



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 2

Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas

**Título: Ruta Habana. Sistema de Información para el  
transporte público en La Habana.**

**Autores:** Lionys Julio Ruiz Casanova  
Pablo Ernesto Miró Martínez

**Tutores:** Ing. Vladimir Milián Núñez  
Ing. Rasiel Ernesto Ciervide Cabalé

**Co Tutora:** Msc. Nora Casanova Casanova

La Habana, junio de 2016  
“Año 58 de la Revolución”



“La innovación es lo que distingue a los líderes de los seguidores”.

Steve Jobs

# Declaración de Autoría

Declaramos ser los únicos autores del trabajo de diploma que tiene como título Ruta Habana. Sistema para la información del transporte público en La Habana para dispositivos que operan con Android y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio del año 2016.

**Lionys Julio Ruiz Casanova**

\_\_\_\_\_

Firma del Autor

**Pablo Ernesto Miró Martínez**

\_\_\_\_\_

Firma del Autor

**Ing. Vladimir Milián Núñez**

\_\_\_\_\_

Firma del Tutor

**Ing. Rasiel Ernesto Ciervide Cabalé**

\_\_\_\_\_

Firma del Tutor

## *Datos de contacto*

### **Tutores**

**Tutor:** Ing. Vladimir Milián Núñez.

**Centro Laboral:** Universidad de la Ciencias Informáticas (UCI).

**Formación Académica:** Ingeniería en Ciencias Informáticas (Julio de 2007).

**Correo Electrónico:** [vmilian@uci.cu](mailto:vmilian@uci.cu)

**Tutor:** Ing. Rasiel Ernesto Ciervide Cabalé.

**Centro Laboral:** Universidad de la Ciencias Informáticas (UCI).

**Formación Académica:** Ingeniería en Ciencias Informáticas (Julio de 2015).

**Correo Electrónico:** [ernesto@uci.cu](mailto:ernesto@uci.cu)

### **Co tutor**

**Co tutora:** Msc. Nora Enriqueta Casanova Casanova.

**Centro Laboral:** Escuela de Iniciación Deportiva (EIDE), Luis Augusto Turcios Lima.

**Formación Académica:** Licenciada en Química (Julio de 1984).

**Correo Electrónico:** [lienory@inder.cu](mailto:lienory@inder.cu)

### **Autores**

**Autor:** Pablo Ernesto Miró Martínez.

**Centro Laboral:** Universidad de la Ciencias Informáticas (UCI).

**Formación Académica:** Ingeniería en Ciencias Informáticas (Julio/2016).

**Correo Electrónico:** [pemiro@estudiantes.uci.cu](mailto:pemiro@estudiantes.uci.cu)

**Autor:** Lionys Julio Ruiz Casanova

**Centro Laboral:** Universidad de la Ciencias Informáticas (UCI).

**Formación Académica:** Ingeniería en Ciencias Informáticas (Julio/2016).

**Correo Electrónico:** [lruiz@estudiantes.uci.cu](mailto:lruiz@estudiantes.uci.cu)

## *Dedicatoria*

### ***De Pablo Ernesto Miró Martínez***

*Quiero dedicar este trabajo de diploma a mis padres por haberme traído al mundo. A mami por cuidar de mí, por darme todo el amor del mundo, por estar siempre que te necesité. A mi papá por formar parte de este trabajo, por aconsejarme cada vez que lo necesité, por ser siempre guía en mis pasos por la vida. A mi hermanita por aguantarme todo el tiempo mis malcriadeces y por cuidarme mucho. A mi queridísimo abuelo que aunque no esté presente, siempre estará conmigo aconsejándome y dándome chuchos sobre todo.*

### ***De Lionys Julio Ruiz Casanova***

*Hoy es un día especial en mi vida y por tanto quiero dedicar esta victoria a aquellas personas que durante estos 24 años no me han dejado caer jamás. En primer lugar dedicar esta tesis a mi madre, por ser la razón de ser de mis días, por brindarme todo el amor que una persona pueda ofrecer a un ser querido, por ser la luz que nunca se ha agotado en mi camino, por educarme, por darme una súper hermana, por prepararme para la vida, por ser ejemplo de sacrificio, deseo y voluntad, por luchar incansablemente por tal de verme feliz, por regañarme cuando fue necesario, en fin por ser mi mami querida. Para ti son hoy las felicitaciones, gracias mamá. En segundo lugar al hombre que encontró a la mejor mujer del mundo para juntos traerme a esta tierra gracias papá, gracias papi por enseñarme el peligro, por guiarme y aconsejarme siempre de la mejor manera posible, por ser ejemplo a seguir, por luchar junto a mami para vernos crecer a mi gordi y a mí en fin gracias papá. En tercer lugar dedicarle este resultado a la mejor hermana que ojos humanos hayan visto. Gracias mi gordi por estar ahí siempre que lo necesité, por darme todo el cariño que me has dado en estos años, por confiar en tu hermano, por defenderme como una leona defiende a sus cachorritos en cualquier marco que te encontrarás, gracias por ser mi hermanita querida, gracias por competir conmigo y obligarme a seguir superando, hoy que soy ingeniero te digo tu tranquila eres la mejor, como tú no hay dos, un beso te adoro. A mi abuela Olga por traer al mundo a mi madre, por quererme tanto y confiar en mí para tirar de una generación que viene creciendo. Por estar pendiente a mis resultados durante toda la vida, por todo esto y mucho más gracias. Un beso grande abuela.*

## *Agradecimientos*

### **De Pablo Ernesto Miró Martínez**

Quiero agradecerle a mi mamá a pesar de que me dio la vida, por estar siempre conmigo en las buenas y en las malas. A mi papá por sus consejos y sobre todo ser un buen profesor para la vida. A mi hermana por aguantarme en este tiempo y por cuidarme mucho. A mi novia Yeneisy por sus regañones que me sirvieron para ser mejor persona. A mi compañero de causa Lionys por ser un compañero de los de verdad. A mi tutor Vladimir por habernos confiado este trabajo de diploma. A Rasiel por ayudarme cada vez que lo molestaba. A Ronny y Jessica por acogerme en su casa cada vez que iba, fueron como mi familia aquí en la escuela. Al Team de mi chicas lindas Elaine, Elisabeth, Maryeris, Jessica Pedraza, Angela, Mayara, y a todas las que en algún momento me brindaron su ayuda. A mi gente del 2503 que fueron más que compañeros en los 4 años que compartí con ellos. A Lisi, Aray, Lachi, Mare, Idalmis, Diane, Viti, Gera, Pèpe, Piedra en fin todos. Al team de la caliente Blade, El chino, Coli, Alcides, Jorge, Los Manuel, Batman, Fi. A todos los quiero y gracias por haber formado parte de mi vida.

### **De Lionys Julio Ruiz Casanova**

Quiero agradecer a mis educadoras del círculo en especial a Lucia por cuidarme tanto. A mis profesoras de la primaria (Lidia, Anay, Rosali y Omaidá), por enseñarme a razonar. A los vecinos del barrio por estar pendientes a mis resultados. A mis excelentes profesores de la ESPA (Deisy, Lala, Juan Carlos, María del Carmen, Mercedes, Gloria, Valido, Oraldo, Lourdes y Valentín), por prepararme para luego poder llegar hasta aquí. A Victoria por enseñarme un poquito Ruso, pero sobre todo matemáticas. A mi madrina Madelis por su apoyo incondicional. A mis tutores Vladimir y Rasiel por su atención durante todo este tiempo. A mis hermanos de la (Pedro, Leandro, Pepe, Jorge, El Migue, Andy y Lisvet), por compartir momentos buenos y malos, para eso estamos. A mis hermanos Reitel, Andy, Malcolm, Luisito, Moisés, Raulito y Juan Darién señores para lo que deseen aquí estoy. A mis niñas del alma (Dania, Daisy e Irmayrs, Aldeyde y Dalianny) con ustedes hasta que se seque la bahía de Matanzas. A mis colegas del 2505, en especial a (Nurys, Jessica, Yilian, Leydis, Reynaldo, Ronny, Jose Messi), a los 2202 en especial a Mayara y a Yesly. A mis compañeras (Maryeris y Arletys) por quererme tanto. A mi bruji Delia por ser mi compañera de mil batallas, te quiero mucho. A la que más me cela en todo el mundo (Angela) te quiero muchísimo. A Eyidey y Paloma por haberme aceptado como amigo y apoyarme en estos últimos meses. A la niña que le entregué parte de mi vida, gracias por aguantar todas mis malcriadeces y siempre estar ahí cerquita de mí, gracias Elaine, te llevo adentro. A mis tías Estrella, Illa, Mirta, tía solita, Mari, Elena, por darme todo el apoyo y el calor del mundo desde chiquito. A mi abuelo Tatá. A mis tíos Arnaldo, Ernesto, Daniel, Eduardín y Roberto por ser para mí la barrera a 2.45 gracias. A mis tías Mercedes y Bety por siempre estar pendiente a mi persona y darme mucho cariño. A mis primos Yurian, Adianez, Helen, Rey, Elvis, Adrián, Adriana, Arelys, Annalie, Maria Elina, Yailenys, Lorena y Claudio por formar parte de mi vida, los súper quiero. A mi primo Yoelvis por enseñarme que en la vida nada cae del cielo, por ser el sol de mis pensamientos. A mis primos Mamote y Argenis y a mis abuelos Niño y Pura que aunque no estén presentes, se sienten orgullosos de mí. A mi vecino Yoel por confiar en mis resultados. A mi compañero de tesis Pablo, por confiar en mí en esta gran batalla, aquí tienes un hermano. A mis madres Maritza, Yolanda y Lazara por aceptarme como otro hijo en casa a las tres las quiero mucho. A mi novia Eilén por ser tan comprensiva y brindarme tanto amor durante estos años, eres realmente especial en mi vida, por eso estas al final de esta gran lista mi quiqui, porque lo mejor se deja para último un besote mi vida.

## **Resumen.**

Con el creciente ritmo de desarrollo de aplicaciones para dispositivos móviles, el hombre dispone de diversos sistemas de localización, siendo el Sistema de Posicionamiento Global (GPS) por sus siglas en inglés el de mayor utilización, ya que permite conocer la ubicación actual de un dispositivo en cualquier momento y lugar donde el mismo se encuentre. Además posibilita el desarrollo de soluciones que permitan una interacción más cercana entre el usuario y su dispositivo independientemente del entorno geográfico en el que se encuentren.

La Universidad de las Ciencias Informáticas como uno de los eslabones fundamentales en la industria cubana del software, se ha propuesto llevar a cabo una investigación sobre la utilización de los sistemas de información geográfica en el mundo de la telefonía móvil.

Procurando finalmente desarrollar un sistema sobre la plataforma Android capaz de solucionar el problema existente respecto a la falta de información a los usuarios sobre los servicios de transporte público en La Habana y ayudar a la población a transitar por la ciudad. Dicho software denominado Sistema para la Información del transporte público en La Habana (Ruta Habana), el cual tendrá sus principales fundamentos en un algoritmo para la obtención de posibles rutas entre dos puntos a partir del uso de la información del transporte público. Dicho algoritmo debe a partir de la posición actual del usuario, mostrar posibles combinaciones de rutas a abordar por el usuario para llegar al destino seleccionado.

**Palabras clave: Android, GPS, Móvil, Sistema de Información Geográfica, Tecnología.**

# Índice de Contenido

<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN .....</b>	<b>11</b>
1.1	CONCEPTOS FUNDAMENTALES A ANALIZAR DURANTE LA INVESTIGACIÓN .....11
1.2	SISTEMAS DE INFORMACIÓN GEOGRÁFICAS PARA DISPOSITIVOS QUE OPERAN CON ANDROID.....13
1.2.1	<i>OsmAnd</i> .....14
1.2.2	<i>Genius Maps – Offline GPS Navigation</i> .....15
1.2.3	<i>GPS Navigation &amp; Maps Sygic</i> .....15
1.2.4	<i>Transit App: (Metro, Bus, Tren)</i> .....16
1.2.5	<i>Andariego</i> .....16
1.2.6	<i>Metro de Madrid</i> .....17
1.2.7	<i>Metro de Paris</i> .....18
1.2.8	<i>SIG-Rutas</i> .....18
1.3	METODOLOGÍAS DE DESARROLLO.....19
1.3.1	<i>Metodologías Ágiles versus Metodologías Tradicionales</i> .....20
1.3.2	<i>Selección de la metodología de software a utilizar</i> .....21
1.4	HERRAMIENTAS Y TÉCNICAS UTILIZADAS PARA DESARROLLAR LA APLICACIÓN RUTA HABANA .....23
1.4.1	<i>Lenguaje de programación utilizado en el desarrollo</i> .....24
1.4.2	<i>Entorno integrado de desarrollo (IDE) utilizado</i> .....24
1.4.3	<i>Bibliotecas de clases para la visualización de mapas</i> .....26
1.3.1	<i>Mapsforge 0.6</i> .....27
1.4.4	<i>Lenguaje de modelado</i> .....28
1.4.5	<i>Herramienta CASE</i> .....29
<b>CAPÍTULO 2: MODELACIÓN, CONTROL Y SEGUIMIENTO DEL SISTEMA .....</b>	<b>31</b>
2.1	DISEÑO DE LA PROPUESTA DE SOLUCIÓN .....31
2.2	DESCRIPCIÓN DEL MODELO DE DOMINIO.....31
2.2.1	<i>Modelo de dominio</i> .....31
2.2.2	<i>Conceptos fundamentales del dominio</i> .....32
2.3	FUNCIONALIDADES DE RUTA HABANA PARA DISPOSITIVOS QUE OPERAN CON ANDROID .....33
2.3.1	<i>Requerimientos funcionales</i> .....34
2.3.2	<i>Requerimientos no funcionales</i> .....35
2.4	ESTÁNDARES DE CODIFICACIÓN SEGÚN LA METODOLOGÍA XP.....36
2.4.1	<i>Comentarios</i> .....37
2.4.2	<i>Declaración de variables</i> .....37
2.4.3	<i>Métodos</i> .....38
2.4.4	<i>Sentencias</i> .....38

2.5 FASE DE PLANIFICACIÓN.....	39
2.5.1 Historias de Usuarios.....	39
2.5.2 Estimación de esfuerzo.....	41
2.5.3 Plan de iteraciones.....	42
2.5.4 Plan de entrega.....	43
<b>CAPÍTULO 3: “ITERACIONES Y PRODUCCIÓN” .....</b>	<b>45</b>
3.1 ARQUITECTURA DE SOFTWARE.....	45
3.2 ESTILO ARQUITECTÓNICO. PATRÓN DE ARQUITECTURA.....	46
3.3 DISEÑO DE LOS ALGORITMOS UTILIZADOS EN LA SOLUCIÓN.....	47
3.4 PATRONES DE DISEÑO UTILIZADOS EN LA IMPLEMENTACIÓN .....	48
3.4.1 Patrones GRASP.....	49
3.4.2 Patrones GoF (The Gang of Four).....	50
3.5 DISEÑO DE LA BASE DE DATOS PARA LA APLICACIÓN RUTA HABANA.....	52
3.6 TAREAS DE INGENIERÍAS REALIZADAS EN EL DESARROLLO.....	54
3.6.1 Primera Iteración.....	54
3.6.2 Segunda Iteración.....	55
3.6.3 Tercera Iteración.....	57
<b>CAPÍTULO 4: “VERIFICACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA” .....</b>	<b>59</b>
4.1 FASE DE PRUEBAS AL SISTEMA.....	59
4.2 PRUEBAS UNITARIAS O PRUEBAS DE UNIDAD.....	59
4.2.1 Primera iteración.....	59
4.2.2 Segunda iteración.....	60
4.2.3 Tercera iteración.....	61
4.3 PRUEBAS DE ACEPTACIÓN.....	61
4.4 IMPACTO DE LA SOLUCIÓN EN EL CAMPO CIENCIA, TECNOLOGÍA Y SOCIEDAD (CTS) .....	62
<b>CONCLUSIONES GENERALES.....</b>	<b>65</b>
<b>BIBLIOGRAFÍA .....</b>	<b>67</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>69</b>
<b>ANEXOS.....</b>	<b>72</b>

# Introducción

## Introducción

El importante despliegue que han tenido las tecnologías de la información y las comunicaciones (TIC) hoy en día supone una visión diferente de la forma en que el ser humano se comunica con su entorno social, a tal punto que la distancia o ubicación geográfica ya no es un impedimento determinante. Con el continuo avance de las tecnologías y gracias a la masiva colaboración de usuarios es posible que se tenga al alcance cualquier información que quizás en épocas anteriores era imposible. Así también es importante tomar en cuenta que en la actualidad el tema de la movilidad cada vez toma más fuerza y conjuntamente con ello las aplicaciones que le permiten al usuario realizar varias acciones como (chat, llamadas, ocio, situaciones de emergencia, avisos, mensajería, geoposicionamiento entre otros), manteniendo presente dos aspectos importantes la portabilidad y la facilidad de uso indistintamente del sistema operativo sobre el cual interactúen estas aplicaciones.

Un sistema operativo móvil se define como el conjunto de programas cuya misión fundamental es la de gestionar los recursos del dispositivo y en consecuencia constituirá el soporte lógico que controla el funcionamiento del equipo físico. Enfocándose mayormente en la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes formas de manejar la información en ellos (1).

Entre los sistemas operativos móviles más utilizados en el mundo se encuentran Symbia OS, Windows Phone, BlackBerry OS, IOs y Android el cual ha ido ganando en la preferencia de los usuarios en los últimos años (2), de manera que se encuentra encabezando la lista con un 47.45% en el mercado de los sistemas operativos móviles a nivel mundial, desplazando de esta forma a su gran competencia (IOs) el cual ha descendido de un 54% a un 42.59% en el último año, evidenciándose de esta forma la fuerza de la plataforma de la compañía Google Inc (3).

Android es una plataforma de Software basada en el núcleo de Linux. Diseñada en principio para dispositivos móviles, la misma permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google Inc mediante el lenguaje de programación Java. Es una solución completa de código abierto bajo la licencia permisiva de Software Apache, la misma brinda la posibilidad al desarrollador de crear y desarrollar aplicaciones escritas con lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de Android).

En el desarrollo de las aplicaciones para dispositivos móviles, una de las categorías que ha ido revolucionando en los últimos años es la de los sistemas de información geográfica (SIG), teniendo en cuenta que la gran vinculación de la tecnología móvil a los sistemas de información geográfica establece una fuerte relación sistema-usuario, posibilitando el intercambio de información y una integración continua y rápida entre sistemas.

En Cuba, a pesar de las limitaciones por las cuales atraviesa el país, fundamentalmente en el desarrollo tecnológico, no se han disminuido las fuerzas, por el contrario se ha ido poco a poco avanzando en tan importante renglón como lo es hoy a nivel mundial la esfera de la informática y las telecomunicaciones,

# Introducción

apoyándose principalmente en la preparación de sus profesionales y en la alta dirección del país. Disímiles son las aplicaciones que se han venido desarrollando en los últimos años relacionadas con el estudio de los sistemas de información geográficas en el país, debido a que estos sistemas son una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográfica referenciada, con el fin de resolver problemas complejos de planificación y gestión geográfica.

