



**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**  
*La Universidad Católica de Loja*

**ÁREA TÉCNICA**

TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y  
COMPUTACIÓN

**Desarrollo de un aplicativo móvil que en base a puntos georeferenciados permita consultar la proximidad de las paradas de buses al acercarse a las mismas. Caso de Estudio rutas de buses UTPL utilizando MVC y Frameworks de Desarrollo Android.**

TRABAJO DE TITULACIÓN.

**AUTOR:** Cadena Zambrano, Edison David

**DIRECTOR:** Guamán Coronel, Daniel Alejandro, Mgs

LOJA – ECUADOR

2016



*Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>*

Septiembre, 2016

## APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magíster.

Daniel Alejandro Guamán Coronel.

### DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: **Desarrollo de un aplicativo móvil que en base a puntos georeferenciados permita consultar la proximidad de las paradas de buses al acercarse a las mismas. Caso de Estudio rutas de buses UTPL utilizando MVC y Frameworks de Desarrollo Android** realizado por **Cadena Zambrano Edison David**, ha sido orientado y revisado durante su ejecución, por se aprueba la presentación del mismo.

Loja, mayo de 2016

f) .....

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo **Cadena Zambrano Edison David** declaro ser autor del presente trabajo de titulación; Desarrollo de un aplicativo móvil que en base a puntos Georeferenciados permita consultar la proximidad de las paradas de buses al acercarse a las mismas. Caso de Estudio rutas de buses UTPL utilizando MVC y Frameworks de Desarrollo Android, de la Titulación Sistemas Informáticos y Computación, siendo Guamán Daniel Alejandro director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad Intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f .....

Autor: Cadena Zambrano Edison David

Cédula: **1721952198**

## **DEDICATORIA**

El presente trabajo de fin de titulación  
está dedicado a mis padres,  
porque gracias a su ejemplo y esfuerzo  
me han enseñado a perseverar  
hasta el final para cumplir todas mis metas.

## **AGRADECIMIENTO**

A Dios todo poderoso, mis padres, mi familia,  
mi novia y a todos mis amigos que de alguna manera  
me apoyaron para conseguir este objetivo.

A mis tutores y todo el personal académico de la  
Universidad Técnica Particular de Loja  
que me compartieron sus conocimientos  
los cuales me han servido para llegar  
a culminar el presente trabajo de fin de titulación.

## ÍNDICE DE CONTENIDOS

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA.....	i
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS.....	vi
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
CAPÍTULO I.....	4
1.1. Arquitectura de software.....	5
1.1.1. Importancia de la arquitectura de software.....	5
1.1.2. Estilos de arquitectura de software.....	6
1.1.3. Selección del estilo arquitectónico de llamada y retorno.....	11
1.2. Metodologías para el desarrollo de software.....	11
1.2.1. Metodologías estructuradas.....	11
1.2.2. Metodologías orientadas a objetos.....	12
1.2.3. Metodologías tradicionales.....	12
1.2.4. Metodologías ágiles.....	13
1.3. Selección de la metodología de desarrollo de software.....	13
1.3.1. Metodología XP (Programación Extrema).....	15
1.3.1.1. <i>Definiciones</i> .....	15
1.3.1.2. <i>Actividades fundamentales de XP</i> .....	15
1.4. Android.....	21
1.4.1. Versiones de Android.....	21
1.4.2. Desarrollo de aplicaciones Android.....	22
1.5. Tecnologías existentes de desarrollo de aplicaciones móviles.....	24
1.5.1. Ventajas.....	24
1.5.2. Desventajas.....	25
1.5.3. Selección del tipo de aplicación móvil.....	25
1.6. Lenguajes de programación y herramientas de desarrollo de software.....	26
1.6.1. Lenguajes de programación desarrollo web.....	26
1.6.1.1. <i>PHP</i> .....	26
1.6.1.2. <i>JavaScript y jQuery</i> .....	27
1.6.2. Lenguajes de programación desarrollo Android.....	27
CAPÍTULO II.....	29

2.	Introducción .....	30
2.1.	Situación actual.....	30
2.2.	Alternativa de solución de software .....	30
2.3.	Herramientas.....	31
2.4.	Iteraciones .....	32
2.5.	Definición del alcance .....	33
2.5.1.	Sistema de administración web.....	33
2.5.2.	Aplicación móvil Android. ....	36
2.5.3.	Resultados esperados.....	37
CAPÍTULO III.....		38
3.	Desarrollo de la solución de software .....	39
3.1.	Planeación.....	39
3.2.	Diseño .....	43
3.2.1.	Tarjetas CRC.....	44
3.2.2.	Modelo Entidad-Relación.....	44
3.2.3.	Diccionario de datos. ....	45
3.2.4.	Diagrama de solución.....	45
3.2.5.	Diagrama arquitectura de software.....	47
3.2.6.	Diagrama de clases.....	48
3.2.7.	Casos de uso.....	49
3.3.	Codificación.....	49
3.3.1.	Solución WEB.....	49
3.3.2.	Aplicación Móvil.....	52
3.4.	Pruebas e implementación .....	55
3.4.1.	Pruebas.....	55
3.4.2.	Implementación de Aplicaciones.....	57
CONCLUSIONES .....		58
RECOMENDACIONES.....		59
BIBLIOGRAFÍA.....		60
ANEXOS.....		63
ANEXO A.....		64
ANEXO B.....		74
ANEXO C.....		76
ANEXO E.....		96
ANEXO F.....		99
ANEXO G.....		117
ANEXO H.....		122



ANEXO I .....	125
---------------	-----

## RESUMEN

En el presente trabajo de titulación se expone el desarrollo de una aplicación web utilizando el patrón arquitectónico Modelo Vista Controlador (MVC) y una aplicación nativa para dispositivos móviles con sistema operativo android la cual permita alertar paradas cercanas de los diferentes recorridos de buses de la Universidad Técnica Particular de Loja.

Como metodología de desarrollo de software se utilizó Programación Extrema (XP). En cuanto a los lenguajes de programación utilizados para la aplicación web se encuentran PHP, JavaScript y para la aplicación móvil Java que es el lenguaje de desarrollo de aplicaciones nativas Android. Además como ente intermedio para la comunicación de ambas aplicaciones se creó un servicio web generado por el framework SLIM el cual permite crear servicios Web RESTful y los muestra en el formato para intercambio de datos JSON.

La aplicación Web y móvil integra un mapa proporcionado por el servidor de aplicaciones de mapas Google Maps, además la aplicación móvil consume los servicios de geolocalización(GPS) para ubicar el lugar donde se encuentra el dispositivo.

**PALABRAS CLAVES:** Android, Controlador, GPS, Google Maps, Java, JavaScript, Json, Metodología Ágil, Modelo, MVC, PHP, RESTful, SLIM, Vista, Web, XP.

## ABSTRACT

Web application development using a Model View Controller architecture and a native application for mobile devices running the Android operating system which will alert users as to university (Universidad Técnica Particular de Loja) bus stops close to them.

The development model used was Extreme Programming (XP). Programming Languages used included PHP and Javascript for the web application and Java for the mobile application. To act as an intermediary between the web application and the mobile application, the SLIM framework was used to create RESTful web services using JSON as a format for data transfers.

The web application and the mobile application as well both implement a map provided by Google Maps, additionally the mobile application uses GPS to figure out where the phone is.

**KEYWORDS:** Agile Development, Android, Controller, GPS, Google Maps, Java, JavaScript, Json, Model, MVC, PHP, RESTful, SLIM, View, Web, XP.

## INTRODUCCIÓN

En el presente trabajo de fin de titulación se propone un prototipo para los usuarios que hacen uso del servicio de recorrido de buses de la Universidad Técnica Particular de Loja y desconocen las paradas en las cuales el bus se estaciona a dejar y recoger pasajeros lo que limita a los usuarios sobre todo visitantes a que hagan uso del servicio que brinda la Universidad.

Por tal razón se ha planteado un prototipo que se dividirá en dos partes: la primera una aplicación web que estará dirigido a los administradores del sistema los cuales gestionen la información, es decir que tengan la posibilidad de agregar, modificar o eliminar nuevas paradas o rutas y la segunda, una aplicación para dispositivos móviles con sistema operativo android que permita alertar las paradas cercanas en una ruta determinada con el objetivo de avisar al usuario en que estación podría desembarcar.

El presente proyecto se encuentra estructurado en 4 capítulos que se complementan uno del otro, realizando el respectivo marco teórico donde se describe una investigación acerca de las metodologías de desarrollo de software, arquitecturas, patrones arquitectónicos, programación para sistemas operativos android entre otros. En el capítulo 2 se expone acerca de la posible solución, definición del alcance, las herramientas a utilizar, etc. Los posteriores capítulos contienen información acerca del desarrollo del proyecto tanto para la aplicación web y móvil además de las pruebas realizadas en las aplicaciones, finalizando con las conclusiones y recomendaciones respectivas.

La propuesta que se plantea parte de una iniciativa generada por el autor del presente proyecto en conjunto con el director de trabajo de fin de titulación, la cual plantea una posible solución a los usuarios que desconocen las paradas del recorrido de buses de la UTPL.

En el transcurso de la realización de todo el proyecto se presentaron una serie de inconvenientes que se enfrentó en el desarrollo y estas se fueron solventando gracias a la amplia gama de información que existe en internet y libros que sirvieron como base fundamental para cumplir con los objetivos planteados desde un principio, los cuales han sido cumplidos en su totalidad.

**CAPÍTULO I**  
**MARCO TEÓRICO**

## **1.1. Arquitectura de software**

Muchos autores exponen varias definiciones acerca de la arquitectura de software, una de ellas es el documento IEEE Std 1471-2000 donde manifiesta que: “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución” (Maier, Emery, & Hilliard, 2001, p. 108). Clements (1996) afirma:

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones. (p.2)

En una definición un poco amplia, David Garlan establece que la AS constituye un puente entre el requerimiento y el código, ocupando el lugar que en los gráficos antiguos se reservaba para el diseño (Garlan, 2000).

Ambos autores relacionan a la arquitectura de software como un sistema que interactúa entre sus componentes. Haciendo una comparación con la arquitectura tradicional la cual se encarga de diseñar los planos de un edificio, su estructura y su diseño, en el software es algo similar ya que se elabora el diseño y la estructura global de manera que se organiza todo el proceso de desarrollo de Software.

### **1.1.1. Importancia de la arquitectura de software.**

Formulando la interrogante de ¿cómo sería la construcción de un edificio sin un plano?, se llegaría a la conclusión que sería muy complicado la realización del mismo, sin haber definido en un principio el diseño así como su estructura.

En el caso de la Arquitectura de Software se refleja algo similar, si se tiene un proyecto de desarrollo de software y por ejemplo no se ha elegido un estilo arquitectónico, o no hay una metodología de desarrollo definida y se empieza con la programación, el resultado final del proyecto será un fracaso con seguridad.

Es por tal motivo que la arquitectura de software cumple un factor importante porque es la encargada de ensamblar todos los componentes, la comunicación entre ellos y la implementación de estructuras de software al más alto nivel.

### 1.1.2. Estilos de arquitectura de software.

Dentro de la arquitectura de software se encuentran los estilos arquitectónicos, según Fielding (2000) un estilo arquitectónico es “un conjunto coordinado de restricciones arquitectónicas que restringe los roles/rasgos de los elementos arquitectónicos y las relaciones permitidas entre esos elementos dentro de la arquitectura que se conforma a ese estilo”(p. 13).

Sin embargo ¿cuál es su diferencia con los patrones de diseño?. La forma de expresar la diferencia entre estos dos temas resulta un poco complicado a tal punto que puede haber una confusión en la que se concluya que es lo mismo, esto sin duda alguna es debido a que la arquitectura de software se encuentra en un proceso de maduración y consolidación. No obstante, si se va a enfocar en los tipos de patrones existentes el tema se ampliará significativamente es por tal motivo que se expone un extracto del documento, **Estilos y Patrones de la Estrategia de Arquitectura de Microsoft** donde los autores Reinoso y Kicillof (2004) afirman:

Los estilos se encuentran en el centro de la arquitectura y constituyen buena parte de su sustancia. Los patrones de arquitectura están claramente dentro de la disciplina arquitectónica, solapándose con los estilos, mientras que los patrones de diseño se encuentran más bien en la periferia, si es que no decididamente afuera.(p. 3)

Entre los estilos arquitectónicos más representativos según Reinoso y Kicillof (2004) se exponen en la figura 1., sin embargo cabe resaltar que en el presente proyecto estará enfocado en los *estilos de llamada y retorno* los cuales se detallará más adelante.

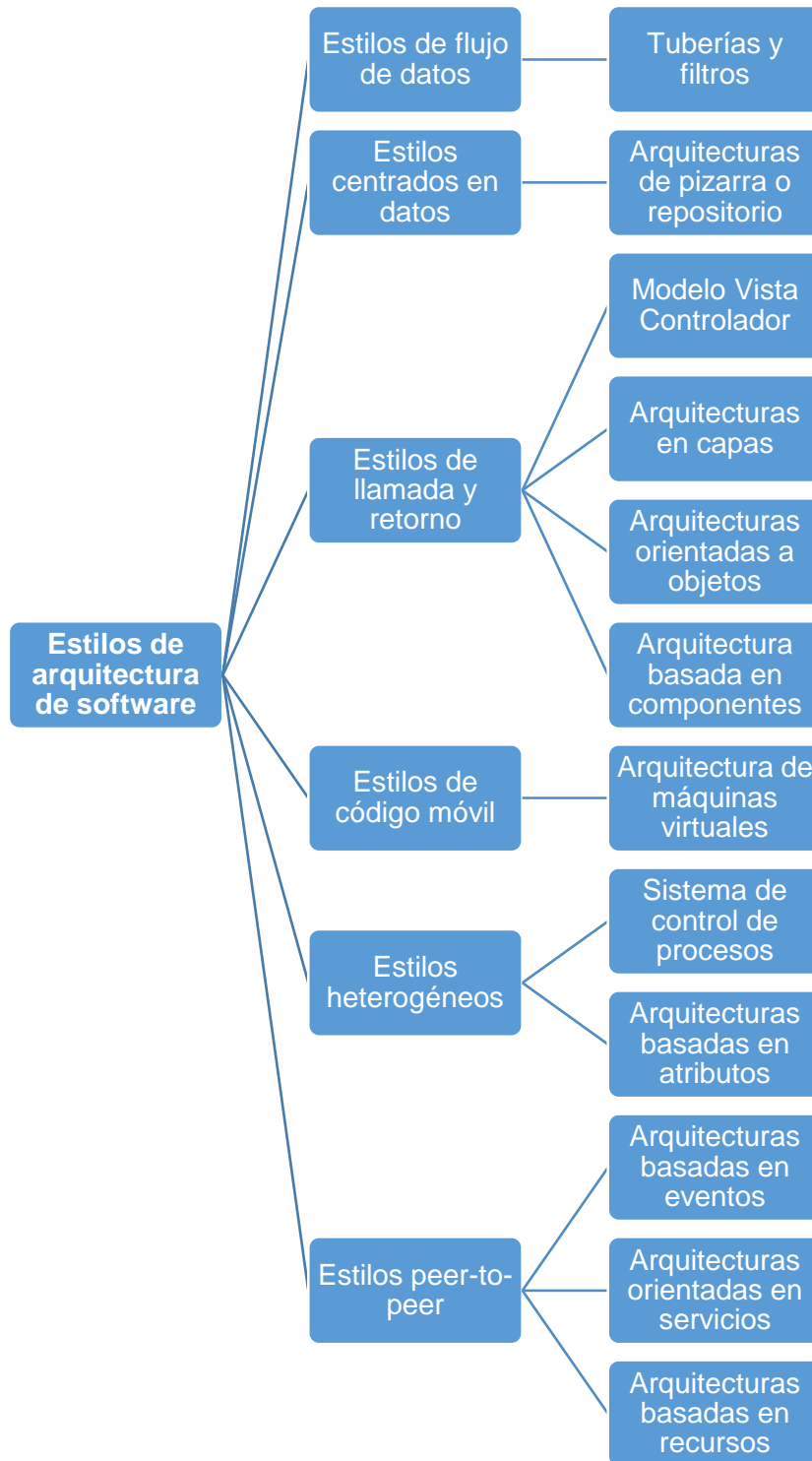


Figura 1. Estilos de arquitectura de software

Fuente: (Reinoso & Kicillof, 2004)

Elaboración: Edison Cadena



### 1.1.2.1. Estilos de llamada y retorno.

El sitio web Ecured (s.f.) describe a estos estilos como la descomposición jerárquica en subrutinas (componentes) que solucionan una tarea o función definida, además permite al diseñador del software construir una estructura de programa relativamente fácil de modificar y ajustar a escala.

Por otro lado Reynoso y Kicillof (2004) afirman la modificabilidad y la escalabilidad de los sistemas, usados en sistemas orientados a objetos, de llamadas a procedimientos remotos y los jerárquicos en capas.

Ambos autores hacen hincapié a que este tipo de estilos se utilizan en sistemas de gran escala por su amplia capacidad de modificación y adaptación al cambio sin duda alguna sería el más elegido por los desarrollares de software.

Dentro de este estilo se encuentran los siguientes estilos arquitectónicos:

#### 1.1.2.1.1. Estilo arquitectónico modelo vista controlador (MVC).

El estilo arquitectónico MVC divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y tiempo reducido (Gutiérrez, 2006).

Hay que resaltar que algunos autores lo llaman patrón arquitectónico en este caso Microsoft (s.f.) también argumenta que “es un patrón fundamental de la separación de la lógica de la interfaz de usuario de la lógica de negocio” (p. 1).

La figura 2 muestra la relación estructural entre los tres objetos.

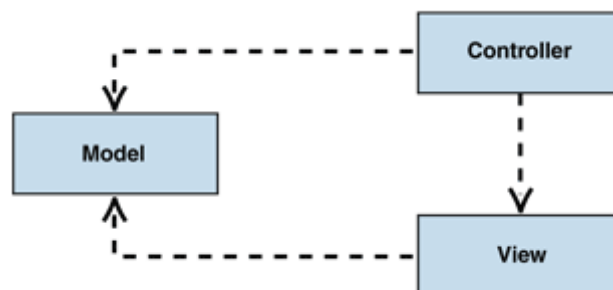


Figura 2. Relación estructural entre objetos.

Fuente: (Enterprise Solution Patterns., s.f.)

Elaboración: Edison Cadena

MVC se basa en dividir la aplicación en 3 partes fundamentales según Burbeck (1992):

**Modelo.-** Incorpora la capa de dominio y persistencia, alojando los datos en un medio de almacenamiento como base de datos, archivo de texto, etc. El modelo es el que se encarga de ser el proveedor de recursos tanto para la vista como también de responder instrucciones que vengan generalmente desde el controlador.

**Vista.-** Encargada de visualizar la información en sistemas web es la parte de cliente donde interviene HTML, cabe resaltar que en esta parte solo se deben efectuar operaciones simples.

**Controlador.-** El controlador interpreta las entradas de ratón y teclado del usuario, informando el modelo y / o la vista para cambiar según el caso. Se puede argumentar que es el mediador entre los cambios de la vista los cuales se los envía al modelo, este a su vez regresa a la vista con los cambios efectuados de modo que se convierte en un ciclo donde cada acción de usuario hace que se inicie un nuevo ciclo.

Este estilo arquitectónico fue diseñado para reducir el esfuerzo de programación debido que el Modelo, la Vista y el Controlador se tratan como entidades separadas, esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

#### 1.1.2.1.2. *Estilo arquitectónico en capas.*

Garlan y Shaw (1994) definen el estilo en capas como “una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior” (p. 28).

Pelaez (2009a) en su blog expone que esta arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica facilitando una forma muy práctica de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

Para que exista una arquitectura en capas mínimo debe haber dos como puede ser el caso de la arquitectura cliente/servidor, siendo un poco más general no orientado a la parte de software puede ser el modelo OSI (Open System Interconnection) con sus 7 niveles como son: nivel físico, vinculo de datos, red, transporte, sesión, presentación y aplicación.

Cuando se habla de arquitectura en capas mayoritariamente se hace énfasis en la arquitectura tres capas la cual tiene tres componentes: la capa de presentación, negocio y de acceso a datos. Este tipo de arquitectura sólo puede hacer referencia a componentes en capas inmediatamente inferiores. Este estilo es de gran utilidad en el desarrollo de sistemas complejos, reduciendo las dependencias de manera que los niveles más bajos no son conscientes de ningún detalle de las superiores.

#### *1.1.2.1.3. Estilo arquitectónico orientado a objetos.*

Los componentes de este estilo son los objetos, en la caracterización clásica de David Garlan y Mary Shaw, los objetos representan una clase de componentes que ellos llaman managers, debido a que son responsables de preservar la integridad de su propia representación (Garlan & Shaw, 1994).

La arquitectura orientada a objetos se basa en el encapsulamiento, herencia y polimorfismo. Las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el eje de las responsabilidades en el diseño de la arquitectura y en la estructura de la aplicación (Reinoso & Kicillof, 2004).

La ventaja más básica concierne a que se puede modificar la implementación de un objeto sin afectar a sus clientes. Además tomar en cuenta que la idea clave de un objeto es ante todo una entidad reutilizable en el entorno de desarrollo.

#### *1.1.2.1.4. Estilo arquitectónico basado en componentes.*

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema, se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. (Pelaez, 2009b).

Otra definición acorde sobre los componentes de software la da Szyperski et al. (1999) donde sintetiza: "Un componente de software es una unidad de composición con interfaces especificadas contractualmente y dependencias del contexto explícitas" (p. 125).

Resumiendo se podría decir que la arquitectura basada en componentes se comunica el uno al otro por medio de las interfaces.

### **1.1.3. Selección del estilo arquitectónico de llamada y retorno.**

Posterior a un análisis de cada uno de los estilos arquitectónicos de llamada y retorno, se realiza la selección del estilo arquitectónico MVC por su principal característica de independizar las 3 capas, (Modelo, Vista, Controlador) lo que permite mayor escalabilidad a los sistemas, es decir facilita realizar actualizaciones y mantenibilidad al software desarrollado bajo este estilo.

Hay que resaltar que en el presente trabajo se sugiere la utilización del estilo arquitectónico MVC, por lo cual se concluye que es el más factible a utilizar al presente proyecto de fin de titulación.

## **1.2. Metodologías para el desarrollo de software**

El desarrollo de software no es una tarea fácil y para emprender un proyecto son necesarias las metodologías las cuales imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Una metodología de desarrollo de software es un proceso de software detallado y completo la cual define con precisión los artefactos, roles y actividades involucrados así como también técnicas y prácticas recomendadas por la metodología. (P. Letelier, 2003)

La clasificación de metodologías para el desarrollo de software no es una tarea fácil debido a que en la actualidad existen gran cantidad de metodologías, sin embargo estas se pueden dividir en 4 grupos: Metodologías Estructuradas, Metodologías Orientadas a Objetos, Metodologías Tradicionales y Metodologías Ágiles. A continuación se detalla cada una de ellas.

### **1.2.1. Metodologías estructuradas.**

Las metodologías estructuradas tienen como objetivo la estructuración y descomposición funcional de problemas en unidades más pequeñas interrelacionadas entre sí. Representan los procesos, flujos y estructuras de datos, de una manera jerárquica y ven el sistema como entradas-proceso-salidas (Hernández, 2014).

Este tipo de metodologías hacen fuerte separación entre los datos y los procesos. Producen una gran cantidad de modelos y documentación y se basan en ciclos de vida en cascada. Como ejemplos de estas metodologías de ámbito gubernamental: MERISE (Francia), MÉTRICA (España), SSADM (P. Letelier, 2003).

### **1.2.2. Metodologías orientadas a objetos.**

Esta clase de metodología va unida a la evolución de los lenguajes de programación orientada a objetos (POO) por sus siglas en inglés, permiten que el software se realice a partir de objetos de comportamiento específico lo que hacen que el producto final sea ampliable y reutilizable.

Booch y Rumbaugh proponen el Método Unificado con la idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más claro, que da lugar al Unified Modeling Language (UML), la notación Orientada a Objetos más popular en la actualidad (Rumbaugh, Jacobson, & Booch, 1999).

Algunas metodologías orientadas a objetos que utilizan la notación UML son: Rational Unified Process (RUP), OPEN, entre otros (P. Letelier, 2003).

### **1.2.3. Metodologías tradicionales.**

Las metodologías no ágiles llamadas también tradicionales o clásicas son aquellas que están dirigidas por una fuerte planificación durante todo el proceso de desarrollo; donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales a excepción de RUP ya que se lo podría configurar para que se adapte a las condiciones del proyecto por lo que podría considerarse ágil (P. Letelier, 2003).

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo de software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto de software. (Hernández, 2014, p. 9)

Ambos autores exponen que este tipo de metodología impone su disciplina de trabajo, además se enfocan en la planificación del proceso de desarrollo de software debido a que los procesos son controlados por políticas y normas.

#### 1.2.4. Metodologías ágiles.

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos con una cercana comunicación), sencillo y adaptable es decir que permite realizar cambios de último momento (P. Letelier, 2003).

Letelier (2003) presenta las principales metodologías ágiles descritas a continuación:

- Extreme Programming.
- Scrum.
- Familia de Metodologías Crystal.
- Feature Driven Development.
- Proceso Unificado Rational, una configuración ágil.
- Dynamic Systems Development Method.
- Adaptive Software Development.
- Open Source Software Development.

Las necesidades del cliente pueden sufrir cambios importantes al momento de contratación de un software por lo que este tipo de metodología ofrece una serie de pautas y principios que junto a sus técnicas harán la entrega del proyecto menos complicada y más satisfactoria tanto para el cliente como para los equipos de entrega permitiendo que los cambios requeridos se efectúen de manera inmediata.

#### 1.3. Selección de la metodología de desarrollo de software

Dentro de los requerimientos planteados para el desarrollo del proyecto de fin de titulación, se han analizado las diferentes metodologías de desarrollo de software, por lo tanto se hará una selección en base a la comparación de metodologías ágiles presentada por Letelier y Penadés (2006) (ver figura 3.), la cual realiza una valoración de 1 como la más baja y 5 como la más alta en base a tres parámetros:

- **Vista del sistema como algo cambiante:** Se realiza una valoración tomando en consideración la metodología que más se adapta a cambios en toda la etapa del proyecto.
- **Colaboración:** Valoración en base a la colaboración entre los miembros del equipo que participan en el proyecto.

