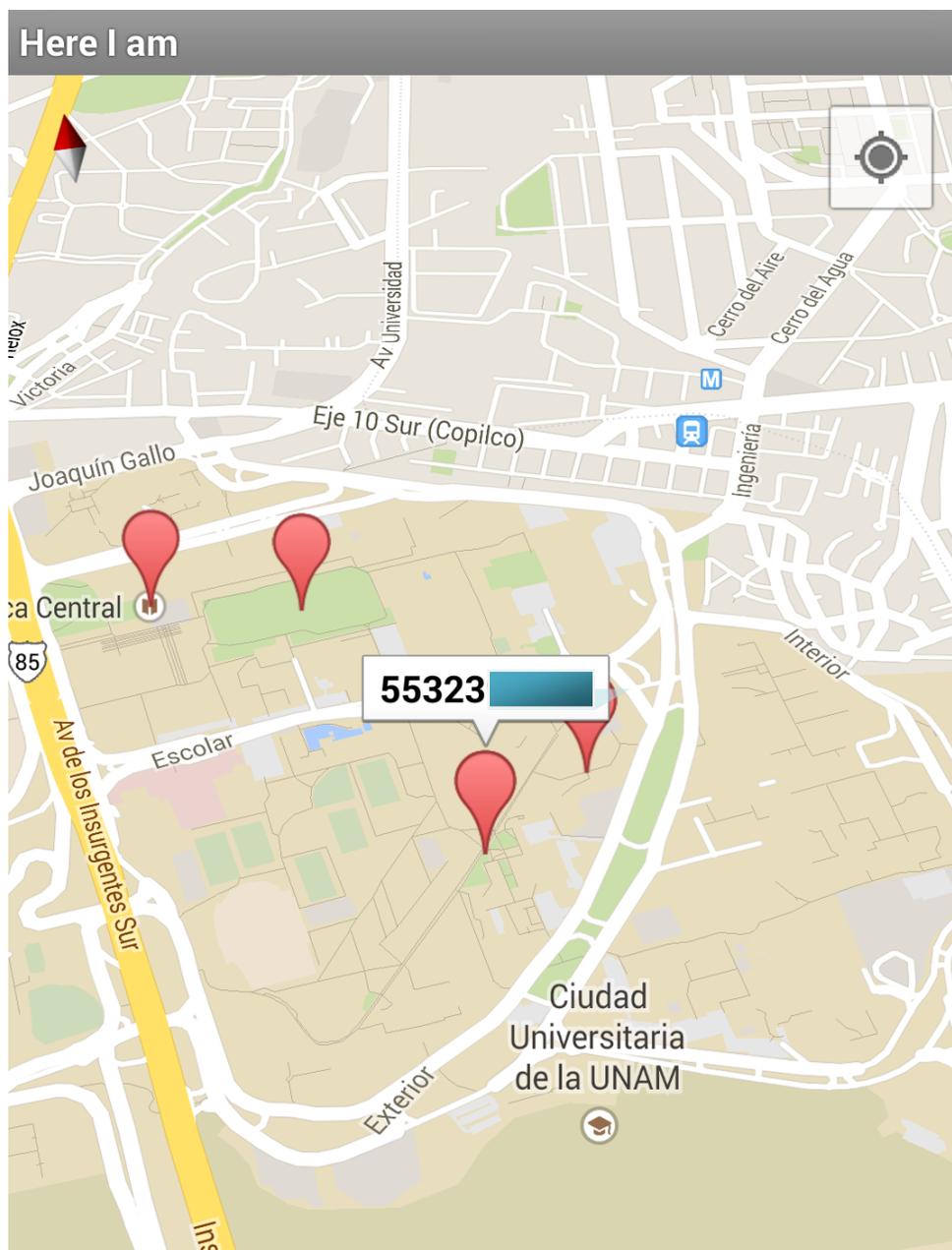


Ilustración 31 Mapa con ubicaciones 2

II. SOPORTE TÉCNICO

Si las especificaciones del dispositivo móvil son las correctas para instalar la aplicación, pero se tienen problemas para instalarla es por los permisos que tiene el dispositivo móvil para instalar aplicaciones que no tengan licencias.

Para permitir la instalación se debe de cambiar la configuración del dispositivo, se tendrán que seguir los siguientes pasos.