## **Situación problemática:**

El transporte de pasajeros es componente imprescindible del proceso de actualización del modelo económico cubano y por ello ha sido refrendado en los Lineamientos de la Política Económica y Social del Partido aprobados en el VI Congreso. Específicamente en el Capítulo X Política para el Transporte, lineamientos (269, 283, 272, 278, 280, 281, 283 y 284), los cuales están referidos al transporte de pasajeros de una forma u otra.

Actualmente el Ministerio de Transporte de Cuba cuenta con un sitio web, en el cual se muestra en una tabla todas las rutas que existen en la ciudad de La Habana; pero no cuenta con un sistema que permita manejar y brindar toda la información referente a: qué ruta puede coger el usuario en un punto determinado de la ciudad, qué parada de ómnibus está más cerca de su ubicación actual, la distancia del recorrido de la ruta que abordó, ubicación de las terminales, las avenidas por las cuales transitan los ómnibus entre otras cuestiones relacionadas con la información solicitada por el viajero.

En La Habana, en el año 2013 se aprobó por parte del Consejo de Ministros una propuesta para el ordenamiento del transporte en la capital (4). Sin embargo, aún existen problemas que haciendo un uso correcto de las tecnologías alineadas con los servicios que brindan los sistemas de información geográfica pudieran ir disminuyendo gradualmente, ejemplo de esto es el problema que existe con la información a los usuarios referente a recorridos, paradas y horarios que a diario utilizan la red de ómnibus en todo el país, principalmente en su capital.

Cuba no está excepta de la introducción de tecnologías de punta para el sector que tributa a las telecomunicaciones y medios a fines, en tanto la inserción de herramientas de interconexión dígame bluetooth, wifi, GPRS, GPS entre otras son cada vez más utilizadas. Tecnologías de uso común por la población mundial como, ordenadores portátiles, tabletas inteligentes, teléfonos inteligentes, relojes, televisores entre otros, los cuales operan sobre diferentes sistemas operativos se han ido adentrando en el país en los últimos años, permitiéndole al usuario utilizar múltiples servicios con solo contar con uno de estos terminales. Por lo general en la isla, la mayor parte de estos dispositivos operan sobre la plataforma Android, según los resultados obtenidos al procesar la información arrojada por una encuesta, aplicada a 100 personas (50 de sexo femenino y 50 del masculino) en puntos de zonas wifi (La Lisa, Vedado y Centro Habana) y paradas de ómnibus de la capital, la cual fue elaborada por el

# Introducción

equipo de trabajo de la presente investigación, con preguntas por lo general cerradas, directas y abiertas.

Además actualmente Android es el sistema operativo de la mayor parte de los dispositivos móviles que tiene a la venta el único proveedor de servicios y telefonía móvil (ETECSA) del país. No obstante al crecimiento de estas tecnologías en Cuba y a la masividad de aplicaciones con las que hoy estos terminales suelen contar, sin embargo dichos terminales no poseen un sistema capaz de suministrar información detallada sobre las rutas e itinerarios de los ómnibus públicos que circulan en la capital.

A raíz de todo lo que se ha investigado respecto al tema del transporte público capitalino hasta el año 2015, el equipo de trabajo de esta investigación, decidió recorrer la ciudad para detectar posibles inconvenientes en cuanto al ordenamiento del transporte capitalino, además aplicar una encuesta inicial a la población que hace uso de la red de transporte de la ciudad, con el objetivo de conocer los criterios de la misma sobre las dificultades y deficiencias que hoy presenta el transporte en la provincia en general. Después de haber culminado el recorrido y haber aplicado la encuesta se pudo concluir que el ordenamiento del transporte en la ciudad de La Habana presenta dificultades evidenciándose fundamentalmente en:

- ✓ La no existencia de un sistema de información sobre las rutas e itinerarios de los ómnibus.
- ✓ Paradas de ómnibus con señalización deficiente y falta de información requerida para los pasajeros.
- ✓ La no existencia de tablillas informativas en las terminales sobre el horario de los recorridos de los ómnibus.
- ✓ Los ómnibus por lo general, no cuentan con medios de comunicación visual para mostrar las avenidas por las cuales circulan durante su recorrido.
- ✓ No existen mapas sobre los cuales la población pueda apoyarse para determinar las rutas a abordar para llegar su destino.

A estos problemas anteriormente mencionados, se le adiciona las pérdidas monetarias para el sector del transporte público debido, a que la población se apoyaría más en el sector no estatal, trayendo como consecuencias que la mayor parte del dinero que las personas destinaran para el transporte quedase fuera del sector estatal, cuya influencia afectaría la economía del organismo y por consiguiente la del país; pudiendo ser empleado este recurso monetario para la inversión en otro tipo de artefactos (neumáticos, motores, puertas entre otros), cuya utilidad sea más apremiante para el transporte en la provincia.

A partir de todo lo anteriormente planteado surge como **problema de investigación**: ¿Cómo facilitar la gestión de la información sobre los itinerarios y las rutas de ómnibus públicos en La Habana a la población que usa dispositivos móviles que operan sobre Android? Para llevar a cabo la investigación se tiene como **objeto de investigación**: Los Sistemas de Información Geográfica para dispositivos

# Introducción

móviles, por lo que se define como **objetivo general**: Desarrollar una aplicación que gestione la información de las rutas e itinerarios del sistema de transporte público en La Habana sobre la plataforma Android. Quedando enmarcada la investigación en el **campo de acción**: Los Sistemas de Información Geográfica para la gestión de itinerarios y rutas del transporte público en la capital cubana sobre dispositivos móviles.

Con el fin de alcanzar el objetivo planteado se propuso desarrollar las **tareas investigativas** que se plantean a continuación:

- ✓ Elaboración del marco teórico de la investigación para definir conceptos fundamentales que ayuden a la comprensión de la misma.
- ✓ Realización del análisis del estado del arte de los sistemas de información geográfica para establecer semejanzas con las soluciones existentes e identificar características o funcionalidades similares a las del sistema a implementar.
- ✓ Análisis de la metodología de desarrollo a utilizar para la descripción de los artefactos que se generan durante la implementación de la aplicación Ruta Habana para dispositivos que operan con sistema operativo Android.
- ✓ Selección de las tecnologías y herramientas necesarias para el desarrollo de la aplicación Ruta Habana para dispositivos que operan con sistema operativo Android.
- ✓ Diseño y descripción de algoritmos para obtención de paradas cercanas a la posición del usuario y la sugerencia de rutas a abordar para llegar a un destino seleccionado.
- ✓ Análisis de los métodos de pruebas existentes para la validación del sistema a partir de pruebas unitarias y de aceptación.

Con el propósito de lograr una mayor organización en la realización de la presente investigación se recurre a los siguientes **métodos científicos**:

**Métodos teóricos**: Permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis (5).

**Analítico – Sintético**: Este permitió el análisis de los procesos, documentos y teorías para la extracción de los elementos más importantes que se relacionan con los sistemas de información geográfica enfocados al desarrollo en Android. Además, para buscar y analizar la información acerca de las tecnologías, metodologías y herramientas a utilizar en el desarrollo del sistema, seleccionando los principales elementos y características.

# Introducción

**Inducción-deducción:** Permitió analizar por separadas las funcionalidades de las aplicaciones móviles, tanto a nivel mundial como nacional desarrolladas para el transporte, para luego determinar que de manera general ninguna satisface las necesidades existentes.

**Histórico - Lógico:** Posibilitó el estudio analítico de la trayectoria histórica de los sistemas de información geográfica que han sido desarrollados para dispositivos inteligentes (móviles, tabletas, entre otros.) en el último quinquenio y facilitó conocer de forma general el funcionamiento y desarrollo de los mismos, permitiendo caracterizar así cada uno en sus aspectos más externos.

**Genético:** Consiste en determinar el campo de acción elemental que se convierte en célula del objeto. En dicha célula están representados todos los componentes del objeto, así como sus leyes más trascendentales (5). La utilización de este método permitió delimitar el campo de acción de la presente investigación, así como deducir y explicar el desarrollo de sistemas más complejos.

**Modelación:** La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad. El modelo como sustituto del objeto de investigación es semejante a él, existiendo una correspondencia objetiva entre el modelo y el objeto, siendo el investigador quien elabora dicho modelo. El modelo es el eslabón entre el sujeto y el objeto intermedio (5). Este método permitió realizar el proceso mediante el cual se crean criterios y alternativas para la propuesta de solución, facilitando la definición de los componentes de la misma así como sus relaciones. Además fue empleado para la realización de los artefactos que requiere la metodología seleccionada: Historias de Usuarios (HU), Tarjetas CRC (Clase, Responsabilidad, Colaborador), Tareas de la Ingeniería, Plan de Iteraciones, Plan de Entregas entre otras.

**Métodos empíricos:** Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional (5).

**Observación:** Se realizó una observación de las diferentes aplicaciones móviles existentes tanto en Cuba como a nivel mundial. Específicamente en aquellas que operan sobre la plataforma Android, destinadas a brindar información al usuario sobre como trasladarse por la ciudades, haciendo uso del transporte público para determinar si algunas de las soluciones existentes daría respuesta al problema en cuestión.

**Métodos particulares:** Son métodos más específicos que están desarrollados en base a las características propias de cada ciencia y para su aplicación están vinculados a técnicas de recolección de datos característicos de ese tipo de investigación. A continuación se describen algunos de estos métodos aplicados en las ciencias sociales (5).

**Entrevista:** La entrevista puede ser individual o colectiva, en ambos casos el entrevistador debe realizar una preparación previa, sobre el tema a tratar, y elaborar una guía para su desarrollo (5). El equipo de trabajo realizó entrevistas no estructuradas a expertos que laboran en las terminales de la

# Introducción

capital para obtener información actualizada sobre los procesos que se estaban llevando a cabo para el mejoramiento del transporte público en la ciudad. La utilización de este método permitió la obtención de información necesaria para el desarrollo de los requerimientos del sistema.

**Encuesta:** Se realiza cuando la información que se necesita puede ser obtenida a partir de la respuesta que una persona o varias puedan dar a un cuestionario preelaborado, y las mismas están dispuestas a colaborar con la investigación (5). Este método ayudó a respaldar la investigación a través de un muestreo estratificado el cual tiene su “fundamento en subdividir una población heterogénea en una serie de subpoblaciones homogéneas para garantizar que todas las características de la población heterogénea estén representadas en la muestra” (5), dicha encuesta fue aplicada a 100 personas en puntos de zonas wifi y paradas de ómnibus en la capital con el fin de conocer cuantitativamente opiniones y sugerencias sobre el transporte capitalino. Para ver el contenido de la encuesta ver **Anexo VI**.

## **Análisis e Interpretación de los resultados de la encuesta.**

A continuación se hará un análisis de los resultados arrojados por las encuestas realizadas a personas que se encontraban en los puntos de zona wifi y paradas de la capital.

### **Género**

**Análisis cuantitativo:** De los datos obtenidos tenemos que del 100% de los encuestados, el 50% pertenecen al sexo masculino, y el 50% restante pertenecen al sexo opuesto, dando como resultados que no existe diferencia entre los géneros.

**Análisis cualitativo:** Como resultado de la aplicación de las encuestas, podemos decir que la mitad de las personas encuestadas pertenece al género masculino, por lo que queda claro que la otra parte pertenece al género femenino, quedando un porcentaje muy equilibrado entre los géneros.

### **Preguntas**

#### **1. ¿Con qué frecuencia utiliza usted un ómnibus público como medio de transporte para trasladarse en la ciudad?**

**Análisis cuantitativo:** Mediante los resultados obtenidos en esta pregunta, podemos analizar que la mayoría de los encuestados los cuales representan el 66% afirman que hacen uso del transporte público a diario; el 28% afirma que la frecuencia con que hacen uso del transporte está entre las 2 y 3 veces en semana; y por último el 6% no hace uso del transporte público en la semana.

**Análisis cualitativo:** Con esta interrogante se puede observar que la mayoría de personas encuestadas se trasladan dentro de la ciudad haciendo uso de los ómnibus públicos, y un importante porcentaje, sobre todo los hombres, hacen uso de la red de transporte capitalina, lo que puede ser tomado como una fortaleza para el presente proyecto, ya que este público objetivo, puede ser orientado a utilizar la aplicación propuesta durante sus traslados en la ciudad.

# Introducción

## 2. ¿Conoce usted los itinerarios completos de los ómnibus en los que acostumbra a viajar?

**Análisis cuantitativo:** De los datos obtenidos en esta interrogante, podemos analizar que del total de los encuestados, el 50% conoce los itinerarios completos de los ómnibus en los que acostumbran a moverse por la ciudad; el 35% desconoce de los recorridos completos, de las rutas que usa frecuentemente; y el 15% afirman que solo conocen el recorrido de algunas de las rutas.

**Análisis cualitativo:** De los resultados obtenidos en esta pregunta podemos observar que la mitad de los encuestados si conocen los itinerarios completos de las rutas que utilizan ya que viven en esta provincia, pero las personas que provienen de otras provincias del país no los conocen, lo que deja abierta la posibilidad para que en el sector se pueda promocionar la aplicación a los visitantes y por lo tanto generar información detallada para la comunidad.

## 3. ¿Cree usted que las paradas de ómnibus poseen la información correcta o necesaria, para ubicar al medio de transporte que debe abordar?

**Análisis cuantitativo:** Mediante los resultados obtenidos en esta pregunta, podemos analizar que la mayoría de los encuestados los cuales representan el 97% afirman que las paradas de ómnibus no poseen la información correcta o necesaria para ubicar el medio de transporte para sus viajes en la ciudad y solo el 3% plantea que la información brindada hasta el momento es suficiente.

**Análisis cualitativo:** Con esta interrogante se puede observar que casi el total de personas encuestadas, afirman que la información mostrada en las paradas hasta este momento no es correcta para ubicar los medios del transporte público. Por lo que queda claro que se hace necesario poner en manos de la población esta información, para así lograr disminuir las inconformidades de la comunidad.

## 4. ¿El ómnibus en el que usted acostumbra a viajar, tiende a llegar en un horario específico?

**Análisis cuantitativo:** Mediante los resultados obtenidos en esta pregunta, podemos analizar que la mayoría de los encuestados los cuales representan el 70%, afirman que los ómnibus no llegan en un horario específico; el 20% plantea que a veces si pasan en determinado horario y solo un 5% de la población encuestada, afirma que los ómnibus en los cuales acostumbra a viajar si pasan en un mismo horario.

**Análisis cualitativo:** de los resultados obtenidos en esta pregunta podemos observar que la séptima parte de los encuestados en la mayoría de los casos, no tienen un ómnibus en el cual se asegura el horario del recorrido, aunque un quinto de la población afirme que no sucede lo mismo, no quita la posibilidad, de lo virtuoso que sería que todas las rutas tuvieran horarios fijos y que a su vez se cumpliera con estos.

## 5. ¿Alguna vez, al utilizar un ómnibus público como su medio de transporte, tomó uno equivocado y tuvo inconvenientes por la falta de conocimiento e información?

# Introducción

**Análisis cuantitativo:** mediante los resultados obtenidos en esta pregunta, podemos analizar que el 77% de la población encuestada ha presentado dificultades durante sus traslados en la ciudad, por la falta de información, en tanto el 23% de las personas, no se ha visto afectada por esta causa.

**Análisis cualitativo:** si vamos a los resultados arrojados por esta interrogante, podemos observar que la información que brinda hoy transporte provincial, respecto a las trayectorias tomadas por los ómnibus, pone a la población en situaciones poco agradables. Evidenciándose que si la investigación en cuestión, cumple con los objetivos propuestos entonces, problemas de esta índole podrán desaparecer.

## 6. ¿Considera usted que la infraestructura y la comunicación visual (señalizaciones) de las paradas de ómnibus es suficiente?

**Análisis cuantitativo:** con los datos obtenidos podemos observar que solo un 7% de la población considera que la infraestructura y la comunicación visual en las paradas es suficiente; mientras que un 18% declara que es insuficiente. Pero el mayor porcentaje, el cual representa el 75% en esta pregunta, estuvo en la respuesta muy insuficiente.

**Análisis cualitativo:** los resultados obtenidos en esta pregunta, demuestran que existen dificultades con la comunicación visual, en las paradas de la capital cubana, problema que de seguir comportándose de esta manera, traería descontento tanto en las personas que viven en la ciudad, como en los visitantes. Poniendo en riesgo la imagen de la ciudad sobre todo al público extranjero que visita el país.

## 7. ¿Es usted poseedor de un Smartphone, o llamado también teléfono inteligente?

**Análisis cuantitativo:** mediante los resultados obtenidos en esta pregunta, podemos analizar que la mayoría de los encuestados los cuales representan el 97% afirman que cuentan con tecnologías con dichas características, y solo la minoría de la muestra analizada la cual representa el 3%, no dispone de un teléfono inteligente.

**Análisis cualitativo:** los resultados obtenidos en esta pregunta, demuestran el ascenso de las tecnologías en la capital y en el país en general, también queda evidenciado que el uso de los dispositivos inteligentes ha crecido perceptiblemente. Esto abre las puertas a la investigación, ya que el desarrollo de una aplicación para estos terminales, podrá ser utilizada por los usuarios.

## 8. ¿Qué sistema operativo utiliza su dispositivo inteligente?

**Análisis cuantitativo:** mediante los resultados obtenidos en esta pregunta, podemos analizar que la mayoría de los encuestados, utilizan dispositivos inteligentes, los cuales operan sobre distintos sistemas operativos. La pregunta arrojó los siguientes resultados, el 68% de los dispositivos opera sobre la plataforma Android, el 23% opera sobre la plataforma iOS, el 5% opera sobre Window Phone y el 4% sobre otras plataformas.

# Introducción

**Análisis cualitativo:** esta pregunta es de suma importancia para la investigación, ya que muestra los sistemas operativos más utilizados por los dispositivos inteligentes con los que hoy cuenta la población capitalina, además demuestra que Android es la plataforma más utilizada por estos terminales, acápíte que se debe explotar durante la investigación, ya que se tiene como objetivo de la misma desarrollar una aplicación sobre la plataforma más utilizada por estos días a nivel mundial y de esta forma proveer un sistema capaz de suministrar información sobre las rutas e itinerarios del transporte capitalino a los usuarios que cuentan con terminales móviles cuyo sistema operativo es Android.

**9. ¿Utilizaría usted una aplicación móvil; que le permita informarse sobre las rutas, tiempo de espera, costo y ubicación geográfica de las paradas de ómnibus más cercanas a su ubicación?**

**Análisis cuantitativo:** esta interrogante es de mucho interés para el equipo de trabajo, porque permitirá validar la necesidad de desarrollar una aplicación que brinde toda la información detallada sobre las rutas, costo y ubicación de las paradas cercanas a la posición de un usuario. La pregunta deja claro que la mayor parte del total de personas encuestadas, la cual representa el 95%, está de acuerdo en utilizar un sistema que suministre toda esta información anteriormente descrita; y solo el 5% afirmó que no la utilizaría.