- **Características de la propia metodología (CM):** Valoración en base a las características de cada metodología ágil como son: simplicidad, excelencia técnica, resultados, adaptabilidad y prácticas de colaboración.

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
-Resultados	2	5	5	4	4	4	5	5
-Simplicidad	1	4	4	3	5	3	5	5
-Adaptabilidad	2	5	5	3	3	4	4	3
-Excelencia técnica	4	3	3	4	4	4	3	4
-Prácticas de colaboración	2	5	5	4	3	3	4	5
<b>Media CM</b>	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
<b>Media Total</b>	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

Figura 3. Comparación Metodologías Ágiles.

Fuente: (Letelier & Penadés, 2006).

Elaboración: (Letelier & Penadés, 2006).

Se realiza una media de todos los indicadores de cada metodología ágil, y se determina cuales obtienen el resultado más alto. XP y ASD (Desarrollo Adaptable de Software) son las metodologías con mejor puntaje seguido de Scrum. Sin embargo XP tiene la principal característica de ser susceptible a modificaciones durante todo el transcurso del proyecto además de mantener una buena colaboración entre todos los miembros del equipo así lo describe Wells (2009). Para el presente proyecto es necesario una metodología que se adapte a los cambios, además permita tener una buena comunicación entre el programador y el cliente. Por tal razón la metodología ágil programación extrema (XP) será la seleccionada para el presente trabajo de fin de titulación.

### **1.3.1. Metodología XP (Programación Extrema).**

Incluida dentro de las metodologías ágiles, XP está centrada en fomentar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, y propiciando un buen clima de trabajo. Dentro de las bases de esta metodología se encuentra la realimentación continua entre el cliente y el equipo de desarrollo, comunicación clara entre todos los participantes, sencillez en las soluciones implementadas y flexible para enfrentar los cambios que se presenten considerando la rectificación de los requerimientos del cliente ya sea al principio, durante o final del proyecto.

#### **1.3.1.1. Definiciones.**

El nombre fue elegido por Kent Beck, el padre de XP; en el libro Ingeniería de Software escrito por Sommerville (2005) el cual expone: “el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo y la participación del cliente en niveles extremos” (p. 364). Pressman (2010) sintetiza: “la Programación Extrema utiliza un enfoque orientado a objetos como su paradigma preferido” (p. 62). Por otra parte Solís (2003) expone que “la programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica” (p. 1).

XP es una metodología de desarrollo ágil y se basa en la participación del cliente así como también en su simplicidad y la comunicación entre todos los miembros del equipo del proyecto, además faculta a sus desarrolladores responder con seguridad a los cambios en las necesidades de los clientes incluso en la etapa final del transcurso del proyecto.

#### **1.3.1.2. Actividades fundamentales de XP.**

La Programación Extrema (XP) abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño codificación y pruebas (Pressman, 2006). En la figura 4 se observa las ideas y tareas claves asociadas con cada actividad del marco de trabajo.



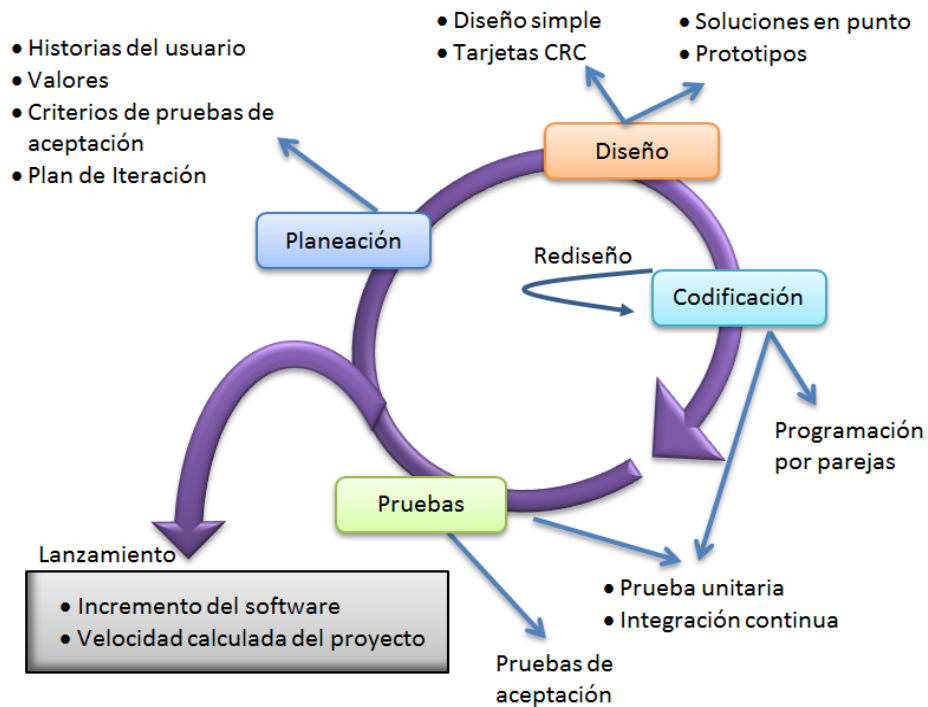


Figura 4. Proceso de la Programación Extrema

Fuente: (Pressman, 2006)

Elaboración: (Pressman, 2006)

### 1.3.1.2.1. Planeación.

La primera actividad planeación comienza creando una serie de historias de usuario, además se pone en práctica los valores, criterios de prueba de aceptación y plan de iteración.

#### ➤ Historias de usuario

Las historias de usuario describen las características y la funcionalidad requerida para el software que se construirá. La historia la escribe el cliente y el mismo le asigna una prioridad basándose en los valores del negocio. Los miembros del equipo de desarrollo evalúan dichas historias y le asignan un costo el cual se mide en semanas de desarrollo, si las historias tienen mucha carga y demoran más de tres semanas de desarrollo se le piden al cliente que la divida en historias menores. Hay que destacar algo importante en esta metodología es que no hay restricción en qué etapa del desarrollo se desee incorporar una historia nueva las cuales pueden escribirse en cualquier momento, estas también se pueden modificar, dividir o eliminar. (Letelier & Penadés, 2006)

Dentro de esta actividad el equipo de desarrollo calcula la velocidad del proyecto, es decir el número de historias de usuario realizadas durante el primer lanzamiento donde se analiza la rapidez del mismo, este a su vez sirve para estimar fechas de entrega y el programa para lanzamientos subsecuentes.

En lo que respecta a la información contenida en las historias de usuario, existen varias plantillas en la actualidad de las cuales se ha seleccionado la expuesta en la tabla 1.

Tabla 1. Formato Historia de Usuario

<b>Historia de Usuario</b>	
<b>Numero:</b>	<b>Nombre Historia de Usuario</b>
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> Alta/Medio/Bajo	<b>Puntos Estimados:</b>
<b>Riesgo en desarrollo:</b> Alta/Medio/Bajo	<b>Programador Responsable:</b>
<b>Descripción:</b>	
<b>Observación:</b>	

Fuente: (Pressman, 2006)

Elaboración: Edison Cadena

### ➤ **Valores**

La Programación Extrema no es un conjunto de reglas a seguir, sino una forma de trabajar en armonía con los valores personales y organizacionales, que tiene su punto de partida en cinco valores fundamentales que son: la comunicación, simplicidad, retroalimentación, coraje y respeto (Wells, 2009).

### ➤ **Criterios de pruebas de aceptación**

Los criterios de las pruebas de aceptación se los realiza en base a las historias de usuario. El cliente especifica los escenarios para poner a prueba si la historia de usuario se ha implementado correctamente y así garantizar la funcionalidad de la obra (Wells, 2009).

➤ **Plan de iteración**

Un plan de iteración es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa (Wells, 2009).

1.3.1.2.2. *Diseño.*

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes. XP recomienda un diseño simple además propone la utilización de tarjetas CRC como se detalla a continuación.

➤ **Diseño simple**

El diseño en XP se basa rigurosamente en mantenerlo simple, esto se prefiere en respecto a una presentación más compleja. Kent Beck se enfoca en que un buen diseño organiza la lógica, de tal forma que un cambio en una parte del sistema no siempre requiere un cambio en otra parte del sistema. (Beck, 1999).

➤ **Tarjetas CRC**

XP ofrece el uso de tarjetas CRC (Colaborador, Responsabilidad, Clase) las cuales cumplen la tarea de identificar y organizar las clases orientadas al objeto, además tiene la responsabilidad de los atributos así como las operaciones de cada clase, finalmente los colaboradores son las clases que se necesitan en otra clase la cual reciba la información y así completar esa responsabilidad. El formato que se utilizará se expondrá en la tabla 2.

Tabla 2. Tarjetas CRC

<b>Nombre de la clase:</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>

Fuente: (Pressman, 2006)

Elaboración: Edison Cadena

Sin embargo esta técnica que ofrece XP se podría complementar con el diagrama de clases, el diagrama entidad relación de la base de datos, casos de uso, diagrama de solución y de arquitectura de software, los cuales se los contemplará como parte de este proyecto.

➤ **Soluciones en punto**

Crear soluciones en punto o de pico se refiere a buscar las respuestas difíciles de problemas técnicos o de diseño. Una solución de pico es un programa muy simple para explorar posibles soluciones, sin embargo construir un pico sólo aborda el problema objeto de examen e ignora todas las demás preocupaciones. XP recomienda la aplicación de este tipo de soluciones en amenazas que retrasen el desarrollo del sistema (Wells, 2009).

➤ **Prototipos**

Los prototipos es una técnica la cual permite obtener una visión preliminar del futuro sistema que se implantará. Un prototipo busca la reacción del cliente permitiendo aceptar sugerencias, y posibles adaptaciones de manera temprana antes de empezar con la codificación del software.

*1.3.1.2.3. Codificación.*

Una de las actividades más importantes y de la que es inevitable prescindir es la codificación. Sin código fuente no hay programa, aunque hay gente que cuenta que existe software en producción del que se perdió el código fuente. Por tanto se necesita codificar y plasmar las ideas a través del código. XP recomienda la programación en parejas el cual se detallará a continuación.

➤ **Programación en parejas**

El concepto clave en esta actividad es la programación en pareja, XP recomienda que dos personas trabajen juntas en una estación de trabajo de computadora para crear el código de la historia de usuario, aplicando el dicho dos cabezas piensan mejor que una, da lugar a la resolución de problemas en tiempo real y el aseguramiento de la calidad en las mismas condiciones así los desarrolladores se mantienen centrados en el problema que se tiene a la mano. El rol que cumple cada uno de los desarrolladores es diferente por ejemplo, una persona puede pensar en los detalles de codificación mientras que la otra valida que se estén aplicando los estándares de codificación.

Cuando los programadores completan su trabajo este se integra con el de otros, en algunos casos este trabajo se encarga diariamente el equipo de integración, en otros los propios programadores son responsables de dicha tarea. Al realizar esta estrategia de integración continua beneficia a evitar problemas de compatibilidad e interfaz los cuales ayudan a descubrir errores desde un principio (Pressman, 2010).

Cabe resaltar que para el presente proyecto no se aplicará la programación en parejas porque solo existe una persona involucrada en el trabajo de fin de titulación.

Respecto a los lenguajes de programación a utilizarse se estará detallando en la sección 1.5.

#### 1.3.1.2.4. Pruebas.

Esta etapa del desarrollo de software cumple un factor importante pues tanto el programador como el cliente realizan pruebas para verificar el funcionamiento de la aplicación. Las pruebas de aceptación es la que recomienda XP, sin embargo se ha visto necesario incluir las pruebas unitarias y las pruebas de campo descritas a continuación:

- **Pruebas de aceptación.-** Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software. (Pressman, 2010)
- **Pruebas unitarias.-** Las pruebas de unidad que se crean deben implementarse en un marco de trabajo que permita ejecutarse de manera fácil y repetida. Well expone: “Corregir pequeños problemas cada cierto número de horas toma menos tiempo que resolver problemas enormes justo antes del plazo final” (Wells, 1999). Este tipo de pruebas busca asegurar que el código funciona de acuerdo con las especificaciones, para ello se divide en módulos y se realiza la pruebas respectivas a cada uno de ellos.
- **Pruebas de campo.-** Correr el sistema en el ambiente real para encontrar errores y validar el producto contra sus especificaciones originales.

## **1.4. Android**

Sistema Operativo con una plataforma abierta basado en el kernel de Linux para dispositivos móviles adquirido por Google y la Open Handset Alliance, su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos. (Polanco, Luis, & Taibo, 2011). Desde su inicio fue pensado especialmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets.

El diccionario GSMSPAIN(Gsms, s.f.) define:

Android es un sistema operativo para dispositivos móviles. Está basado en GNU/Linux e inicialmente fue desarrollado por Google. Esta plataforma permite el desarrollo de aplicaciones por terceros (personas ajenas a Google), para lo cual, los desarrolladores deben de escribir código gestionado en el lenguaje de programación Java y controlar los dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google. (p. 1)

El concepto de Android está generalizado y todos definen que Android es un sistema operativo que desde un inicio fue pensado para dispositivos móviles y que actualmente está disponible para otras plataformas como: televisores, vehículos, tabletas, computadores, etc.

### **1.4.1. Versiones de Android.**

Los nombres que Google ha puesto según el sitio oficial Android (Android, 2014), a sus versiones de Android son muy peculiares, estos han sido enfocados a las diferentes clases de postres empezando con tarta de manzana (Apple Pie) hasta su más reciente versión paleta de dulce (Lollipop). En la tabla 3 se expone todas las versiones existentes de Android

Tabla 3. Versiones del sistema operativo Android.

<b>Nombre</b>	<b>Versión</b>
Apple Pie	v.1.0
Banana Bread	v.1.1
Cupcake	v.1.5
Donout	v.1.6
Eclair	v.2.0/2.1
Froyo	v.2.2
Gingerbread	v.2.3
Honeycomb	v.3.0 /v3.1/ v3.2
Ice Cream Sandwich	v.4.0
Jelly Bean	v.4.1/ v4.2/ v.4.3
KitKat	v.4.4
Lollipop	v.5.0

Fuente: (Android, 2014)

Elaboración: Edison Cadena

Cada versión requiere un hardware mínimo en los diferentes dispositivos para poder ejecutar el sistema operativo limitando algunas funcionalidades y recursos del dispositivo cuando se desarrolla en versiones antiguas, por lo tanto para el presente trabajo de fin de titulación se utilizará desde la versión 4.0 la cual se expondrá en el capítulo II a más detalle.

#### **1.4.2. Desarrollo de aplicaciones Android.**

El desarrollo de aplicaciones Android ha ido evolucionando en los últimos años ante su principal competidor iOS, esto según lo revela la empresa VisionMobile la cual realiza desde hace tiempo su informe Developer Economics y expone un análisis en cuanto a la elección de los desarrolladores entre las plataformas existentes (figura 5).

## DEVELOPER LOYALTY SPLIT BETWEEN IOS & ANDROID

% of developers using each platform as their main platform (n=6,311)

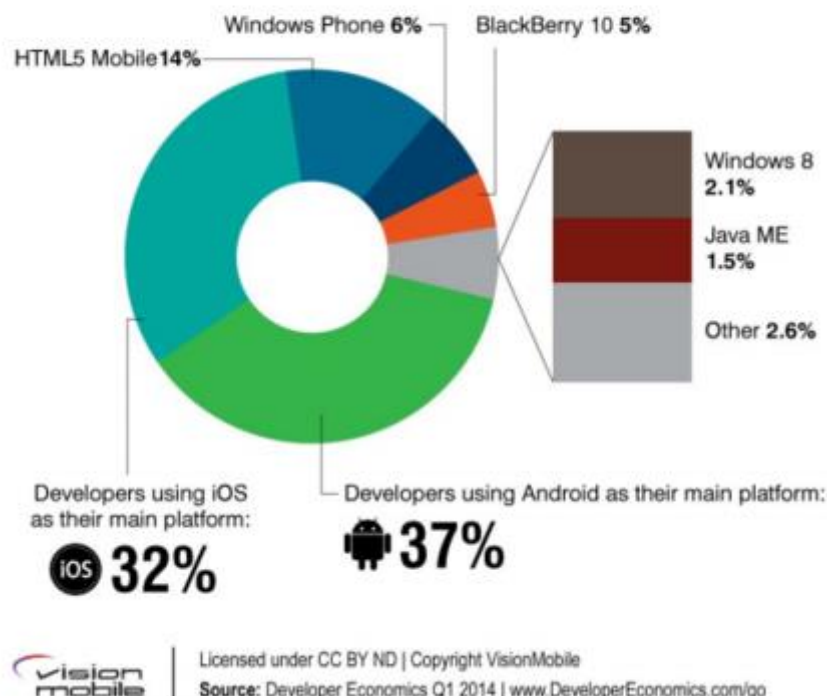


Figura 5. Prioridades de plataformas para los desarrolladores.

Fuente: (Pappas, 2014)

Elaboración: (Pappas, 2014)

La figura anterior resalta que los desarrolladores priorizan Android con un 37%, mientras que iOS se mantiene un 32% de interés en priorizar para la plataforma de Apple. En el caso de HTML5, la prioridad se sitúa en un 14%, mientras que en el tercer trimestre de 2013 era del 17%: la falta de herramientas de desarrollo maduras parece haber sido la causa de ese descenso (Pastor, 2014).

La gran cantidad de usuarios que utilizan el sistema operativo Android da como resultado que los desarrolladores se inclinen más por esta tecnología al observar más mercado. Sin duda alguna Android es una de las mejores opciones a elegir cuando se trata de desarrollar aplicaciones móviles.



## 1.5. Tecnologías existentes de desarrollo de aplicaciones móviles

Respecto a las tecnologías existentes para el desarrollo de aplicaciones Android existe una gran gama de lenguajes de programación, así como del tipo de aplicaciones que son: nativas, web e híbridas descritas a continuación (Fernández, 2013).

- **Aplicaciones nativas.-** Son aquellas que están programadas en el entorno de desarrollo específico para cada sistema operativo, su principal característica es que es funcional sin depender de conexión a internet, además tienen un rendimiento optimizado así como una interfaz mucho más adaptada al sistema operativo al cual el usuario está acostumbrado.
- **Aplicaciones web.-** Aplicaciones responsivas desarrolladas en HTML que se ejecutarán dentro del propio navegador web del dispositivo, lo que permite ser compatible con cualquier sistema operativo que disponga de conexión a internet.
- **Aplicaciones híbridas.-** Es una combinación de las dos anteriores es decir son aplicaciones desarrolladas en parte con el entorno de desarrollo nativo y en parte con el lenguaje web, con su principal ventaja de poder acceder a características del hardware del dispositivo y permite su uso en diferentes plataformas.

Cada tipo de aplicación móvil tiene sus ventajas y desventajas que pueden influir al momento de tomar una decisión para el desarrollo, es por ello que se debe seleccionar con cuidado el tipo de tecnología, el mismo que se elegirá dependiendo de los requerimientos de la aplicación. A continuación se describen las ventajas y desventajas de cada una de las tecnologías según el punto de vista del sitio web (LanceTalent, 2015).

### 1.5.1. Ventajas.

#### Aplicaciones nativas

- Mayor rendimiento.
- Acceso completo al hardware del teléfono (cámara, sensores, GPS, etc.).
- Funcionalidad sin necesidad de acceso a internet.
- Mantenimiento más sencillo.
- Mejor experiencia de usuario.

### **Aplicaciones web**

- Multiplataforma.
- Desarrollo más económico.
- El mismo código fuente para varias plataformas.
- Última versión siempre disponible para el usuario.

### **Aplicaciones híbridas**

- Es posible su distribución en las tiendas de iOS Android.
- El mismo código fuente para varias plataformas.
- Acceso a parte del hardware del teléfono.

#### **1.5.2. Desventajas.**

### **Aplicaciones nativas**

- Desarrollo costoso.
- El código del cliente no es reutilizable para las diferentes plataformas.
- Actualización manualmente por el usuario.

### **Aplicaciones web**

- Requiere de conexión a internet.
- Acceso limitado al hardware del teléfono.
- Tiempo de respuesta menor al de una aplicación nativa.

### **Aplicaciones híbridas**

- Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.
- En su mayoría requiere conexión a internet.

#### **1.5.3. Selección del tipo de aplicación móvil.**

Para el presente trabajo de fin de titulación es necesario explotar el hardware del dispositivo específicamente el sensor GPS, además se requiere una aplicación con rendimiento optimizado y fácil de usar para el usuario final. Es por tal motivo que se ha planteado desarrollar una aplicación nativa para dispositivos móviles con sistema operativo Android, la cual tiene todas las características necesarias para ajustarse a los requerimientos planteados.

## **1.6. Lenguajes de programación y herramientas de desarrollo de software**

Los lenguajes de programación y herramientas de desarrollo de software son elementos fundamentales del proyecto, por tal razón se expondrán a detalle cada uno de ellos, mismos que serán utilizados en el presente trabajo de fin de titulación.

Dentro de cada uno de los lenguajes de programación existen varias herramientas de programación que se complementan entre sí, como es el caso de los diferentes IDE de desarrollo, frameworks, librerías, APIs, servicios web entre otros.

### **1.6.1. Lenguajes de programación desarrollo web.**

En la actualidad existe un catálogo muy extenso en cuanto al desarrollo de aplicaciones web, cada vez nuevas tecnologías o la evolución de las mismas se incorporan al mundo de la web. En el caso particular se hablará de PHP que es uno de los lenguajes de programación más populares de la Web. De acuerdo a Netcraf (2013) se ejecutaba en más de 245 millones de servidores Web hasta enero 2013.

#### **1.6.1.1. PHP.**

Es un lenguaje de programación para crear sitios web dinámicos e interactivos, PHP (Hypertext Preprocessor) se ejecuta en un servidor Web y sirven páginas web a visitantes que lo solicitan. Una de las características que resalta es que se puede embeber código PHP en páginas Web HTML, facilitando la creación de contenido rápidamente (Doyle, 2010).

Por otra parte Gutiérrez y Bravo (2005) define que “PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor ” (p. 65). PHP es un lenguaje de script del lado del servidor lo que quiere decir que los scripts se ejecutaran en un servidor Web.

PHP será utilizado como lenguaje del lado del servidor para el presente trabajo de fin de titulación, al ser multiplataforma es decir que pueda ser ejecutado en sistemas operativos Windows, Linux, Mac Os X, entre otros además de ser de código abierto por lo que no habrá necesidad de adquirir licencias de pago para su utilización.

### **1.6.1.2. JavaScript y jQuery.**

JavaScript es el lenguaje interpretado más utilizado, principalmente en la construcción de páginas Web, con una sintaxis muy semejante a Java y a C. Al contrario de Java no se trata de un lenguaje orientado a objetos propiamente dicho, sino que éste está basado en prototipos, ya que las nuevas clases se generan clonando las clases base y extendiendo su funcionalidad (Flanagan, 2006).

Crockford (2008) por su parte define a JavaScript como un lenguaje de desarrollo importante porque es el lenguaje del navegador Web. Está asociado con los navegadores más populares y de hecho este es lenguaje de programación del lado del cliente más popular del mundo.

jQuery es una librería de JavaScript así lo sintetiza Sawyer (2012) “es el arma secreta de los diseñadores Web en la batalla de la programación con JavaScript. Con jQuery, puede realizar tareas en una única línea de código que código de JavaScript corresponderían cientos de líneas de código y horas de prueba en los navegadores” (p. 88).

Luego de analizar los diferentes conceptos de ambos lenguajes, se determina que por ser los más soportados por todos los navegadores, y además no se necesita adquirir licencias para su funcionamiento en vista de que son de código abierto, se ha planteado utilizarlos en el presente trabajo de fin de titulación.

### **1.6.2. Lenguajes de programación desarrollo Android.**

El sitio oficial de Android revela que desde un inicio y hasta la actualidad Java se ha consolidado como lenguaje de desarrollo oficial para aplicaciones nativas Android, además Google lanzó como entorno de desarrollo Android Studio. (Android, 2015).

Existen herramientas de desarrollo de terceros como: Android APIMiner, AndroWish, App Inventor for Android, Basic4android, HyperNext Android Creator, Kivy entre otras, cada una de ellas con un diferente lenguaje de programación, lo cual abre las puertas a desarrolladores a elegir el entorno de desarrollo y lenguaje que más se adapte a sus necesidades.

Sin embargo hay que recalcar, que el lenguaje de programación Java sigue siendo líder en el desarrollo de aplicaciones nativas Android, además la cantidad de información y documentación que existe en esta tecnología impulsa a más desarrolladores a unirse, esto lo revela Spectrum IEEE la cual realiza un estudio de las tendencias que existen en cuanto a lenguajes de programación la cual se ilustra en la figura 6 (Diakopoulos, Nick & Cass, 2015).

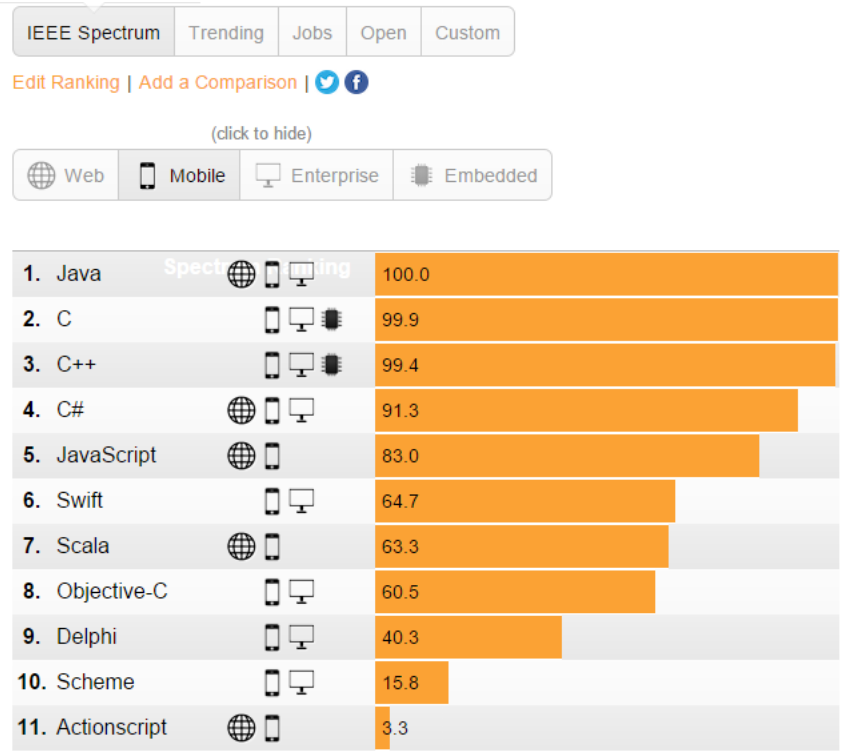


Figura 6. Top de lenguajes de programación

Fuente: (Diakopoulos, Nick & Cass, 2015)

Elaboración: (Diakopoulos, Nick & Cass, 2015)

**CAPÍTULO II**  
**SITUACIÓN INICIAL, ALTERNATIVA DE SOLUCIÓN DE SOFTWARE**  
**Y PROPUESTA DE SOLUCIÓN**

## **2. Introducción**

El presente capítulo abarca los temas relacionados a la situación actual que tiene la UTPL en lo que corresponde al recorrido de buses y así proponer una alternativa de software. Además se expone las herramientas a utilizarse tanto para el desarrollo de la aplicación web y aplicación móvil. Finalmente se plantea los alcances y resultados esperados.