Ir a configuración del dispositivo, después al apartado de Seguridad y habilitar la opción de Orígenes desconocidos, así se permitirá instalar esta aplicación.

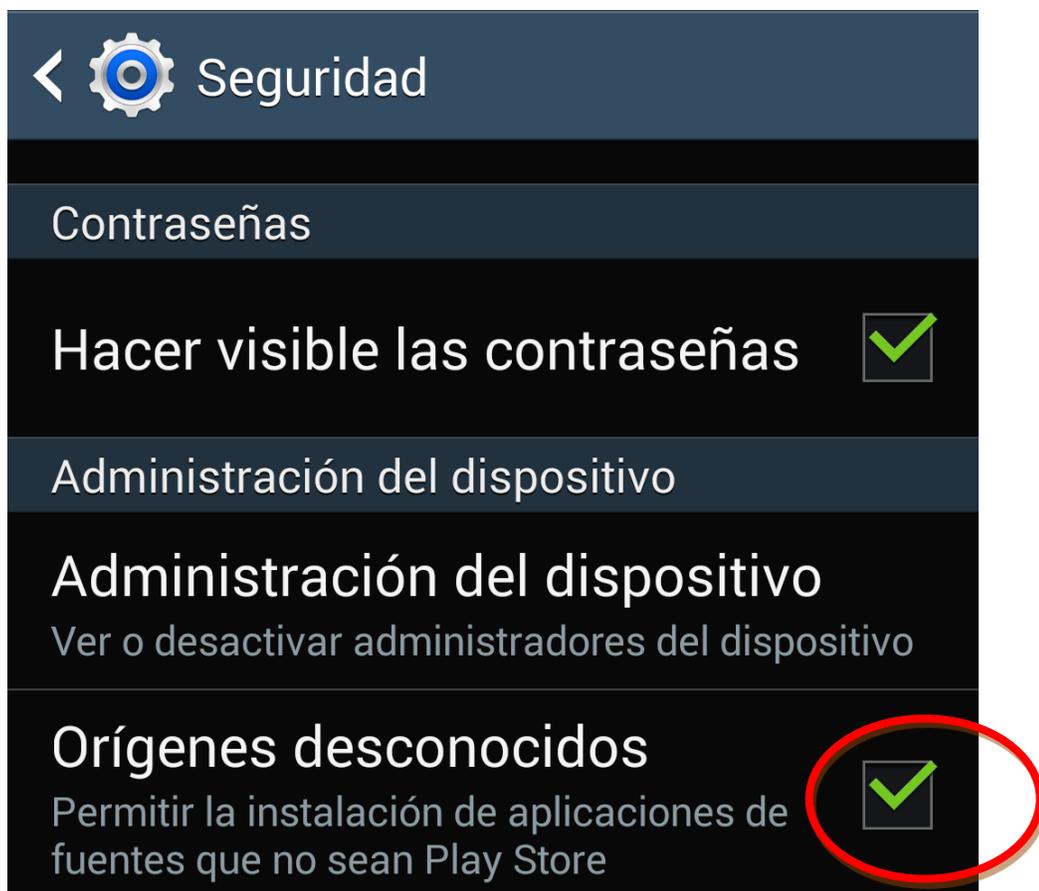


Ilustración 32 Configuraciones del teléfono

Al querer ingresar a la aplicación y sabes que tu usuario y contraseña son las correctas, se debe de verificar que el dispositivo tenga conexión a internet ya que así seguirá enviando un mensaje que el usuario y contraseña son incorrectos, como se muestra en la siguiente imagen.