**Análisis cualitativo:** Los resultados obtenidos en esta pregunta, muestran la importancia que le atribuyen los usuarios a una aplicación de este tipo, debido a que con solo instalarla en sus dispositivos podrán contar con una información actualizada de los recorridos, costos y ubicación tanto de paradas, como de terminales de ómnibus. Mejorando gradualmente la satisfacción de la población con los servicios que brinda el transporte público.

Luego de haber analizado cuantitativa y cualitativamente los resultados de las encuestas se describirá la estructura de la investigación para una mejor comprensión de la misma:

**Capítulo 1: Fundamentación teórica de la investigación.** En este capítulo se hará un análisis del estado del arte de los Sistemas de Información Geográfica, así como de las principales aplicaciones para dispositivos móviles que hacen uso del GPS para ayudar a los usuarios a trasladarse en las ciudades. Además se analizarán los principales elementos teóricos entre los que se encuentra la metodología de desarrollo para el sistema, se detallarán y analizarán las herramientas y tecnologías que se utilizan en el proceso de desarrollo de un SIG, se conformará la propuesta de tecnologías y herramientas a utilizar y se expondrán los elementos analizados, como criterios de selección a la hora de elegir entre todas las opciones posibles.

**Capítulo 2: Modelación, control y seguimiento del sistema Ruta Habana.** En este capítulo se abordará el entorno de desarrollo definido, se explicarán las diferentes características que va a presentar el sistema a implementar y sus funcionalidades siguiendo los aspectos que plantea la metodología ágil XP en sus fases de Exploración y Planificación.

# Introducción

**Capítulo 3: Iteraciones y Producción del sistema.** En este capítulo se puntualizarán las iteraciones llevadas a cabo durante la etapa de construcción del sistema, definiéndose la arquitectura del mismo, los patrones de diseños utilizados, las tarjetas Clase Responsabilidad Colaboradores (CRC), las tareas de ingenierías generadas por cada historia de usuario (HU). Además se describirán los algoritmos de mayor relevancia para la investigación, diseñados por el equipo de desarrollo.

**Capítulo 4: Verificación y Validación de la solución propuesta.** En este capítulo se documentarán las fases de pruebas según lo propuesto por la metodología de desarrollo XP. Las pruebas llevadas a cabo fueron: pruebas unitarias y de aceptación respectivamente. Para de esta forma demostrar que el sistema desarrollado cumple con todas las funcionalidades definidas en las fases anteriores, quedando listo para ser entregado y posteriormente desplegado para su utilización por parte de los usuarios. También se hace una valoración del impacto de la aplicación propuesta en el campo ciencia, tecnología y sociedad (CTS) por sus siglas.

### **Capítulo 1: Fundamentación teórica de la investigación**

En este capítulo se enunciarán los conceptos fundamentales referentes a un SIG y a sus componentes. Se llevará a cabo un análisis de las diferentes aplicaciones informáticas existentes tanto en Cuba como a nivel mundial, dedicadas a los SIG donde el objetivo principal de las mismas sea, brindarles información a los usuarios sobre cómo llegar a un lugar en un momento determinado. Se describirán las características del equipo de desarrollo. Se fundamentarán las herramientas y tecnologías a utilizar durante todo el desarrollo del software. Además quedarán expuestas las razones de la selección de la metodología de desarrollo a utilizar.

#### **1.1 Conceptos fundamentales a analizar durante la investigación**

A continuación se enuncian los principales conceptos relacionados con un sistema de información geográfica para ayudar a una mejor comprensión de la temática abordada en el trabajo.

**Tecnología:** La tecnología es una red que abarca los más diversos sectores de la actividad humana, un modo de vivir, de comunicarse, de pensar, un conjunto de condiciones por las cuales el hombre es dominado ampliamente, mucho más que tenerla a su disposición (6).

**Sistema informático:** Un sistema informático es un conjunto de partes o recursos formados por el hardware, software y las personas que lo emplean, que se relacionan entre sí para almacenar y procesar información con un objetivo en común (7).

**Teléfonos inteligentes:** Los teléfonos inteligentes (Smartphone) son teléfonos móviles que incorporan características de una computadora personal. Suelen permitir al usuario instalar nuevas aplicaciones, aumentando así sus funcionalidades. Esto es posible porque ejecutan un sistema operativo potente de fondo. Pueden incluir un mini teclado, una pantalla táctil, un lápiz óptico, etc. Además tienden a incluir acceso a internet, servicios de e-mail, cámara integrada, navegador web, procesador de textos, entre otras muchas aplicaciones (8).

**Sistema de información geográfica:** Un Sistema de Información Geográfica (SIG) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica (9).

Dicha tecnología puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, etc.

En el año 1962 se vio la primera utilización real de los SIG en el mundo, concretamente en Ottawa (Ontario, Canadá) y a cargo del Departamento Federal de Silvicultura y Desarrollo Rural. Desarrollado por Roger Tomlinson, el llamado Sistema de Información Geográfica de Canadá fue utilizado para

## Capítulo 1: Fundamentación teórica de la investigación

almacenar, analizar y manipular datos recogidos para el Inventario de Tierras Canadá; una iniciativa orientada a la gestión de los vastos recursos naturales del país con información cartográfica relativa a tipos y usos del suelo, agricultura, espacios de recreo, vida silvestre, aves acuáticas y silvicultura. Se añadió, así mismo, un factor de clasificación para permitir el análisis de la información (9).

### **Componentes de un SIG**

Para lograr el funcionamiento de un SIG se debe contar con ciertos componentes como son:

**Hardware:** El hardware a emplear varía desde servidores para guardar la información así como computadoras conectadas a la red o individuales para capturar los datos. Para lograr un funcionamiento correcto dentro de una organización se debe utilizar hardware específico que solventa las necesidades de la aplicación (velocidad, costo, seguridad, etc.) (10).

**Software:** Los principales componentes que los SIG proporcionan son los Sistemas de Manejo de Base de Datos (SMBD), la Interfaz Gráfica de Usuarios (GUI) por sus siglas en inglés para facilitar el acceso a las herramientas, instrumentos de captura y manejo de información geográfica y los de soporte de consultas, análisis y visualización de datos geográficos (10).

**Datos:** Son el componente más importante de un SIG. Los datos geográficos y alfanuméricos pueden obtenerse de varias formas, una de ellas es a través de los propios recursos y otra mediante otros proveedores (10).

**Personal:** Se hace necesario contar con personal experto que garantice la actualización de la información, de esta forma, el SIG no pierde valor y no se cometen errores que pueden afectar a otros individuos (10).

**Métodos:** Para lograr la exitosa implementación de un SIG hay que tener en cuenta un buen diseño y reglas bien definidas.

Algunas de las ventajas que brindan los SIG son (10):

1. La sustitución del mapa impreso en papel al utilizar los mapas digitales.
2. La manipulación de elementos de un mapa, relacionando cada objeto con una información más amplia.
3. El análisis matemático y salidas gráficas visualizando resultados.
4. La generación de muchos mapas con los mismos datos.

El campo de aplicación de los SIG se hace cada vez mayor, llegándose a utilizar en muchas de las circunstancias de la vida cotidiana. La arqueología, la cartografía, la planificación urbana, la evaluación de las redes de servicio y el transporte son algunas de las ramas donde pueden ser útiles estos sistemas.

### **Sistema de Posicionamiento Global**

El sistema de posicionamiento global (GPS) es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. Fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

Los sistemas GPS (Global Positioning System) por sus siglas en inglés, hacen uso de veinticuatro satélites a cincuenta y cinco grados sobre la línea del Ecuador que se encuentran girando en una órbita alrededor de la tierra, además utilizan seis direcciones diferentes para rodearla, dando cada uno aproximadamente dos vueltas diarias a la tierra. Cada satélite pasa continuamente enviando información (posición del satélite, hora del mensaje, hora de la zona horaria donde se encuentra) a los dispositivos GPS, con esta información los dispositivos determinan la distancia a la que se encuentran de la tierra. Si se realiza el procedimiento anterior con tres satélites o más, conociendo las intersecciones (triangulación) de las esferas generadas con centro en los satélites, se podrá determinar con exactitud la posición en el globo terráqueo (11).

### **1.2 Sistemas de información geográficas para dispositivos que operan con Android**

El sistema operativo Android surge como resultado de la Open Handset Alliance un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil y con el compromiso de comercializar dispositivos móviles con este sistema operativo. El desarrollo viene avalado principalmente por Google (tras la compra de Android Inc. en 2005) y entre las compañías se pueden encontrar: de software, operadores, fabricantes de móviles (Motorola, Samsung, LG, HTC, Huawei y otros) o fabricantes de Hardware (12).

Las aplicaciones para Android se escriben y desarrollan en Java aunque con unas interfaces de programación de aplicaciones propias o APIS como son conocidas comúnmente, por lo que las aplicaciones escritas en Java para computadoras personales o PC's (acrónimo de las palabras inglesas Personal Computer) y demás plataformas ya existentes no son compatibles con este sistema. Estas se ejecutan en Dalvik, una máquina virtual Java desarrollada por Google específicamente para ser utilizada en los dispositivos que operan con Android con un conjunto de características que la diferencian del resto de las máquinas virtuales Java (12).

En la actualidad, con la creciente influencia de las tecnologías informáticas, existen varios sistemas a nivel internacional que tienen entre otras funcionalidades, brindar al usuario la información necesaria para transitar por las ciudades, ya sea mediante la utilización de medios particulares o del transporte público con el cual cuenta la misma, en dispositivos que operan con el sistema operativo Android. El uso de estos sistemas ha aumentado de forma exponencial en los últimos dos años, y como

consecuencia, han pasado del total desconocimiento a la práctica cotidiana en el mundo de los negocios, en las universidades y en los organismos gubernamentales.

A continuación se brinda una muestra de soluciones de sistemas que se utilizan para llevar a los usuarios la información necesaria para trasladarse por las ciudades; teniendo en cuenta que en el momento de realizar la investigación estas aplicaciones eran las más completas en cuanto a funcionalidades. Además de ser los de más demanda por la población que hace uso de los mismos.

### **1.2.1 OsmAnd**

OsmAnd es una aplicación, que permite el uso de un mapa en cualquier lugar que lo necesite sin tener una conexión de datos. La misma cuenta con el enrutamiento en línea y la navegación guiada, es de código abierto y está siendo desarrollada activamente. Todo el mundo puede contribuir a la aplicación para informar fallos, mejorar las traducciones, o la codificación de nuevas características. El proyecto se encuentra en un estado muy animado, de mejora continua por todas estas formas de desarrollo y la interacción del usuario. El avance del proyecto también se basa en las contribuciones financieras para financiar el desarrollo, codificación, y prueba de nuevas funcionalidades. OsmAnd ofrece enrutamiento, con el guiado óptico y la voz, un coche, bicicleta y peatones. Las principales funcionalidades operan tanto en línea como fuera de línea. Los datos fuera de línea se basan en OpenStreetMap - el mapa wiki libre (OSM) (13).

#### **Principales características (13)**

- ✓ Realiza búsquedas de lugares de dirección, según el tipo (restaurante, hotel, gasolinera, museo), o mediante coordenadas geográficas.
- ✓ Permite visualizar la posición y orientación en el mapa.
- ✓ Muestra diferentes superposiciones como pistas GPX gira / navegación y mapas adicionales con la transparencia personalizable.
- ✓ Visualiza opcionalmente el límite de velocidad, en caso de que se excedan notifica de esto al usuario.
- ✓ Los mapas incluyen pie, senderismo y rutas en bicicleta, ideal para actividades al aire libre.
- ✓ Encaminamiento y modos de visualización especiales para ciclistas y peatones.
- ✓ Muestra de forma opcional las paradas de transporte público (autobús, tren), incluyendo los nombres de las líneas.
- ✓ Calcula la distancia y el tiempo aproximado existente entre puntos determinados por el usuario.
- ✓ Tiene cobertura de mapas muy aproximadas y con buena calidad en muchos países del mundo principalmente en Europa occidental.

## **1.2.2 Genius Maps – Offline GPS Navigation**

Genius Maps – Offline GPS Navigation es una aplicación gratuita para dispositivos móviles. Tiene entre sus principales funcionalidades planificación de rutas, la exploración y la navegación fuera de línea por lo que no requiere de una conexión a Internet móvil para buscar y navegar. Todos los mapas se almacenan en el dispositivo, por lo que no hay costes de itinerancia y su batería dura más tiempo. Genius Maps ofrece una prueba gratuita de 7 días totalmente funcional y de tráfico en vivo que se puede activar en cualquier momento. Además brinda la posibilidad de optar por actualizar la orientación pro y el tráfico en tiempo real durante o después del juicio o, simplemente, seguir utilizando los mapas gratuitos de nivel profesional para la planificación de rutas y explorar con la navegación a pie sin ningún costo (14).

### **Principales características (14)**

- ✓ Permite la descarga de mapas libremente, removido y redefinidos tantas veces como se desee e incluso después de completar el ensayo.
- ✓ Orientación pro, esta realiza notificaciones paso a paso como: instrucciones de voz en el idioma seleccionado, desvíos automáticos, alertas de límites de velocidad, puntos de interés a lo largo de las actualizaciones de rutas y mapas ilimitados para una región seleccionada.
- ✓ Calcula tráfico en tiempo real permitiendo conocer acerca de la congestión del mismo, obras viales y ocurrencia de accidentes, y luego vuelve a enrutar su viaje de forma automática.
- ✓ Tiene cobertura de mapas muy aproximadas y con buena calidad en varios países del mundo.
- ✓ Requiere de una versión Android: 2.2 o superior y su tamaño es 23 Mega Byte.

## **1.2.3 GPS Navigation & Maps Sygic**

GPS Navigation & Maps Sygic es la aplicación de navegación sin conexión más instalada del mundo y se basa en la tecnología de TomTom Maps. La misma, brinda la posibilidad de disfrutar de acceso gratuito permanente a: mapas de TomTom sin conexión, puntos de interés (PDI), planificador de rutas y actualizaciones de mapas gratuitas. Los mapas de alta calidad de TomTom y otros proveedores se almacenan en su dispositivo Android o en la tarjeta de memoria, para que puedan ser usados sin conexión a Internet. Además cuenta con una versión Premium que permite disfrutar siempre de: mapas 3D, navegación con instrucciones de voz giro a giro, instrucciones de carriles, advertencias de límite de velocidad y visión de los cruces con flechas indicadoras de carriles en los cruces complicados (15).

### **Principales características (15)**

- ✓ Servicio de tráfico para evitar retrasos en los desplazamientos diarios.
- ✓ Avisador Premium de radares online con 300 000 ubicaciones de cámaras móviles cada mes.
- ✓ El Head Up Display (HUD) proyecta la navegación GPS en el parabrisas.
- ✓ Muestra zonas urbanas y rurales en 3D para facilitar la orientación.

## Capítulo 1: Fundamentación teórica de la investigación

- ✓ Cuenta con gráficos optimizados para tabletas y pantallas de alta definición.
- ✓ Permite la integración con el sistema de audio del vehículo ya sea a través del cable o mediante una conexión entre el dispositivo y el vehículo por bluetooth.
- ✓ Se encuentra disponible, en su versión 15.6.7 para Android 4.0 o superior (Ice Cream Sandwich, API 14), la cual fue actualizada por última vez el 11 de febrero de 2016.

### **1.2.4 Transit App: (Metro, Bus, Tren)**

Transit App es una aplicación desarrollada específicamente para el transporte público, fue elegida aplicación oficial por la Toronto Transit Commission (TTC) y premiada como aplicación de búsqueda por la New York City Transit Authority (Autoridad de Tránsito de la ciudad de Nueva York). Una vez accionada, Transit despliega todas las opciones de transporte cercanas así como los horarios de salida brindando acceso instantáneo a la información que más se necesita. Se encuentra disponible en 87 ciudades principalmente en la mayoría de las grandes ciudades de América, Canadá, México y también en varias ciudades europeas. La misma continúa en expansión, dando un nuevo aire fresco al transporte público (16).

#### **Principales características (16)**

- ✓ Realiza informes sobre la llegada del próximo metro o autobús (incluye previsiones en tiempo real).
- ✓ Planifica con facilidad viajes desde un punto A hasta un punto B (incluye todas las modalidades bus, bicicleta, metro, uber, tren, ferry, vagoneta y tranvías).
- ✓ Muestra exactamente en tiempo real dónde se encuentra en el mapa el tren o autobús.
- ✓ Accede a horarios e itinerarios de ruta e incluso estando en modo sin conexión.
- ✓ Marca como favoritas las rutas más importantes para el usuario.
- ✓ Accede en el mapa a los puntos de bicicletas públicas.
- ✓ Cuenta con un total de instalaciones de 1, 000,000 a 5, 000,000 y su tamaño, versión actual y la versión de Android varía según el dispositivo.

### **1.2.5 Andariego**

Es una aplicación para teléfonos móviles con sistema operativo Android, posee referencias de la cartografía de toda Cuba y permite a los usuarios acceder a un programa localizador que proporciona, entre otros, la distancia entre La Habana y todos los municipios del país, además, ubica los centros de salud que se soliciten, lugares de alojamiento, tiendas, paradas de ómnibus, agencias de viajes, entre otras informaciones. Fue creada por GeoSí, una empresa estatal cubana del Grupo Empresarial GEOCUBA perteneciente al Ministerio de las Fuerzas Armadas, responsable de la creación y

## Capítulo 1: Fundamentación teórica de la investigación

mantenimiento de la Cartografía Digital a nivel nacional, y del desarrollo y comercialización de soluciones informáticas aplicadas al medio geográfico (17).

### **Principales características (17)**

- ✓ Selecciona las provincias que se deseen visitar, disponible por el momento solo para La Habana.
- ✓ Busca, localiza y visualiza la información de lugares de interés (breve descripción, datos de contactos, servicios, horarios, imágenes, videos y más).
- ✓ Permite localizar la posición del usuario y muestra el recorrido mientras se mueve haciendo uso del servicio brindado por el GPS.
- ✓ Gestiona rutas de vehículos entre dos o más lugares contenidos en la base de datos.
- ✓ Muestra la descripción del recorrido entre los lugares o puntos seleccionados, expresando las distancias entre los puntos y la distancia total.
- ✓ Disponible para anuncios estatales y de cuentapropistas, con precios competitivos.
- ✓ Muestra algunas paradas de los diferentes tipos de ómnibus con los que cuenta el transporte público en la capital.