### **2.1. Situación actual**

La Universidad Técnica Particular de Loja dentro de sus servicios en el campus universitario cuenta con recorrido privado de buses para docentes, estudiantes y público en general. Este servicio cuenta con varias flotas vehiculares de propiedad de la Universidad las cuales cubren ciertos sectores de la ciudad de Loja de forma y horarios estratégicos. Los usuarios para hacer uso de este servicio tendrán que desplazarse a las paradas establecidas por la universidad, debido a que los buses no recogen pasajeros en lugares no autorizados.

Actualmente algunas personas que visitan la UTPL desconocen las paradas de buses donde recogen y dejan pasajeros, dando como resultado que prefieran utilizar otros medios de transporte externos como taxis o buses urbanos que cubren las rutas a varios metros del campus Universitario.

### **2.2. Alternativa de solución de software**

Como iniciativa se ha planteado desarrollar dos aplicaciones web y móvil con la finalidad de informar al usuario las diferentes rutas y paradas establecidas por la Universidad donde puedan desembarcar o embarcar hacia un lugar determinado de la ciudad de Loja.

Aprovechando las nuevas tecnologías como los teléfonos inteligentes y su gran acogida en el mercado, se ha visto necesario desarrollar una aplicación para dispositivos móviles Android la cual servirá de apoyo a los visitantes de manera interactiva y así puedan obtener información del servicio de recorrido de buses que brinda la Universidad.

Sin embargo es necesario que la aplicación móvil tenga información actual, por lo que será necesario un usuario encargado de gestionar la información del recorrido de buses de la UTPL. Es por tal razón que se ha planteado desarrollar un sistema web con las herramientas necesarias para gestionar las diferentes paradas y rutas planteadas por la Universidad.

### 2.3. Herramientas

Las herramientas necesarias para el desarrollo de la aplicación web se muestran en la tabla 4. Cabe destacar que las mismas han sido seleccionadas en base a su tipo de licencia que deben de ser de código abierto, es decir que sean libres y no genere un costo por su uso además de ser las más utilizadas en el desarrollo de aplicaciones web.

Tabla 4. Herramientas Desarrollo Web

CARACTERÍSTICAS	HERRAMIENTAS	VERSIÓN
Php + MySql	Xampp	3.2.1
Base de datos	MYSQL	5.6.21
Interfaz de administración BD	PhpMyAdmin	4.2.11
Depurador de errores	Google Chrome Console	43.0.2357.65
IDE de Desarrollo	Sublime Text	3
Framework estilos	Bootstrap	3.3.4
Framework webservice	Slim	2
Diagramas UML	Enterprise Architect	8.0
Librería de seguridad captcha	Phpcaptcha	3.6.2

Elaboración: Edison Cadena

En la tabla 5. se muestran las herramientas y características mínimas que serán necesarias para la ejecución de la aplicación web. Estas herramientas son indispensables para poner en funcionamiento la aplicación web.

Tabla 5. Herramientas de ejecución para Aplicación Web

CARACTERÍSTICAS	HERRAMIENTAS	VERSIÓN
Servidor Web + PHP	Apache	2.4.10
Base de datos	MySql	5.6.21
Navegador	Google Chrome Mozilla Firefox	40 o superior 30 o superior

Elaboración: Edison Cadena



Las diferentes herramientas que se utilizarán para el desarrollo de la aplicación nativa Android se muestran en la tabla 6, las mismas que han sido seleccionadas por su gran utilización en el mercado de desarrollo de aplicaciones nativas Android.

Tabla 6. Herramientas Desarrollo Android

CARACTERÍSTICAS	HERRAMIENTAS	VERSIÓN
Lenguaje de Programación	Java	
IDE Desarrollo	Android Studio	1.1.0
Herramientas de desarrollo	JDK	8.0
Depuración	Teléfono Alcatel Idol 3	
Diagramas UML	Enterprise Architect	8.0
Depurador de errores	Logcat- Android Studio	

Elaboración: Edison Cadena

En la tabla 7. se muestran las herramientas mínimas que serán necesarias para la correcta ejecución de la aplicación móvil.

Tabla 7. Herramientas para ejecución de aplicación móvil

CARACTERÍSTICAS	HERRAMIENTAS	VERSIÓN
Sistema Operativo	Android	4.0 o superior
GPS	Teléfono Compatible	
Internet	Wifi o red de datos	

Elaboración: Edison Cadena

Todas las herramientas seleccionadas tienen licencia de código abierto, razón principal por la cual se ha hecho la elección correspondiente de cada una de las herramientas necesarias tanto para su desarrollo como su ejecución.

## 2.4. Iteraciones

Se ha planteado un plan de iteraciones ordenado por prioridad de desarrollo en el proyecto los cuales se ordenan en base a requerimientos principales que sirvan de base para las iteraciones siguientes. Cada iteración tiene una duración aproximada de 3 semanas para su desarrollo, si el tiempo excede se deberá dividir la iteración en 2 partes. De esta forma se logra estimar el tiempo de finalización del proyecto en base a cada historia de usuario culminada. El orden de iteraciones según la prioridad se muestra en la tabla 8.

Tabla 8. Orden de iteraciones

ITERACION	HISTORIA DE USUARIO
1	Inicio de Sesión
2	Administración de Rutas
3	Administración de paradas y recorridos
4	Administración de usuarios
5	Alertar paradas
6	Buscar Paradas

Elaboración: Edison Cadena

## 2.5. Definición del alcance

El alcance del presente proyecto constará en la creación de un sistema de administración web y una aplicación para dispositivos móviles con sistema operativo Android las cuales permitan consultar paradas cercanas desde el lugar donde se encuentran así mismo les permita alertar la parada próxima a embarcar o desembarcar dentro de una ruta de recorrido de buses UTPL.

El presente trabajo de fin de titulación no pretende dejar en producción el producto final sino más bien generar una iniciativa en donde las autoridades de la Universidad deberán analizar su implementación. En lo que respecta a las funcionalidades de la aplicación web y móvil se describirán en la sección 2.6.1 y 2.6.2.

### 2.5.1. Sistema de administración web.

Sistema web en el cual el usuario administrador podrá gestionar todas las paradas que integran una ruta el cual va a ser consumido por los usuarios finales desde los dispositivos móviles. Los diferentes componentes que integraran la aplicación web son: Ingreso, Principal, Gestión de paradas, Gestión de rutas, Gestión de trayectoria, Graficar Trayectoria, Gestión de Administradores, Gestión de Horarios y Modificar Perfil.

#### ➤ Ingreso

Interfaz de ingreso en el cual el usuario accederá al sistema a través de un usuario y contraseña. La primera vez de ingreso el usuario por defecto será “administrador” y la contraseña de acceso “administrador”.

### ➤ **Principal**

El componente principal generará una vista de bienvenida al sistema y la visualización en general de cuál es el número de paradas y rutas creadas en el sistema.

### ➤ **Gestión paradas**

Componente para la creación, edición y eliminación de paradas que existen en el recorrido de buses de la UTPL.

Dentro de los campos que contemplan este componente están:

- **Nombre:** El nombre de la parada a registrar.
- **Dirección:** Calle principal de ubicación de la parada a registrar.
- **Intersección:** Calle secundaria de ubicación de la parada a registrar.
- **Latitud:** Latitud georeferenciada.
- **Longitud:** Longitud georeferenciada.

Búsqueda por cualquier tipo de campo para localizar una parada registrada.

### ➤ **Gestión rutas**

En el presente componente se gestionará las diferentes rutas que existen para el recorrido de buses de la UTPL con los siguientes campos descritos a continuación:

- **Nombre:** El nombre de la ruta a registrar.
- **Sentido Ruta:** Trayectoria del bus de la UTPL ya sea que se encuentre bajando, subiendo o ambos.

Búsqueda por cualquier tipo de campo para localizar una ruta registrada.

Creación de la trayectoria del recorrido de bus en base a puntos georeferenciados en el mapa, así como también la creación de los diferentes horarios.

### ➤ **Gestión trayectoria (paradas-rutas)**

En el presente componente se creará la trayectoria conformada por la agrupación de paradas a una ruta en particular la cual guiará al usuario final para que visualice el recorrido de los buses de la UTPL, los campos que se incluirán en este componente se describen a continuación:

- **Nombre Ruta.-** Selección de la ruta que pertenece la trayectoria.
- **Nombre Parada.-** Selección de la parada a la cual pertenece la trayectoria.
- **Sentido.-** El sentido de la ruta previamente creado.
- **Horario.-** El horario por el cual el bus se estaciona a recoger o dejar pasajeros.

Búsqueda por cualquier tipo de campo para localizar una trayectoria registrada.

➤ **Graficar Trayectoria**

Componente para la creación y eliminación de puntos georeferenciados los cuales al ser asignados a una ruta determinada se graficará la ruta por la cual el bus realiza su recorrido.

Dentro de los campos que contemplan este componente están:

- **Nombre Ruta:** El nombre de la parada a registrar.
- **Latitud:** Latitud Georeferenciada del punto de trayectoria.
- **Longitud:** Longitud Georeferenciada del punto de trayectoria.

Búsqueda por cualquier tipo de campo para localizar una trayectoria registrada.

➤ **Gestión administradores**

Componente de creación, edición, modificación y eliminación de los usuarios con el rol de administrador que manejaran el sistema web. Los campos que se incluirán en este módulo serán los siguientes:

- **Nombres:** Nombres completos del usuario (administrador).
- **Apellidos:** Apellidos completos del usuario (administrador).
- **Usuario:** El usuario o alias con el cual iniciará sesión el administrador.
- **Contraseña:** Contraseña del usuario con la cual iniciará sesión.
- **Correo:** Correo electrónico del usuario (administrador).

Búsqueda por cualquier tipo de campo para localizar un usuario previamente registrado.

➤ **Gestión de horarios**

Componente de creación, edición, modificación y eliminación de los horarios disponibles para asignar a una parada determinada. El campo que se incluirá en el siguiente módulo será el siguiente:

- **Horario:** Horario disponible en el cual el bus recoge pasajeros.

Búsqueda por cualquier tipo de campo para localizar un horario previamente registrado.

➤ **Modificar Contraseña**

Panel de administración donde el usuario podrá modificar su contraseña por motivos de seguridad.

➤ **Cerrar sesión**

Cierra la sesión iniciada previamente del usuario.

**2.5.2. Aplicación móvil Android.**

Aplicación móvil en la cual el usuario será capaz de buscar paradas cercanas desde el lugar que se encuentre, además podrá ser alertado de las paradas próximas a desembarcar dentro de una ruta determinada. Otra funcionalidad de la aplicación es la de informar los horarios aproximados por la cual el bus recoge pasajeros en cada parada.

Cabe mencionar que toda información que el usuario final maneje será proporcionada por la aplicación web, razón por la cual los administradores serán responsables de la veracidad de la información que ingresen.

➤ **Interfaz principal.**

El usuario tendrá un pequeño menú para seleccionar la acción tanto como para alertar o buscar paradas cercanas.

**a) Alertar paradas cercanas**

Con la principal característica de visualizar un mapa embebido (servicio dado por Google Maps), en la cual el usuario podrá seleccionar la ruta y visualizar las paradas asignadas a la misma.

i. ***Iniciar Recorrido***

El botón que estará incluido en la interfaz principal el cual al ser presionado empezará alertar las paradas próximas de viaje previo la selección de la ruta, también forzara la activación del GPS del dispositivo móvil ya que dicho sensor es obligatorio para el correcto funcionamiento de la aplicación.

ii. ***Detener Recorrido***

Al igual que el botón de iniciar recorrido este botón también estará incluido en la interfaz principal el cual al ser presionado desactivará las alertas a las paradas cercanas, y desactivara el consumo del GPS.

iii. ***Configuraciones***

Interfaz de configuraciones donde el usuario personalizará algunas funciones de la aplicación como el tipo de alertas ya sean por voz o mediante mensajes por pantalla, así como también modificar la distancia en la cual se activarán las alertas al acercarse a una parada, dando como resultado que la aplicación sea flexible y escalable al usuario final.

## **b) Buscar paradas cercanas**

Interfaz en la cual se mostrará un mapa con todas las paradas creadas por el administrador, con el fin de ubicar la posición geográfica de la misma y mostrar las 3 paradas más cercanas desde la ubicación actual del usuario final.

### **i. Localizar parada**

Se mostrará las 3 paradas más cercanas desde la ubicación actual del usuario haciendo diferencia de las demás y se graficará una línea de referencia para señalar las paradas cercanas.

### **ii. Mostrar distancia**

Se visualizará el tiempo aproximado que tardaría en llegar caminando así como también la distancia a las paradas cercanas.

### **iii. Detalles de paradas**

Cada vez que se selecciona una parada graficada en el mapa se desplegará información como nombre de la parada, ruta y hora.

## **2.5.3. Resultados esperados.**

Con el desarrollo del presente proyecto se espera plantear una solución de software que facilite la información necesaria a quienes visiten por primera vez o desconocen el servicio de recorrido de buses que presta la Universidad. La solución de software se dividirá en dos aplicaciones la primera web que permita administrar las paradas-rutas del recorrido de buses de la Universidad y la segunda la aplicación móvil que permita alertar y buscar las paradas cercanas desde el lugar que se encuentre el usuario.

**CAPÍTULO III**  
**DESARROLLO DE LA SOLUCIÓN DE SOFTWARE**

### 3. Desarrollo de la solución de software

En el presente capítulo se abarcará todo lo relacionado al proceso de desarrollo de software utilizando la metodología de desarrollo Programación Extrema la cual se descompone en 4 actividades fundamentales que son: Planeación, Diseño, Codificación y Pruebas.

#### 3.1. Planeación.

La primera actividad de la metodología XP como ya se ha mencionado anteriormente, es la planeación la cual utiliza historias de usuario para describir a grandes rasgos las características principales que debe tener el sistema desde el punto de vista del cliente. En el caso nuestro se ha planteado los siguientes requerimientos expuestos en las tablas 9, conforme a la propuesta inicial planteada.

A continuación se describe las historias de usuario necesarias para el presente proyecto:

##### ➤ Historias de usuario de la aplicación web

Tabla 9. Historia de Usuario 1

Historia de Usuario	
<b>Numero:</b> 1	<b>Nombre Historia de Usuario</b> Inicio de Sesión
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1-5
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2
<b>Riesgo en desarrollo:</b> Bajo	<b>Programador Responsable:</b> Edison Cadena
<b>Descripción:</b> <ul style="list-style-type: none"><li>➤ Se deberá crear una interfaz de inicio de sesión para el ingreso de usuarios.</li><li>➤ Se deberá contar con un verificador de seguridad que restrinja el acceso a robots que intenten ingresar por fuerza bruta.</li></ul>	
<b>Observación:</b>	

Elaboración: Edison Cadena



Tabla 10. Historia de Usuario 2

<b>Historia de Usuario</b>	
<b>Numero:</b> 2	<b>Nombre Historia de Usuario:</b> Administración de Rutas
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 2-5
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3
<b>Riesgo en desarrollo:</b> Baja	<b>Programador Responsable:</b> Edison Cadena
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Componente de administración de rutas donde los administradores podrán crear, actualizar y eliminar registros con la información correspondiente al nombre y sentido de la ruta.</li> <li>➤ Componente de administración que servirá para crear la trayectoria que corresponde a una ruta determinada. En este componente se ingresaran varios puntos georeferenciados que serán parte de la gráfica de trayectoria por la cual circula el bus, mientras más puntos se cree en esta sección más preciso será la gráfica.</li> <li>➤ Interfaz de administración en la cual se crearán los horarios aproximados por los cuales los buses embarcan pasajeros dentro de una parada predeterminada.</li> </ul>	
<b>Observación:</b>	

Elaboración: Edison Cadena

Tabla 11. Historia de Usuario 3

<b>Historia de Usuario</b>	
<b>Numero:</b> 3	<b>Nombre Historia de Usuario</b> Administración de paradas y recorridos.
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Usuario final	<b>Iteración Asignada:</b> 3-5
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en desarrollo:</b> Bajo	<b>Programador Responsable:</b> Edison Cadena
<b>Descripción:</b> <ul style="list-style-type: none"> <li>➤ Componente de administración de paradas donde los administradores podrán crear y eliminar registros con la información correspondiente al nombre, dirección, calle de intersección y puntos georeferenciados (latitud, longitud).</li> <li>➤ Gestión de recorrido de bus que permitirá agregar las paradas que corresponde a una ruta, así como también el horario por la cual el bus embarca pasajeros. En este componente se enlaza toda la información de rutas, paradas y horarios la cual servirá para mostrar al usuario final desde la aplicación móvil.</li> </ul>	
<b>Observación:</b>	

Elaboración: Edison Cadena

Tabla 12. Historia de Usuario 4

<b>Historia de Usuario</b>	
<b>Numero:</b> 4	<b>Nombre Historia de Usuario</b> Administración de usuarios
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Usuario final	<b>Iteración Asignada:</b> 4-5
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en desarrollo:</b> Bajo	<b>Programador Responsable:</b> Edison Cadena
<b>Descripción:</b> <ul style="list-style-type: none"> <li>➤ Componente de administración para crear, actualizar y eliminar usuarios administradores del sistema.</li> <li>➤ Actualización de contraseña y opción para cerrar sesión por parte del usuario que se encuentre iniciado sesión.</li> </ul>	
<b>Observación:</b>	

Elaboración: Edison Cadena

➤ **Historias de usuario de la aplicación móvil**

Tabla 13. Historia de Usuario 5

<b>Historia de Usuario</b>	
<b>Numero:</b> 5	<b>Nombre Historia de Usuario</b> Alertar paradas
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Usuario final	<b>Iteración Asignada:</b> 4-5
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en desarrollo:</b> Bajo	<b>Programador Responsable:</b> Edison Cadena
<b>Descripción:</b> <ul style="list-style-type: none"> <li>➤ Componente que alertará una parada próxima a llegar en función de la posición actual y la posición de la parada. Se comparará la distancia que existe entre cada una de ellas y se emitirá un mensaje de voz o de texto cuando se encuentre cerca. Para empezar un recorrido de alerta de paradas se deberá seleccionar la ruta por la cual circulará el bus.</li> </ul>	

<ul style="list-style-type: none"> <li>➤ En este componente se podrá configurar la forma de emisión de mensaje de alerta (voz o texto por pantalla) y la distancia que se emitirá el mensaje.</li> </ul>
<b>Observación:</b>

Elaboración: Edison Cadena

Tabla 14. Historia de Usuario 6

<b>Historia de Usuario</b>	
<b>Numero:</b> 6	<b>Nombre Historia de Usuario</b> Buscar paradas
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b> Usuario final	<b>Iteración Asignada:</b> 4-5
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en desarrollo:</b> Bajo	<b>Programador Responsable:</b> Edison Cadena
<b>Descripción:</b> <ul style="list-style-type: none"> <li>➤ Componente que mostrará todas las paradas de buses de la UTPL en donde recogen y dejan pasajeros.</li> <li>➤ Componente que buscará las tres paradas más cercanas al lugar donde se encuentre el usuario y se resaltarán para identificarlas.</li> <li>➤ El componente también permitirá mostrar los horarios que existen por cada parada, además de guiar al usuario hacia una parada previamente seleccionada.</li> </ul>	
<b>Observación:</b>	

Elaboración: Edison Cadena

### 3.2. Diseño

En esta etapa se realizará el diseño de la aplicación utilizando las tarjetas CRC que recomienda XP, además de algunos artefactos adicionales como son: diagrama de clases (Anexo B), diagrama entidad relación de la base de datos (figura 7), casos de uso (Anexo C), diagrama de solución (figura 8) y de arquitectura de software (figuras 9 y 10).

### 3.2.1. Tarjetas CRC.

Una vez hecho el análisis de las historias de usuario detalladas en la planeación (sección 3.1) se ha desarrollado las diferentes tarjetas CRC cuyo objetivo de la misma es hacer mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y determinar que el diseño sea lo más simple posible. Las tarjetas CRC que se utilizarán para el presente proyecto se expondrán en el **Anexo D**.

### 3.2.2. Modelo Entidad-Relación.

El modelo Entidad-Relación (ER) es una técnica muy utilizada para el modelado de datos que permite representar las entidades y las relaciones que existen entre ellas dando como resultado la creación del diseño de la base de datos. En la figura 7 se ilustra el modelo entidad relación del presente proyecto.

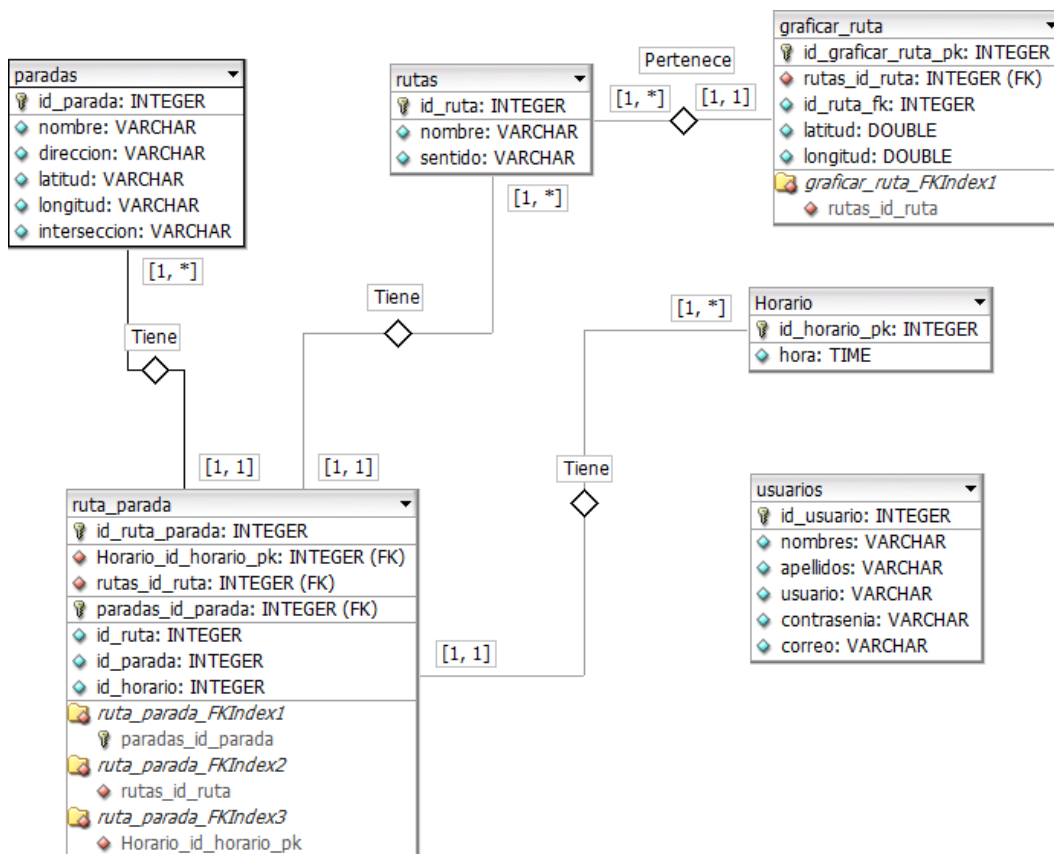


Figura 7. Modelo Entidad-Relación.

Elaboración: Edison Cadena

### **3.2.3. Diccionario de datos.**

El diccionario de datos contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre de campo, descripción, tipo de dato, tipo de restricción y procedencia. El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades, por tal razón se ha contemplado desarrollar el diccionario de datos, el mismo que estará disponible en el **Anexo E**.

### **3.2.4. Diagrama de solución.**

En la figura 8 se mostrará todos los componentes que intervienen, su funcionalidad y la relación que existe con cada uno de ellos a nivel general. El diagrama se compone por dos partes la aplicación web y la aplicación móvil.

En la aplicación web intervienen 3 capas que son el Modelo, Vista y Controlador los mismos que se comunican por protocolo HTTP, además los lenguajes de programación que se utilizarán son Javascript del lado del cliente y PHP del lado del servidor los cuales se crearan en el editor de texto Sublime Text. Las librerías y frameworks utilizados son: Bootstrap, Datatables, y jQuery a más de los servicios de Google Maps que serán consumidos por la aplicación web.

La aplicación web creará un webservice con la información extraída desde la base de datos MySQL, este servicio web se lo creará con la ayuda de la librería Slim. La aplicación móvil consumirá este servicio para recoger toda la información generada en la aplicación web. La aplicación móvil para dispositivos con sistema operativo Android será desarrollada en el entorno de desarrollo Android Studio con el lenguaje de programación Java, la misma consumirá el sensor GPS del teléfono para ubicar con mayor precisión la localización del dispositivo.