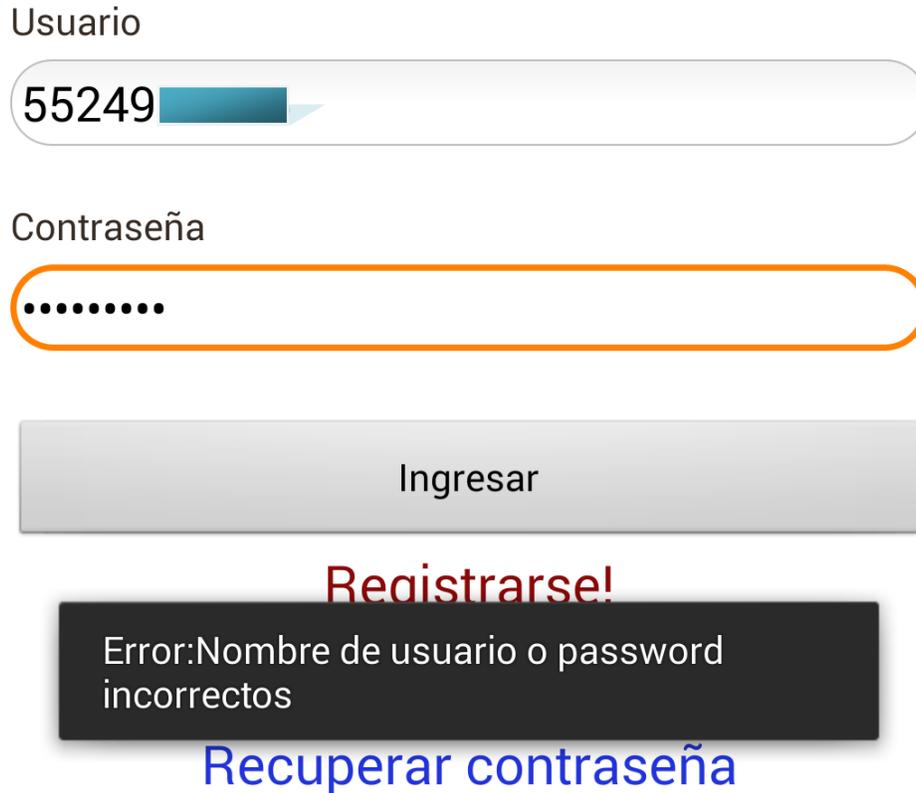


Ilustración 33 Error al ingresar el usuario



Ilustración 34 Icono de una señal de datos móviles



Ilustración 35 Icono de una señal WiFi

Si te aparecen las siguientes imágenes cuando ingresas a la aplicación es por que debes prender tu GPS para que la aplicación pueda obtener tu ubicación y pueda mostrarte la de tus contactos.