### **1.2.6 Metro de Madrid**

Metro de Madrid es una aplicación de uso sencillo e inmediato a través de la cual se puede descubrir: el mapa actualizado de la totalidad de la red, el plano turístico, la información de las líneas con las correspondencias y servicios de la estación y el estado de la circulación en tiempo real, toda la información de servicios de la estación que selecciones, y además, al poder localizar por GPS, proporciona información sobre la estación más próxima a nuestra ubicación, permitiendo elegir una estación de destino o la estación más próxima al destino elegido recomendándote el trayecto más idóneo con menos estaciones o con menos trasbordos. También indica el tiempo aproximado del trayecto y permite planificar tus desplazamientos introduciendo directamente una estación de Origen y otra de Destino pudiendo seleccionar la opción de menos estaciones o de menos trasbordos e indicándote el tiempo que se tarda en realizarlo. Permite consultar abonos, tarifas y horarios así como la información de contacto (18).

### **Principales características (19)**

- ✓ Obtiene ubicación aproximada a partir de la red.
- ✓ Obtiene ubicación precisa a partir del GPS y de la red.
- ✓ Realiza llamadas directamente a números de teléfonos.
- ✓ Impide que el dispositivo entre en modo suspensión mientras se ejecuta en primer plano.
- ✓ Requiere de una versión de Android 2.2 o superior.
- ✓ En su versión 1.13 actualizada el 7 de abril del 2015 su tamaño es de 54 MB.

### **1.2.7 Metro de Paris**

Visitar París en Metro es la aplicación oficial de la RATP. Se encuentra totalmente traducida a varios idiomas (ruso, brasileño, chino, inglés, alemán, neerlandés, italiano, español y japonés), está diseñada para todos los usuarios que deseen visitar París y transitar la ciudad en metro. La RATP incluye el metro, el ferrocarril, el tranvía y el autobús. La misma da acceso a una multitud de servicios para desplazarse fácilmente por París y descubrir los lugares más hermosos de la capital. La mayoría de las funcionalidades son accesibles incluso en modo sin conexión para que pueda disfrutar de la capital sin consumir los megas de internet de su móvil (20).

#### **Principales características (21)**

- ✓ Calcula itinerarios desde y hacia las estaciones de metros y los lugares turísticos catalogados.
- ✓ Planifica fácilmente traslados desde las estaciones y los aeropuertos de París, gracias a la información disponible: mapas e itinerarios, estaciones comunicadas por los transportes públicos, horarios, tarificación, entre otros.
- ✓ Encuentra los planos de la red férrea sin conexión una vez haya sido descargado el mapa de Paris.
- ✓ Realiza una geo localización para encontrar los emplazamientos culturales cercanos al usuario.
- ✓ La última versión (1.6.12.4), determina ubicación aproximada según la red y una ubicación precisa a partir del GPS.
- ✓ Lee la configuración del servicio de Google.
- ✓ Impide que el dispositivo entre en modo suspensión.
- ✓ Requiere de una versión de Android 2.2.3 o superior a esta.
- ✓ Tamaño de la misma es de 32 MB.

### **1.2.8 SIG-Rutas**

SIG-Rutas es un SIG sobre las rutas que brindan servicio de transportación a los trabajadores de la comunidad universitaria de la UCI. Este sistema está diseñado y desarrollado para que funcione sobre una plataforma web. Entre las funcionalidades que tiene se pueden mencionar la navegación (acercar, alejar, mover, recentrar, etc.), cálculo de áreas, medir distancias, visualización de capas y personalización de rutas. Además, cuenta con un sistema de alerta y permite conocer la localización de direcciones, municipios, paradas y rutas. Además brinda a los usuarios la posibilidad de conocer las paradas más cercanas a partir de diferentes criterios introducidos en el sistema (22).

#### **Principales características (22)**

- ✓ La velocidad de procesamiento de la información, la actualización y la recuperación dependen de la cantidad de información que tenga que procesar el sistema.

## Capítulo 1: Fundamentación teórica de la investigación

- ✓ La construcción de la aplicación funcionó bajo los conceptos de arquitectura cliente/servidor. Por tanto, el usuario final debe tener como requerimientos mínimos de software un navegador que cumpla con los estándares de la Word Wide Web Consortium (W3C, por sus siglas en inglés), se recomienda utilizar el Mozilla Firefox en su versión 3 o superior.
- ✓ Para los Servidores los sistemas operativos GNU/Linux, servidor web Apache v2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- ✓ Postgre SQL v3.0 como Sistema Gestor de Base de Datos.
- ✓ En cuanto al hardware las PC's utilizadas por los clientes deben tener tarjeta de red, al menos 128 MB de memoria RAM, se requiere al menos 100 MB de disco duro y procesador 512 MHz.
- ✓ En el caso de las PC's servidoras deben contar con tarjeta de red, para el Servidor de Mapas mínimo de 2 GB de RAM y 2 GB de disco duro, para el Servidor de BD 2 GB de RAM y 10 GB de disco duro en conjunto de un procesador 3 GHz.

De manera general estas aplicaciones informáticas son de uso liberado para los usuarios, pero no sucede así con el código fuente de las mismas, por lo que no pueden ser reutilizadas y adaptadas al problema existente. Cabe destacar que los sistemas analizados, dotan de información al usuario en cuanto a: planificación de traslados dentro de la ciudad, ubicación actual en el mapa, visualización de puntos de interés, entre otras.

Si bien en su totalidad brindan un amplio volumen de información acerca de las ciudades y del transporte, aún no existe ningún sistema de este tipo, que mediante el uso específicamente de los mapas de Cuba pueda brindar toda la información requerida a la población que se traslada en La Habana. Por lo que hacer uso de uno de los sistemas anteriormente mencionados, no facilitaría la toma de decisiones al usuario respecto a la travesía a seguir para llegar a un sitio determinado dentro de la provincia de la capital, haciendo uso de las redes de ómnibus y paradas de la ciudad. A pesar que varias de las aplicaciones mencionadas en este capítulo, ya tengan actualizada paradas y rutas que hacen estacionamiento en las mismas, no son capaces de brindar una información real y actualizada sobre la red del transporte en general.

### **1.3 Metodologías de desarrollo**

Para asegurar el éxito durante el desarrollo de software no es suficiente contar con notaciones de modelado y herramientas, hace falta un elemento importante: la metodología de desarrollo, la cual provee una dirección a seguir para la correcta aplicación de los demás elementos. Generalmente el proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema "tradicional" para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño respecto a tiempo y recursos, donde por lo general se exige

# Capítulo 1: Fundamentación teórica de la investigación

un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema está proclive a cambios frecuentemente y se exige reducir drásticamente los tiempos de desarrollo, manteniendo una elevada calidad, por lo que no se considera su utilización para la implementación de la propuesta solución, ya que durante las iteraciones de producción del software, pueden aparecer cambios con gran frecuencia. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo renuncian a prescindir de las buenas prácticas de la ingeniería de software, asumiendo el riesgo que ello conlleva. En este contexto, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, en los cuales se considera más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan, por lo que las mismas constituyen una solución acorde para ese entorno aportando una elevada simplificación, que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto.

### 1.3.1 Metodologías Ágiles versus Metodologías Tradicionales

La **Tabla 1** (23), evidencia las principales diferencias entre las metodologías ágiles y a las tradicionales. Estas diferencias que afectan no sólo al proceso en sí, sino también al contexto del equipo así como a su organización.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparadas para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles

# Capítulo 1: Fundamentación teórica de la investigación

Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
---	--

Tabla 1: Diferencias entre Metodologías Ágiles y Tradicionales.

## 1.3.2 Selección de la metodología de software a utilizar

El proceso XP se basa en la perspectiva orientada a objetos como paradigma y contiene un conjunto de reglas y prácticas que suceden en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas.

**Planeación:** La actividad comienza escuchando al cliente para recabar requerimientos que permiten que los miembros técnicos del equipo entiendan el contexto del negocio y adquieran la sensibilidad de la salida, características principales y funcionalidad que se requieren. Escuchar permite recoger información para crear las historias del usuario que describen la salida necesaria, características y funcionalidad. Cada Historia de Usuario es descrita por el cliente y colocada en una ficha técnica. El cliente le asigna una prioridad (valor) a la historia en base al valor general de la característica o función. Luego los miembros del equipo XP evalúan cada HU y le asignan un costo, medido en semanas de desarrollo. Los clientes y desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega (el siguiente incremento de software). Una vez que se llega a un compromiso acerca de la entrega, el equipo XP ordena las historias que serán desarrolladas.

**Diseño:** Se debe utilizar un diseño sencillo para representar tareas con elevado nivel de complejidad. XP estimula el uso de las tarjetas Clase Responsabilidad Colaborador, estas identifican y organizan las clases relevantes para los requerimientos del sistema, mecanismo eficaz para pensar en el software en un ámbito orientado a objetos. Además es el único producto del trabajo de diseño que se genera como parte del proceso de desarrollo de software, según lo planteado por la metodología XP. Un concepto central en la programación extrema es que el diseño ocurre tanto antes como después de que comienza la codificación.

**Codificación:** Después de elaboradas las HU y realizado el trabajo de diseño preliminar, el equipo no comienza la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las Historias de Usuario que se van a incluir en la entrega en curso. La prueba unitaria es una manera de probar el correcto funcionamiento de un módulo de código. Esto ayuda para que cada parte o módulo funcione correctamente e independientemente.

**Pruebas:** Las pruebas unitarias que se crean deben contemplar el uso de estructuras que permitan su automatización. Frente a esto se requiere estructurar las llamadas pruebas de integración como una técnica sistemática para la construcción de la arquitectura del software. Las pruebas de integración son aquellas pruebas de todos los elementos que componen un proceso. También se deben realizar las pruebas de validación junto con las de integración, consiguiendo que el equipo pueda tener indicadores

## Capítulo 1: Fundamentación teórica de la investigación

de avances y señales de alertas cuando las situaciones no se desarrollen normalmente. De esta forma la corrección de los problemas será más rápida, que si se espera al final del desarrollo para realizar las mismas. Las pruebas de validación son el proceso de revisión del software que cumple con las especificaciones requeridas. Las llamadas pruebas de aceptación son definidas por el cliente y orientadas a las características y funcionalidades que debe cumplir el sistema.

De la mano de Kent Beck, XP ha conformado un extenso grupo de seguidores en todo el mundo, publicando gran cantidad de libros a los que dio comienzo el mismo Beck. Inclusive Addison-Wesley ha creado una serie de libros denominada The XP Series (23) para fomentar el conocimiento sobre dicha metodología. XP se distingue, fundamentalmente, por su pronta, concreta y continua retroalimentación en ciclos cortos de tiempo, así como por su enfoque de planificación incremental, el cual permite definir un plan general que se espera evolucione durante la vida del proyecto (24). XP posee facilidad para flexibilizar la implementación de funcionalidades, respondiendo a los cambios que el negocio necesite. Se encuentra bien documentada con innumerables recursos disponibles en línea, comunidades libres y grupos de noticias, encontrándose gran cantidad de proyectos desarrollados con esta metodología.

La metodología XP ostenta de gran dependencia de las pruebas automatizadas escritas por los propios programadores y de la supervisión por parte del cliente en el proceso de desarrollo para permitir la evolución del sistema y capturar los defectos antes de finalizar cada iteración. Además manifiesta la dependencia de un proceso de diseño evolutivo que dura todo el tiempo en que se desarrolla el sistema y su estrecha y cercana colaboración de los programadores con habilidades ordinarias.

Se adapta perfectamente a las características del equipo de desarrollo y del proyecto. Así mismo, se caracteriza por centrarse en las necesidades del cliente, siguiendo una serie de reglas para lograr un producto de buena calidad en poco tiempo, proponiendo iteraciones muy cortas. Esta metodología también es muy efectiva cuando se exige reducir los tiempos de desarrollo mientras mantiene una alta calidad, además es considerada como una metodología postmoderna cuyas capacidades se generan a través de procesos emergentes.

Por todo lo anteriormente descrito, se decide emplear la metodología Programación Extrema (XP, eXtreme Programming) como guía para lograr el proceso de desarrollo de la propuesta de solución.

### **Características del equipo de desarrollo**

La Tabla N° 2 recoge esquemáticamente las principales aptitudes del equipo de desarrollo del sistema de información para la información del transporte público en La Habana.

Características	Valoración
Composición del equipo de trabajo	4 Personas ✓ 2 clientes ✓ 2 desarrolladores

# Capítulo 1: Fundamentación teórica de la investigación

Experiencia productiva	(alta, media, baja) ✓ Media
Conocimientos de metodologías ágiles	(bajo, medio, avanzado) ✓ cliente (medio)
Conocimientos de metodologías tradicionales	(bajo, medio, avanzado) ✓ cliente (medio) ✓ desarrolladores(bajo)
Conocimientos del Lenguaje de Programación Java	(bajo, medio, avanzado) ✓ cliente (medio) ✓ desarrolladores(medio)

Tabla 2: Características del equipo de desarrollo.

A continuación se muestran las razones de la selección:

- ✓ El proyecto es pequeño y todo el trabajo es llevado a cabo por una pareja de programadores.
- ✓ La generación de artefactos y roles en exceso no es necesaria ya que el proyecto es pequeño y está diseñado para ser realizado en el menor tiempo posible.
- ✓ Los requisitos suelen ser cambiados con frecuencia en la medida en que avanza el desarrollo del proyecto, así el cliente puede ir añadiendo HU, dividir las para agilizar el trabajo o eliminarlas simplemente. Esta metodología permite al equipo de trabajo modificar todos los planes conforme a lo anterior.
- ✓ Tanto cliente como desarrolladores forman parte del equipo de trabajo de forma tal que se logra una integración cliente, equipo de desarrollo permitiendo la retroalimentación, corrección de errores y finalmente la realización de un producto capaz de satisfacer todas las necesidades.
- ✓ Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Estas versiones ya deben constituir un resultado de valor para el negocio.
- ✓ Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- ✓ Se tiene la ventaja de ir desarrollando pequeñas partes del sistema y siendo estas probadas, permitiendo terminar todas las funcionalidades y finalmente integrar todo el producto final.

## 1.4 Herramientas y técnicas utilizadas para desarrollar la aplicación Ruta Habana

Para obtener un producto final que cumpla tanto con los requisitos funcionales como no funcionales que fueron identificados durante el desarrollo, se utilizaron un conjunto de herramientas, técnicas y lenguajes de programación, los cuales quedarán descritos a continuación.

### **1.4.1 Lenguaje de programación utilizado en el desarrollo**

Un lenguaje de programación es un lenguaje formal que permite controlar el comportamiento físico y lógico de un ordenador mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes (25).

Los lenguajes de programación se clasifican según el paradigma que usan en: procedimentales, orientados a objetos, funcionales, lógicos, híbridos, etc. Debe distinguirse de “lenguaje informático”, que es una definición más amplia, puesto que estos incluyen otros lenguajes como son el HTML o PDF que dan formato a un texto y no es programación en sí misma (26).

#### **¿Por qué Java?**

Java es un lenguaje de programación orientado a objetos, el cual agrupa en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. Fue desarrollado por James Gosling y sus compañeros de Sun Microsystems al principio de la década de los 90, con el objetivo de desarrollar software altamente confiable. El software desarrollado en Java es compilado en byte code, el cuál es ejecutado posteriormente por la máquina virtual Java.

Después de realizar un análisis se determinó utilizar como lenguaje de programación Java 8.0 para la implementación del sistema para la información del transporte público en La Habana para dispositivos que operan con Android teniendo en cuenta que es un lenguaje orientado a objetos de gran potencialidad, robustez, seguro y multiplataforma, el equipo de desarrollo tiene conocimientos sobre este lenguaje y posee experiencias, capacidades y habilidades con el desarrollo de aplicaciones en Java. Además por ser este el lenguaje nativo en el desarrollo de aplicaciones Android.

### **1.4.2 Entorno integrado de desarrollo (IDE) utilizado**

Un Entorno de Desarrollo Integrado, traducido del inglés Integrated Development Environment (IDE) es una herramienta informática que comprende un entorno de programación amigable para uno o varios lenguajes, brindando facilidades al desarrollador, tales como: un editor de código fuente, herramientas de construcción automáticas, un depurador, un compilador y opcionalmente un constructor de interfaz gráfica. Los IDE están pensados y diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software.

#### **¿Cuál es el mejor IDE?**

Cuando se habla de encontrar un entorno de desarrollo mejor que otro para determinado software, esto se convierte en una tarea complicada. Depende del lenguaje y el tipo de desarrollo que se quiera realizar, ya que no existe uno que sea mejor que otro, por lo general se trata de utilizar el que tenga

## Capítulo 1: Fundamentación teórica de la investigación

herramientas que el desarrollador domine en mayor cuantía, y que estas puedan ser utilizadas para obtener el resultado que se espera, con la calidad requerida y que cumpla con los requisitos previamente definidos por el cliente, no siempre un IDE por ser muy completo es el más conveniente. No es lo mismo que se utilice un IDE para aprender a programar, que en todo caso cualquiera es excelente, a que previamente se tenga un propósito bien claro, por ejemplo si se busca desarrollar para la plataforma Android, al buscar en la documentación se encuentra que tanto Android Studio como Eclipse pueden ser utilizados para esta plataforma, ya que ambas soportan el lenguaje en el cual se desarrolla para el sistema operativo Android (Java); sin embargo depende mucho, con cual se sienta mejor el equipo de desarrollo o el programador a la hora de la elección. Evidenciándose de esta forma que no por tener un IDE completo en total funcionamiento, se tiene el problema resuelto.

### **Android Studio**

Android Studio es un entorno de desarrollo integrado (IDE) multiplataforma de código abierto. Cuenta con capacidad de completamiento de código, ambiente flexible y rápido. Es un producto singular por la versatilidad que ofrece, y todas estas características están determinadas por la arquitectura con la cual fue diseñado. Basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, programado en Java.

Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionaran las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de plugins).

Android Studio se mantuvo durante un tiempo en versión beta, pero desde el 8 de diciembre de 2014, que se liberó la versión estable de Android Studio 1.0, Google lo ha ratificado como el IDE de programación oficial para desarrollar aplicaciones para su sistema operativo, dejando el plugin ADT para Eclipse de estar en desarrollo activo (27).

### **Principales características que incluye Android Studio: (27)**

- ✓ Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- ✓ Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- ✓ Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para

dispositivos de gama baja con limitaciones de memoria interna).

- ✓ Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de pruebas, compilación o empaquetado.
- ✓ Nueva interfaz específica para el desarrollo en Android.
- ✓ Permite la creación de nuevos módulos dentro de un mismo proyecto, sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos, algo habitual en Eclipse.
- ✓ Posibilita el control de versiones accediendo a un repositorio desde el que se puede descargar Mercurial, Git, Github o Subversion.
- ✓ Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- ✓ Vista previa en diferentes dispositivos y resoluciones.
- ✓ Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo XML.

Complementando lo señalado el equipo de desarrollo posee experiencia, conocimientos, habilidades y capacidades en el trabajo con este IDE, por ello fue seleccionado para desarrollar la propuesta de solución siguiendo las políticas de seguridad que rige la universidad, al mismo tiempo por ser un IDE que se comercializa bajo la licencia de software libre. Otro factor fundamental para la decisión es el tiempo limitado que se dispone para el desarrollo del sistema lo que dificulta la posibilidad de seleccionar un IDE el cual se desconoce por completo.