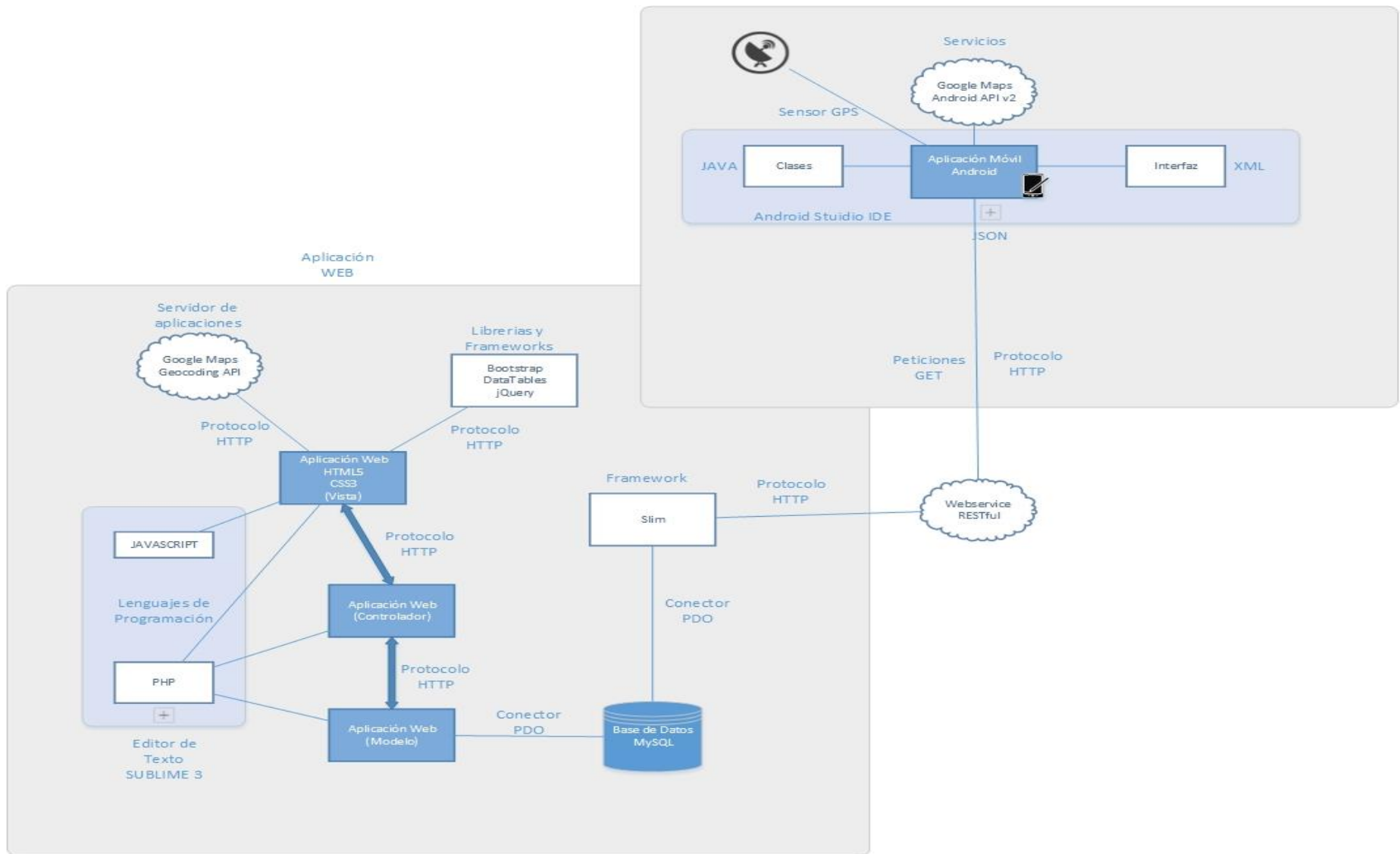


Figura 8. Diagrama de solución.

Elaboración: Edison Cadena

### 3.2.5. Diagrama arquitectura de software.

Este diagrama brinda más detalles sobre la vinculación de los componentes y las capas de software. El presente proyecto consta de dos partes aplicación web y aplicación móvil los cuales se dividirán en dos diagramas que son: arquitectura de software aplicación web y arquitectura de software aplicación móvil.

#### ➤ Arquitectura de software aplicación web

Este tipo de diagrama (figura 9) muestra la relación que tiene el usuario con la aplicación el cual interactúa a través de interfaces, seguidamente la aplicación se comunica con el motor de base de datos realizando peticiones de consulta, inserción, actualización y eliminación de datos.

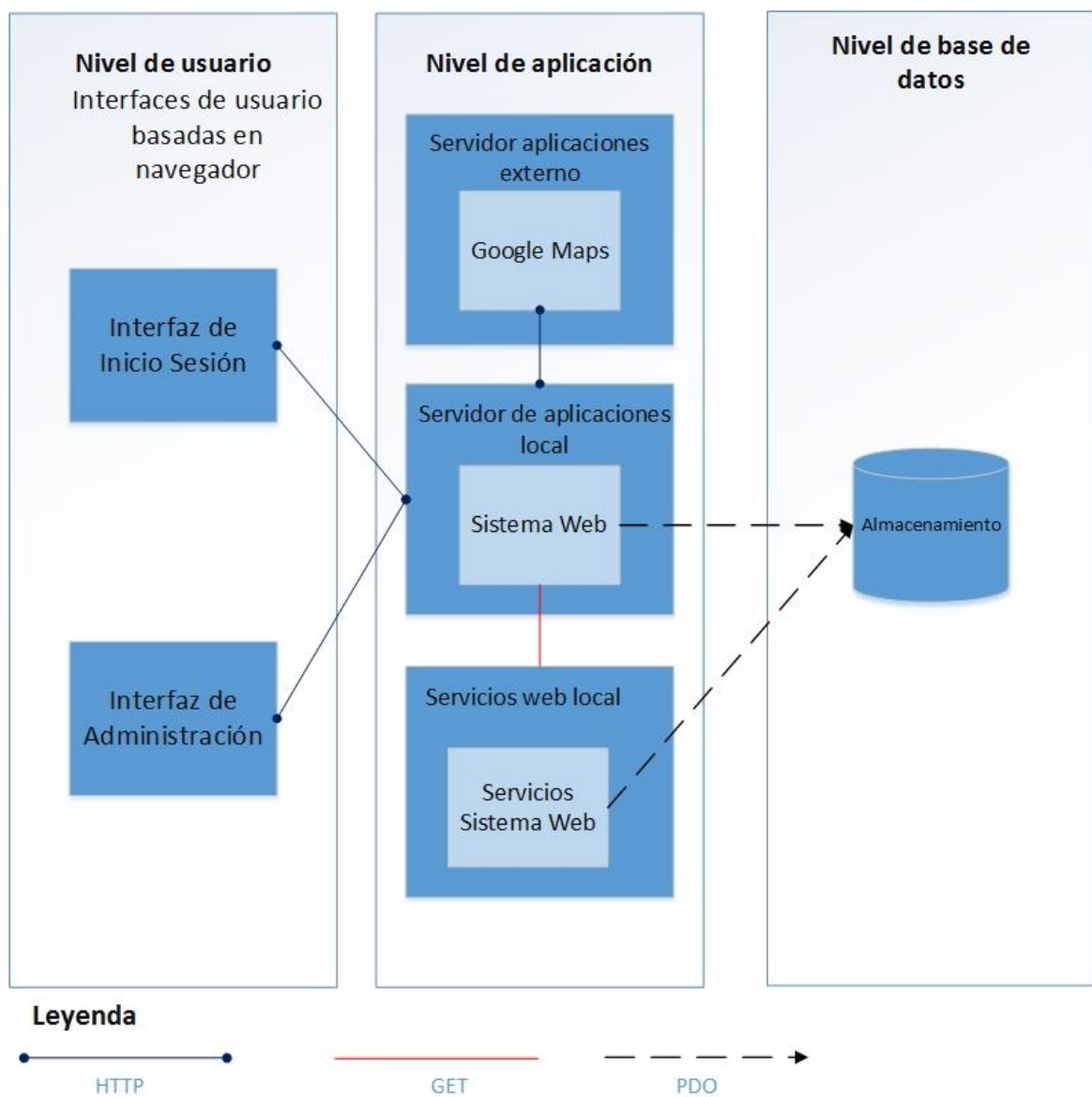


Figura 9. Arquitectura de software aplicación web.

Elaboración: Edison Cadena



### ➤ Arquitectura de software aplicación móvil

El diagrama expuesto en la figura 10 ilustra la relación e iteración que existe entre el usuario, aplicación, datos y hardware. El usuario se comunica con la aplicación a través de interfaces, la aplicación por su parte se comunica a la información a través de servicios web y accede al hardware del teléfono específicamente al GPS.

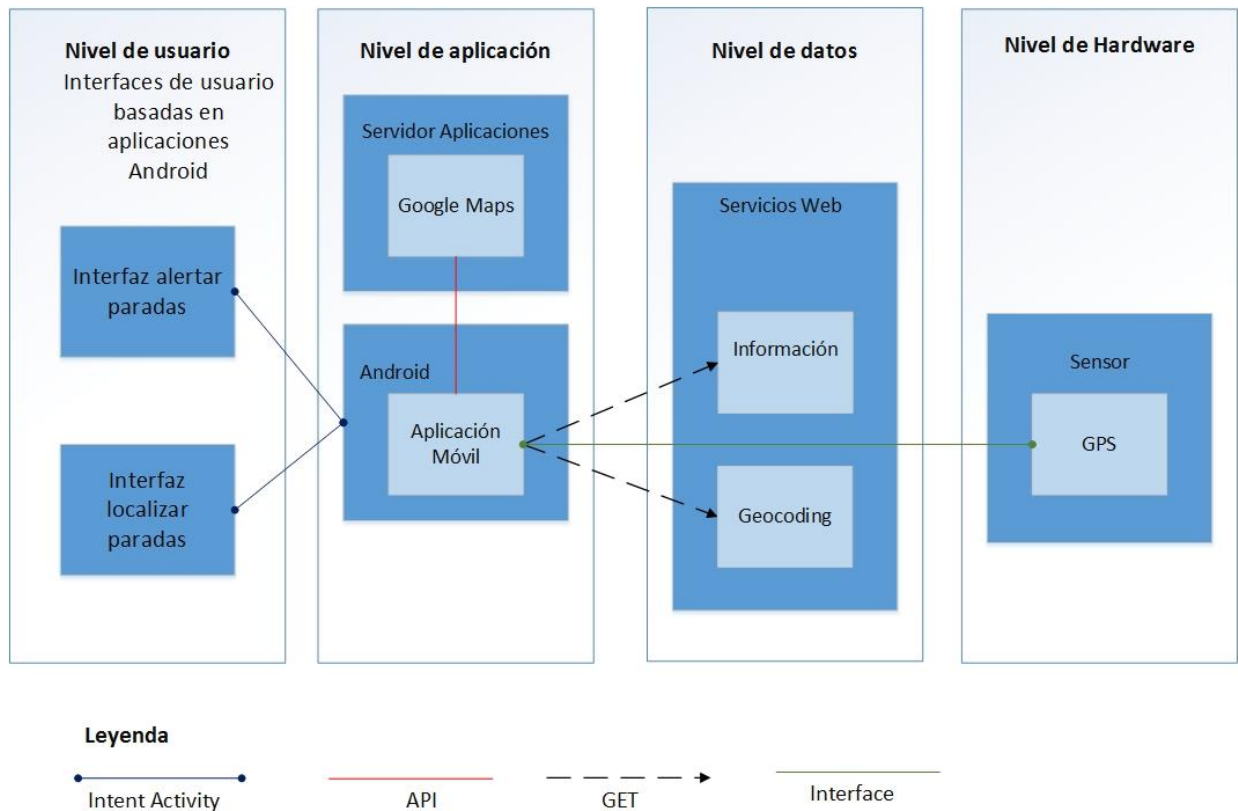


Figura 10. Arquitectura de software aplicación móvil.

Elaboración: Edison Cadena

### 3.2.6. Diagrama de clases.

Los diagramas de clases son el pilar básico del modelado con UML y su principal uso es mostrar lo que el sistema puede hacer antes de ser construido. La metodología XP no contempla al diagrama de clases dentro de su estructura, sin embargo se lo ha contemplado debido a que muestran las clases del sistema y sus interrelaciones permitiendo así tener una visión más clara en cuanto al diseño de la aplicación. El presente diagrama estará expuesto en el Anexo B.

### **3.2.7. Casos de uso.**

Los casos de uso ofrecen una perspectiva más amplia para comprender la iteración de los usuarios dentro de un sistema. Es necesario que se tenga en consideración las acciones que realizan los usuarios tanto el administrador como el usuario final. Por tal razón se ha contemplado implementar los casos de uso tanto de la aplicación web como de la aplicación móvil los cuales estarán descritos en el Anexo C.

### **3.3. Codificación**

El presente proyecto contempla el desarrollo de dos soluciones la primera WEB y la segunda para dispositivos móviles con Sistema Operativo Android, ambos utilizando la metodología de desarrollo ágil XP como ya se ha descrito en los capítulos anteriores.

#### **3.3.1. Solución WEB.**

La solución WEB que se propone será usada por los administradores los cuales gestionarán todas las paradas, rutas, usuarios entre otros componentes del sistema.

Para el desarrollo de la presente solución se ha utilizado el patrón arquitectónico Modelo Vista Controlador además de la implementación de frameworks y varias librerías y servicios necesarios para el correcto funcionamiento del sistema.

##### **➤ Patrón Arquitectónico Modelo Vista Controlador**

En el primer capítulo se mencionó acerca del patrón arquitectónico y sus ventajas a la hora de desarrollar un sistema. A continuación se describe cada una de las capas donde se ha trabajado.

**Capa 1 (Modelo).**- El modelo es la capa donde se trabajará con los datos por lo tanto contiene funciones de insertar, modificar, eliminar y actualizar información sobre la base de datos. Se ha planteado un modelo autónomo con funciones para trabajar con dicha información utilizando PHP como lenguaje de programación y PDO como driver para establecer la conexión a MySQL. En la figura 11 se muestra una parte de código de la capa modelo.

```

1  <?php
2  class horarios_model{
3      private $db;
4
5      public function __construct(){
6          $this->db=Conectar::conexion();
7      }
8
9      public function nuevo_horario($hora)
10     {
11         $consulta = $this->db->prepare("CALL insertHorario('$hora')");
12         $consulta->execute();
13         $resultado="Nuevo horario creada satisfactoriamente";
14         return $resultado;
15     }
16

```

Figura 11. Código fuente de la capa modelo del módulo horario.

Elaboración: Edison Cadena

**Capa 2 (Vista).**- La vista como se explicó en el primer capítulo, la cual se encarga de visualizar la información hacia el usuario final. Normalmente el código que se visualiza en esta capa no son clases si no archivos HTML. En la figura 12 se muestra un extracto de la capa vista del módulo horario.

```

<div class="container informacionContenedor ">
  <div class="row clearfix informacionTexto">
    <div class="col-md-1 column">
      
    </div>
    <div class="col-md-4 column">
      <center><h2>Gestión de horarios</h2></center>
    </div>
    <div class="col-md-6 column"><br>
      <a data-placement="top" data-toggle="tooltip"><button class="bt
    </div>

```

Figura 12. Extracto de código de la capa vista del módulo horario.

Elaboración: Edison Cadena

**Capa 3 (Controlador).**- El controlador actúa como ente intermediario entre el modelo, vista y viceversa es decir todas las peticiones que se hagan deben ser atendidas por el controlador para que se comunique con el modelo y este sea quien devuelva la información al controlador para entregársela a la vista. En la figura 13 se muestra una parte del código de la capa controlador.

```

if(!empty($_POST['opcion'])){

    switch ($_POST['opcion']){
        case 'nuevo':
            $horariosMsj=$horarios->nuevo_horario($_POST['hora']);
            $horariosSel=$horarios->get_horarios();
            break;
        case 'actualizar':
            $horariosMsj=$horarios->editar_horario($_POST['txtId'],$
            $horariosSel=$horarios->get_horarios();
            break;
        case 'eliminar':
            $horariosMsj=$horarios->eliminar_horario($_POST['txtId']
            $horariosSel=$horarios->get_horarios();
            break;
        default:
    }
}

```

Figura 13. Extracto de código de la capa controlador del módulo horario.

Elaboración: Edison Cadena

Los métodos desarrollados en esta parte se muestran en el anexo A.

### ➤ Frameworks

Un framework a nivel general es una estructura tecnológica definida la cual contiene una serie de artefactos o módulos que sirven de base para el desarrollo de software. Dentro de la programación WEB existen una amplia gama de frameworks sin embargo se han seleccionado por ser tecnologías de código abierto y ajustarse según a los requerimientos del sistema. A continuación se detallará cada uno de ellos:

- **Slim.-** Framework que permite la generación de un servicio web RESTful en formato JSON a través del lenguaje de programación PHP.
- **Bootstrap.-** Framework de estilos desarrollado por Twitter que permite crear interfaces responsivas, es decir que se adapta a cualquier tamaño del dispositivo que se visualice.

### ➤ Librerías

Las librerías son archivos con extensión js que incluyen una serie de funciones y métodos para simplificar tareas complejas de las cuales se han utilizado las siguientes:

- **jQuery.-** Dentro del diseño de la aplicación se ha utilizado el framework Bootstrap el cual necesita esta librería para su funcionamiento.

- **DataTables.-** Librería que permite convertir una tabla simple en una tabla con componentes adicionales como la paginación la cual brinda al usuario una experiencia de usuario más amigable.

➤ **Servidor de aplicaciones**

Google Maps como servidor de aplicaciones permite embeber dentro de la aplicación web un mapa y a través del mismo utilizar diferentes API como Geocoding que permiten convertir de coordenadas GPS a nombres de calles y viceversa.

➤ **Lenguajes de programación**

Para el desarrollo de la aplicación web se ha utilizado como lenguaje de programación PHP lo que permite el acceso a la información de la base de datos.

JAVASCRIPT por su parte trabaja del lado del cliente y se ha utilizado en el presente proyecto para realizar eventos sobre los mapas y objetos de la vista respectivamente.

➤ **Estándar de programación**

Para el desarrollo de la solución web se ha utilizado el estándar de programación PSR-0, PSR-1 a continuación en la figura 14 se muestra un extracto del código fuente en donde se ha empleado el estándar antes mencionado.

```
public function nueva_ruta_parada($idRuta, $idParada)
{
    $consulta = $this->db->prepare("CALL inserRutaParada('$idRuta', '$idParada')");
    $consulta->execute();
    $resultado="Nueva ruta-parada creada satisfactoriamente";
    return $resultado;
}
```

Figura 14. Estándar de programación

Elaboración: Edison Cadena

### 3.3.2. Aplicación Móvil.

Para el desarrollo de la solución móvil se ha empleado el IDE (Ambiente de Desarrollo Integrado) Android Studio el cual se muestra un extracto de la interfaz en la figura 15. JAVA se ha utilizado como lenguaje de programación base además del consumo de servicios web y sensores del dispositivo como GPS para el funcionamiento total de la aplicación.

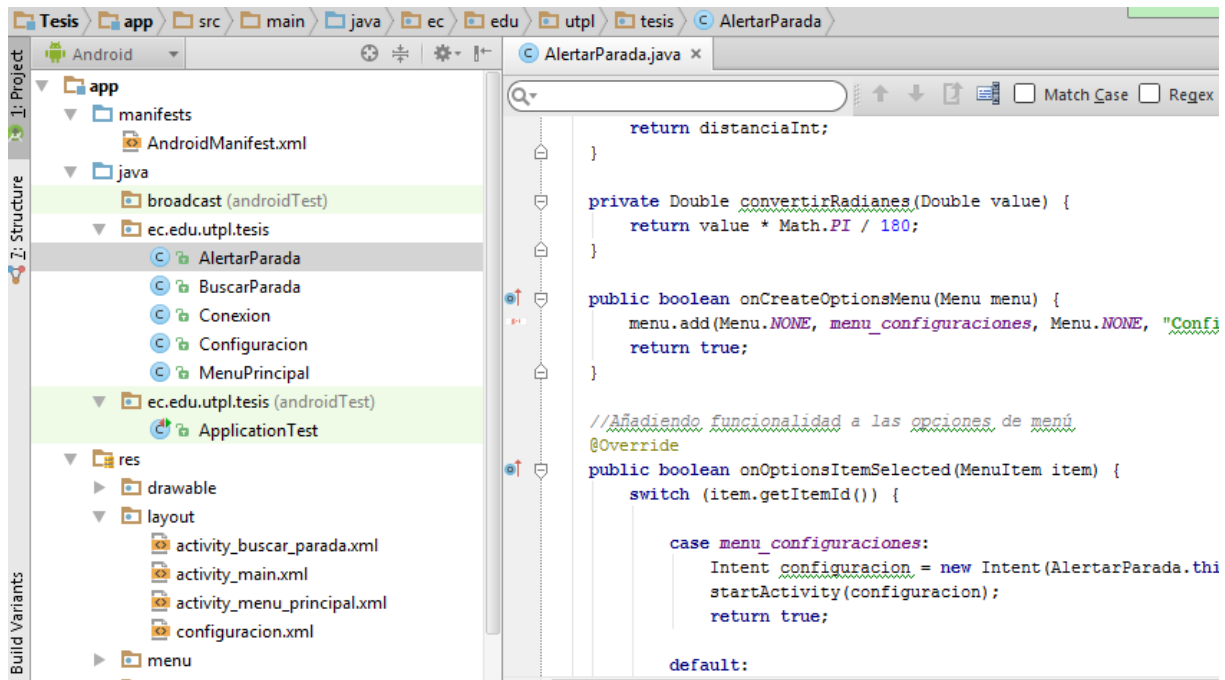


Figura 15. Interfaz Android Studio.

Elaboración: Edison Cadena

### ➤ Lenguaje de programación

El lenguaje de programación utilizado para el desarrollo de la aplicación móvil es JAVA con el paradigma de programación POO (Programación Orientada a Objetos). Cabe resaltar que no se ha utilizado el patrón arquitectónico MVC para esta sección y solo se ha trabajado con la teoría de programación estructurada utilizando subrutinas y secuencias tanto condicionales como repetitivas.

### ➤ Diseño y modelado de interfaces

Las interfaces cumplen un rol importante en cuanto a la funcionalidad de la aplicación ya que de ellas depende que el sistema sea de fácil manejo para el usuario final, mientras más sencillo su diseño mucho mejor.

Para el diseño de las interfaces se ha utilizado el framework Bootstrap el cual permite a la aplicación adaptarse a los diferentes tamaños de pantalla es decir que sea responsiva, además de su diseño amigable y elegante a la vez. En lo que respecta a la presentación de la información en tablas se ha utilizado el framework Datables el cual estructura la información, realiza la paginación y contiene un buscador dando como resultado una experiencia de usuario agradable.

En el Anexo F se encuentra el manual de usuario de la aplicación móvil y web donde se explica cada una de las interfaces creadas.

#### ➤ **Servicios Web**

En lo que concierne a la generación de servicios web, se ha utilizado el framework Slim el cual crea servicios RESTful utilizando el lenguaje de programación PHP. El objetivo del servicio web será para la comunicación entre la aplicación móvil y los datos generados en la aplicación web utilizando el protocolo HTTP. La aplicación móvil consumirá los servicios a través del formato JSON y así poder obtener información como la posición, nombre entre otras características de cada parada así como también de las rutas.

#### ➤ **Sensores**

Los teléfonos inteligentes actualmente incorporan una serie de sensores que permiten responder a varios eventos que genera el usuario como el acelerómetro que permite conocer en qué posición se encuentra el teléfono al girar la pantalla. Entre ellos se encuentra el GPS que tiene la principal característica de comunicarse con los satélites y ubicar la posición exacta del dispositivo, también proporciona otras variables como altitud, velocidad, entre otras.

El sensor GPS se ha utilizado para reconocer la ubicación del dispositivo y mostrarla en tiempo real en un mapa proporcionado por Google Maps, en donde se hará el proceso de encontrar la parada del recorrido de bus UTPL próxima a llegar y alertar al usuario ya sea por mensaje de voz o mensaje por pantalla.

#### ➤ **Servidor de aplicaciones**

Al igual que en la solución web se ha utilizado Google Maps como servidor de aplicaciones de mapas. Google dispone de una Consola de Desarrolladores la cual dispone diferentes APIs que son base fundamental en soluciones informáticas que utilicen servicios de Google. En el presente proyecto se ha empleado Google Maps Android API V2, además se ha generado la clave API respectiva la cual permite embeber un mapa en el dispositivo móvil con el fin de graficar las diferentes paradas y rutas.

### 3.4. Pruebas e implementación

Finalmente la última actividad de la metodología Programación Extrema corresponde a las pruebas, las cuales validen el correcto funcionamiento del sistema así como también los pasos necesarios que se debe seguir para la puesta en producción de ambas aplicaciones.

#### 3.4.1. Pruebas.

El presente proyecto parte de una necesidad motivo por el cual los requerimientos han sido planteados en base a las necesidades que existen de los usuarios que utilizan el servicio de recorrido de buses de la UTPL, por lo que las pruebas que se han realizado en esta etapa han sido validadas por el tutor de proyecto explotando las diferentes funcionalidades de la misma.

Dentro de lo expuesto en el primer capítulo acerca de la metodología XP la cual manifiesta que el cliente es parte fundamental en toda la etapa del desarrollo de software, sin embargo para el presente proyecto al ser una propuesta de solución se ha coordinado con el tutor del presente proyecto para validar las pruebas respectivas.

##### 3.4.1.1. Pruebas de aceptación.

XP recomienda la utilización de las pruebas de aceptación para garantizar que lo desarrollado cumple con lo requerido del cliente. Las pruebas de aceptación se realizan al finalizar cada iteración con el objetivo de validar la funcionalidad del sistema de esta manera se logra determinar en forma acertada los errores y cambios a realizar. Es necesario tener en cuenta que para el presente proyecto al ser una propuesta no existe un cliente final para realizar las validaciones sin embargo se ha realizado las pruebas de aceptación con el tutor del presente trabajo de fin de titulación. A continuación se detalla en la tabla 15 el formato a utilizar.

Tabla 15. Formato prueba de aceptación.