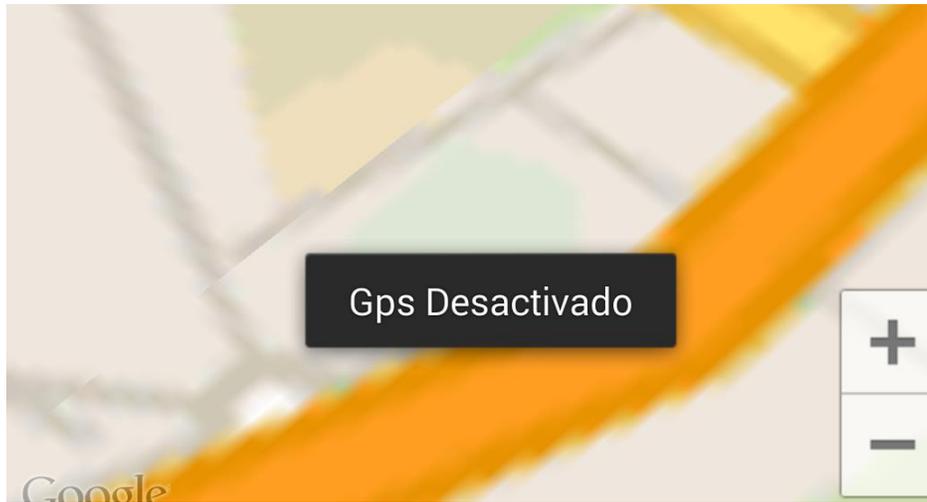


Ilustración 36 GPS Desactivado

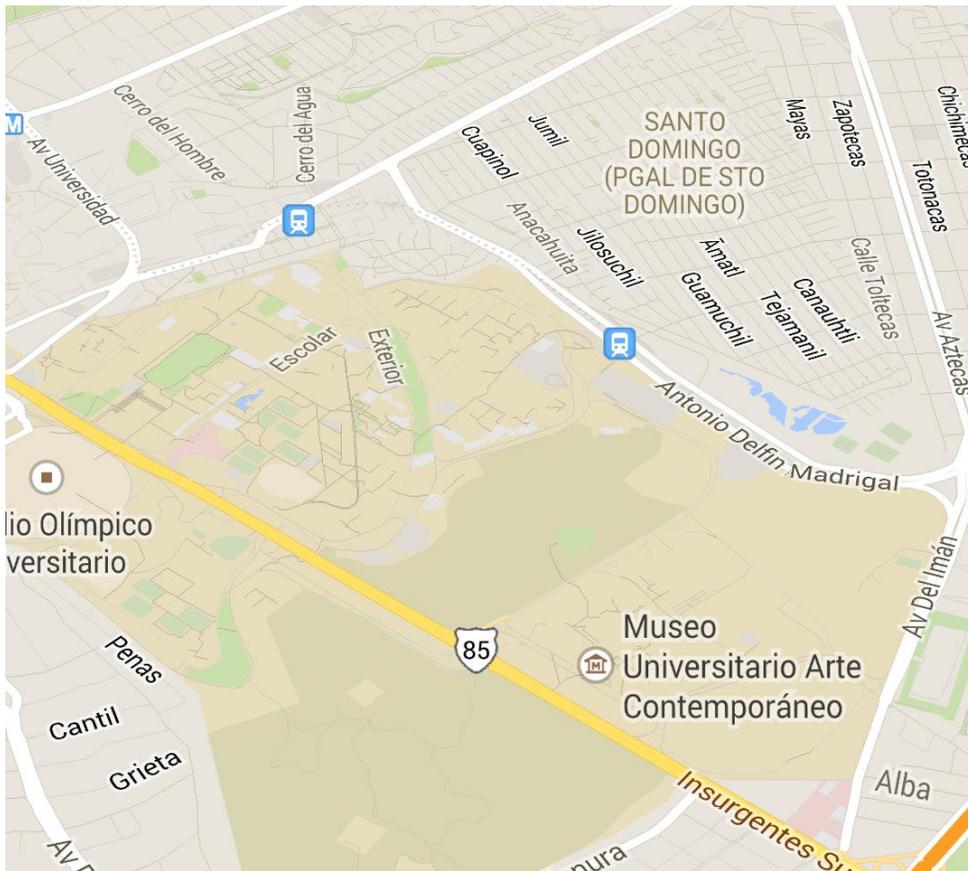


Ilustración 37 Mapa sin ubicaciones de los contactos

III. INTERFAZ INTEGRADA DE DESARROLLO ANDROID

Para comenzar a desarrollar aplicaciones para el sistema operativo Android se solicita instalar un IDE y el SDK de Android, en el capítulo 4.2 se describen los diferentes IDE's, en este apartado se dará las breves instrucciones de como instalar el IDE Eclipse y como configurar el SDK.

Google nos permite descargar estas herramientas en su página oficial, <http://developer.android.com/sdk/> en esta página existe un apartado en donde podemos descargar las siguientes herramientas en un solo paquete:

- Eclipse + ADT plugin
- Android SDK Herramientas
- Android Platform-tools
- Una versión de la plataforma Android
- Una versión de la imagen del sistema Android para el emulador

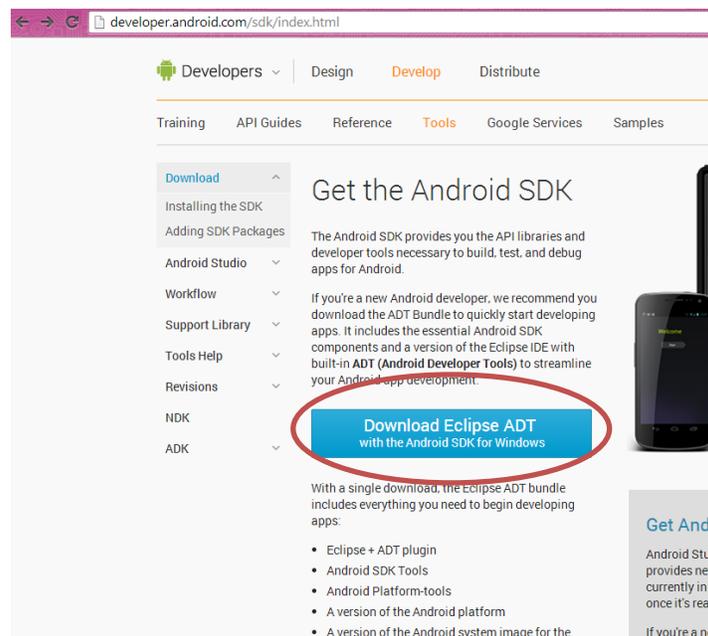


Ilustración 38 Página para descargar Eclipse y SDK Android

Posteriormente de darle clic para descargar el paquete, Google pedirá que aceptes sus términos y condiciones y que se elija los bits del equipo de cómputo en donde se va a instalar Eclipse ya sea 32 o 64 bits.