### **1.4.3 Bibliotecas de clases para la visualización de mapas**

Para facilitar el desarrollo de aplicaciones sobre la plataforma Android, Google proporciona un SDK muy completo. El mismo contiene las clases necesarias para visualizar mapas utilizando el servicio de mapas de Google directamente desde Internet. Existen otras alternativas que permiten que la información cartográfica se encuentre almacenada en el dispositivo móvil. A continuación se presentan dos de las más utilizadas:

**OsmDroid:** Es empleado en gran cantidad de proyectos como AndNav2 y OsmAnd. Está disponible bajo licencia GPL. Utiliza la información geográfica disponible en OpenStreetMap, permite mostrar mapas tanto descargados directamente de Internet como almacenados previamente en el dispositivo (28). La herramienta Mobile Atlas Creator provee una interfaz visual cómoda e intuitiva para preparar los archivos de mapas en el formato necesario, estos son almacenados como imágenes. Los diferentes niveles de zoom se almacenan por separado lo que hace que el tamaño de estos archivos crezca considerablemente teniendo en cuenta que los móviles no presentan gran cantidad de memoria.

### 1.3.1 Mapsforge 0.6

El proyecto mapsforge fue iniciado en 2008 en el instituto de ciencias de la computación Freie Universität Berlin en la universidad de Berlín. Mapsforge es una librería libre, offline y de código abierto que permite gestionar la visualización de mapas de OpenStreetMap en los dispositivos Android. Cuenta con una API fácil de usar, además con ella los desarrolladores pueden crear sus propias aplicaciones de mapeo en menos de 30 líneas de código. Fue diseñada y desarrollada con el objetivo de proveer herramientas libres y abiertas que permitan a la comunidad la creación de aplicaciones basadas en OpenStreetMap de una forma sencilla. Estas herramientas y APIs proporcionadas incluyen soluciones para representación en mapas, planificación y navegación de rutas, indexado y búsqueda de POI, capas en los mapas entre otras opciones.

#### **Principales características (29)**

- ✓ Representación gráfica rápida de los datos de OpenStreetMapFast en dispositivos.
- ✓ Potente y flexible API para capas.
- ✓ Formato de archivo compacto para el almacenamiento de datos de mapas.
- ✓ Soporte para gestos multitáctiles.
- ✓ Detalle muy elevado (soporte para aproximadamente 200 etiquetas de OSM).
- ✓ Se puede ejecutar en cualquier dispositivo con Android 1.5 o superior.
- ✓ Librería de tamaño reducido (aproximadamente 400 KB, incluyendo código fuente e imágenes).
- ✓ API fácil de utilizar (similar al API GoogleMaps para móviles).
- ✓ 100% libre y de código abierto (licencia LGPL).
- ✓ Se pueden personalizar los estilos de renderización de los mapas.

Durante el análisis de la documentación referente a las librerías que existen para realizar el mapeo, haciendo uso de los mapas que brinda OpenStreetMap se encontró que no solo **Mapsforge** es la única librería desarrollada para esto. También existen otras como **OsmDroid**, la cual permite personalizar el estilo de renderización de los mapas y gestionar mapas sin conexión pero a diferencia de **Mapsforge** los almacena como tramas y a la vez ocupa mucho espacio en el dispositivo, y **MapQuest** la cual dispone de una opción básica libre y opciones de pago a través de una clave de API, permite calcular rutas y girar la visualización del mapa pero no permite el trabajo con mapas sin conexión. Por todo lo anteriormente descrito y luego de haberse hecho una comparación entre las librerías que permiten el trabajo con mapas de OpenStreetMap se decide utilizar Mapsforge en el desarrollo de la aplicación Ruta Habana.

#### **Sistemas gestores de base de datos**

Un sistema gestor de bases de datos (SGBD) se puede definir como una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Esta colección de datos,

## Capítulo 1: Fundamentación teórica de la investigación

normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente (30). En otras palabras los SGBD, son aplicaciones que permiten a los usuarios definir, crear, mantener la base de datos, igualmente proporcionan un acceso controlado a la misma.

### **SQLite 3.0**

SQLite es un manejador de código abierto de bases de datos que combina una interfaz limpia de SQL. Permite trabajar con poca memoria y con una velocidad bastante rápida, peculiaridades que son necesarias cuando se habla de entornos móviles. Soporta las características estándar de las bases de datos relacionales, como la sintaxis basada en SQL, transacciones y la elaboración de consultas. Debido a esto, cualquier desarrollador que haya trabajado con bases de datos sin importar el entorno, no encontrará una dificultad especial en trabajar con bases de datos locales en Android.

### **Principales características del gestor de bases de datos (31):**

- ✓ Tamaño: tiene pequeña memoria y necesita de una biblioteca única para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- ✓ Portabilidad: se ejecuta en múltiples plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- ✓ Interfaces: cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby y Groovy.
- ✓ Costo: es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.
- ✓ No posee configuración: de la forma en que fue creado y diseñado, no necesita ser instalado.
- ✓ No es necesario prender, reiniciar o apagar un servidor, e incluso configurarlo; esta cualidad permite que no haya un administrador de base de datos para crear las tablas, vistas, asignar permisos o bien la adopción de medidas de recuperación de servidor por cada caída del sistema.

En aditamento a las características señaladas, SQLite es el SGBD mayormente utilizado en el desarrollo de aplicaciones Android para las cuales se hace necesario el uso de una base de dato para el buen funcionamiento de las mismas; por lo que se decide utilizarlo en su versión 3.0 para almacenar los datos necesarios para el correcto funcionamiento del sistema para el transporte público en La Habana.

### **1.4.4 Lenguaje de modelado**

Para modelar las entidades se empleará el Lenguaje Unificado de Modelado (UML) por sus siglas en inglés, es un lenguaje utilizado para especificar, visualizar, construir y documentar artefactos de un sistema de software. Está constituido por un conjunto de diagramas y es necesario contar con todos

## Capítulo 1: Fundamentación teórica de la investigación

esas representaciones dado que cada una se dirige a cada tipo de persona implicada en el sistema. Un modelo UML indica qué es lo que se supone que deba hacer el sistema, no como lo hará (32).

UML facilita el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan. Por las características antes mencionadas y debido a que el equipo de desarrollo cuenta con experiencia en la utilización del lenguaje se decide emplearlo en su versión 2.1.

Para efectuar el modelado de los procesos de negocio se empleó BPMN (Notación para el Modelado de Procesos de Negocio), la cual es una notación gráfica estandarizada para el modelado de procesos de negocio. BPMN sirve como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

### **1.4.5 Herramienta CASE**

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE) por sus siglas en inglés, son empleadas con el fin de automatizar los aspectos claves de todo el proceso de desarrollo de software de un sistema, desde su inicio, hasta completar todo el proceso de implementación.

Para el desarrollo de la propuesta de solución se empleará Visual Paradigm for UML, una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: negocio, descripción de los requisitos, análisis y diseño orientados a objetos, implementación, pruebas y despliegue.

#### **Principales características de Visual Paradigm for UML (33):**

- ✓ Entorno de creación de diagramas UML 2.1.
- ✓ Lenguaje estándar, común a todo el equipo de desarrollo.
- ✓ Existen varias versiones compatibles con diferentes plataformas, usadas para necesidades específicas del equipo de desarrollo.
- ✓ Permite la integración con Android Studio.

Como demuestran las características antes mencionadas la herramienta Visual Paradigm for UML en su versión 8.0 es un software potente por lo que es utilizada a nivel mundial. Además posibilita la realización satisfactoria de los diagramas en dependencia del flujo de trabajo de la metodología que se utilice. Por estas razones es la herramienta seleccionada para confeccionar el diseño de la base de datos y el diagrama del modelo del dominio.

### **Conclusiones del capítulo**

En este capítulo fue realizado un análisis de los sistemas de información geográfica existentes específicamente, aquellos que se enfocan en brindar información al usuario a través de una aplicación móvil, sobre el uso de los medios del transporte público, concluyendo que ninguno ofrece solución a la totalidad de los objetivos propuestos. Se definieron las herramientas y tecnologías mayormente

## Capítulo 1: Fundamentación teórica de la investigación

adecuadas para dar solución al problema planteado, teniendo en cuenta las características de estas y las necesidades descritas con anterioridad a las que da respuesta la propuesta de solución.

Se seleccionó Android Studio en su versión 1.3.2 como IDE de desarrollo, Java 8 como lenguaje de programación, UML 2.1 como lenguaje de modelado, Visual Paradigm 8.0 como herramienta para el modelado y SQLite 3.0 como sistema gestor de base de datos. Además, se investigó sobre las metodologías de desarrollo de software más utilizadas en este tipo de aplicaciones, seleccionándose XP como la metodología de desarrollo a utilizar.

### **Capítulo 2: Modelación, control y seguimiento del sistema**

En este capítulo se aborda el entorno de desarrollo definido, se explican las diferentes características que va a presentar la aplicación Ruta Habana para dispositivos que operan bajo la plataforma Android a implementar y sus funcionalidades. Para ello se siguen los aspectos que plantea la metodología ágil XP en su fase de Planificación, la cual guiará las pautas a seguir para lograr mayor organización en la distribución de las actividades, tiempo y artefactos a desarrollar, confeccionándose las Historias de Usuarios (HU), las cuales proporcionarán un mayor entendimiento y comprensión del sistema. Además se tratan los principales artefactos generados, la planificación del tiempo y el esfuerzo de las fases posteriores.

#### **2.1 Diseño de la propuesta de solución**

Utilizando toda la información recopilada en el capítulo anterior, se propone como solución la implementación de una aplicación móvil para la plataforma Android, para aquellos usuarios que hagan uso del transporte público de La Habana y porten con ellos un dispositivo cuyo sistema operativo sea el anteriormente mencionado y además contengan GPS integrado. La misma será capaz de brindarle al usuario una valiosa información a la hora de transitar por la ciudad sin necesidad de realizar una activación de los datos móviles o de la redes inalámbricas, debido a que dicha aplicación está pensada, diseñada e implementada para hacer uso de los mapas sin conexión almacenados en el dispositivo; por ejemplo en que punto de la ciudad se encuentra una vez que el GPS esté activo, que parada le está más cercana tanto a su posición actual como a determinado punto en la ciudad, que medios de transporte tienen paradas oficiales en determinada parada y con que otros medios de transporte existen puntos de enlaces en una misma parada. También la aplicación podrá determinar y mostrar finalmente al usuario una o varias rutas para trasladarse de un lugar a otro en la ciudad. Además brindará a los usuarios información referentes a puntos de interés y como llegar a los mismos haciendo uso de la red provincial de transporte de la ciudad.

#### **2.2 Descripción del Modelo de Dominio**

A continuación se describirá el modelo del dominio realizado durante la fase de exploración y planificación de la aplicación, aunque este artefacto no esté entre los propuestos por la metodología de desarrollo utilizada, se propone su realización para una mejor comprensión del dominio del sistema por parte del cliente y del equipo de desarrollo.

##### **2.2.1 Modelo de dominio**

El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos que se manejan en el dominio del sistema. Los objetos o conceptos incluidos en el Modelo de Dominio

no describen clases u objetos del software; sino entidades o conceptos del mundo real que están asociados al problema en cuestión. Dicho modelo podrá ser utilizado como una base de las abstracciones relevantes en el proceso de construcción del sistema. El Modelo de Dominio ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación. El proceso para su elaboración tiene tres pasos. En primer lugar identificar las clases conceptuales, después dibujarlas en un diagrama de clases y finalmente añadir relaciones y atributos (34).

Resumiendo, se denomina “Modelo de Dominio” a la representación visual de los conceptos u objetos que son de interés, sus características y las relaciones entre dichos conceptos. Es el mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes. Es un diccionario visual del dominio del problema. Teniendo en cuenta todo lo explicado anteriormente y debido a que no se tienen bien definidas las fronteras del negocio para la aplicación a desarrollar, se decide realizar un Modelo de Dominio.

Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación, pudiendo mostrar (35):

- ✓ Objetos del dominio o clases conceptuales.
- ✓ Asociaciones entre las clases conceptuales.
- ✓ Atributos de las clases conceptuales.
- ✓ Además se procederá a explicar cada uno de los conceptos que conforman al mismo.

### **2.2.2 Conceptos fundamentales del dominio**

A continuación se proporcionará una breve descripción de los conceptos encontrados en el ámbito del problema, con el fin de facilitar una mejor comprensión del Diagrama del Modelo de Dominio.

**Usuario:** Persona que por medio de un dispositivo móvil puede acceder a la aplicación para navegar en la misma y obtener la información que necesite en un cualquier momento.

**Provincia:** Es un espacio ubicado dentro del sistema que permitirá al usuario seleccionar la provincia en la cual se encuentre.

**Base datos de Ruta Habana:** Es el espacio donde se encuentran todos los datos referentes a los recorridos, puntos de interés, paradas y terminales de La Habana.

**Ruta:** Es el espacio dentro del sistema que permitirá al usuario visualizar la información sobre una ruta.

**Operador:** Es el encargado dentro del sistema de gestionar el tipo de transporte que puede abordar el usuario.

**Lugar:** Es un espacio dentro del sistema que permitirá visualizar la información referente a los POI.

**Categoría:** Define dentro del sistema la categoría de un lugar de interés para el usuario.



con que son detallados los requerimientos que permitirán el diseño y posteriormente la implementación del sistema. Existen dos clasificaciones de requerimientos (Funcionales y No funcionales), a través de los cuales se define el alcance del sistema en cuanto a las acciones que debe realizar.

### **2.3.1 Requerimientos funcionales**

Los requisitos funcionales (RF) denotan una funcionalidad del sistema, son características requeridas que expresan una capacidad de acción del mismo o una determinada función; generalmente expresada en una declaración en forma verbal. Sirven de base para estimar el coste, el tiempo necesario para desarrollar el software y para planificar los contenidos técnicos de las iteraciones posteriores (36). A partir de varias reuniones programadas con el cliente, se identificaron los requisitos a cumplir por el sistema, los cuales se muestran a continuación:

RF 1: Realizar navegación en el mapa. El sistema debe permitir que el usuario pueda navegar por el mapa transformando la visualización del mismo para obtener una mejor representación de los objetos.

RF 2: Mostrar ubicación en el mapa. El sistema debe permitir que el usuario pueda visualizar su ubicación en el mapa una vez activado el GPS.

RF 3: Localizar parada más cercana a la posición del usuario. El sistema debe permitir que el usuario pueda visualizar la ubicación en el mapa de la parada más cercana a su posición.

RF 4: Localizar parada más cercana al punto de destino. El sistema debe permitir que el usuario pueda visualizar la ubicación en el mapa de la parada más cercana a su destino.

RF 5: Mostrar rutas de enlaces en una parada. El sistema debe permitir que el usuario pueda visualizar las rutas que arriban a determinada parada, para que el usuario pueda tomar decisiones a fines con su viaje

RF 6: Sugerir posibles rutas a abordar. El sistema debe permitir que el usuario pueda conocer si existe al menos una combinación de rutas para trasladarse desde su lugar de origen hacia su lugar de destino.

RF 7: Mostrar el recorrido de las rutas en el mapa. El sistema debe ser capaz de visualizar el recorrido de ida y vuelta de una ruta seleccionada.

RF 8: Mostrar la ubicación de las terminales. El sistema debe visualizar la ubicación de las terminales en el mapa, para que el usuario tenga conocimiento de las mismas.

RF 9: Realizar llamada a la terminal. El sistema debe permitir que el usuario llame a una terminal seleccionada, una vez que el usuario se encuentre visualizando el listado de terminales y presione el botón llamar, que aparece al lado de cada terminal.

RF 10: Gestionar puntos de interés en el mapa. El sistema debe brindar al usuario información sobre (dirección, ubicación, detalles del lugar y el teléfono en caso que exista), correspondiente a cada punto de interés ubicado en el mapa, además permite la inserción de un punto de interés para el usuario, así como eliminar puntos agregados cuando el mismo desee.

RF 11: Cambiar idioma. El Sistema debe permitir que el usuario elija el idioma de su preferencia (Español/Ingles).

### **2.3.2 Requerimientos no funcionales**

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el equipo de desarrollo no debe obviar, puesto que estos son los encargados de proporcionar la terminación del producto. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

#### **Diseño e implementación:**

- ✓ Se utilizará el lenguaje de programación Java en su versión 1.8.
- ✓ Se empleará como IDE Android Studio en su versión 1.3.2 junto al SDK 23.0.0 para el desarrollo de las funcionalidades.

#### **Hardware:**

- ✓ Para un buen funcionamiento del sistema, se hace necesario contar con un dispositivo con las siguientes características:
  - ✓ Debe contar con sistema de posicionamiento global (GPS) integrado.
  - ✓ Será necesario disponer de un terminal, dígame tabletas inteligentes, móviles, ordenadores entre otros.
  - ✓ Memoria RAM de 512 MB o superior.
  - ✓ Microprocesador Dual Core a 2.0 GHz o superior.
  - ✓ El dispositivo deberá tener mínimo 40 MB de almacenamiento disponible.

#### **Software:**

- ✓ El dispositivo móvil debe tener el sistema operativo Android en su versión 3.0 o superior.

#### **Interfaz:**

- ✓ El sistema se mostrará de forma vertical en todos los dispositivos que se instale.
- ✓ Los colores que predominan son (naranja, rojo, azul y blanco).

#### **Disponibilidad:**

- ✓ La aplicación debe estar disponible en todo momento que el usuario necesite acceder y manejar la información contenida en la misma.

#### **Usabilidad:**

- ✓ Para la navegación en la aplicación, se requiere un mínimo dominio de la tecnología móvil, por lo que posibilitará al usuario sin experiencia una rápida adaptación para operar con el sistema. La aplicación deberá presentar una interfaz que permita la fácil interpretación por el usuario, además debe posibilitar que el usuario visualice de manera rápida la información buscada.

### **Rendimiento:**

- ✓ El sistema debe responder en el menor tiempo posible, ante las solicitudes de información por parte de los usuarios. La eficiencia de la aplicación estará determinada en gran medida por la velocidad que se ejecuten las consultas a la base de datos, las mismas dependerán de las características (software y hardware) del dispositivo móvil donde se encuentra ejecutándose la aplicación.

### **Soporte:**

- ✓ El sistema contará con un manual de ayuda, con el objetivo de explicar cómo funciona la búsqueda de la información sobre recorridos, paradas y puntos de interés. También explicará cómo se debe proceder para encontrar paradas cercanas y las posibles rutas a abordar.

### **Seguridad:**

- ✓ La base de datos donde se encuentra toda la información suministrada al usuario sobre el transporte en la capital y los puntos de interés, debe estar empaquetada dentro de la extensión (.apk), además será encriptada para que el usuario no pueda modificarla y poder mantener la integridad de los datos.