Prueba de aceptación	
Número:	Número Historia de Usuario
Descripción:	
Condiciones de ejecución	
Pasos de ejecución	
Resultados esperados	
Evaluación de la prueba	

Fuente: (Pressman, 2006)

Elaboración: Edison Cadena



## Detalle

- ✓ **Número:** Número de la prueba de aceptación
- ✓ **Número Historia de Usuario:** Número de la historia de usuario que se hace referencia.
- ✓ **Descripción.-** Resumen de la historia de usuario a aprobarse.
- ✓ **Condiciones de ejecución.-** Condiciones previas requeridas.
- ✓ **Pasos de ejecución.-** Pasos que el usuario deberá hacer para ejecutar la acción aprobar.
- ✓ **Resultados esperados.-** Respuesta obtenida del sistema posterior a la ejecución de la acción.
- ✓ **Evaluación de la prueba.-** Nivel de satisfacción con respecto al resultado obtenido del sistema.

### **3.4.1.2. Pruebas unitarias.**

Es necesario realizar las pruebas de unidad o unitarias las cuales consisten en probar cada uno de los métodos y verificar que todo funcione bien. Para ejecutar estas pruebas se ha utilizado la herramienta Android Instrumentation Test, el cual está incorporado en el entorno de desarrollo Android Studio. En el caso de la aplicación web se ha utilizado la librería SimpleTest, la misma que permite realizar pruebas de unidad para el lenguaje de programación PHP.

### **3.4.1.3. Pruebas de calidad de software.**

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. Para realizar estas pruebas se ha utilizado el software SonarQube que examina el código fuente y emite resultados de los cuales se puede validar la calidad de software. Los resultados de las pruebas realizadas estarán presentes en el Anexo I.

### **3.4.1.4. Pruebas de campo.**

Este tipo de pruebas las realiza los interesados del proyecto en conjunto con el desarrollador de software, las cuales permiten verificar en ambiente real que la aplicación funcione correctamente, así como también que los requerimientos planteados inicialmente estén plasmados en su totalidad. Las validaciones de este tipo de prueba, estará disponible en un video que forma parte del presente proyecto. El detalle de las pruebas de aceptación y unitarias estarán expuestas en el Anexo G.

### **3.4.2. Implementación de Aplicaciones.**

Una vez realizadas las respectivas pruebas se ha hecho la implementación de ambas aplicaciones (web y móvil) las cuales se detalla a continuación:

#### **➤ Implementación Aplicación Web**

La implementación de la aplicación se ha realizado siguiendo los pasos detallados en el anexo H. Seguidamente se describe a manera general los pasos respectivos:

- ✓ Instalar XAMPP.
- ✓ Copiar los archivos de la aplicación en el directorio /htdocs.
- ✓ Crear la base de datos en MySQL con el nombre “tesis” e importar el archivo .sql

#### **➤ Instalación Aplicación Móvil**

Para la instalación respectiva en los diferentes dispositivos móviles con sistema operativo android igual o superior a la versión Android 4.0 (ICE CREAM SANDWICH).

Dentro del alcance del proyecto no se ha contemplado la subida de la aplicación a la tienda Play Store de Google, motivo por el cual se tendrá que activar la opción orígenes desconocidos y posteriormente instalar la aplicación con extensión apk, en el anexo H se hace una explicación más detallada acerca de la instalación.

## CONCLUSIONES

- ✓ La utilización de la metodología XP es una buena alternativa para el desarrollo de software, ya que a medida que el proyecto avanzó permitió adaptarse a los cambios de manera ágil y oportuna.
- ✓ La utilización de la herramienta SonarQube sirvió para obtener métricas que puedan ayudar a mejorar la calidad del código.
- ✓ Las pruebas de aceptación permiten validar lo que el cliente pidió ya que el mismo experimenta la funcionalidad de la aplicación y permite observar las reacciones del sistema al ingreso de los datos exitosos o erróneos.
- ✓ La utilización del patrón arquitectónico MVC (Modelo Vista Controlador) ha permitido un desarrollo más organizado, permitiendo realizar mejoras o cambios sin comprometer todo el código fuente si no solamente a la capa correspondiente, esto porque el patrón permite dividir el código fuente en 3 partes: el modelo que gestiona los datos, la vista que muestra todo el diseño de la aplicación y que a su vez realiza peticiones al controlador quien es ente intermedio para que exista comunicación entre el modelo y la vista.
- ✓ La programación nativa para Android es mucho mejor en comparación con las aplicaciones híbridas o web, debido que permite explotar todos los recursos del dispositivo (cámara, gps, acelerómetro, etc.), además las aplicaciones no necesariamente dependen de conexión a internet para su funcionamiento y se obtiene una mejor experiencia por parte del usuario.
- ✓ La utilización del IDE Android Studio en comparación con Eclipse es mucho mejor empezando porque Android Studio está basado en IntelliJ IDEA uno de los IDE para java de primer nivel, además Android Studio incorpora Gradle el cual facilita la reutilización de código y gestiona las dependencias de una forma cómoda y potente.
- ✓ El framework Slim permite crear servicios web con rapidez, gracias a su esencia, Slim es un despachador que recibe una solicitud HTTP, invoca una rutina de devolución de llamada apropiada, y devuelve una respuesta HTTP en formato JSON. De esta forma fue pieza fundamental para la comunicación entre la aplicación web y la aplicación móvil.
- ✓ Con las pruebas realizadas se ha podido determinar que el proyecto ha pasado todas las pruebas que determinan la calidad de software.

## RECOMENDACIONES

- ✓ Se recomienda la utilización del patrón arquitectónico MVC en proyectos similares por la gran ventaja de dividir el proceso de desarrollo en varias capas, aprovechando esto se podría trabajar con varios desarrolladores que trabajen en cada capa sin tener dependencia total de los demás.
- ✓ Se deben fijar una serie de reglas generales en la comunicación con el cliente debido a que por el grado de informalidad que la metodología presenta, pueden surgir diferencias que pongan en peligro la culminación exitosa del proyecto.
- ✓ Se recomienda el uso de metodologías ágiles para el desarrollo de software, estas permiten enfocarse principalmente en la programación sin descuidar la parte de documentación.
- ✓ Se recomienda el uso de software libre para el desarrollo e implementación de aplicaciones ya que permite la reducción de costo del proyecto.
- ✓ La actualización de la información por parte de los usuarios administradores es de suma importancia para el desempeño de la aplicación móvil.
- ✓ Se recomienda adquirir un servidor físico o en la nube (internet) con ip pública ya que es indispensable para la comunicación entre la aplicación web y móvil.
- ✓ Los dispositivos móviles deberán tener el servicio de internet móvil ya que la información necesaria para el funcionamiento de la aplicación se encuentra en el servidor con ip pública (internet).
- ✓ Se recomienda la programación nativa para dispositivos móviles con el IDE Android Studio, sobre todo en aplicaciones donde sea necesario el consumo de los sensores del dispositivo.
- ✓ Como base de datos se recomienda MySQL que es muy usada en la actualidad así también como PostgreSQL, ambas son muy robustas y de código abierto.
- ✓ Se recomienda instalar equipos GPS en los diferentes buses los cuales permitan enviar información hacia un servidor, con el objetivo de mostrar en tiempo real del dispositivo móvil el lugar aproximado que se encuentra cada bus de la UTPL.
- ✓ La creación de una nueva funcionalidad en la aplicación móvil que sea capaz de activar una alarma que alerte a los usuarios del próximo bus a llegar a una parada específica.

## BIBLIOGRAFÍA

- Android. (2014). History. Retrieved from <http://www.android.com/history/>
- Android. (2015). Developer Android. Retrieved from <http://developer.android.com/index.html>
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change. XP Series*.  
<http://doi.org/10.1136/adc.2005.076794>
- Burbeck, S. (1992). Applications Programming in Smalltalk-80 ( TM ): How to use Model-View-Controller ( MVC ). *Smalltalk-80 v2, 80(Mvc)*, 1–11. Retrieved from [http://www.dgp.toronto.edu/~dwigdor/teaching/csc2524/2012\\_F/papers/mvc.pdf](http://www.dgp.toronto.edu/~dwigdor/teaching/csc2524/2012_F/papers/mvc.pdf)
- Clements, P. C. (1996). A Survey of Architecture Description Languages. In *8th International Workshop on Software Specification and Design (IWSSD 96)* (pp. 16–25).  
<http://doi.org/10.1109/IWSSD.1996.501143>
- Crockford, D. (2008). *JavaScript: The Good Parts. Online* (Vol. 44).  
<http://doi.org/10.1241/johokanri.44.584>
- Diakopoulos, Nick & Cass, S. (2015). The Top Programming Languages 2015. Retrieved from <http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2015#index/2015/0/1/0/0/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/>
- Doyle, M. (2010). Beginning PHP 5.3. In *Beginning PHP 5.3* (Vol. 21, p. 38). Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/22996945>
- Ecured. (n.d.). Retrieved from [http://www.ecured.cu/index.php/Estilos\\_arquitectónicos](http://www.ecured.cu/index.php/Estilos_arquitectónicos)
- Enterprise Solution Patterns: (n.d.). Model-View-Controller. Retrieved from <https://msdn.microsoft.com/en-us/library/ff649643.aspx>
- Fernández, A. (2013). Aplicaciones web vs. aplicaciones nativas vs. aplicaciones híbridas. Retrieved from <http://blogthinkbig.com/aplicaciones-web-nativas-hibridas/>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures. Building*. Retrieved from <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- Flanagan, D. (2006). *JavaScript: The Definitive Guide. Chemistry*.  
<http://doi.org/10.1007/s13398-014-0173-7.2>
- Garlan, D. (2000). Software architecture: a roadmap. ... of the *Conference on the Future of Software ...*, 91–101. <http://doi.org/10.1145/336512.336537>
- Garlan, D., & Shaw, M. (1994). An Introduction to Software Architecture. In *Advances in software engineering and knowledge engineering* (pp. 1–40). Retrieved from <http://portal.acm.org/citation.cfm?id=865128>
- Gsms. (n.d.). Android. Retrieved from <http://www.gsmspain.com/glosario/?palabra=ANDROID>

- Gutiérrez, J. (2006). ¿ Qué es un framework web?, 1–4. Retrieved from [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)
- Gutiérrez, J&Bravo, G. (2005). *PHP 5 a través de ejemplos*.
- Hernández, J. I. (2014). *Análisis y Desarrollo Web*. Retrieved from <https://books.google.com.ec/books?id=nYDVBQAAQBAJ>
- LanceTalent. (2015). Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes. Retrieved from <http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>
- Letelier, P., & Penadés, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica Administrativa*, 5, 17. <http://doi.org/1666-1680>
- Maier, M. W., Emery, D., & Hilliard, R. (2001). Software architecture: Introducing IEEE standard 1471. *Computer*, 34, 107–109. <http://doi.org/10.1109/2.917550>
- Netcraft. (2013). PHP just grows & grows. Retrieved from <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>
- P. Letelier. (2003). Proceso de desarrollo de software, 1–14. Retrieved from 1 <http://www.ceis.cl/Gestacion/Gestacion.htm> (5.3.2003) 2 <http://standishgroup.com/> (5.3.2003)
- Pappas, A. (2014). Developer Economics: Ecosystem wars drawing to a close. Retrieved from <http://www.visionmobile.com/blog/2014/02/developer-economics-q1-2014/>
- Pastor, J. (2014). Desarrollo de aplicaciones móviles (I): así está el mercado. Retrieved from <http://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado>
- Pelaez, J. (2009a). Arquitectura basada en capas. Retrieved from <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>
- Pelaez, J. (2009b). Arquitectura basada en Componentes. Retrieved from <http://geeks.ms/blogs/jkpelaez/archive/2009/04/18/arquitectura-basada-en-componentes.aspx>
- Polanco, K. M., Luis, J., & Taibo, B. (2011). “Android” El Sistema Operativo De Google Para Dispositivos Móviles. *Negottun*, 7, 79–96.
- Pressman, R. S. (2006). *Ingeniería de Software. Un enfoque práctico*. Computer.
- Pressman, R. S. (2010). *Ingeniería del software, 7ma Ed.*
- Reinoso, C., & Kicillof, N. (2004). Estilos y Patrones en la Estrategia de Arquitectura de Microsoft, 1–73.
- Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *El Lenguaje Unificado de Modelado. Elements*. Retrieved from <http://portal.acm.org/citation.cfm?id=993859&dl=>
- Sawyer, D. (2012). *Programación JavaScript y jQuery*.

Solís, M. C. (2003). Una explicación de la programación extrema (XP).

Sommerville, I. (2005). *Ingeniería del software*. *danielr.obolog.es*. Retrieved from [http://danielr.obolog.es/ingenieria-software-355416\nhttp://fondoeditorial.uneg.edu.ve/citeg/numeros/c02/c02\\_art10.pdf](http://danielr.obolog.es/ingenieria-software-355416\nhttp://fondoeditorial.uneg.edu.ve/citeg/numeros/c02/c02_art10.pdf)

Szyperski, C., Szyperski, C., Bosch, J., Bosch, J., Weck, W., & Weck, W. (1999). Component-Oriented Programming. *Object-Oriented Technology ECOOP'99 Workshop Reader*.

Wells, D. (1999). Unit Tests. Retrieved from <http://www.extremeprogramming.org/rules/unittests.html>

Wells, D. (2009). *Extreme Programming: A gentle introduction*. Retrieved from <http://www.extremeprogramming.org/>

## **ANEXOS**



## ANEXO A.

### Métodos de la aplicación Web por capas (Modelo, Vista, Controlador)

#### Modelo.

Nombre de clase: paradas\_model

Métodos	Parámetros	Retorna	Descripción
get_paradas		Objeto	Listado de todas las paradas registradas en la base de datos.
nueva_parada	nombre direccion latitud longitud intersección	String	Crea un nuevo registro en la base de datos.
eliminar_parada	idParada	String	Eliminar un registro en función del id.

Nombre de clase: rutas\_model

Métodos	Parámetros	Retorna	Descripción
get_rutas		Objeto	Listado de todas las rutas registradas en la base de datos.
nueva_ruta	nombre sentido	String	Modifica una ruta almacenada en la base de datos en función de la clave primaria (id).
editar_ruta	idRuta nombre sentido	String	Crea un nuevo registro en la base de datos.
eliminar_ruta	idRuta	String	Eliminar un registro en función del id.

Nombre de clase: ruta\_parada\_model

Métodos	Parámetros	Retorna	Descripción
get_ruta_parada		Objeto	Listado de todas las rutas-paradas registradas en la base de datos.
editar_ruta_parada	idRutaParada, idRuta, idParada	String	Modifica un registro en la base de datos en función de la clave primaria (id).
nueva_ruta_parada	idRuta idParada	String	Crea un nuevo registro en la base de datos.
eliminar_ruta_parada	idRuta	String	Eliminar un registro en función del id.

Nombre de clase: graf\_tray\_model

Métodos	Parámetros	Retorna	Descripción
get_trayectoria		Objeto	Listado de todas los puntos que conforman una trayectoriar.
nueva_punto_trayectoria	idRuta latitud longitud	String	Inserta un punto que se asignara a una trayectoria.
eliminar_trayectoria	idGrafRuta	String	Elimina punto de trayectoria.

Nombre de clase: usuarios\_model

Métodos	Parámetros	Retorna	Descripción
get_usuarios		Objeto	Listado de todos los usuarios registrados en la base de datos.
nuevo_usuario	Nombres, Apellidos, Usuario, Contraseña, Correo	String	Modifica un registro en la base de datos en función de la clave primaria (id).
editar_usuario	idUsuario, Nombres, Apellidos, Usuario, Correo	String	Crea un nuevo registro en la base de datos.
eliminar_usuario	idUsuario	String	Eliminar un registro en función del id.

Nombre de clase: horarios\_model

Métodos	Parámetros	Retorna	Descripción
get_horarios		Objeto	Listado de todos los horarios registrados en la base de datos.
nuevo_horario	Hora	String	Modifica un registro en la base de datos en función de la clave primaria (id).
editar_horario	idhorario, hora	String	Crea un nuevo registro en la base de datos.
eliminar_horario	idhorario	String	Eliminar un registro en función del id.

Nombre de clase: login\_model

Métodos	Parámetros	Retorna	Descripción
iniciar_sesion	usuario, contraseña	String	Función que valida el usuario y contraseña para acceder al sistema.

Nombre de clase: modificar\_psw\_model

Métodos	Parámetros	Retorna	Descripción
modificarContrasenia	idUsuario, contraseniaAnt, contraseniaNue, contraseniaRepNue	String	Función que modifica la contraseña de un usuario.

### Vista.

Vista	Descripción	Controlador y método relacionado
vw_paradas	Pantalla para la administración de las paradas	ct_paradas/ get_paradas ct_paradas/nueva_parada ct_paradas/eliminar_parada
vw_rutas	Pantalla para la administración de las rutas	ct_rutas/ get_rutas ct_rutas/ nueva_ruta ct_rutas/ editar_ruta ct_rutas eliminar_ruta
vw_ruta_parada	Pantalla para la administración de la trayectoria al unir las rutas y paradas respectivamente	ct_ruta_parada/ get_ruta_parada ct_ruta_parada/ nueva_ruta_parada ct_ruta_parada/ editar_ruta_parada ct_ruta_parada/ eliminar_ruta_parada

vw_usuario	Pantalla para la administración de los usuarios administradores del sistema web.	ct_usuarios/get_usuarios ct_usuarios/nuevo_usuario ct_usuarios/editar_usuario ct_usuarios/eliminar_usuario
vw_graf_tray	Pantalla de gestión de los puntos de una trayectoria.	ct_graf_tray get_trayectoria ct_graf_tray/nueva_punto_trayectoria ct_graf_tray /get_rutas
vw_horario	Pantalla de gestión de horarios de una parada	ct_usuarios/ get_horarios ct_usuarios/nuevo_horario ct_usuarios/ editar_horario ct_usuarios/eliminar_horario
vw_login	Formulario de acceso al sistema, mediante el usuario y contraseña como	ct_login/ iniciar_sesion
vw_modificar_psw	Formulario de actualización de contraseña de cada usuario que ha iniciado sesión.	ct_modificar_psw/ modificarContrasenia

### Controlador.

Nombre del controlador: ct\_paradas

Opción	Descripción	Vista relacionada	Modelo relacionado
nuevo	Llama a la función crear nueva_parada y envía las variables recibidas desde la vista.	vw_paradas	md_paradas
eliminar	Llama a la función eliminar_parada y envía el id del registro a eliminar.	vw_paradas	md_paradas

Nombre del controlador: ct\_rutas

Opción	Descripción	Vista relacionada	Modelo relacionado
nuevo	Llama a la función crear nueva_ruta y envía las variables recibidas desde la vista.	vw_rutas	md_rutas
actualizar	Llama a la función editar_ruta y envía las variables recibidas desde la vista para su actualización mediante el id	vw_rutas	md_rutas
eliminar	Llama a la función eliminar_ruta y envía el id del registro a eliminar.	vw_rutas	md_rutas

Nombre del controlador: ct\_ruta\_parada

Opción	Descripción	Vista relacionada	Modelo relacionado
nuevo	Llama a la función crear nueva_ruta_parada y envía las variables recibidas desde la vista. Llama a la función obtener rutas y paradas complementándose con los modelos de cada clase respectivamente.	vw_ruta_parada	md_rutas md_paradas md_ruta_parada
actualizar	Llama a la función editar_ruta_parada y envía las variables recibidas desde la vista para su actualización mediante el id.	vw_ruta_parada	md_rutas md_paradas md_ruta_parada
eliminar	Llama a la función eliminar_ruta_parada y envía el id del registro a eliminar.	vw_ruta_parada	md_rutas md_paradas md_ruta_parada

Nombre del controlador: ct\_graf\_tray

Opción	Descripción	Vista relacionada	Modelo relacionado
nueva_punto_trayectoria	Llama a la función crear nueva_punto_trayectoria y envía todos las variables recibidas desde la vista.	vw_graf_tray	md_graf_tray md_rutas
eliminar	Llama a la función eliminar_trayectoria y envía el id del registro a eliminar.	vw_graf_tray	md_graf_tray md_rutas

Nombre del controlador: ct\_horarios

Opción	Descripción	Vista relacionada	Modelo relacionado
nuevo	Llama a la función crear nuevo_horario y envía las variables recibidas desde la vista.	vw_horarios	md_horarios
actualizar	Llama a la función editar_horario y envía las variables recibidas desde la vista para su actualización mediante el id	vw_horarios	md_horarios
eliminar	Llama a la función eliminar_horario y envía el id del registro a eliminar.	vw_horarios	md_horarios



Nombre del controlador: ct\_usuarios

Opción	Descripción	Vista relacionada	Modelo relacionado
nuevo	Llama a la función crear nuevo_usuario y envía las variables recibidas desde la vista.	vw_usuarios	md_usuarios
actualizar	Llama a la función editar_usuario y envía las variables recibidas desde la vista para su actualización mediante el id	vw_usuarios	md_usuarios
eliminar	Llama a la función eliminar_usuario y envía el id del registro a eliminar.	vw_usuarios	md_usuarios

Nombre del controlador: ct\_modificar\_psw

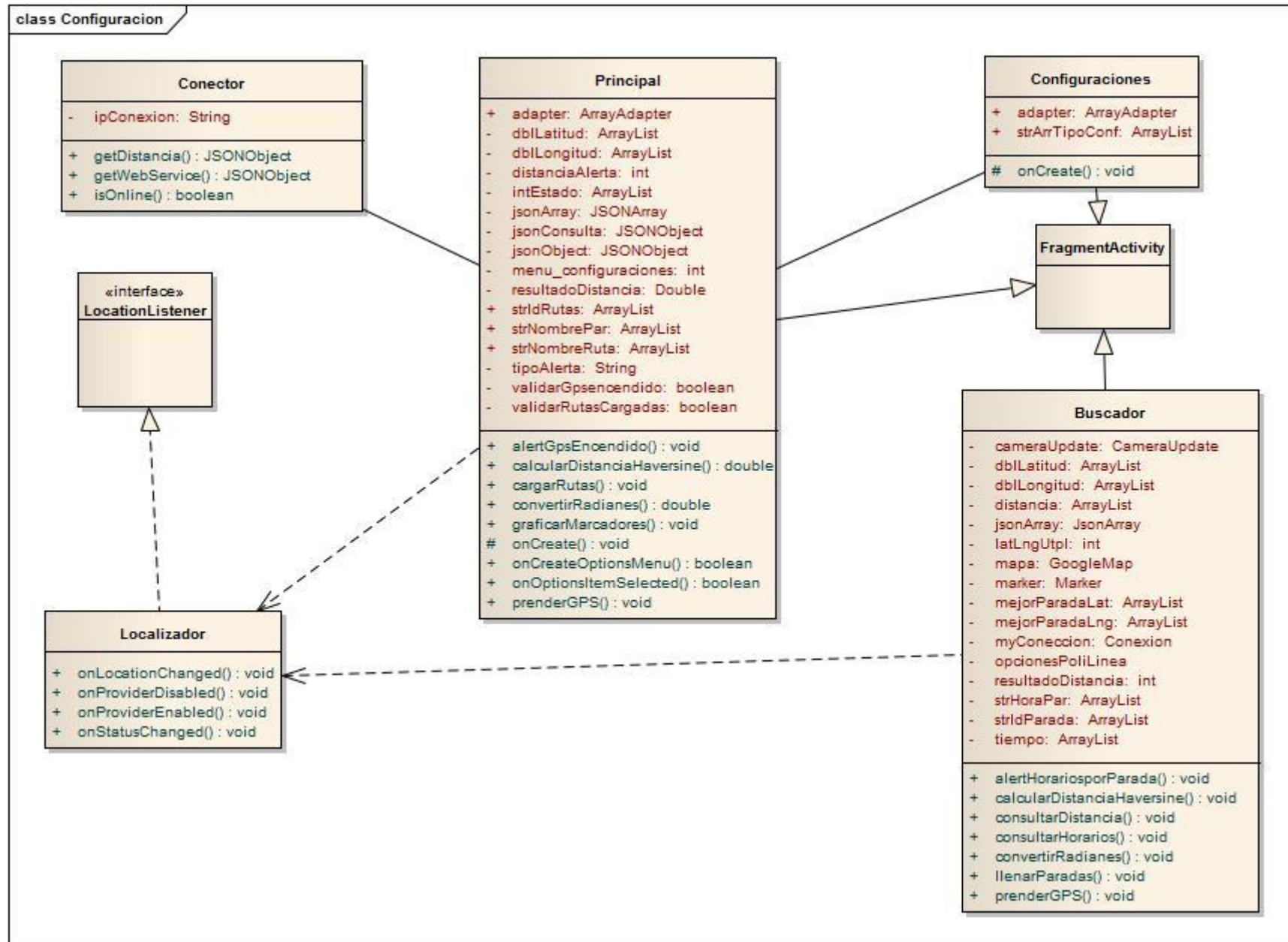
Opción	Descripción	Vista relacionada	Modelo relacionado
modificar_contra	Llama a la función modificarContrasenia y envía las variables recibidas desde la vista.	vw_modificar_psw	md_modificar_psw

Nombre del controlador: ct\_login

Opción	Descripción	Vista relacionada	Modelo relacionado
iniciarSesion	Llama a la función iniciar_sesion y envía las variables recibidas desde la vista.	vw_modificar_psw	md_modificar_psw

## ANEXO B.

### Diagrama de clases aplicación móvil.



En el diagrama anterior se puede observar todas las clases e interfaces que componen la aplicación android las cuales detallo a continuación.

**AlertarParada.-** Contiene la mayor cantidad de funciones y métodos con los cuales la aplicación trabajará, esta a su vez es asociada con la clase Conexion, Configuración y FragmentActivity.

**BuscarParada.-** Clase la cual contiene funciones que sirven para localizar paradas cercanas al lugar actual del usuario final, así como desplegar los horarios que existen.

**Conexión.-** La clase cual se encarga de establecer comunicación con el webservice así como también comprobar si existe conexión a internet.

**ObtieneUbicacion.-** Es la clase encargada de establecer conexión con el sensor GPS del dispositivo móvil y de esta forma obtener toda la información. También hace una implementación de la interfaz LocationListener que es la cual brinda los métodos necesarios para leer la información.

**Configuración.-** Contiene métodos y funciones que permitan al usuario modificar la forma de alerta ya sea por mensaje de voz o por pantalla. Además se gestiona la distancia en metros en la cual el usuario desea recibir la alerta.