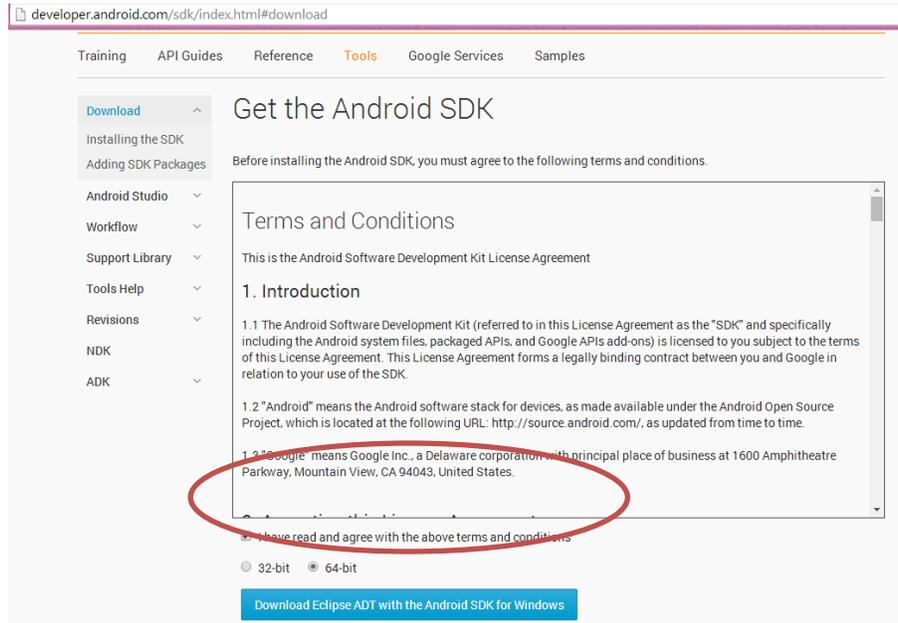


Ilustración 39 Requerimientos para descargar el paquete de Eclipse y SDK Android

Después de elegir estas opciones se comenzara a descargar el paquete.



Ilustración 40 Descarga del paquete Eclipse y SDK Android

Seguir los siguientes pasos:

1. Descomprimir el archivo .zip ya descargado.
2. Abrir la carpeta llamada eclipse esta se encuentra dentro de la carpeta que se acaba de descomprimir.
3. Abrir el archivo llamado eclipse 

4. Elegir la ruta en donde deseamos guardar los proyectos.

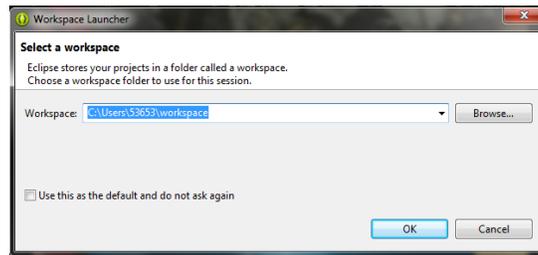


Ilustración 41 Ruta para guardar los proyectos

El SDK de Android no incluye todo lo necesario para empezar a desarrollar. El SDK separa herramientas, plataformas y otros componentes en paquetes se pueden descargar según sea necesario utilizando el Administrador de Android SDK.

5. Haga clic SDK Manager en la barra de herramientas. Así se podrán descargar herramientas conforme nos vaya requiriendo el proyecto a realizar.

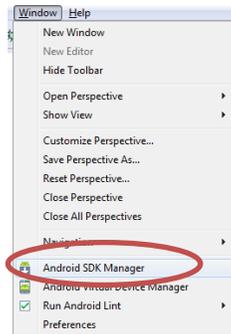


Ilustración 42 SDK Manager para descargar más herramientas

Para comenzar a desarrollar un nuevo proyecto seguir los siguientes pasos

1. Dar clic en File → New → Project
2. Se elige que el tipo de proyecto, en este caso será Android
3. Nombrar el proyecto, el paquete y elegir para que versión del sistema operativo Android se va a orientar la aplicación.