## **2.4 Estándares de codificación según la metodología XP**

La metodología XP promueve la programación basada en estándares, de forma tal que todo el equipo de desarrollo pueda comprenderla y de esta manera facilitar la recodificación en caso de ser necesario. Los estándares de codificación son pautas de programación que no se encuentran enfocadas a la lógica del programa sino a la estructura y la apariencia física del mismo facilitando la lectura. Mejoran la legibilidad del código al mismo tiempo que permiten una rápida comprensión. Permite que cualquier programador entienda lo que está desarrollado y así pueda darle un buen mantenimiento al sistema. En la mayoría de las ocasiones el software es mantenido, por un equipo técnico de trabajo, entre los cuales raras veces hay presencia de los desarrolladores del mismo.

El establecimiento de los estándares o convenciones de programación empleados en el desarrollo de software sobre la plataforma Java está basado en los estándares recomendados por Sun Microsystems, que han sido difundidos y aceptados ampliamente por toda la comunidad Java, y que han terminado por consolidarse como un modelo estándar de programación (25).

A continuación se muestran las reglas y recomendaciones a seguir para que el código del sistema Ruta Habana para dispositivos que operan con sistema operativo Android, sea comprendido por los desarrolladores del mismo o el futuro equipo que realice próximas versiones.

### 2.4.1 Comentarios

Todo fichero fuente debe comenzar con un comentario que incluya el nombre del sistema, nombre de la clase en la cual se encuentra el desarrollado, información sobre la versión del código, la fecha y el copyright. El copyright indica la propiedad legal del código, el ámbito de distribución, el uso para el que fue desarrollado y su modificación.

Dentro de estos comentarios iniciales podrían incluirse adicionalmente comentarios sobre los cambios efectuados sobre dicho fichero (mejoras, incidencias, errores, y otros). Estos comentarios son opcionales si los ficheros están bajo un sistema de control de versiones bien documentado, en caso contrario se recomienda su uso. Estos comentarios constituyen el historial de cambios del fichero. Este historial es único para cada fichero y permitirá conocer rápidamente el estado y la evolución que ha tenido el fichero desde su origen. La figura 2 muestra un ejemplo de comentario de inicio aplicado en la implementación de Ruta Habana.

```
/*
***** Ruta Habana *****
*
*
***** Version 1.0*****
***** Marzo de 2016*****
***** MapaActiviy.java*****
*
*
***** Departamento de Ciencias Básicas de la Facultad 2 *****
***** Universidad de las Ciencias Informáticas (UCI) *****
*/
```

Figura 2: Ejemplo de estándar de comentario.

### 2.4.2 Declaración de variables

Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas. Las variables nunca podrán comenzar con el carácter "\_" o "\$". Los nombres de variables deben ser cortos y sus significados tienen que expresar con suficiente claridad la función que desempeñan en el código. Debe evitarse el uso de nombres de variables con un sólo carácter, excepto para variables temporales (consultar figura 3 para ver ejemplo de estándar de declaración de variables).

```
Toolbar toolbar;  
Toolbar toolbar_base;  
private Carousel carousel;  
private List<OperadorObj> ope = new ArrayList<>();  
private List<Drawer> dra = new ArrayList<>();  
ActionBarDrawerToggle mDrawerToggle;  
DrawerLayout mDrawerLayout;  
ListView mDrawerList;  
AlertDialog dialog;  
LinearLayout acercaDe;  
Dialog customDialog = null;
```

Figura 3: Ejemplo de estándar de declaración de variables.

### 2.4.3 Métodos

Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas como se muestra en **la figura 3**.

```
@TargetApi(Build.VERSION_CODES.KITKAT)  
public void getCambiarLocale(String locale) {  
    if (Objects.equals(locale, ""))  
        return;  
    Locale miLocale = new Locale(locale);  
    Locale.setDefault(miLocale);  
    android.content.res.Configuration config = new android.content.res.Configuration();  
    config.locale = miLocale;  
    getBaseContext().getResources().updateConfiguration(config,  
        getBaseContext().getResources().getDisplayMetrics());  
}
```

Figura 4: Ejemplo de estándar de métodos.

### 2.4.4 Sentencias

Las sentencias pertenecientes a un bloque de código estarán tabuladas un nivel más a la derecha con respecto a la sentencia que las contiene. El carácter inicio de bloque "{" debe situarse al final de la línea que inicia el bloque de instrucciones. El carácter final de bloque "}" debe situarse en una nueva línea tras concluir el último fragmento de código perteneciente al bloque de instrucciones y alineado con el primer carácter de dicho bloque. Todas las sentencias de un bloque deben encerrarse entre llaves "{...}", aunque el bloque conste de una única sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves. En las figuras 4 y 5 se muestran ejemplos de estándar de declaración de sentencias.

```
switch (position){
    case 0:
        Intent intent = new Intent(getApplicationContext(), Trayectos_Metrobus.class);
        startActivity(intent);
        overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
        break;
    case 1:
        Intent intent1 = new Intent(getApplicationContext(), Trayectos_Guaguas.class);
        startActivity(intent1);
        overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
        break;
    case 2:
        Intent intent2 = new Intent(getApplicationContext(), Trayectos_Ruteros.class);
        startActivity(intent2);
        overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
        break;
    default:
        break;
}
```

Figura 5: Ejemplo de sentencia Switch.

### 2.5 Fase de planificación

La fase de planificación es la primera fase definida por la metodología XP, en esta se define el alcance real del sistema permitiendo una familiarización del equipo de desarrollo con las herramientas, tecnologías y procesos. Esta fase comienza por la creación de una serie de historias, llamadas historias de usuario (HU) las cuales definen mediante su redacción las verdaderas necesidades del cliente y es aquí donde los programadores estiman el tiempo de desarrollo (37).

#### 2.5.1 Historias de Usuarios

Las HU son la técnica utilizada en XP para especificar las funcionalidades del sistema, brindan detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación. El cliente es el encargado de asignarle una prioridad a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este costo se traduce en las semanas que llevará desarrollar las mismas. Si las HU según lo planificado demoran en desarrollarse, se sugiere dividirla en HU más pequeñas. Es importante destacar que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

#### La prioridad en el negocio

- ✓ **Alta:** Son aquellas HU que constituyen funcionalidades fundamentales en el desarrollo el sistema, a las que el cliente define como principales para el control integral del sistema.
- ✓ **Media:** Son las funcionalidades a tener en cuenta por el cliente, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- ✓ **Baja:** Es otorgada a las HU que son funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo.

Para hacer más comprensible las HU, a continuación el equipo de desarrollo enuncia su leyenda:

**Número:** Número de identificación para las HU, sería incremental en el tiempo.

**Nombre Historia de Usuario:** Es el nombre de la HU, sirve para identificarla fácilmente tanto para los desarrolladores como para los clientes.

**Modificación de Historia de Usuario Número:** Cantidad de modificaciones que se le ha realizado a la HU, de no tener cambios se debe escribir la palabra ninguna,

**Usuario:** Nombre del programador encargado de implementar la HU.

**Prioridad del negocio:** Qué tan importante es para el cliente, se clasifica en Alta, Media y Baja.

**Iteración asignada:** Iteración en la que se desarrollará la HU.

**Puntos estimados:** Tiempo en semanas que se le asignará. (Estimado)

**Descripción:** Es la descripción de la historia, detallando las operaciones del usuario y las respuestas del sistema.

**Riesgo en desarrollo:**

- ✓ Las historias de usuarios que, en caso de tener algún error de implementación, puedan afectar el funcionamiento o la disponibilidad de la aplicación cada vez que el usuario haga uso de la misma, tienen riesgo de desarrollo **alto**.
- ✓ Las historias de usuarios que puedan presentar errores y retrasan la entrega de la versión, tienen riesgo de desarrollo **medio**.
- ✓ Las historias de usuario que puedan presentar errores, pero estos son tratados con facilidad y no afectan en desarrollo del proyecto, tienen riesgo de desarrollo **bajo**.

**Observaciones:** Informaciones de interés, como glosarios, detalles del usuario, entre otros.

**Prototipo de Interfaz:** Contiene la imagen de una de las interfaces de usuario relacionadas con la HU. Para dar cumplimiento al desarrollo de las funcionalidades del sistema fueron definidas 11 HU. A continuación se muestran varias de las HU definidas por los usuarios, por ser funcionalidades de prioridad media y alta en el negocio. En el **Anexo I: Historias de Usuario** pueden encontrarse las HU restantes implementadas en el desarrollo de la aplicación.

Historia de usuario	
<b>Número:</b> HU-1	<b>Nombre de la historia de usuario:</b> Realizar navegación en el mapa
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova Pablo Miro Martínez	<b>Iteración Asignada:</b> 1
<b>Prioridad en el negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Media	
<b>Descripción:</b> El sistema debe permitir que el usuario pueda navegar por el mapa transformando la visualización del mismo, realizando acciones de zoom para obtener una mejor representación de los objetos.	

<b>Observaciones:</b>
<b>Prototipo de Interfaz:</b>



Tabla 3: HU-Realizar navegación en el mapa

Historia de usuario	
<b>Número:</b> HU-2	<b>Nombre de la historia de usuario:</b> Mostrar ubicación en el mapa
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova Pablo Miro Martínez	<b>Iteración Asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alta	
<b>Descripción:</b> El sistema debe mostrar la ubicación del usuario en el mapa, a partir del posicionamiento obtenido por el sistema de posicionamiento global.	
<b>Observaciones:</b>	
<b>Prototipo de Interfaz:</b>	



Tabla 4: HU-Mostrar Mapa.

### 2.5.2 Estimación de esfuerzo

La medida utilizada para la estimación del esfuerzo asociado a la implementación de cada una de las Historias de Usuario es el punto (máximo esfuerzo), la cual queda establecida por el equipo de desarrollo. Un punto equivale a una semana ideal de programación. Esta medida generalmente toma

valores de 1 a 3 en dependencia de la complejidad de la HU, partiendo de los factores que el equipo decida para clasificarlas. Estos factores por lo general son: la complejidad, el esfuerzo y el riesgo que demande cada una de ellas. Cuando una HU tiene tiempo de duración inferior a una semana significa que está muy bien descrita y que puede ser combinada con otra si muchas dificultades. (24) Además el equipo de desarrollo podrá mantener un registro de la “rapidez” con que se están efectuando las tareas, establecidas en puntos por iteración, apoyándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. (37) La planificación se puede realizar basándose en el tiempo o el alcance. La rapidez del proyecto es utilizada para establecer cuantas historias de usuarios se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias de usuarios. La **Tabla 5** contiene la estimación del esfuerzo por HU según el orden a realizar.

Número	Nombre de la Historia de Usuarios	Puntos estimados en días/semanas
HU-1	Realizar navegación en el mapa.	1
HU-2	Mostrar ubicación en el mapa.	1
HU-3	Localizar parada más cercana a la posición del usuario.	1
HU-4	Localizar parada más cercana al punto de destino.	0.4
HU-5	Mostrar puntos de enlaces en una parada.	0.6
HU-6	Sugerir posibles rutas a abordar.	2
HU-7	Mostrar el recorrido de las rutas en el mapa.	1
HU-8	Mostrar la ubicación de las terminales	0.4
HU-9	Realizar llamada a la terminal.	0.4
HU-10	Mostrar puntos de interés en el mapa.	0.6
HU-11	Cambiar idioma.	0.4
<b>Total</b>	<b>11</b>	<b>44 días/9 semanas</b>

Tabla 5: Estimación del esfuerzo por Historias de Usuarios.

### 2.5.3 Plan de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. La misma incluye varias iteraciones del sistema antes de la entrega del primer lanzamiento. El calendario es dividido en un número de iteraciones de tal manera de que cada iteración tome de una a cuatro semanas de implementación. En la primera iteración se crea un sistema que abarca los aspectos más importantes de la arquitectura global. Esto se logra seleccionando las historias que hagan referencia a la construcción de la estructura de todo el

sistema. El cliente decide que historias van a ser implementadas para cada iteración. Además, se realizan las pruebas funcionales, realizadas por el cliente, al final de cada iteración. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance. Al final de la última iteración el sistema estará listo para ser utilizado.

Teniendo en cuenta el riesgo para desarrollar cada una de las historias de usuario, el tamaño del equipo de desarrollo, así como otros factores subjetivos, se decidió dividir el proyecto en tres iteraciones, detalladas a continuación:

**Iteración 1:** En esta iteración se implementarán las HU de prioridad media, funcionalidades que permitirán brindar toda la información referente a paradas, recorridos y sitios de interés al usuario a través del uso del mapa.

**Iteración 2:** En esta iteración se implementarán las HU de prioridad alta, las cuales constituyen funcionalidades indispensables para cubrir las necesidades del cliente.

**Iteración 3:** En esta iteración se implementarán las HU de prioridad baja, las cuales no son menos importantes que las de la iteración anterior.

**La tabla 6** muestra la relación de HU que se realizarán por cada iteración, así como la duración total en semanas de estas.

Iteraciones	Orden de las HU por iteraciones	Duración de las iteraciones
Iteración 1	HU-1, HU-7, HU-8, HU-10	3 semanas
Iteración 2	HU-2, HU-3, HU-4, HU-5, HU-6	5 semanas
Iteración 3	HU-9, HU-11	1 semanas
Total		9 semanas

Tabla 6: Relación de Historias de Usuarios por cada iteración.

### 2.5.4 Plan de entrega

En el momento en que culmina la elaboración de las HU, se inicia el proceso de creación de un plan de entrega. El cual tiene como objetivo fundamental la obtención por parte de los programadores de una estimación detallada del período de tiempo que deben tener en cuenta para la implementación. El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión “Juego de planeamiento” (“Planning Game”), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planeamiento, “Planning meeting”). (38) Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por

los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario. En la siguiente tabla se muestra el **Plan de Entregas** por cada una de las iteraciones con las fechas aproximadas de las entregas correspondientes.

Aplicación	Entrega 1ra Iteración	Entrega 2da Iteración	Entrega 3ra Iteración
Ruta-Habana para Android	8 de abril de 2016	6 de mayo de 2016	20 de mayo de 2016

*Tabla 7: Plan de entrega por iteraciones.*

### **Conclusiones del capítulo**

Al concluir este capítulo fueron descritas las fases de Exploración y Planeación para la propuesta de solución, elemento que posibilitó la obtención de los artefactos necesarios para dar cumplimiento a la planificación establecida para el desarrollo del SIG Ruta Habana para dispositivos que operan con sistema operativo Android, en correspondencia a la metodología seleccionada tales como: Historias de Usuario, el esfuerzo estimado por HU, el plan de iteraciones así como el plan de entregas correspondiente para las iteraciones planteadas favoreciendo el orden en que se les dará cumplimiento según las necesidades del cliente. El desarrollo de estas etapas permitió además que el proceso fuera dividido en tres iteraciones en las que se desarrollaron las HU de prioridad media durante la primera iteración por ser las que permiten obtener información indispensable para el control de los recursos de hardware y software en dispositivos que operan con Android, posteriormente fueron desarrolladas las de prioridad alta, estas HU son indispensables para el desarrollo del sistema y finalmente las de prioridad baja que brindan la información restante de las características de hardware del dispositivo. Determinándose que el proceso de desarrollo durará un total de 9 semanas, divididas en iteraciones de 3, 5 y 1 semana respectivamente, de manera tal que el equipo de desarrollo y el cliente obtendrían un producto funcional al concluir cada iteración, posibilitando a este último plantear su conformidad o no con la aplicación obtenida.

### **Capítulo 3: “Iteraciones y Producción”**

En el presente capítulo se define la arquitectura de la aplicación y el estilo arquitectónico utilizado, conjuntamente se describirán los dos algoritmos diseñados por los desarrolladores ya que sobre los mismos se encuentra el mayor peso de la solución propuesta para lograr el objetivo de la investigación, el primero para la obtención de paradas cercanas a la posición del usuario y el segundo utilizado para la sugerencia de posibles rutas a abordar por el usuario para trasladarse de un lugar a otro en la ciudad haciendo uso del transporte público. Luego se describen los patrones de diseños y el diseño de la base de datos del sistema. Acto seguido se detallan las iteraciones realizadas durante la etapa de construcción de la propuesta de solución. Con este fin son definidas las tarjetas CRC como artefacto muy útil para respaldar el enfoque orientado a objetos. Son expuestas además las tareas de ingeniería resultantes por cada HU. Los artefactos que se obtendrán serán el resultado del seguimiento de la filosofía determinada por la metodología XP, la cual plantea que la implementación debe ejecutarse de manera iterativa e incremental, logrando al término de cada iteración un producto funcional que debe ser testeado de conjunto con el cliente; y a su vez es garantizada la retroalimentación entre los desarrolladores y el mismo.

#### **3.1 Arquitectura de software**

La arquitectura de un software es el diseño de más alto nivel de la estructura de un sistema. Toda arquitectura de software debe definir diversos aspectos del software, y generalmente, cada uno de estos se describe de una manera más comprensible si se utilizan distintos modelos o vistas (39).

Para el desarrollo del sistema se empleará la arquitectura en tres capas, la misma consiste en dividir el sistema en varias capas lográndose de esta forma reducir el grado de complejidad. Las aplicaciones contienen códigos para la presentación, procesamiento y almacenamiento de datos, la diferencia se encuentra en la organización del mismo. Ofrece ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La utilización de esta arquitectura permite desarrollar capas en paralelo, además posibilita un mantenimiento más sencillo ya que se puede modificar un componente de ser necesario en lugar de toda la aplicación y permite agregarle nuevos módulos aumentando su flexibilidad. El sistema se divide en las siguientes capas lógicas:

**Interfaz:** Representa la información proporcionada por el acceso a datos en un formato que permite la interacción entre el usuario y el sistema, se puede evidenciar en las interfaces de usuario.

**Lógica del negocio:** Procesa las solicitudes realizadas por el usuario y se envían las respuestas produciendo generalmente cambios en el sistema. Esta capa sirve como intermediaria entre la interfaz y el acceso a datos.

## Capítulo 3: Iteraciones y Producción

**Acceso a datos:** Se encarga de acceder a los ficheros que almacenan los datos persistentes para proveer de información al usuario o a la aplicación misma, o para realizar cambios en la misma.

En la arquitectura se definen niveles que se corresponden con la forma en que las capas son distribuidas de forma física. Como las capas van a residir en el dispositivo móvil entonces la arquitectura de solución es de tres capas y un nivel.

### **3.2 Estilo arquitectónico. Patrón de arquitectura**

Involucrados en una arquitectura, se encuentran los estilos arquitectónicos. Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software (35).

Existen numerosos estilos arquitectónicos entre los que se encuentra el estilo de llamada y retorno y dentro de él mismo, el patrón de arquitectura **Modelo-Vista-Controlador** (MVC). Éste separa la lógica de control, la interfaz de usuario y los datos de una aplicación en tres componentes distintos. Se utiliza puesto que se tiene una capa de acceso a datos donde van a estar todas las clases que de una forma u otra piden información a una base de datos.