**ANEXO C.**

**Especificaciones de casos de uso Aplicación Web**

<b>UC- 01</b>	<b>Registro de Rutas</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear, actualizar y eliminar registros con la información correspondiente al nombre y sentido de la ruta.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario se autentifica correctamente en el sistema.	
	2		El sistema presenta las opciones: a) Gestión de Rutas. • Registro Rutas • Graficar Trayectoria • Registro Horario b) Registro de Paradas. c) Gestión de Recorrido. d) Administración de Usuarios.
	3	El usuario selecciona Gestión de Rutas la sub opción Registro de Rutas.	
	4		El sistema presenta la lista de registros con las opciones de: a) Agregar Ruta b) Editar Ruta c) Eliminar Ruta
	5	El usuario selecciona la opción Agregar Ruta, Editar Ruta o Eliminar Ruta.	

	6		Validar los datos del formulario y guarda si los datos son correctos se procede a agregar, actualizar o eliminar la ruta, caso contrario emitirá un mensaje con la excepción correspondiente.
	7	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.	
	5	Si el sistema detecta que el formulario está mal llenado o que exista algún registro con las mismas características, o si se trata de eliminar un registro que depende de otro se emitirá un mensaje de alerta y no se ejecutará la acción.	
<b>Frecuencia esperada</b>	2 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 02</b>	<b>Graficar de puntos de trayectoria</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear y eliminar registros la cual graficará la trayectoria de una ruta determinada. Los diferentes puntos georeferenciados que se vayan agregando irán graficando la ruta en el orden de ingreso.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema. Conexión a internet para carga de mapa Google Maps.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario se autentifica correctamente en el sistema.	

	2		El sistema presenta las opciones:  a) Gestión de Rutas. • Registro Rutas • Graficar Trayectoria • Registro Horario b) Registro de Paradas. c) Gestión de Recorrido. d) Administración de Usuarios.
	3	El usuario selecciona Gestión de Rutas la sub opción Graficar Trayectoria.	
	4		El sistema presenta la lista de registros con las opciones de:  a) Agregar Trayectoria b) Eliminar Trayectoria
	5	El usuario selecciona la opción Agregar Trayectoria.	
	6		Mapa de Google con un marcador el cual el usuario tendrá que desplazarlo para obtener el punto geo referenciado.
	7	El usuario selecciona la opción Eliminar Trayectoria.	
	8		Mensaje de confirmación de eliminar registro
	9	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	

	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.
	5	Si el sistema detecta que falta seleccionar la ruta en el formulario emitirá un mensaje.
<b>Frecuencia esperada</b>	10 veces/día	
<b>Estabilidad</b>	alta	
<b>Comentarios</b>	Es necesario que los puntos se ingresen en orden debido a que se irán ordenando conforme el ingreso.	

<b>UC- 03</b>	<b>Registro de Horario</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear, actualizar y eliminar registros con la información correspondiente al horario ingresado.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario se autentifica correctamente en el sistema.	
	2		El sistema presenta las opciones: <ul style="list-style-type: none"> <li>a) Gestión de Rutas. <ul style="list-style-type: none"> <li>• Registro Rutas</li> <li>• Graficar Trayectoria</li> <li>• Registro Horario</li> </ul> </li> <li>b) Registro de Paradas.</li> <li>c) Gestión de Recorrido.</li> <li>d) Administración de Usuarios.</li> </ul>



	3	El usuario selecciona Gestión de Rutas la sub opción Registro de Horario.	
	4		El sistema presenta la lista de registros con las opciones de:  a) Agregar Horario b) Editar Horario c) Eliminar Horario
	5	El usuario selecciona la opción Agregar Horario, Editar Horario o Eliminar Horario.	
	6		Validar los datos del formulario y guarda si los datos son correctos se procede a agregar, actualizar o eliminar la ruta, caso contrario emitirá un mensaje con la excepción correspondiente.
	7	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.	
	5	Si el sistema detecta que el formulario está mal llenado o si se trata de eliminar un registro que depende de otro se emitirá un mensaje de alerta y no se ejecutará la acción.	
<b>Frecuencia esperada</b>	15 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 04</b>	<b>Registro de Paradas</b>
<b>Tipo</b>	Implementación Obligatoria
<b>Prioridad</b>	Alta

<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear y eliminar registros la cual agregará las paradas que después serán agregadas a una ruta determinada.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario se autentifica correctamente en el sistema.	
	2		El sistema presenta las opciones: a) Gestión de Rutas. <ul style="list-style-type: none"> <li>• Registro Rutas</li> <li>• Graficar Trayectoria</li> <li>• Registro Horario</li> </ul> b) Registro de Paradas. c) Gestión de Recorrido. d) Administración de Usuarios.
	3	El usuario selecciona Registro de Paradas.	
	4		El sistema presenta la lista de registros con las opciones de: a) Agregar Parada b) Eliminar Parada
	5	El usuario selecciona la opción Agregar Parada.	

	6		Mapa de Google con un marcador el cual el usuario tendrá que desplazarlo para obtener el punto geo referenciado y así obtener la dirección, latitud y longitud. El nombre de la parada así como la intersección se escribirá manualmente en el formulario.
	7	El usuario selecciona la opción Eliminar Parada.	
	8		Mensaje de confirmación de eliminar registro
	9	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.	
	5	Si el sistema detecta que el formulario está mal llenado o si se trata de eliminar un registro que depende de otro se emitirá un mensaje de alerta y no se ejecutará la acción.	
<b>Frecuencia esperada</b>	15 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 05</b>	<b>Gestión de Recorrido</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear, actualizar y eliminar registros que consistirá en agregar un recorrido en base a las paradas, rutas y horarios, rutas y horarios.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema.  Se deben ingresar con anterioridad las rutas, paradas y horarios correspondientes.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario se autentifica correctamente en el sistema.	
	2		El sistema presenta las opciones:  a) Gestión de Rutas. • Registro Rutas • Graficar Trayectoria • Registro Horario b) Registro de Paradas. c) Gestión de Recorrido. d) Administración de Usuarios.
	3	El usuario selecciona Gestión de Recorrido.	
	4		El sistema presenta la lista de registros con las opciones de:  a) Agregar Recorrido b) Actualizar Recorrido c) Eliminar Recorrido
	5	El usuario selecciona la opción Agregar Recorrido.	

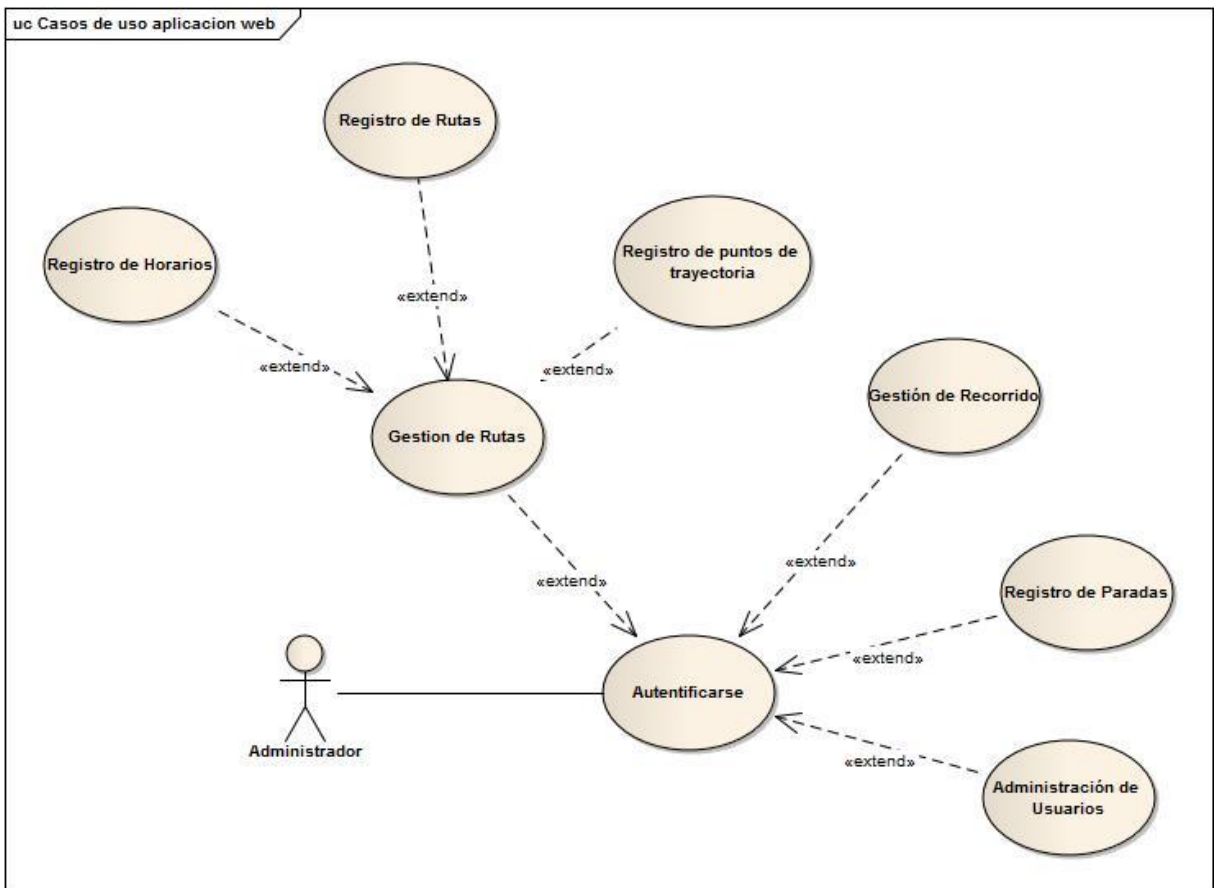
	6		<p>Despliega todas las paradas, rutas y horarios.</p> <p>Validar los datos del formulario y guarda si los datos son correctos se procede a agregar, actualizar o eliminar el recorrido, caso contrario emitirá un mensaje con la excepción correspondiente.</p>
	7	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.	
	5	Si el sistema detecta que el formulario está mal llenado o si se trata de eliminar un registro que depende de otro se emitirá un mensaje de alerta y no se ejecutará la acción.	
<b>Frecuencia esperada</b>	20 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 06</b>	<b>Administración de Usuarios</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Administrador		
<b>Descripción</b>	En este caso de uso se especifica la gestión para crear, actualizar y eliminar registros de usuarios que serán los administradores del sistema.		
<b>Precondición</b>	El usuario se ha autenticado en el sistema.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>

Normal	1	El usuario se autentifica correctamente en el sistema.	
	2		El sistema presenta las opciones: a) Gestión de Rutas. <ul style="list-style-type: none"> <li>• Registro Rutas</li> <li>• Graficar Trayectoria</li> <li>• Registro Horario</li> </ul> b) Registro de Paradas. c) Gestión de Recorrido. d) Administración de Usuarios.
	3	El usuario selecciona Administración de Usuarios.	
	4		El sistema presenta la lista de registros con las opciones de: a) Agregar Usuario b) Actualizar Usuario c) Eliminar Usuario
	5	El usuario selecciona la opción Agregar Usuario.	
	6		Validar los datos del formulario y guarda si los datos son correctos se procede a agregar, actualizar o eliminar un usuario, caso contrario emitirá un mensaje con la excepción correspondiente.
	7	Cerrar Sesión	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y contraseña no son correctos o el código de verificación no coincide se emitirá un mensaje de inicio de sesión incorrecto.	

	5	Si el sistema detecta que el formulario está mal llenado o si se trata de eliminar un registro que depende de otro se emitirá un mensaje de alerta y no se ejecutará la acción.
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	alta	
<b>Comentarios</b>		

### Casos de uso aplicación web



## Especificaciones de casos de uso Aplicación Móvil

<b>UC- 01</b>	<b>Alertar Paradas</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Usuario final		
<b>Descripción</b>	En este caso de uso se especifica la gestión para alertar al usuario las paradas próximas a llegar dentro de una ruta determinada.		
<b>Precondición</b>	Sensor GPS activado. Conexión a internet obligatorio.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario abre la aplicación	
	2		La aplicación presenta las opciones:  a) Alertar Paradas. b) Localizar Paradas.
	3	El usuario selecciona la opción Alertar Paradas.	
	4		La aplicación presenta un mapa embebido de Google Maps con las opciones de seleccionar ruta, empezar recorrido y detener recorrido
	5	El usuario selecciona una ruta.	
	6		Grafica la ruta en el mapa de google maps, así como también muestra las paradas que integra la ruta.
	7	El usuario selecciona la opción empezar recorrido.	



	8		El dispositivo móvil empieza a reconocer la ubicación actual y empieza alertar según la aproximación hacia alguna parada de la ruta.
	9	El usuario selecciona la opción detener recorrido	
	10		El sistema deja de alertar las paradas próximas a llegar.
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	7	Si el usuario no activa el sensor GPS, la aplicación antes de empezar el recorrido emitirá una alerta para activar el sensor.	
	5	Si no se ha empezado un recorrido la opción detener recorrido estará deshabilitada.	
<b>Frecuencia esperada</b>	100 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 02</b>	<b>Configuraciones de Variables</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Usuario final		
<b>Descripción</b>	En este caso de uso se especifica la gestión para modificar el tipo de alerta ya sea por voz o por pantalla, así como modificar la distancia que se alertará una parada próxima.		
<b>Precondición</b>			
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario abre la aplicación	

	2		La aplicación presenta las opciones:  a) Alertar Paradas. b) Localizar Paradas.
	3	El usuario selecciona la opción Alertar Paradas.	
	4		La aplicación presenta un mapa embebido de Google Maps con las opciones de seleccionar ruta, empezar recorrido y detener recorrido
	5	El usuario selecciona la opción configuración la cual se visualiza al presionar el botón físico o virtual de menú.	
	6		La aplicación presenta un formulario para realizar los cambios respectivos en cuanto al tipo de alerta y la distancia.
	7	Cambio de parámetros y grabar los cambios con el botón de aceptar.	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	7	Si el valor ingresado en la distancia es menor a 50 metros no permitirá guardar. La distancia tiene que ser mayor o igual a 50 metros.	
<b>Frecuencia esperada</b>	20 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 03</b>	<b>Buscar parada</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Usuario final		
<b>Descripción</b>	En este caso de uso se especifica la gestión para localizar paradas cercanas al lugar donde se encuentra		
<b>Precondición</b>	Sensor GPS activado. Conexión a internet obligatorio.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario abre la aplicación	
	2		La aplicación presenta las opciones:  a) Alertar Paradas. b) Localizar Paradas.
	3	El usuario selecciona la opción Localizar Paradas.	
	4		La aplicación presenta un mapa embebido de Google Maps con todas las paradas registradas, las mismas que se podrán ocultar o mostrar según sea el caso en el mapa. Además se presenta las opciones de buscar parada y mostrar distancia.
	5	El usuario selecciona la opción buscar parada.	
	6		La aplicación identifica la posición actual del dispositivo y resalta con diferente color los marcadores de 3 paradas cercanas.
	7	El usuario selecciona la opción mostrar distancia.	

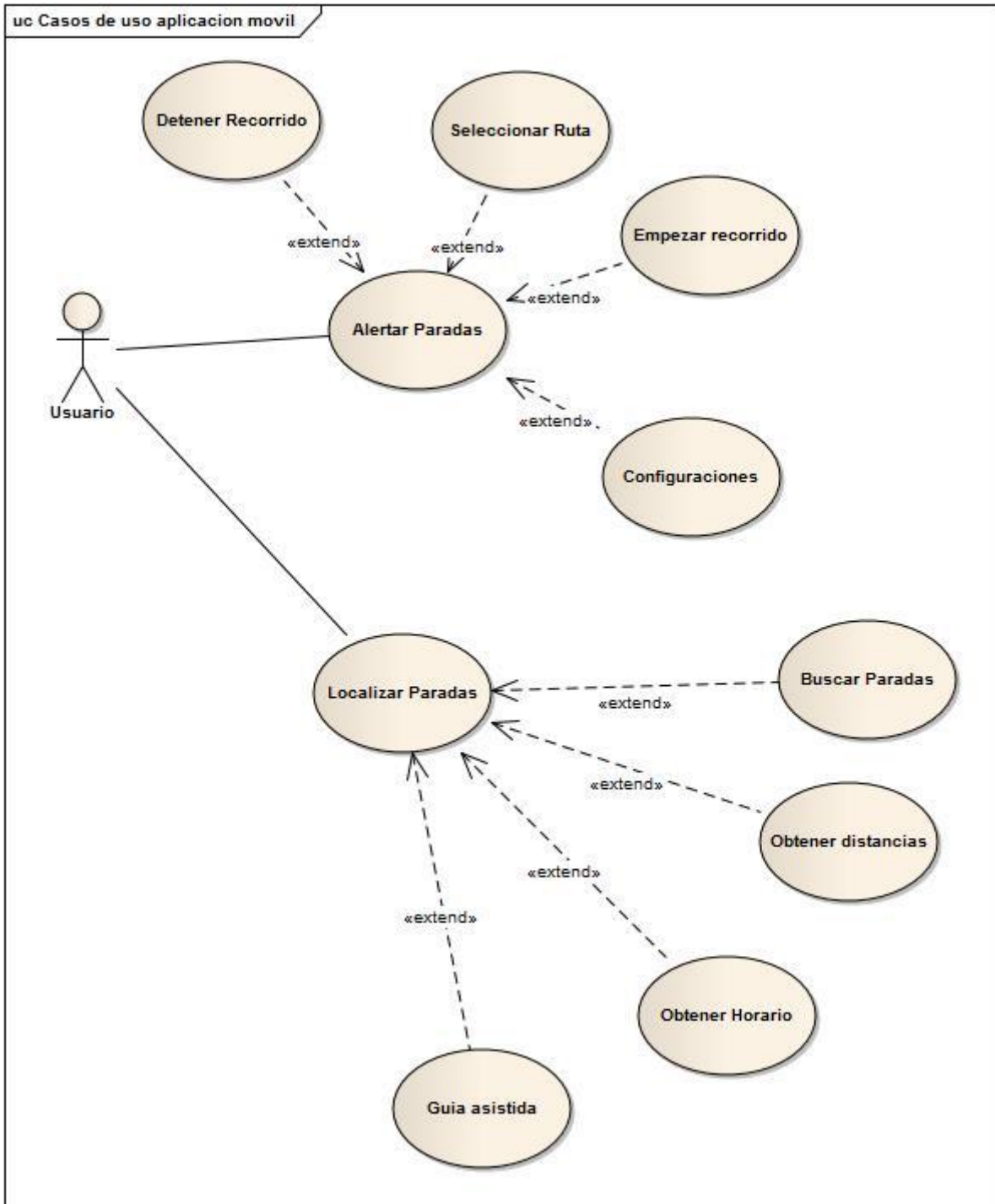
	8		La aplicación mostrará la distancia y tiempo aproximado de llegada a pie de las 3 paradas más cercanas de la posición del dispositivo.
	9	El usuario mantiene presionado un marcador correspondiente a una parada.	
	10		Muestra los horarios asignados a la parada.
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	5	Si el usuario no activa el sensor GPS, la aplicación antes de empezar a buscar las paradas cercanas emitirá un mensaje de alerta.	
	7	Si no se ha seleccionado buscar paradas la opción mostrar distancias estará deshabilitada.	
<b>Frecuencia esperada</b>	100 veces/día		
<b>Estabilidad</b>	alta		
<b>Comentarios</b>			

<b>UC- 04</b>	<b>Guía asistida</b>		
<b>Tipo</b>	Implementación Obligatoria		
<b>Prioridad</b>	Alta		
<b>Actores Involucrados</b>	Usuario final		
<b>Descripción</b>	En este caso de uso se especifica la gestión para utilizar el servicio de navegación asistida que ofrece Google Maps.		
<b>Precondición</b>	Sensor GPS activado. Conexión a internet obligatorio.		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	<b>Respuesta del sistema</b>
Normal	1	El usuario abre la aplicación	

	2		La aplicación presenta las opciones:  e) Alertar Paradas. f) Localizar Paradas.
	3	El usuario selecciona la opción Localizar Paradas.	
	4		La aplicación presenta un mapa embebido de Google Maps con todas las paradas registradas, las mismas que se podrán ocultar o mostrar según sea el caso en el mapa. Además se presenta las opciones de buscar parada y mostrar distancia.
	5	El usuario selecciona una parada	
	6		Se muestra un ícono en la parte inferior derecha para utilizar el servicio de navegación asistida de Google maps.
	7	El usuario selecciona la opción para empezar la navegación o guía asistida.	
	8		La aplicación detectará la posición actual y la de destino, seguidamente empezará a guiar al usuario al lugar que tendrá que desplazarse para llegar a la parada previamente seleccionada.
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	5	Si el usuario no activa el sensor GPS, la aplicación antes de empezar a guiar al usuario hacia la parada seleccionada emitirá un mensaje de alerta.	

<b>Frecuencia esperada</b>	100 veces/día
<b>Estabilidad</b>	alta
<b>Comentarios</b>	

### Casos de uso aplicación móvil.



**ANEXO D.**

**Tarjetas CRC aplicación móvil**

<b>Clase: AlertarParada</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Carga Rutas.	Conexion
Grafica Marcadores sobre el mapa.	Conexión
Activa el servicio de recepción de datos GPS del dispositivo móvil.	ObtieneUbicacion
Verifica si el GPS se encuentra activado.	
Calcula la distancia entre 2 puntos de inicio y 2 puntos de fin (latitud, longitud) respectivamente.	
Inicializa las funciones para empezar el recorrido.	Conexion
Detiene los servicios de recepción GPS.	

<b>Clase: BuscarParada</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Grafica Marcadores sobre el mapa.	Conexión
Activa el servicio de recepción de datos GPS del dispositivo móvil.	ObtieneUbicacion
Verifica si el GPS se encuentra activado.	
Calcula la distancia entre 2 puntos de inicio y 2 puntos de fin (latitud, longitud) respectivamente.	
Inicia la operación de buscar paradas cercanas	Conexion
Reinicio de operación.	

<b>Clase: Conexión</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Petición de servicio web para acceder a la información que será utilizada para la funcionalidad de la aplicación móvil.	
Valida la conexión hacia el servidor.	

<b>Clase: ObtieneUbicacion</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Obtiene información acerca del servicio de geolocalización del dispositivo móvil (GPS).	Principal

<b>Clase: Configuracion</b>	
<b>Responsabilidad</b>	<b>Colaborador</b>
Cambia tipo de alerta ya sea a voz o por pantalla.	Principal
Configura la distancia que se alertara la parada.	Principal



## ANEXO E.

### Diccionario de datos

Nombre de la tabla: paradas

<b>Campo</b>	<b>Descripción</b>	<b>Tipo de Datos</b>	<b>Tipo de Restricción</b>	<b>Procedencia</b>
id_parada	Identificador único del registro parada.	INT(11)	PK NOT NULL	Generado por el sistema
nombre	Nombre completo de la parada	VARCHAR(100)	NOT NULL	Ingresado por el usuario.
direccion	Calle Principal de la parada	VARCHAR(100)	NOT NULL	Generado por geocoder de Google Maps a través de la iteración del usuario.
latitud	Latitud georeferenciada por GPS	DOUBLE	NOT NULL	Generado por Google Maps a través de la iteración del usuario.
longitud	Longitud georeferenciado por GPS	DOUBLE	NOT NULL	Generado por google maps a través de la iteración del usuario.
interseccion	Calle Secundaria de la parada	VARCHAR(150)	NULL	Ingresado por el usuario.

Nombre de la tabla: rutas

<b>Campo</b>	<b>Descripción</b>	<b>Tipo de Datos</b>	<b>Tipo de Restricción</b>	<b>Procedencia</b>
id_ruta	Identificador único del registro ruta.	INT(11)	PK NOT NULL	Generado por el sistema
nombre	Nombre de la ruta.	VARCHAR(150)	NOT NULL	Ingresado por el usuario.
sentido	Sentido por el cual circula el transporte ya sea si sube o baja del campus UTPL.	VARCHAR(50)	NOT NULL	Ingresado por el usuario.

Nombre de la tabla: ruta\_parada

<b>Campo</b>	<b>Descripción</b>	<b>Tipo de Datos</b>	<b>Tipo de Restricción</b>	<b>Procedencia</b>
id_ruta_parada	Identificador único del registro parada_ruta.	INT(11)	PK NOT NULL	Generado por el sistema
id_ruta	Identificador único de la ruta.	INT(11)	FK NOT NULL	Ingresado por el usuario.
id_parada	Identificador único de la parada.	INT(11)	FK NOT NULL	Ingresado por el usuario.
Id_horario	Identificador único del horario asignado.	INT (11)	FK NOT NULL	Ingresado por el Usuario

Nombre de la tabla: usuarios

<b>Campo</b>	<b>Descripción</b>	<b>Tipo de Datos</b>	<b>Tipo de Restricción</b>	<b>Procedencia</b>
id_usuario	Identificador único del registro usuario.	INT(11)	PK NOT NULL	Generado por el sistema
nombres	Nombres del usuario administrador.	VARCHAR(100)	NOT NULL	Ingresado por el usuario.
apellidos	Apellidos del usuario administrador.	VARCHAR(100)	NOT NULL	Ingresado por el usuario.
usuario	Alias con el cual se identificara al usuario.	VARCHAR(50)	NOT NULL	Ingresado por el usuario.
contrasenia	Contraseña de acceso al sistema.	VARCHAR(200)	NOT NULL	Ingresado por el usuario.
correo	Correo electrónico del usuario administrador.	VARCHAR(150)	NOT NULL	Ingresado por el usuario.

Nombre de la tabla: graficar\_ruta

<b>Campo</b>	<b>Descripción</b>	<b>Tipo de Datos</b>	<b>Tipo de Restricción</b>	<b>Procedencia</b>
id_graficar_ruta_pk	Identificador único del registro graficar_ruta.	INT(11)	PK NOT NULL	Generado por el sistema
id_ruta_fk	Identificador único de la ruta.	INT(11)	FK NOT NULL	Ingresado por el usuario.
latitud	Latitud de la ubicación del punto a graficar.	Double	NOT NULL	Ingresado por el usuario.
longitud	Longitud de la ubicación del punto a graficar.	Double	NOT NULL	Ingresado por el Usuario

Nombre de la tabla: horarios

Campo	Descripción	Tipo de Datos	Tipo de Restricción	Procedencia
id_horario	Identificador único del registro graficar_ruta.	INT(11)	PK NOT NULL	Generado por el sistema
hora	Hora de parada que posteriormente se asignará.	TIME	NOT NULL	Ingresado por el usuario.