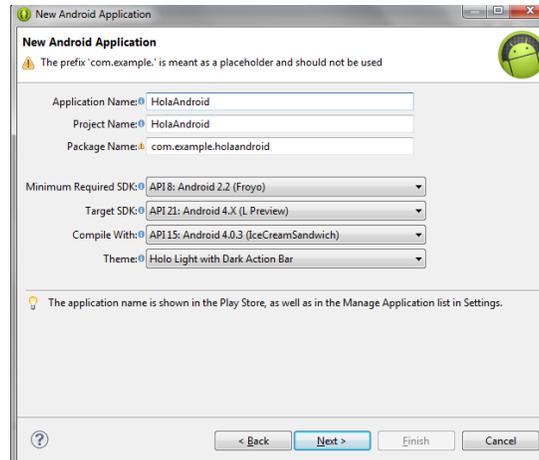


Ilustración 43 Creación de un nuevo proyecto

4. Darle clic en Finalizar.

Realizando los pasos anteriores ya se podrá empezar a desarrollar una aplicación en Android.

Para hacer una aplicación sencilla que solo muestre en pantalla el texto “Hola Android” y ejecutarla en la máquina virtual de Eclipse, procede con los siguientes pasos:

1. Después de que se abrió el proyecto que se acaba de crear abrir el archivo activity_main.xml que se encuentra en la carpeta layout.

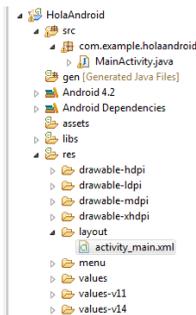


Ilustración 44 Ubicación del archivo activity_main.xml

2. Cambiar el texto que nos crea por default en la parte de TextView, android:text.



Ilustración 45 activity_main.xml

3. Posteriormente abrir el archivo AndroidManifest.xml y mandar a llamar la actividad que se va a ejecutar para que muestre el texto.



Ilustración 46 AndroidManifest.xml

4. Para correr la aplicación se debe crear una máquina virtual. Abrir Window→Android Virtual Device Manager y dar clic en New.

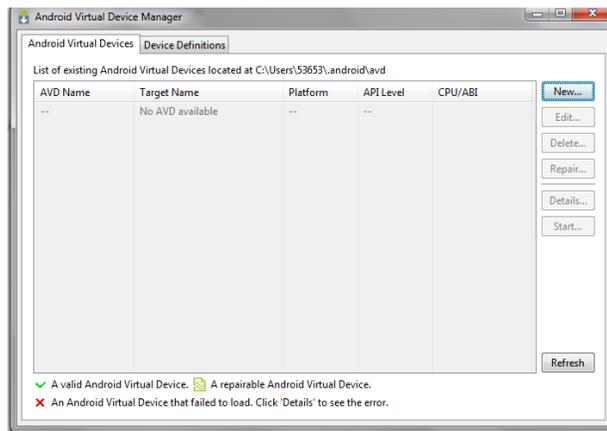


Ilustración 47 Android Virtual Device Manager

5. Nombrar la máquina virtual, elegir el tamaño del display y la versión de Android que soportara, todas estas características van conforme a los requerimientos del proyecto que se esté realizando.