La capa **Modelo** compuesta por el paquete **cu.uci.pabloylionysapp.model** contiene todas las clases que tienen el código relacionado con el acceso a datos, para que este sea lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. Se incluirán consultas a las bases de datos y validaciones de entrada de datos. Se evidencia dentro del modelo la clase **RutaDB**, la cual contiene varios métodos que envían a ejecutar consultas SQL al gestor obteniendo y devolviendo los resultados de las mismas para luego ser procesados.

La capa **Vista** compuesta por el paquete **cu.uci.pabloylionysapp.interfaces** contiene todas las clases que poseen el código representado o lo que es igual a la parte que será visualizada en pantalla por el usuario. Se evidencian dentro de este paquete las clases **BaseRuta** y **Terminales**, interfaces visuales que son mostradas al usuario para que el mismo interactúe con la aplicación.

La capa **Controladora** compuesta por el paquete **cu.uci.pabloylionysapp.controladoras** que contiene las clases que ejecutan la lógica de la aplicación, realizan llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Se evidencian en este paquete varias clases; **GpsManager**, **MyLocationOverlay**, **OperadoresAdapter**, **ParadasAdapter**, las cuales se encargan de recopilar toda la información y enviar los datos a las clases de la capa Vista. Estas controladoras se programan en lenguaje Java conformando la base de la aplicación. A continuación se muestra en la (Figura 7) dicho patrón arquitectónico para comprender mejor su funcionamiento.

## Capítulo 3: Iteraciones y Producción

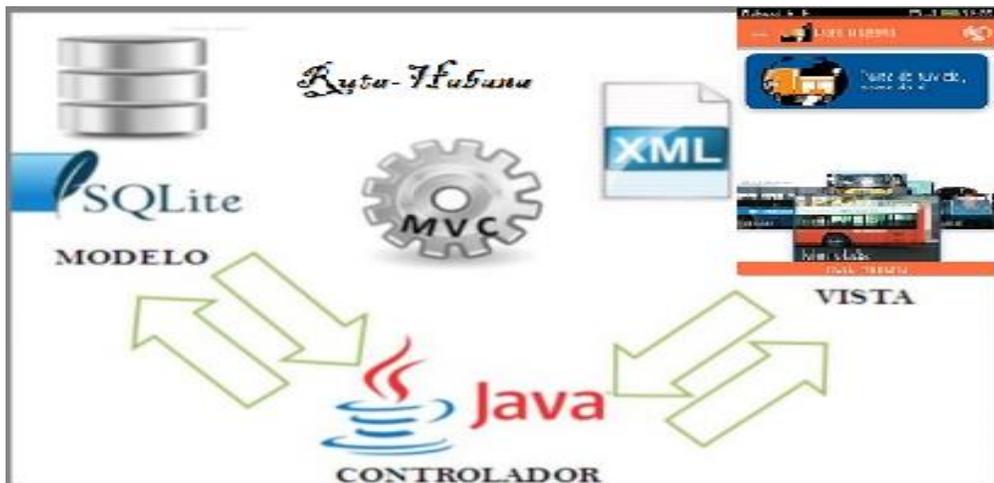


Figura 6: Patrón Arquitectónico del sistema.

Con el uso de este patrón se persigue mejorar la reusabilidad del código y que las modificaciones en las vistas impacten en menor medida en la lógica de negocio o de datos.

### 3.3 Diseño de los algoritmos utilizados en la solución

Para lograr que el sistema Ruta Habana, cumpliera con los objetivos y expectativas del cliente en conjunto con el equipo de desarrollo, fue necesario diseñar e implementar algoritmos, que respondieran y cumplieran acorde a los resultados deseados. Principalmente para la obtención de la parada más cercana a la posición del usuario, así como para sugerir posibles rutas a abordar por la población para trasladarse en la ciudad. A continuación se describen los algoritmos diseñados e implementados en la solución.

#### Descripción del algoritmo para la obtención de paradas cercanas a la posición del usuario

Para la realización del algoritmo anteriormente mencionado se siguieron los siguientes pasos:

1. Se declaró el método de tipo void, showStop (final int vprovinciaid), el cual es el encargado de mostrar la parada más cercana a la posición del usuario. El mismo recibe por parámetro el identificador de la provincia seleccionada por el usuario.
2. Se capturan las coordenadas del usuario a partir de las coordenadas obtenidas por el GPS.
3. Se crea una lista de municipios.
  - 3.1 Se almacena en la variable puntoMunicipio, el municipio en el cual se encuentra el usuario. Para ello se invoca al método calcularMasCercano (), el cual espera una lista de objetos de tipo municipio.
4. Ejecución del método calcularMasCercano ().
  - 4.1 Se verifica que la lista pasada por parámetros no esté vacía.
  - 4.2 Si la lista contiene solo un elemento, se devuelve el objeto.
  - 4.3 Si no, se itera la lista desde i hasta n, calculando la distancia mínima que existe del punto

## Capítulo 3: Iteraciones y Producción

en el cual se encuentra el usuario a todos los puntos (municipios) de la lista, para finalmente conocer el municipio en el cual se encuentra el usuario.

5. Luego de obtener el municipio en el cual se encuentra el usuario se crea una lista con las paradas pertenecientes a dicho municipio.

5.1 Se invoca al método `calcularMasCercano()`, pasándole la lista de paradas, el funcionamiento es el mismo que se describió anteriormente, pero en este caso se compara el punto de ubicación del usuario con los puntos de cada parada del municipio en el que se encuentra el usuario.

5.2 Se devuelve finalmente el punto (parada), más cercano a la posición del usuario con sus detalles (listado de rutas que pasan por el punto, dirección y el nombre).

### **Descripción del algoritmo para sugerir las posibles rutas a abordar**

1. Se captura la posición actual del usuario a partir del servicio GPS, o porque el usuario marcó en el mapa su ubicación.
2. Se captura la posición del lugar de destino del usuario, una vez que este selecciona a donde desea ir.
3. Luego se buscan las paradas más cercanas a ambos puntos (origen y destino).
4. Se obtienen las listas de rutas correspondientes a cada punto (parada).
  - 4.1 Se iteran ambas listas para buscar rutas que coincidan en ambos puntos.
  - 4.2 Se compara cada elemento  $i$  del arreglo de rutas de la parada origen, con todos los elementos  $k$  del arreglo de rutas de la parada de destino.
  - 4.3 Si el elemento  $i$  del arreglo de rutas del punto origen, coincide con el elemento  $k$  del arreglo de rutas del punto de destino.
  - 4.4 Se adiciona el elemento  $i$  al arreglo resultado.
5. Si el arreglo resultado, tiene elementos repetidos.
6. Eliminar elementos repetidos.
7. Se devuelve un mensaje al usuario con las posibles rutas a abordar.

### **3.4 Patrones de diseño utilizados en la implementación**

Según Craig Larman los patrones de diseño son una pareja de problema-solución al que se le ha asignado un nombre y que es aplicables en varios contextos (35). Estos encapsulan buenos principios y sugerencias ante problemas recurrentes.

Según Tedeschi los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. En otras palabras, “brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares” (40).

Según Roberto Canales Mora un patrón de diseño es una descripción de clases y objetos

## Capítulo 3: Iteraciones y Producción

comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Son a menudo mal interpretados como aplicables sólo a la programación en los grandes sistemas, pero en realidad, se pueden aplicar a la solución de problemas en la programación pequeña como en la implementación de estructuras de datos o algoritmos simples (41).

Después de haber realizado un análisis sobre los criterios de expertos en el tema, se puede resumir, que los patrones de diseño se logran combinar en componentes que resuelven grandes problemas. Así como adaptar a problemas de menor complejidad siempre y cuando se haga un uso correcto de los mismos. Además cabe destacar que existen varios grupos de patrones de diseño, pero la solución se basa en los más usados en función del objetivo que se ha planteado en este trabajo.

### **3.4.1 Patrones GRASP**

Los patrones generales de software para la asignación de responsabilidades (GRASP) describen los principios esenciales de la asignación de responsabilidades a objetos, por tanto fueron considerados en la confección de las clases que componen el modelo de diseño. A continuación se describen brevemente algunos de los más importantes según lo define (Craig Larman) (35).

**Experto:** Define el principio fundamental en virtud del cual se asignan las responsabilidades en un diseño orientado a objetos. Plantea que la responsabilidad debe recaer en la clase que cuenta con la información necesaria para cumplir con ella. De forma tal que el sistema sea más fácil de entender, mantener y ampliar, permitiendo reutilizar los componentes creados en futuras aplicaciones. La **(Figura 18 en el Anexo VIII)** evidencia el uso de este patrón en el desarrollo de la aplicación.

**Creador:** Define quien debe ser el responsable de crear una nueva instancia de una clase. Plantea que debe recaer en la clase que agrega, contiene, registra, utiliza o tiene los datos de inicialización de la nueva instancia. La prioridad en caso de que exista más de una posibilidad es en el orden enunciado. En otras palabras el propósito fundamental de este patrón es asignar responsabilidades relacionadas con la creación de objetos producidos en cualquier evento. Además, facilita un Bajo Acoplamiento, supone menor dependencia respecto al mantenimiento y mejores oportunidades de reutilización. La **(Figura 19 en el Anexo VIII)** muestra cómo fue empleado el patrón creador en la implementación de la aplicación.

**Bajo Acoplamiento:** Define como fomentar una menor dependencia y una mayor reutilización. Plantea mantener la fuerza de las conexiones entre clases lo más bajo posible. Este parámetro es conocido como acoplamiento. No puede verse de forma aislada como Experto o Alta Cohesión. El acoplamiento tal vez no sea muy importante, sino se busca la reutilización. En términos generales, suelen tener escaso acoplamiento las clases muy genéricas y con grandes posibilidades de ser reutilizadas. La **(Figura 20 en el Anexo VIII)** muestra cómo fue empleado el patrón bajo acoplamiento en la implementación de la aplicación.

## Capítulo 3: Iteraciones y Producción

**Alta Cohesión:** Define que las responsabilidades asignadas a una misma clase deben guardar relación y estar enfocadas al mismo objetivo. Fomenta de esta forma que las clases no adquieran una complejidad excesiva o realicen un trabajo enorme, evitando así que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas. La **(Figura 21 en el Anexo VIII)** muestra cómo fue empleado el patrón alta cohesión en la implementación de la aplicación.

**Controlador:** Define quien debería encargarse de atender un evento del sistema. Asigna la responsabilidad de administrar estos mensajes a una clase que represente una de las siguientes opciones: el negocio o la organización global, el sistema, un controlador de papeles o una clase artificial que represente el caso de uso. Además sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (42). La **(Figura 22 en el Anexo VIII)** evidencia el uso de este patrón en la aplicación.

### **3.4.2 Patrones GoF (The Gang of Four)**

La línea base en el tema de patrones de diseño la impone el catálogo “Design Patterns: Elements of Reusable Object-Oriented Software” (43), en este libro se presenta un conjunto de 23 patrones de diseño identificados a partir del estudio y la experiencia del grupo GOF, quienes se dedicaron a analizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupación a partir de dos criterios, su propósito y alcance.

Los patrones Banda de los cuatro (GoF por sus siglas en inglés) describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos que ayudan a realizar tareas complejas. Facilitan el aprendizaje y la comunicación entre programadores y diseñadores. Estos patrones se clasifican en tres tipos: creacionales, estructurales y de comportamiento. (43)

1. Los patrones **Creacionales** se ocupan del proceso de creación de clases y objetos, son los encargados de “abstraer el proceso de instanciación o creación de objetos, ayudan a que el sistema sea independiente de cómo sus objetos son creados, integrados y representados”. (44) Los patrones que forman parte de esta categoría son cinco (5): Método de Fábrica, Fábrica Abstracta, Constructor, Prototipo y Solitario. Dentro de los patrones Creacionales se utilizó:
  - ✓ **Solitario:** Está diseñado para restringir la creación de objetos pertenecientes a una clase o al valor de un único objeto. Su intención consiste en garantizar que una clase solo tenga

## Capítulo 3: Iteraciones y Producción

una instancia y proporcionar un punto de acceso global a ella. Las situaciones más habituales donde podemos ver la aplicación de este patrón son aquellas en las que dicha clase controla a un recurso físico único o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación. En el sistema propuesto se restringe a una sola instancia de las clases GpsManager y RenderTheme, contribuyendo así al ahorro de memoria, elemento muy importante en los dispositivos móviles. A continuación se muestra un ejemplo de la utilización de este patrón en la aplicación. Para ello ver la (**Figura 23 en el Anexo VIII**).

2. Los patrones **Estructurales** separan la interfaz de la implementación y se encargan de guiar la forma en que las clases y los objetos se combinan para formar nuevas estructuras más complejas y proporcionar nuevas funcionalidades. (43) También se encargan de lograr que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones existentes entre objetos. Los patrones pertenecientes a esta categoría son los siguientes: Adaptador, Apoderado, Decorador, Puente, Composición, Fachada y Peso Mosca. Dentro de los patrones Estructurales se utilizó:

✓ **Decorador:** Define una dependencia de uno a muchos entre objetos, tal que, cuando un objeto cambia de estado todos los que son dependientes de este son actualizados automáticamente (43). Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Es aplicado a la generación de vistas. En la aplicación se utiliza este patrón para asegurar que cuando el objeto de tipo GpsManager obtenga una nueva actualización de la ubicación donde se encuentra el usuario, todas las clases interesadas en conocer dicho evento sean actualizadas. A continuación se muestra un ejemplo de la utilización de este patrón en la aplicación. Para ello ver la (**Figura 24 en el Anexo VIII**).

3. Los patrones de **Comportamiento** caracterizan la forma en que las clases o los objetos interactúan y distribuyen las responsabilidades. “Son los encargados de las opciones de comportamiento de la aplicación, permitiendo que el comportamiento varíe en tiempo de ejecución, sin estos patrones entonces sería necesario diseñar e implementar el resto de los comportamientos cada uno por separados” (44). En esta categoría se agrupan once: Intérprete, Mediador, Iterador, Observador, Visitante, Recuerdo, Cadena de Responsabilidad, Acción, Estado, Estrategia y Método de Plantilla. Dentro de los patrones de Comportamiento se utilizó:

✓ **Observador:** Define una dependencia de uno a muchos entre objetos, tal que, cuando un objeto cambia de estado todos los que son dependientes de este son actualizados automáticamente. (43) En la aplicación se utiliza este patrón para asegurar que cuando el objeto de tipo GpsManager obtenga una nueva actualización de la ubicación donde se

## Capítulo 3: Iteraciones y Producción

encuentra el usuario, todas las clases interesadas en conocer dicho evento son actualizadas.

### **3.5 Diseño de la base de datos para la aplicación Ruta Habana**

Al realizar el estudio de los artefactos generados por la metodología de software XP, seleccionada para guiar el proceso de desarrollo del sistema de información para el transporte público de La Habana, queda claro para el equipo de trabajo, que la representación del modelo físico de la base de datos, no es generado por la metodología utilizada. Sin embargo el equipo de trabajo inmerso en esta tarea, consideró realizar una descripción del diseño físico de la base de datos, para que esta información quede archivada y pueda ser consultada por los mismos desarrolladores o por el equipo que posteriormente brinde soporte al sistema.

El modelado de datos permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. Es un lenguaje gráfico capaz de describir conceptos. Los elementos esenciales del diagrama son las entidades, los atributos y las relaciones entre las entidades.

- ✓ Las entidades son objetos de los que el sistema necesita guardar información.
- ✓ Los atributos son las características asociadas a una entidad. Estos pueden ser clasificados en obligatorios, opcionales, claves foráneas y claves primarias.
- ✓ Las relaciones, por su parte, muestran la forma en que dos entidades se asocian. Se representan mediante una línea que une a las dos entidades implicadas.

Para su realización se utilizó como guía el diagrama del modelo del dominio descrito en el anterior capítulo y además se agregaron las tablas: **Recorrido\_Parada** y **Lugar\_Parada**, generadas a partir de la relación mucho a mucho entre las clases: **Recorrido**, **Parada** y **Lugar** respectivamente. Además se utilizó el patrón de diseño llave subrogada, encargado de almacenar un valor numérico único para cada fila de la tabla actuando como una clave sustituta, de forma totalmente independiente a los datos La Figura 14 muestra el diseño de la base de datos.

## Capítulo 3: Iteraciones y Producción

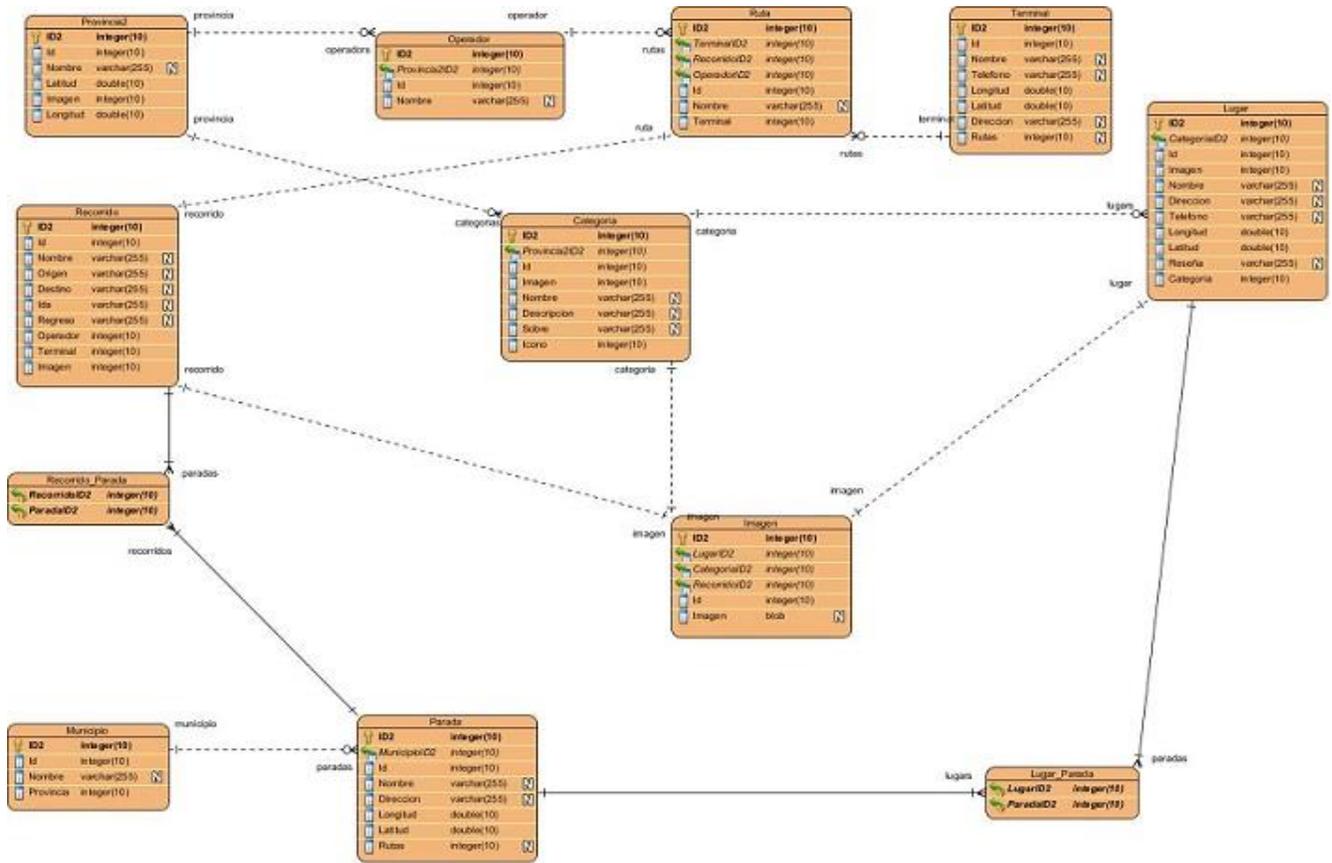


Figura 7: Modelo físico de la Base de Datos.