## ANEXO F.

### Manual de usuario

#### Aplicación web

Diseño de interfaz ingreso al sistema



En la figura anterior se muestra la interfaz de inicio de sesión donde el usuario digitara el alias y la contraseña respectivamente.

Por primera vez de ingreso el usuario será “administrador” y la contraseña “administrador”.

Diseño de interfaz pantalla principal.

Tesis! Gestión Rutas Registro de Paradas Paradas-Rutas Administración Usuarios Bienvenido Administrador!

Gestión rutas-paradas + Agregar Parada-Ruta

Mostrar 10 registros Buscar:

Id Ruta	Nombre Ruta	Nombre Parada	Sentido	Hora	Editar	Eliminar
13	Campus UTPL-Terminal Terrestre	Estacion Principal Buses UTPL	Baja y Sube	09:05:00		
14	Campus UTPL-Terminal Terrestre	Hipervalle	Baja y Sube	09:10:00		
15	Campus UTPL-Terminal Terrestre	Terminal Terrestre Baja	Baja y Sube	09:15:00		
16	Campus UTPL-Terminal Terrestre	Parada del Valle	Baja y Sube	09:20:00		
17	Campus UTPL-Terminal Terrestre	Cuxibamba	Baja y Sube	09:30:00		
18	Campus UTPL-Terminal Terrestre	Puerta de la Ciudad	Baja y Sube	09:35:00		
19	Campus UTPL-Terminal Terrestre	Parada Santiago de las montañas	Baja y Sube	09:40:00		
20	Campus UTPL-Terminal Terrestre	Parada Santiago de las montañas	Baja y Sube	09:45:00		

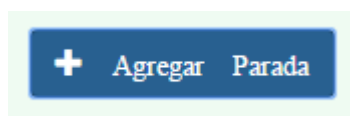
© 2015 - Build by: Edison Cadena - Todos los derechos reservados

En la figura anterior se muestra la interfaz de la pantalla principal la cual está dividida en 3 secciones:

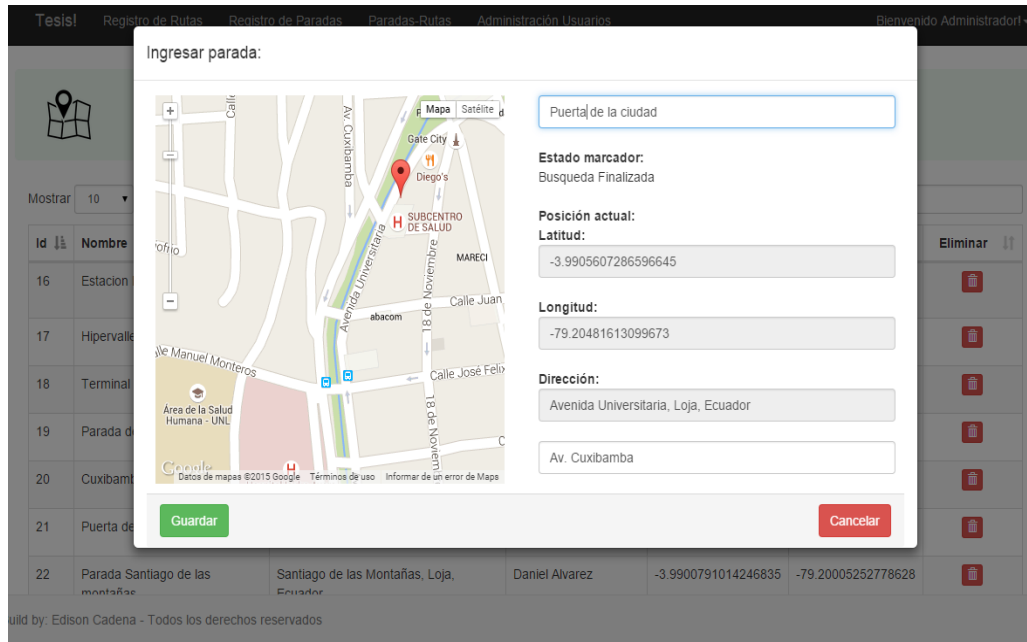
- **Sección de menú.-** El menú de opciones disponibles.
- **Sección de trabajo.-** Se visualizará la información correspondiente a la selección realizada por el usuario en el menú.
- **Sección de pie de página.-** Se muestra el nombre del desarrollador.

### Ingreso nueva parada

1. Escoger la opción dentro del menú “Registro de Paradas”.
2. Dar clic en el botón Agregar Parada.



3. Mover el marcador que esta sobre el mapa y ubicar el lugar donde se encuentra la parada.
4. Llenar los campos “Nombre” e “Intersección” respectivamente.



5. Clic en el botón “Guardar” para guardar cambios.

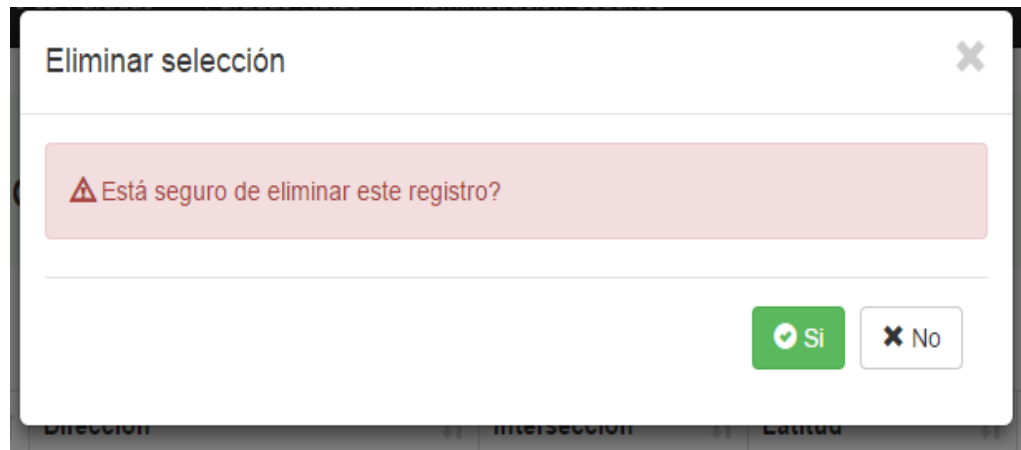
## Eliminar parada

1. Buscar la parada a eliminar.

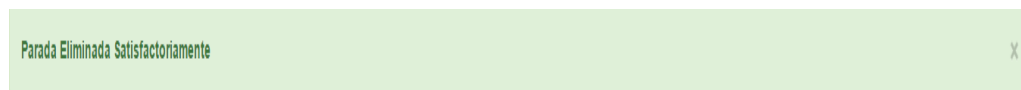


2. Clic en el icono  del registro a eliminar.

3. Confirmar la eliminación del registro.



4. Presionar Si para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.



### Ingreso nueva ruta

1. Escoger la opción dentro del menú "Gestión de Rutas".
2. Escoger la opción "Registro de Rutas"
3. Dar clic en el botón Agregar Ruta.



4. Llenar los campos "Nombre" y seleccionar el sentido de la ruta ya sea que el bus suba, baje o baje y suba al campus Universitario.



A form modal dialog with a blue header bar containing an information icon and the text "Por favor llene la siguiente información:". Below the header, the form is titled "Datos Ruta:". It has two input fields: a text box labeled "Nombre" and a dropdown menu labeled "Sentido de Ruta" with "Sube" selected. At the bottom, there are two buttons: a green "Guardar" button and a red "Cancelar" button.

5. Clic en el botón "Guardar" para guardar cambios.
6. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.


## Actualizar ruta

1. Buscar la ruta a editar.

Mostrar 10 registros      Buscar: camp

Id	Nombre	Sentido Ruta	Editar	Eliminar
5	Campus UTPL-Terminal Terrestre	Baja y Sube		

Mostrando 1 - 1 de 1 registros (filtrados 2 total registros)      Anterior 1 Siguiente

2. Clic en el icono  del registro a editar.
3. Llenar los datos del registro actualizar.

**Por favor llene la siguiente información:**

**Datos Ruta:**

**Sentido de Ruta**



4. Presionar Guardar para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

Ruta Actualizada satisfactoriamente


## Eliminar ruta

1. Buscar la ruta a eliminar.

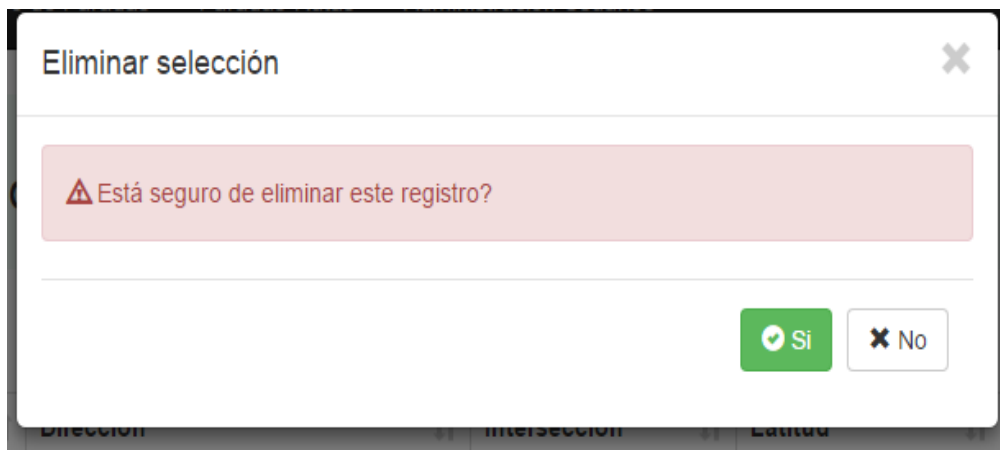
Mostrar 10 registros      Buscar: camp

Id	Nombre	Sentido Ruta	Editar	Eliminar
5	Campus UTPL-Terminal Terrestre	Baja y Sube		

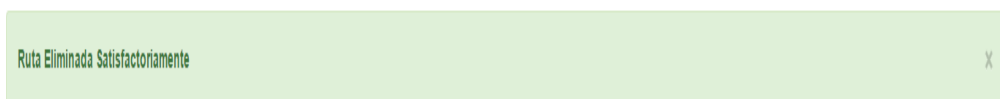
Mostrando 1 - 1 de 1 registros (filtrados 2 total registros)      Anterior 1 Siguiente

2. Clic en el icono  del registro a eliminar.
3. Confirmar la eliminación del registro.



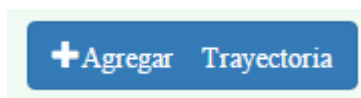


4. Presionar Si para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

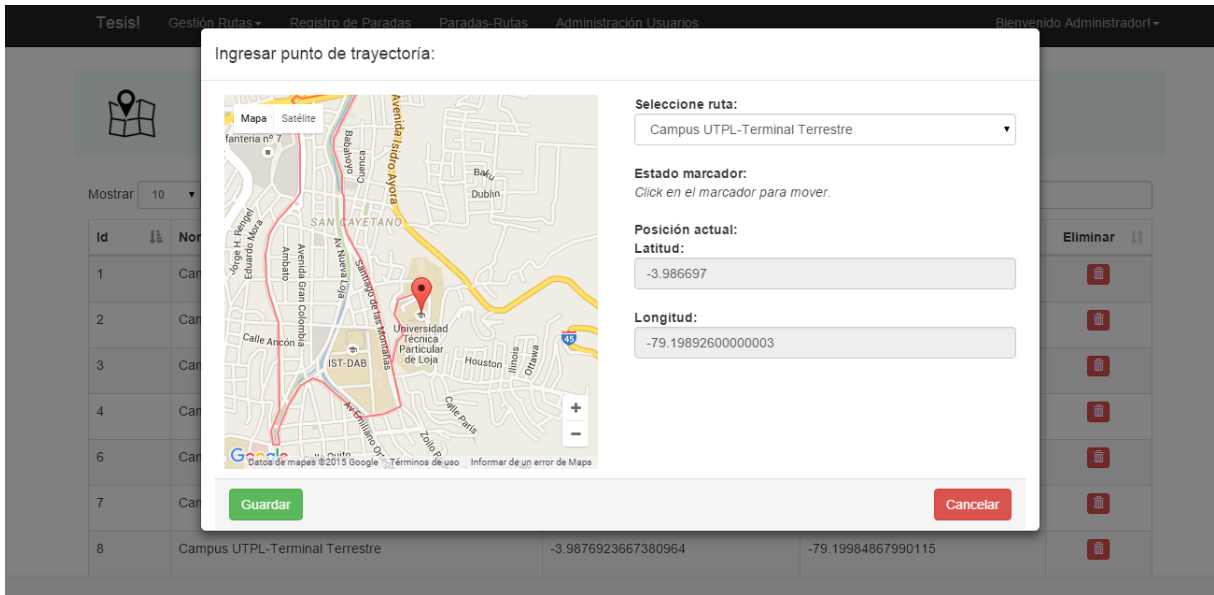


### Ingreso de Trayectoria

1. Escoger la opción dentro del menú "Gestión de Rutas".
2. Escoger la opción "Graficar Trayectoria".
3. Dar clic en el botón Agregar Trayectoria.



4. Seleccionar la ruta a la cual desea agregar los puntos de trayectoria.
5. Mover el marcador que esta sobre el mapa y ubicar el lugar de punto de trayectoria siguiente.



6. Clic en el botón “Guardar” para guardar cambios.

## Eliminar Trayectoria

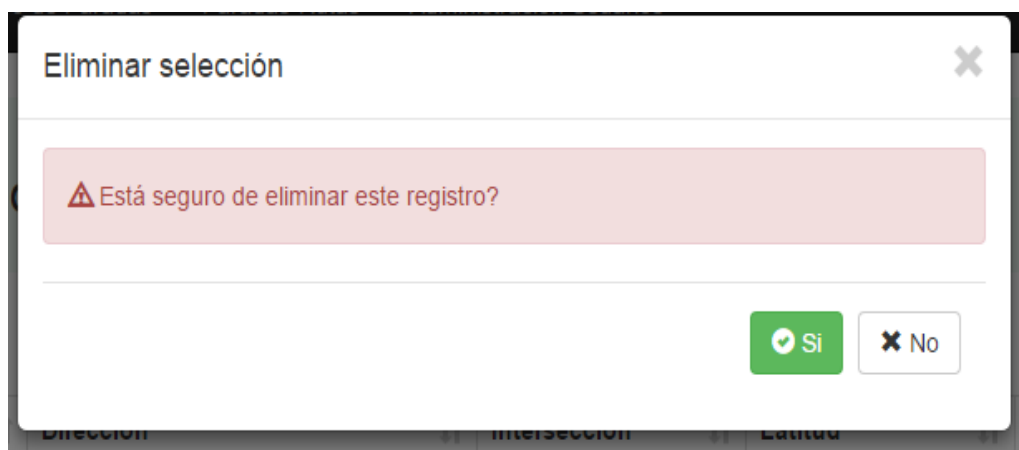
1. Buscar la parada a eliminar.

Mostrar 10 registros Buscar: Campus UTPL

Id	Nombre Ruta	Latitud	Longitud	Eliminar
1	Campus UTPL-Terminal Terrestre	-3.985434059823529	-79.19845393121341	

2. Clic en el icono  del registro a eliminar.

3. Confirmar la eliminación del registro.

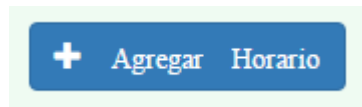


4. Presionar Si para confirmar o no para cancelar

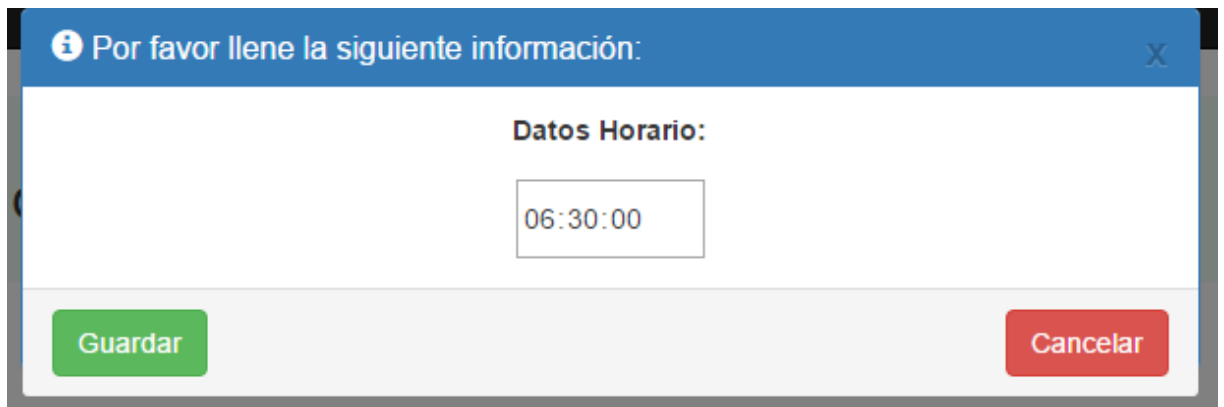
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

## Ingreso horario

1. Escoger la opción dentro del menú “Gestión de Rutas”.
2. Escoger la opción “Registro de Horario”
3. Dar clic en el botón Agregar Horario.



4. Agregar un horario para posteriormente ser asignado a una parada.

Una ventana modal con un encabezado azul que dice "Por favor llene la siguiente información:". El título del formulario es "Datos Horario:". Hay un campo de texto con el valor "06:30:00". En la parte inferior hay dos botones: "Guardar" (verde) y "Cancelar" (rojo).



5. Clic en el botón “Guardar” para guardar cambios.
6. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

## Actualizar horario

1. Buscar el horario a editar.


Mostrar  registros

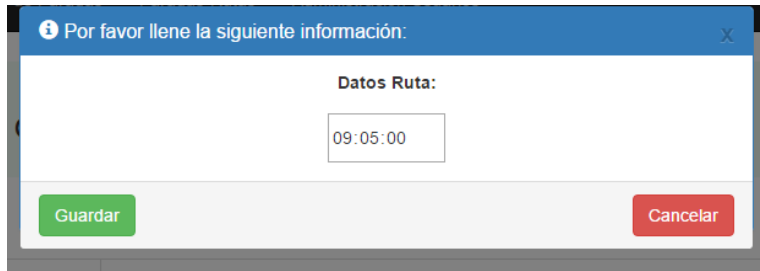
Buscar:

Id	Horario	Editar	Eliminar
1	09:05:00		

Mostrando 1 - 1 de 1 registros (filtrados 9 total registros)

Anterior **1** Siguiente

2. Clic en el icono  del registro a editar.
3. Llenar los datos del registro actualizar.





4. Presionar Guardar para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.


### Eliminar horario

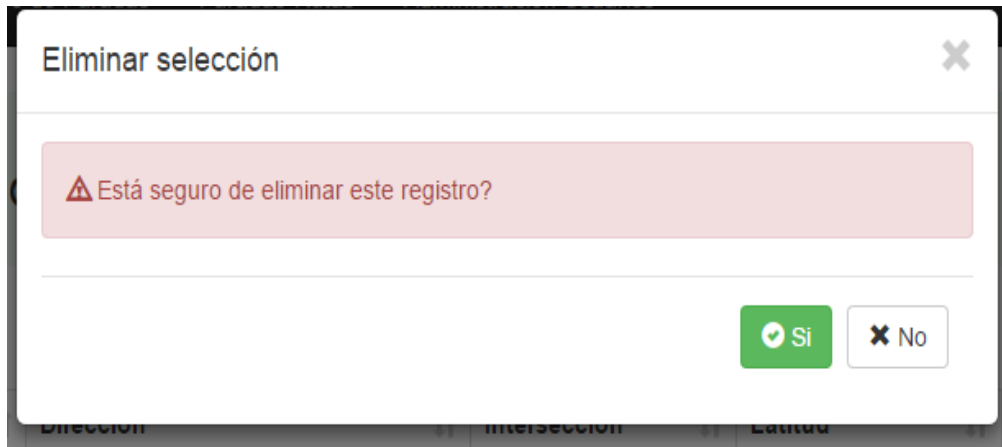
1. Buscar el horario a eliminar.

Mostrar  registros Buscar:

Id	Horario	Editar	Eliminar
1	09:05:00		

Mostrando 1 - 1 de 1 registros (filtrados 9 total registros) Anterior **1** Siguiente

2. Clic en el icono  del registro a eliminar.
3. Confirmar la eliminación del registro.



4. Presionar Si para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

### Ingreso nueva parada-ruta

1. Escoger la opción dentro del menú "Paradas-Rutas".
2. Dar clic en el botón Agregar Paradas-Rutas.

**+ Agregar Parada-Ruta**

3. Seleccionar la ruta, parada y horario que conforma la trayectoria o recorrido.

**Por favor llene la siguiente información:**

**Datos Ruta-Parada:**

**Agregar ruta**  
Campus UTPL-Terminal Terrestre

**Agregar parada**  
Estacion Principal Buses UTPL

**Agregar Horario**  
09:05:00

**Guardar** **Cancelar**

4. Clic en el botón “Guardar” para guardar cambios.
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.


## Actualizar parada-ruta

1. Buscar el registro a editar.

Mostrar 10 registros      Buscar: Paraj

Id Ruta	Nombre Ruta	Nombre Parada	Sentido	Editar	Eliminar
5	Campus UTPL-Terminal Terrestre	Parada del Valle	Baja y Sube		
5	Campus UTPL-Terminal Terrestre	Parada Santiago de las montañas	Baja y Sube		

Mostrando 1 - 2 de 2 registros (filtrados 7 total registros)      Anterior 1 Siguiente

2. Clic en el icono  del registro a editar.
3. Llenar los datos del registro actualizar.

**Por favor llene la siguiente información:**

**Datos Ruta-Parada:**

**Modificar ruta**  
 Campus UTPL-Terminal Terrestre

**Modificar parada**  
 Estacion Principal Buses UTPL

**Modificar Horarios**  
 09:05:00

**Guardar** **Cancelar**

4. Presionar Guardar para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

Ruta-Parada Actualizada satisfactoriamente

### Eliminar parada-ruta

1. Buscar el registro a eliminar.


Mostrar 10 registros

Buscar: Para

Id Ruta	Nombre Ruta	Nombre Parada	Sentido	Editar	Eliminar
5	Campus UTPL-Terminal Terrestre	Parada del Valle	Baja y Sube		
5	Campus UTPL-Terminal Terrestre	Parada Santiago de las montañas	Baja y Sube		

Mostrando 1 - 2 de 2 registros (filtrados 7 total registros)

Anterior 1 Siguiente

2. Clic en el icono  del registro a eliminar.
3. Confirmar la eliminación del registro.

**Eliminar selección**

**Está seguro de eliminar este registro?**

**Si** **No**

4. Presionar Si para confirmar o no para cancelar

5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

### Ingreso nuevo usuario

1. Escoger la opción dentro del menú “Administración Usuarios”.
2. Dar clic en el botón Agregar Usuario.



3. Llena los campos necesarios para la creación del usuario.

**Por favor llene la siguiente información:**

**Datos Usuario:**

Nombres

Apellidos

Usuario

Correo



Contraseña

**Guardar** **Cancelar**

4. Clic en el botón “Guardar” para guardar cambios.
5. Si mostrará un mensaje de confirmación para validar si se realizó la transacción.


### Actualizar usuario

1. Buscar el registro a editar.

Nombres	Apellidos	Usuario	Correo	Editar	Eliminar
Daniel	Guaman	daniel	daniel@gmail.com		

Mostrando 1 - 1 de 1 registros (filtrados 2 total registros)

Anterior **1** Siguiente

2. Clic en el icono  del registro a editar.
3. Llenar los datos del registro actualizar.

Por favor llene la siguiente información:

**Datos Usuario:**

Daniel

Guaman

daniel

daniel@gmail.com

Guardar Cancelar

4. Presionar Guardar para confirmar o no para cancelar
5. Si mostrará un mensaje de confirmación para validar si se realizó la transacción.

Usuario Actualizado satisfactoriamente


## Eliminar usuario

1. Buscar el registro a eliminar.

Nombres	Apellidos	Usuario	Correo	Editar	Eliminar
Daniel	Guaman	daniel	daniel@gmail.com		

Mostrando 1 - 1 de 1 registros (filtrados 2 total registros)

Anterior 1 Siguiente

2. Clic en el icono  del registro a eliminar.
3. Confirmar la eliminación del registro.

Eliminar selección

Está seguro de eliminar este registro?

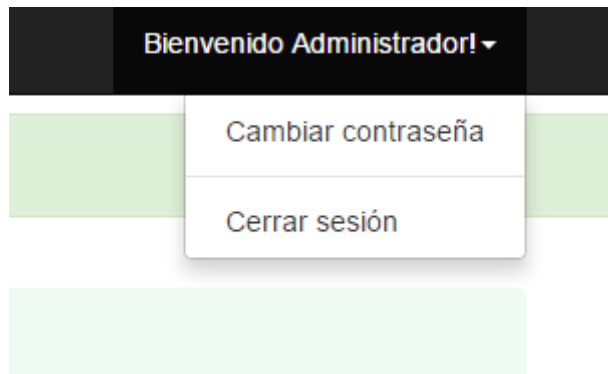
Si No



4. Presionar Si para confirmar o no para cancelar
5. Se mostrará un mensaje de confirmación para validar si se realizó la transacción.

### Modificar Contraseña

1. Desplazarse al menú y dar clic en el nombre de usuario que ha iniciado sesión para desplegar el sub-menú.



2. Dar clic en Cambiar contraseña.
3. Llenar los campos necesarios para la actualización de la contraseña.

Un formulario web con un fondo gris claro y un borde gris. En el centro, el título "Modificar Contraseña" está escrito en una fuente grande y oscura. Debajo del título, hay tres campos de entrada de texto con bordes redondeados y un fondo blanco. Los campos están etiquetados como "Contraseña Actual", "Nueva Contraseña" y "Repita Nueva Contraseña". En la parte inferior derecha del formulario, hay un botón rectangular azul con el texto "Guardar" en blanco.