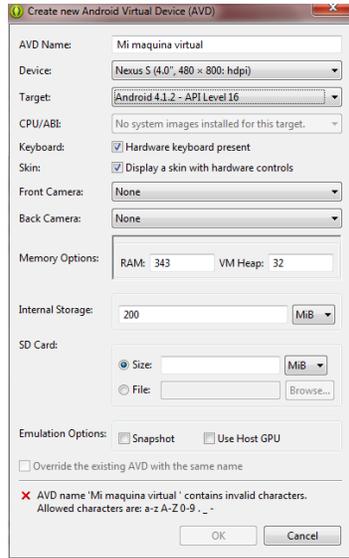


Ilustración 48 Configuración de la máquina virtual

6. Agregada la máquina ya se puede ejecutar la aplicación.



Ilustración 49 Botón para ejecutar la aplicación

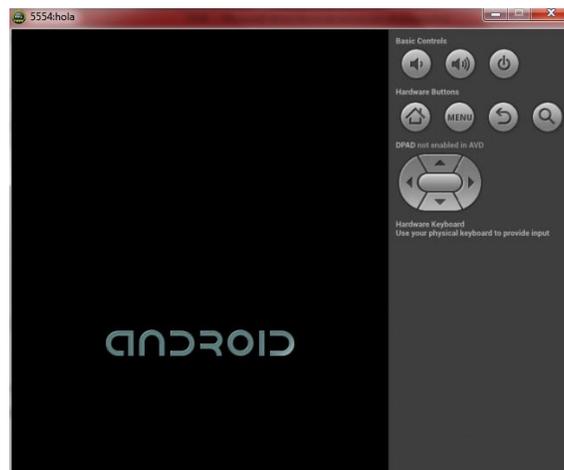


Ilustración 50 Iniciando la máquina virtual de Android

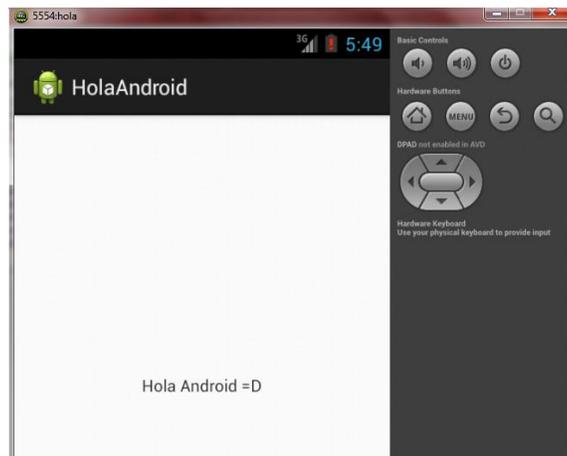
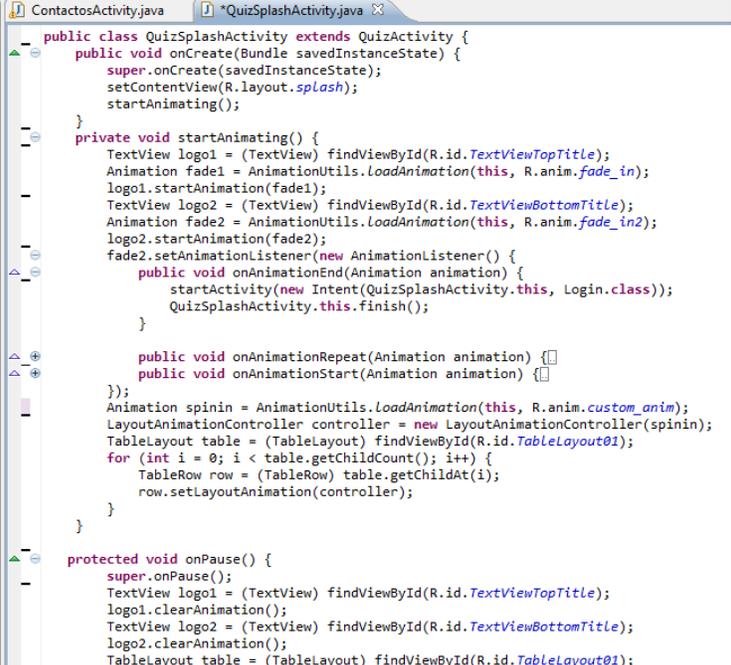


Ilustración 51 Ejecutando el proyecto

IV. CODIFICACIÓN

La siguiente imagen contiene fragmentos de la clase SplashActivity, esta clase es la encargada de desplegar el splash de la aplicación.



```

public class QuizSplashActivity extends QuizActivity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        startAnimating();
    }

    private void startAnimating() {
        TextView logo1 = (TextView) findViewById(R.id.TextViewTopTitle);
        Animation fade1 = AnimationUtils.loadAnimation(this, R.anim.fade_in);
        logo1.startAnimation(fade1);
        TextView logo2 = (TextView) findViewById(R.id.TextViewBottomTitle);
        Animation fade2 = AnimationUtils.loadAnimation(this, R.anim.fade_in2);
        logo2.startAnimation(fade2);
        fade2.setAnimationListener(new AnimationListener() {
            public void onAnimationEnd(Animation animation) {
                startActivity(new Intent(QuizSplashActivity.this, Login.class));
                QuizSplashActivity.this.finish();
            }
        });

        public void onAnimationRepeat(Animation animation) {}
        public void onAnimationStart(Animation animation) {}
    });
    Animation spinin = AnimationUtils.loadAnimation(this, R.anim.custom_anim);
    LayoutAnimationController controller = new LayoutAnimationController(spinin);
    TableLayout table = (TableLayout) findViewById(R.id.TableLayout01);
    for (int i = 0; i < table.getChildCount(); i++) {
        TableRow row = (TableRow) table.getChildAt(i);
        row.setLayoutAnimation(controller);
    }
}

protected void onPause() {
    super.onPause();
    TextView logo1 = (TextView) findViewById(R.id.TextViewTopTitle);
    logo1.clearAnimation();
    TextView logo2 = (TextView) findViewById(R.id.TextViewBottomTitle);
    logo2.clearAnimation();
    TableLayout table = (TableLayout) findViewById(R.id.TableLayout01);
}