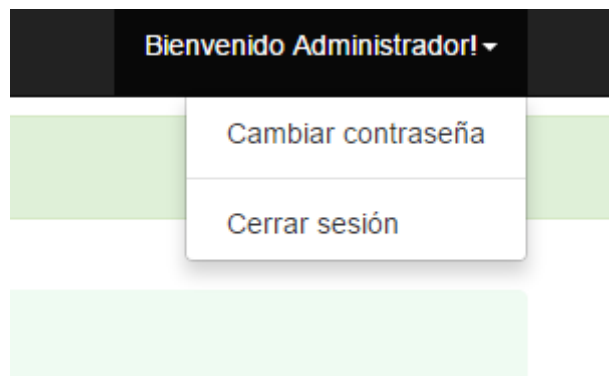
4. Clic en el Botón Guardar.
5. Se mostrará mensaje de confirmación de actualización de contraseña.

La contraseña se actualizo correctamente.

X

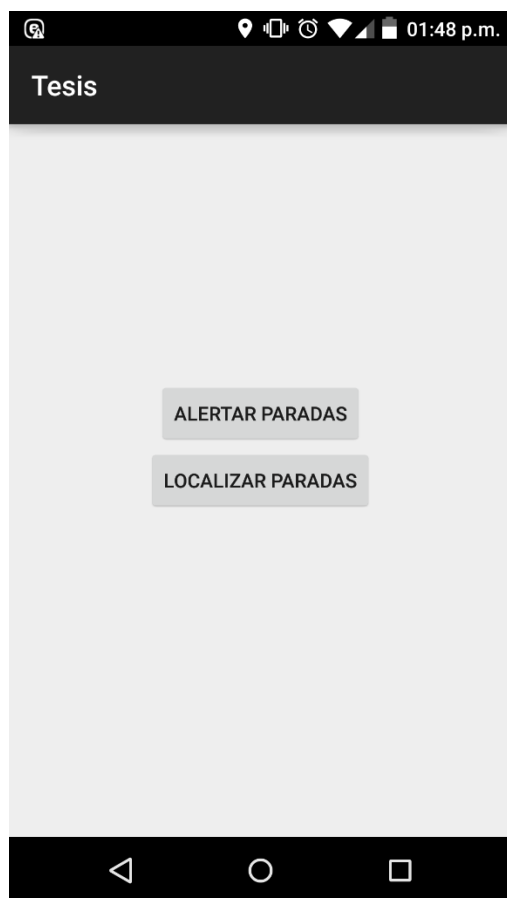
## Cerrar sesión

1. Desplazarse al menú y dar clic en el nombre de usuario que ha iniciado sesión para desplegar el sub-menú.



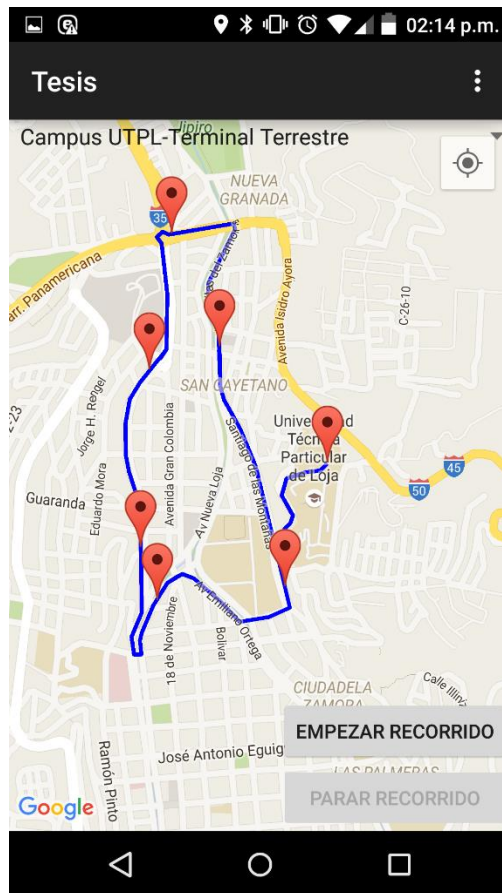
2. Clic en Cerrar sesión.

## Aplicación Móvil



1. Seleccionar la opción correspondiente.
  - a. Alertar Paradas.- Alertar paradas cercanas dentro de una ruta determinada.
  - b. Localizar Paradas.- Busca paradas cercanas desde el lugar de donde se encuentra.

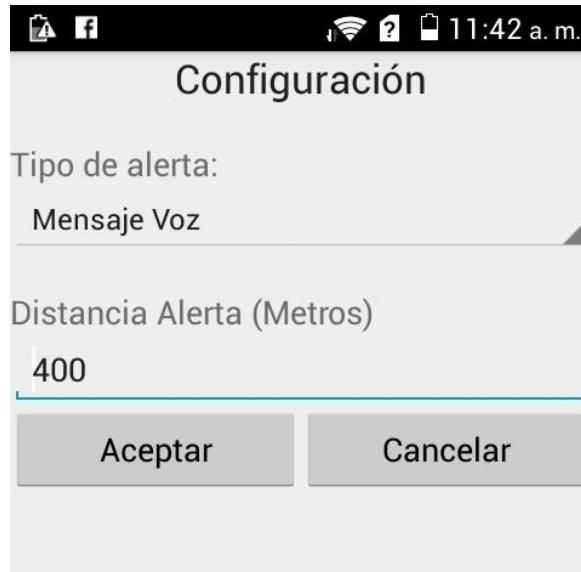
## Alertar Paradas



1. Seleccione la ruta a viajar.
2. Clic en el botón Empezar Recorrido.
3. Para finalizar el clic en el botón Parar Recorrido.

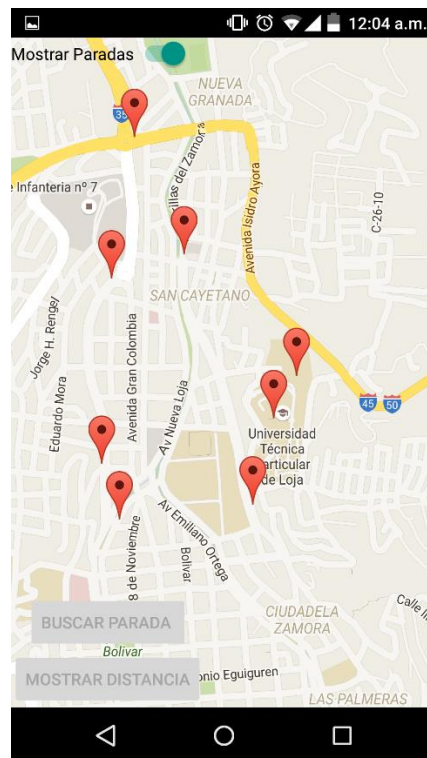
### Configuraciones.

1. Una vez abierta la aplicación presionar el botón físico de menú del dispositivo para visualizar la opción de configuración.
2. Seleccionar el tipo de Alerta (Mensaje de Voz, Mensaje Por Pantalla).
3. Distancia alertar la parada.



4. Clic en el botón Aceptar para guardar los cambios.

### Buscar Paradas



1. Esperar la carga de todas las paradas.
2. Prender GPS.
3. Esperar posición actual.
4. Presionar botón Buscar Parada.
5. Presionar botón mostrar distancia.

## ANEXO G.

### Pruebas de aceptación

Prueba de aceptación	
Número: 1	Número Historia de Usuario: 1
<b>Descripción:</b> Acceso al sistema por a través de un navegador.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"><li>• Se debe digitar el acceso a la aplicación.</li><li>• Disponer de conexión a internet obligatoriamente.</li><li>• Ingresar el usuario y contraseña respectivamente.</li></ul>	
<b>Pasos de ejecución:</b> Ingresar <a href="http://&lt;nombrservidor&gt;/tesis">http://&lt;nombrservidor&gt;/tesis</a>	
<b>Resultados esperados</b> Ingreso al sistema satisfactoriamente.	
<b>Evaluación de la prueba</b> Satisfactoria	

<b>Prueba de aceptación</b>	
<b>Número:</b> 2	<b>Número Historia de Usuario:</b> 2
<b>Descripción:</b> Gestión en la administración del sistema.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• Haber iniciado sesión previamente.</li> </ul>	
<b>Pasos de ejecución:</b> <p>El usuario:</p> <ul style="list-style-type: none"> <li>• Ingresa al menú “Registro de Paradas”. En la opción seleccionada el usuario realiza operaciones de creación y eliminación de los datos.</li> <li>• Ingresa al menú “Registro de Rutas”. En la opción seleccionada el usuario realiza operaciones de creación, actualización y eliminación de los datos.</li> <li>• Ingresa al menú “Paradas-Rutas”. En la opción seleccionada el usuario realiza operaciones de creación, actualización y eliminación de los datos.</li> <li>• Ingresa al menú “Administración de Usuarios”. En la opción seleccionada el usuario realiza operaciones de creación, actualización y eliminación de los datos.</li> <li>• Ingresa al menú donde se presenta el nombre de usuario e ingresa al submenú Cambiar Contraseña.</li> <li>•</li> </ul>	
<b>Resultados esperados</b> Gestión de la administración del sistema.	
<b>Evaluación de la prueba</b> Satisfactoria	

Prueba de aceptación	
<b>Número:</b> 3	<b>Número Historia de Usuario:</b> 3
<b>Descripción:</b> Funcionalidad de la aplicación móvil.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• Instalar Aplicación.</li> <li>• Conexión a internet.</li> <li>• Activar Gps.</li> </ul>	
<b>Pasos de ejecución:</b> El usuario: <ul style="list-style-type: none"> <li>• Selecciona la ruta por la cual desea movilizarse.</li> <li>• Da clic en el botón Iniciar Recorrido para empezar.</li> <li>• Para configurar da clic en el botón físico de menú y selecciona configuraciones.</li> <li>• Cambia el tipo de notificación y la distancia que desee que se le notifique.</li> </ul>	
<b>Resultados esperados</b> <ul style="list-style-type: none"> <li>• La aplicación notifica al usuario que está próximo a llegar a una parada determinada ya sea por mensaje de voz o por pantalla.</li> <li>• Muestra la ubicación del usuario y de las paradas que contiene el recorrido.</li> </ul>	
<b>Evaluación de la prueba</b> Satisfactoria	



Prueba de aceptación	
Número: 4	Número Historia de Usuario: 4
<b>Descripción:</b> Funcionalidad de la aplicación móvil.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• Conexión a internet.</li> <li>• Activar Gps.</li> </ul>	
<b>Pasos de ejecución:</b> El usuario: <ul style="list-style-type: none"> <li>• Presiona botón buscar parada.</li> <li>• Presiona botón mostrar parada</li> </ul>	
<b>Resultados esperados</b> <ul style="list-style-type: none"> <li>• La aplicación mostrará las paradas cercanas disponibles que se encuentren en base a la posición actual del usuario.</li> <li>• Muestra las distancias y el tiempo que tarda en llegar caminando a las paradas</li> </ul>	
<b>Evaluación de la prueba</b> Satisfactoria	

## Pruebas de unitarias

Captura de las diferentes pruebas de unidad que se realizaron en la aplicación móvil con Android Studio, las cuales se ejecutaron satisfactoriamente.

ApplicationTest: 10 total, 10 passed

ec.edu.utpl.testunitario.tests.ApplicationTest	
test_calcularDistanciaHaversine	passed
test_cargarRutas	passed
test_consultarDistancia	passed
test_consultarHorarios	passed
test_getDistancia	passed
test_getWebService	passed
test_graficarMarcadores	passed
test_isOnline	passed
testApplicationTestCaseSetUpProperly	passed
testAndroidTestCaseSetUpProperly	passed

Para la aplicación web se utilizó la librería SimpleTest que permite realizar pruebas de unidad en el lenguaje de programación PHP. Se realizó los diferentes test para probar la funcionalidad de cada módulo. A continuación se presenta un extracto de las pruebas realizadas:

En la presente figura la prueba de unidad ejecutó un fallo debido a que el usuario es incorrecto.

## testmodulos.php

Fail: PruebasUnitariasPhp -> testInicioSesion -> Equal expectation fails at character 0 with [Lo sentimos usua

1/1 test cases complete: 3 passes, 1 fails and 0 exceptions.

Se ha realizado las correcciones respectivas para poder ejecutar las pruebas de unidad sin inconvenientes.

## testmodulos.php

1/1 test cases complete: 4 passes, 0 fails and 0 exceptions.

En la siguiente figura se presenta un extracto de la programación de las pruebas respectivas

```
require_once('librerias/simpletest/autorun.php');
require_once('modulos/conexion/bd_conexion.php');
require_once('modulos/horarios/modelo/md_horarios.php');
require_once('modulos/paradas/modelo/md_paradas.php');
require_once('modulos/login/modelo/md_login.php');

class PruebasUnitariasPhp extends UnitTestCase{
    function testConexion(){
        $conexion= new Conectar();
        $this->assertTrue('modulos/conexion/bd_conexion.php');
    }

    function testHorariosModelo(){
        $horarios= new horarios_model();
        $numeroElementos=count($horarios->get_horarios());
        //Si el número de elementos es mayor o igual a 0 esta bien
        $this->assertTrue($numeroElementos>=0);
    }

    function testParadasModelo(){
        $paradas= new paradas_model();
        $numeroElementos=count($paradas->get_paradas());
        //Si el número de elementos es mayor o igual a 0 esta bien
        $this->assertTrue($numeroElementos>=0);
    }

    function testInicioSesion(){
        $login= new Login_model();
        $this->assertEqual($login->iniciar_sesion('administrador','administrador'),'Accediendo');
    }
}
```

## ANEXO H.

### Instalación aplicación web

Nota: La presente instalación se realizó en el Sistema Operativo Windows 8.1, si se desea instalar en alguna distribución de Linux se recomienda utilizar apache como servidor web.

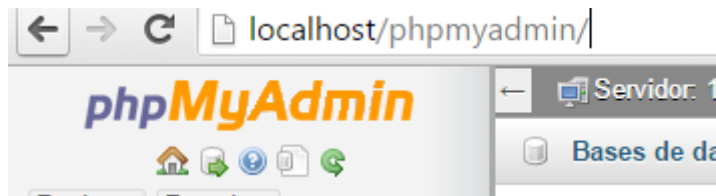
1. Descargar e instalar XAMPP que contiene las aplicaciones necesarias para instalar el proyecto como: apache, mysql entre otros.
2. Copiar los archivos de código fuente a la ruta <directorio>/xampp/htdocs/.

▶ Este equipo ▶ Disco local (C:) ▶ xampp ▶ htdocs

Quedará de la siguiente forma:

tesis 06/07/2015 14:47 Carpeta de archivos

3. Iniciar phpmyadmin para crear la base de datos.



4. Seleccionar base de datos y crear con el nombre “tesis” con cotejamiento “utf8\_spanish\_ci”.

Crear base de datos ⓘ

tesis utf8\_spanish\_ci Crear

5. Una vez seleccionada la base de datos se importara el archivo .sql y hacer clic en continuar.

Bases de datos SQL Estado actual Usuarios Exportar Importar

## Importando al servidor actual

**Archivo a importar:**

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.  
 Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador:  tesis.sql (Máximo: 2,048KB)

Conjunto de caracteres del archivo:

**Importación parcial:**

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo grandes; *sin embargo, puede dañar las transacciones.*

Omitir esta cantidad de consultas (en SQL) o líneas (en otros formatos) desde la primera:

**Formato:**

**Opciones específicas al formato:**

Modalidad SQL compatible:

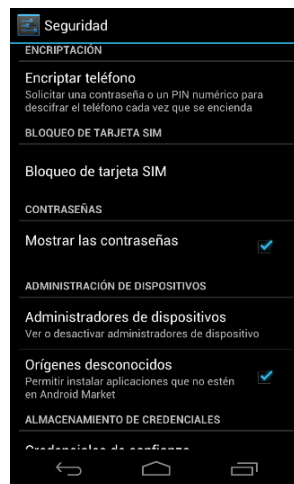
No utilizar AUTO\_INCREMENT con el valor 0

6. Verificar que los servicios de apache y mysql estén corriendo.

Modules				
Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	2372 7064	4430, 8080	<input type="button" value="Stop"/>
<input type="checkbox"/>	MySQL	2904	3306	<input type="button" value="Stop"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/>

## Instalación aplicación móvil

1. Verificar que el sistema operativo sea igual o superior a 4.0
2. Activar la opción orígenes desconocidos que se encuentra dentro de ajustes/seguridad.



3. Abrir archivo .apk
4. Seleccionar Instalar.
5. Aceptar permisos.
6. Fin.

## ANEXO I

### Pruebas de calidad de software

Estas pruebas se las realizo utilizando la herramienta de software SonarQube que valida el código fuente y emite resultados de mantenibilidad, seguridad, eficiencia, usabilidad, portabilidad, entre otros.

Las figuras siguientes son capturas de pantalla realizadas al código fuente que está desarrollado en el lenguaje de programación PHP.

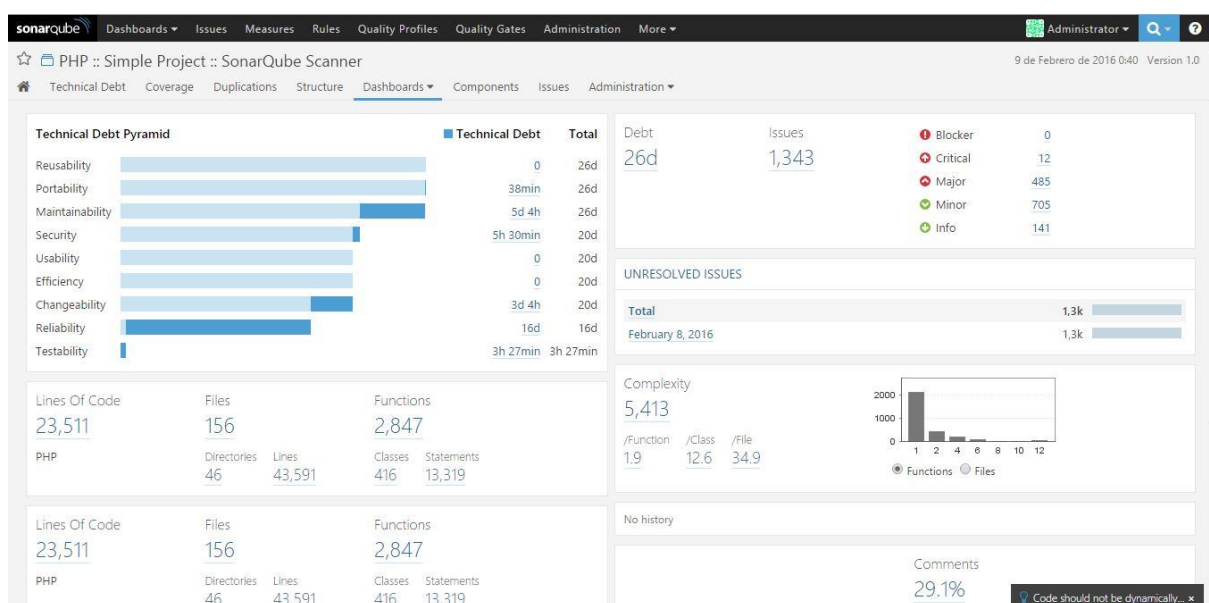


Figura. Tablero de resultados

La figura **tablero de resultados** se muestra diferentes variables como: número de líneas de código, número de funciones, etc. La gráfica Technical Debt Pyramid permite determinar qué nivel de mantenibilidad, seguridad, eficiencia, portabilidad, reusabilidad entre otros tiene el código fuente. Con estos valores se puede verificar el nivel de calidad de software del proyecto.

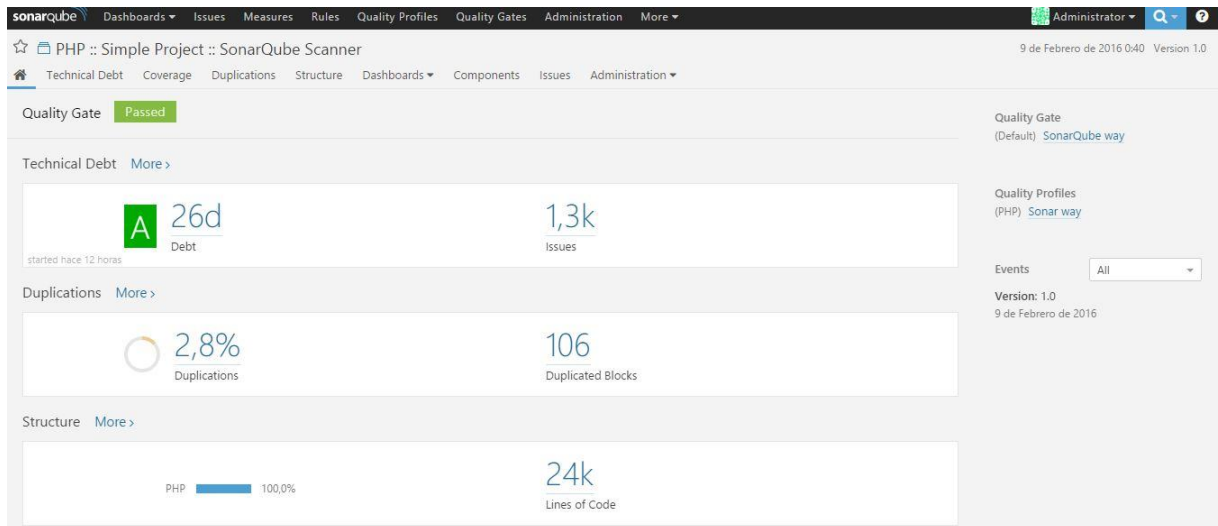


Figura calidad de software.

La figura **calidad de software** se muestra una captura de pantalla de la herramienta SonarQube en la cual se ha realizado la calificación en cuanto a la calidad y los resultados han sido satisfactorios, el proyecto ha “pasado” el análisis así mismo tiene una calificación de A. Los resultados obtenidos permiten validar que nivel de calidad se software tiene el proyecto.

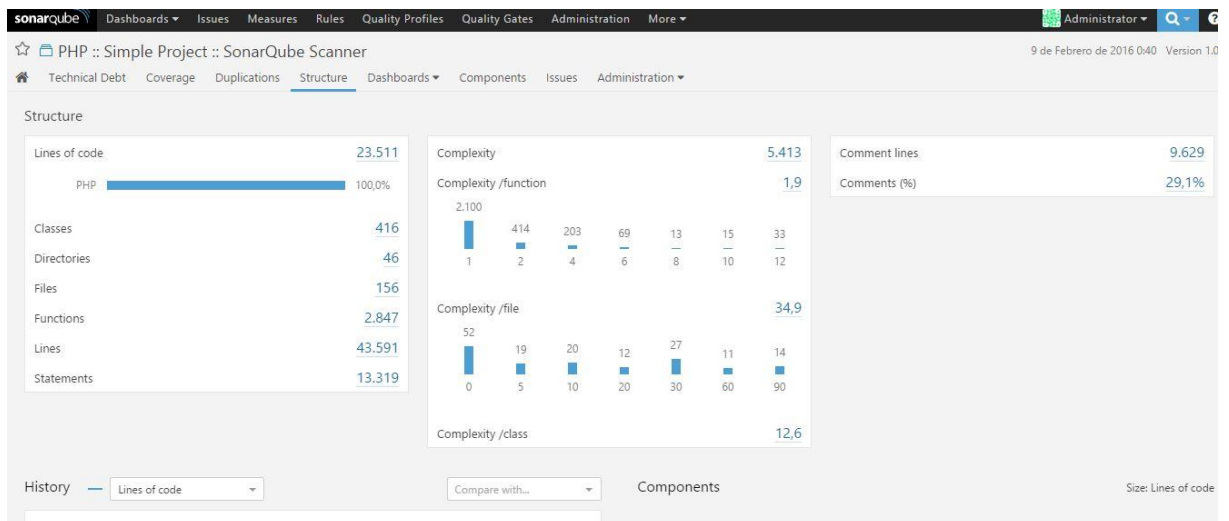


Figura información general estructura.

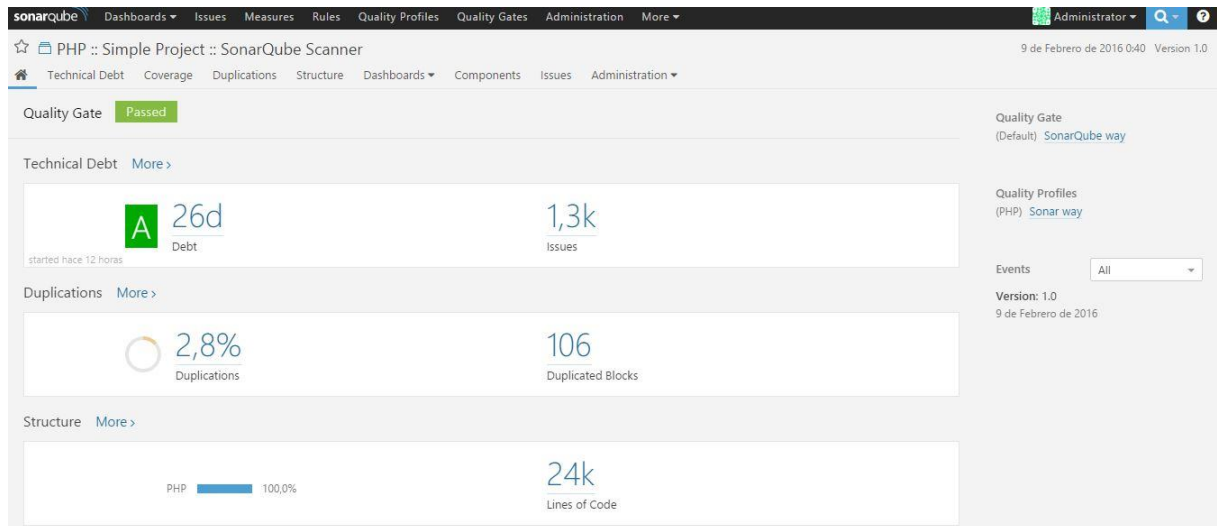


Figura información general calidad de software.

Las figuras **información general estructura** e **información general calidad de software** se muestra un resumen en cuanto al número de líneas de código, funciones, clases entre otras variables que tiene el proyecto, así como también el porcentaje de código duplicado que existe en el caso presentado se puede observar que es mínimo.