```

Ilustración 52 Código clase SplashActivity

Código de la clase ContactosActivity, esta clase realiza la lectura de los contactos del dispositivo móvil.

```

*ContactosActivity.java
public void readContacts()
{
    StringBuffer sb = new StringBuffer();
    sb.append("");
    ContentResolver cr = getContentResolver();
    Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI, null, null, null, null);
    String phone = null;
    String image_uri = "";
    Bitmap bitmap = null;
    if (cur.getCount() > 0)
    {
        while (cur.moveToNext())
        {
            String id = cur.getString(cur.getColumnIndex(ContactsContract.Contacts._ID));
            String name = cur.getString(cur.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
            if (Integer.parseInt(cur.getString(cur.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER))) > 0)
            {
                System.out.println("name : " + name + ", ID : " + id);
                sb.append("\n Contact Name:" + name);
                Cursor pCur = cr.query( ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
                    ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = ?", new String[] { id }, null);
                while (pCur.moveToNext())
                {
                    phone = pCur.getString(pCur.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
                    sb.append("\n Phone number:" + phone);
                    System.out.println("phone" + phone);
                }
                pCur.close();
            }
        }
    }
}

```

Ilustración 53 Código clase ContactosActivity

La clase Login realiza la autenticación del usuario así también como el registro y recuperación de contraseña.

```

*ContactosActivity.java  *MapaActivity.java  *Login.java
public void onClick(View view) {
    String usuario = user.getText().toString();
    String passw = pass.getText().toString();

    if (checklogindata(usuario, passw) == true) {
        // si pasamos esa validacion ejecutamos el AsyncTask pasando
        // el usuario y clave como parametros
        new asynclogin().execute(usuario, passw);
    } else {
        // si detecto un error en la primera validacion vibrar y
        // mostrar un Toast con un mensaje de error.
        err_login();
    }
}

});

olvpasword.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        // Abre el navegador al formulario pass.php
        String url = "http://" + IP_Server + "/HereIam/pass.php";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
});

registrar.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        // Abre el navegador al formulario adduser.html
        String url = "http://" + IP_Server + "/HereIam/adduser.html";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
    }
});
}
}
}

```

Ilustración 54 Código clase Login

La imagen siguiente muestra el código de la clase Mapa, esta clase es la que realiza el llamado de los mapas de Google y pinta la ubicación de los contactos y usuario.

```
public class MapaActivity extends Activity {

    TextView textDetail;
    MarkerOptions mk;
    GoogleMap mMap;
    EditText user;
    HttpPostaux post;
    String numeros = "";

    String IP_Server = "132.248.141.209"; // IP DE NUESTRO PC
    String URL_connect = "http://" + IP_Server + "/HereIam/local.php";
    String URL_connect2 = "http://" + IP_Server + "/HereIam/consulta.php";

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map);

        textDetail = (TextView) findViewById(R.id.textview1);
        readContacts();

        LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        LocationListener miloclistener = new MiLocationListener();
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
            miloclistener);

        LatLng CU = new LatLng(19.31981, -99.184484);
        float zoom = 14;

        mMap = ((MapFragment) getSupportFragmentManager().findFragmentById(R.id.map))
            .getMap();
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(CU, zoom));

        mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        CameraPosition camPos = new CameraPosition.Builder().target(CU)
            .zoom(14).bearing(45).tilt(70).build();

        CameraUpdate camUpd3 = CameraUpdateFactory.newCameraPosition(camPos);
```

Ilustración 55 Código clase Mapa