El desarrollo de cualquier proyecto requiere de un buen diseño de sus clases para de esta forma realizarlo con la mejor calidad posible y así el cliente quede satisfecho. En la metodología XP el diseño de las clases se realiza a través de las tarjetas CRC, para de esta forma ayudar al refinamiento de las clases. De forma organizada cada tarjeta representa una clase, donde se describen las responsabilidades que tiene y las clases colaboradoras que se relacionan con la misma. Además ayudan a diseñar el sistema en conjunto entre todo el equipo de desarrollo, aunque su principal objetivo es propiciar el enfoque orientado a objetos y reducir el modo de pensar procedimental. Están diseñadas en cuatro secciones: nombre de la clase, descripción, responsabilidades y colaboradores.

Una clase describe un objeto o evento del sistema, mediante sus atributos y métodos. Las responsabilidades de estas se describen por las tareas que realiza o por los métodos y los colaboradores son las demás clases con las que interactúa para cumplir con sus responsabilidades.

A continuación se describen algunas de las tarjetas CRC diseñadas para la implementación del sistema Ruta Habana para dispositivos que operan bajo la plataforma Android. Para consultar el resto de las tarjetas ver **Anexo II: Tarjetas CRC**.

**Nombre de la clase:** Terminal

**Descripción:** Se encarga de crear la interfaz donde se muestran todas las terminales de La Habana.

## Capítulo 3: Iteraciones y Producción

Responsabilidad	Colaboradores
Mostrar toda la información referente a las terminales (teléfono, dirección, nombre y ubicación).	Provincia Ruta

Tabla 8: Tarjeta CRC de la clase Terminal.

Nombre de la clase: Categoría	
<b>Descripción:</b> Define los tipos de categoría en los que se puede clasificar un punto de interés.	
Responsabilidad	Colaboradores
Mostrar toda la información referente a las categorías que se puede clasificar un punto de preferencia para el usuario (deporte, cultura, compra, educación entre otros).	Provincia Lugar

Tabla 9: Tarjeta CRC de la clase Categoría.

### 3.6 Tareas de ingenierías realizadas en el desarrollo.

Las Tareas de Ingeniería (TI), se encuentran asociadas fuertemente a las funcionalidades y características que se deben cumplir. Cada tarea pertenece a una HU en específico. Se consideran como la entrada de trabajo para el equipo de programadores. Es una ficha que contiene el número identificador de la tarea, el identificador de la HU con la que está relacionada, el nombre de la tarea, la fecha de inicio, la fecha de fin, el programador responsable y la descripción. También se realizan para llevar a cabo la correcta implementación de las HU descritas por el cliente, que se diseñan en cada una de las iteraciones. Las TI permiten a los desarrolladores obtener un nivel de detalle más avanzado que el propiciado por las HU. El desarrollo del software se planificó en tres iteraciones de trabajo.

#### 3.6.1 Primera Iteración

Durante la primera iteración se abordaron las HU de prioridad media donde se construyó la base arquitectónica del producto con las funcionalidades fundamentales y necesarias que permitieron brindar al cliente, información referente a paradas, recorridos y sitios de interés para ser mostradas y obtener de esa forma una retroalimentación rápida y amplia. En la **Tabla 10** se describen las HU correspondientes a esta iteración, el tiempo estimado y el tiempo real utilizado para la realización de las tareas de ingeniería.

Historia de Usuario	Estimación en días	Real en días
Realizar navegación en el mapa.	5	5
Mostrar el recorrido de las rutas en el mapa.	5	5
Mostrar la ubicación de las terminales en el mapa	2	2

## Capítulo 3: Iteraciones y Producción

Mostrar puntos de interés en el mapa.	3	3
---------------------------------------	---	---

*Tabla 10: HU implementadas en la primera iteración.*

En la **Tabla 11** se encuentran descritas las tareas de ingenierías a implementar por cada HU perteneciente a esta iteración.

Historia de Usuario	Tareas de Ingeniería
Realizar navegación en el mapa.	Implementar la funcionalidad navegación en el mapa.
Mostrar el recorrido de las rutas en el mapa.	Implementar la funcionalidad mostrar recorrido de las rutas en el mapa.
Mostrar la ubicación de las terminales en el mapa	Implementar la funcionalidad mostrar ubicación de las terminales en el mapa.
Mostrar puntos de interés en el mapa.	Implementar la funcionalidad mostrar POI en el mapa.

*Tabla 11: TI a implementar durante la primera iteración.*

A continuación la tabla 16 refleja una muestra de la tarea de ingeniería asociada a la HU “Realizar navegación en el mapa” perteneciente a la primera iteración. El resto de las tareas pueden ser localizadas en el **Anexo III: Tareas de Ingeniería** del presente trabajo.

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	Nombre de la HU: Realizar navegación en el mapa.
<b>Nombre de la tarea:</b> Implementar la funcionalidad navegación en el mapa.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1/1
Fecha de inicio: 21/03/2016	<b>Fecha fin:</b> 25/03/2016
<b>Programador(es):</b> Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, realizar la navegación en el mapa sin dificultad alguna (Realizar zoom en el mapa, desplazar el mapa para ver puntos, direcciones y lugares de interés según la acción de preferencia del usuario).	

*Tabla 12: TI # 1 Implementar la funcionalidad navegación en el mapa.*

### 3.6.2 Segunda Iteración

Durante esta iteración se abordaron las HU de alta prioridad. Al culminar se consta de un producto casi listo para su puesta en funcionamiento con la mayoría de sus funcionalidades más críticas ya implementadas. A continuación se describe en la **Tabla 13** las HU implementadas con la estimación y el tiempo real de cada una:

## Capítulo 3: Iteraciones y Producción

Historia de Usuario	Estimación en días	Real en días
Mostrar ubicación en el mapa.	5	5
Localizar parada más cercana a la posición del usuario.	5	5
Localizar parada más cercana al punto de destino.	2	2
Mostrar puntos de enlaces en una parada.	3	3
Sugerir posibles rutas a abordar.	10	10

Tabla 13: HU Implementadas en la segunda iteración.

En la **Tabla 14** se encuentran descritas las tareas de ingenierías a implementar por cada HU perteneciente a esta segunda iteración.

Historia de Usuario	Tareas de Ingeniería
Mostrar ubicación en el mapa.	Implementar mostrar ubicación en el mapa.
Localizar parada más cercana a la posición del usuario.	Implementar la funcionalidad localizar parada cercana a la posición actual o a una posición marcada en el mapa.
Localizar parada más cercana al punto de destino.	Implementar la funcionalidad localizar parada cercana al destino.
Mostrar puntos de enlaces en una parada.	Implementar la funcionalidad mostrar puntos de enlaces en una parada.
Sugerir posibles rutas a abordar.	Implementar búsqueda de rutas de una parada a otra.

Tabla 14: TI a implementar en la segunda iteración.

A continuación la **Tabla 15** refleja una muestra de la tarea de ingeniería asociada a la HU “Mostrar ubicación en el mapa” perteneciente a la segunda iteración. El resto de las tareas pueden ser localizadas en el **Anexo III: Tareas de Ingeniería** del presente trabajo.

Tarea de Ingeniería	
Número de tarea: 5	Nombre de la HU: Mostrar ubicación en el mapa.
Nombre de la tarea: Implementar mostrar ubicación en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/1
Fecha de inicio: 11/04/2016	Fecha fin: 15/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación y activado el GPS. Obtener su localización actual y real en el mapa. Inclusive aunque el usuario este en constante movimiento. Para ello será necesario hacer uso de la clase GpsManager y obtener los permisos de los servicios de localización.	

Tabla 15: TI # 5 Implementar mostrar ubicación en el mapa.

## Capítulo 3: Iteraciones y Producción

### 3.6.3 Tercera Iteración

En esta iteración serán implementadas las HU de la tercera iteración cuyo resultado final será un producto listo para su funcionamiento. A continuación se describe en la **Tabla 16** las HU implementadas con la estimación y el tiempo real de cada una:

Historia de Usuario	Estimación en días	Real en días
Realizar llamada a la terminal.	2	2
Cambiar idioma.	2	2

*Tabla 16: HU Implementadas en la tercera iteración.*

En la **tabla 17** se encuentran descritas las tareas de ingenierías a implementar por cada HU perteneciente a esta segunda iteración.

Historia de Usuario	Tareas de Ingeniería
Realizar llamada a la terminal.	Implementar realizar llamada a la terminal.
Cambiar idioma.	Implementar cambiar idioma.

*Tabla 17: TI a implementar en la tercera iteración.*

A continuación la **tabla 18** refleja una muestra de la tarea de ingeniería asociada a la HU “Realizar llamada a la terminal” perteneciente a la tercera iteración.

Tarea de Ingeniería	
Número de tarea: 10	Nombre de la HU: Realizar llamada a la terminal.
Nombre de la tarea: Implementar realizar llamada a la terminal.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4/1
Fecha de inicio: 16/05/2016	Fecha fin: 17/05/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación realizar llamadas a las terminales para conocer los horarios de salidas de sus respectivas rutas.	

*Tabla 18: TI # 10 Implementar realizar llamada a la terminal.*

### Conclusiones del capítulo

En este capítulo se realizó la construcción de la solución propuesta para el Sistema de Información para el transporte público de La Habana para dispositivos que operan bajo Android, donde se desarrollaron los principales artefactos a llevar a cabo en el proceso de diseño e implementación para la construcción del sistema. Se definieron los patrones arquitectónicos y de diseño usados con el objetivo de lograr una mayor organización en los elementos que conforman la aplicación. La modelación

## Capítulo 3: Iteraciones y Producción

del modelo de datos permitió obtener una visión detallada del formato físico de la base de datos, de igual forma el diagrama de despliegue permitió una mejor visibilidad de la distribución de los nodos físicos que serán necesarios para el despliegue del sistema. Así como también se confeccionaron las Tarjetas CRC debido a su gran importancia para la implementación de la aplicación.

### **Capítulo 4: “Verificación y Validación de la solución propuesta”**

En el presente capítulo se documenta la fase de prueba según lo propone la metodología de desarrollo XP. Las pruebas llevadas a cabo serán: pruebas unitarias y de aceptación respectivamente. Para de esta forma demostrar que el sistema desarrollado está listo para ser entregado y posteriormente desplegado para su utilización por parte de los usuarios.

#### **4.1 Fase de pruebas al sistema**

La fase de pruebas del sistema permite evaluar la eficacia del mismo, estas son las encargadas de verificar la consumación de los objetivos trazados durante la etapa de implementación del sistema. Para realizar la verificación y validación de un programa de cómputo se llevan a cabo un conjunto de procedimientos, actividades, técnicas y herramientas que se utilizan, paralelamente a su desarrollo asegurando que el producto resuelve el problema inicialmente planteado.

En concordancia con lo descrito por la metodología XP, la cual plantea que la **Fase de Prueba** debe ser constantemente como sea posible, permitiendo un aumento de la calidad del sistema desarrollado y reduciendo el número de errores no detectados (45). XP es una metodología que divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. La estrategia a seguir para la realización de las pruebas al sistema Ruta Habana implementado para dispositivos cuyo sistema operativo sea Android contempla dos niveles de pruebas: pruebas de Unidad y pruebas de Aceptación

#### **4.2 Pruebas Unitarias o Pruebas de Unidad**

Al desarrollar un nuevo software o sistema de información, la primera etapa de pruebas a considerar es la etapa de pruebas unitarias o también llamada pruebas de caja blanca, estas pruebas también son llamadas pruebas modulares ya que permiten determinar si un módulo del programa está listo y correctamente terminado, estas pruebas no se deben confundir con las pruebas informales que realiza el programador mientras está desarrollando el módulo.

XP propone realizar las HU en pequeñas iteraciones para permitir aumentar la calidad del sistema y reducir el número de errores no detectados disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Para la realización de estas pruebas se utilizó el marco de trabajo JUnit 3.0 que está integrado con el SDK de Android. Este marco de trabajo permite realizar la ejecución de clases Java de manera controlada, para poder evaluar el funcionamiento de cada uno de los métodos de la clase.

##### **4.2.1 Primera iteración**

Al culminar la implementación de las funcionalidades propuestas para ser desarrolladas en la primera iteración, las mismas fueron sometidas a un proceso riguroso de pruebas para asegurar que el código

## Capítulo 4: Verificación y Validación de la solución propuesta

propuesto estaba listo para continuar con el desarrollo del sistema, cumpliendo de esta forma con lo que propone la metodología XP, “No pasar a una etapa superior mientras existan errores por corregir” (38). Dichas pruebas arrojaron los siguientes resultados:

### **1. Primera fase**

- ✓ No se mostraba el mapa en la interfaz (Paradas en La Habana).
- ✓ No se mostraba el recorrido de regreso de las rutas.
- ✓ No coincidían los puntos de las paradas en el mapa con la ubicación real de las mismas.
- ✓ No se mostraban los puntos de enlaces entre las rutas.
- ✓ No coincidían la ubicación de las terminales en el mapa con las ubicaciones reales de las mismas.
- ✓ No permitía realizar llamadas a las terminales.
- ✓ No coincidían los POI ubicados en el mapa con la ubicación que debían tener los mismos.
- ✓ No se mostraban los detalles de los puntos de interés.

### **2. Segunda fase**

- ✓ No permitía aplicar el zoom+ al mapa para una mejor visualización de un área seleccionada.
- ✓ No se mostraban las avenidas por las cuales transitaba una ruta durante su recorrido.
- ✓ No permitía realizar las llamadas a las terminales.
- ✓ No se podía acceder a la información de las diferentes rutas pertenecientes a las terminales.

### **3. Tercera fase**

En esta fase fueron corregidos todos los errores, dejando todo listo para seguir con el desarrollo de la aplicación.

#### **4.2.2 Segunda iteración**

Al igual que en la primera iteración se realizaron varias fases de pruebas a la implementación de las funcionalidades propuestas para ser desarrolladas en la segunda iteración, las mismas fueron sometidas a un proceso riguroso de pruebas para asegurar que el código propuesto estaba listo para continuar con el desarrollo del sistema. Dichas pruebas arrojaron los siguientes resultados:

### **1. Primera fase**

- ✓ La aplicación no brindaba acceso a los permisos para activar el GPS.
- ✓ No se mostraba la ubicación actual del usuario en el mapa.

## Capítulo 4: Verificación y Validación de la solución propuesta

- ✓ No se mostraba la parada más cercana a una posición dada por el usuario.
- ✓ No se mostraban los puntos de enlaces en las paradas.
- ✓ No se le mostraba al usuario una posible ruta a coger para trasladarse entre dos puntos seleccionados.

### **2. Segunda fase**

En esta fase fueron corregidos todos los errores, dejando todo listo para seguir con el desarrollo de la aplicación.

#### **4.2.3 Tercera iteración**

Al igual que en las anteriores iteraciones el sistema fue sometido a pruebas para su posterior despliegue. De esta forma todo quedará listo para que la población haga uso de la aplicación Ruta Habana en su versión 1.0.

##### **1. Primera fase**

- ✓ Al seleccionar otro idioma el sistema no realizaba los cambios correspondientes.
- ✓ Al seleccionar la opción llamar la aplicación se reiniciaba.

##### **2. Segunda fase**

En esta fase fueron corregidos todos los errores, dejando todo listo para la puesta en marcha de la aplicación desarrollada. Para ver las pruebas realizadas ver **Anexo IV**.

Las pruebas unitarias realizadas al código fuente del sistema, permitieron entregar un software con las características esperadas, ya que el mismo no presenta errores que puedan ocasionar problemas una vez que sea desplegado y utilizado por los usuarios. Se sugiere ver **gráfico 1 en Anexo IV** el cual muestra los resultados por iteraciones de las pruebas unitarias realizadas.

#### **4.3 Pruebas de Aceptación**

Después de toda la planificación, diseño, reuniones, revisiones de control y las pruebas internas, solo queda un acto final antes de que el cliente acepte el software o solución desarrollada. El equipo ha trabajado meses en tratar de perfeccionar su labor, pero la verdadera prueba para decir que está listo, es cuando sea efectuada la prueba de aceptación con los usuarios reales del sistema.

Las pruebas funcionales o de aceptación son especificadas por el cliente, se centran en los requerimientos funcionales del software y van dirigidas a las características generales y las funcionalidades del sistema. Son realizadas por el usuario final permitiendo la valoración del producto, donde el cliente confirma que las funcionalidades exigidas y descritas en la HU funcionan correctamente (45).

Las pruebas de aceptación se realizaron en diferentes teléfonos con tecnología Android, para evaluar funcionalidad y robustez en diferentes plataformas de hardware y a la vez verificar que el producto

## Capítulo 4: Verificación y Validación de la solución propuesta

cumple con los requerimientos previamente definidos. Las pruebas se llevaron a cabo con los usuarios, dándole un uso real a la aplicación. Los errores detectados fueron reportados como no conformidades. Para el desarrollo de las pruebas se emplearon los casos de prueba de aceptación donde se describe el proceso de prueba a realizar. La **tabla 19** muestra un ejemplo de caso de pruebas realizado a la aplicación, para ver los resultados finales de las iteraciones de las pruebas de aceptación realizadas consultar el **Anexo V: Pruebas de Aceptación**.

Caso de Prueba de Aceptación	
<b>Código:</b> HU1-P1	<b>Historia de usuario:</b> Realizar navegación en el mapa.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario pueda navegar por el mapa, transformando la visualización del mismo, realizando acciones de zoom para obtener una mejor representación de los objetos.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> Verificar que el mapa sea cargado y mostrado al usuario.	
<b>Resultado esperado:</b> La aplicación muestra en las interfaces que hacen uso del mapa (Paradas en La Habana, Puntos de Interés, Recorridos), la visualización del mismo.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 19: Caso de Prueba de Aceptación HU1-P1.

Algunas de las deficiencias detectadas en las iteraciones fueron las que se describen a continuación:

- ✓ No se mostraban con claridad los detalles de los puntos de interés.
- ✓ El seguimiento de los recorridos no llegaba hasta el final.
- ✓ No se mostraba el Marker (Marcador) que define la posición dada por el GPS.
- ✓ No se mostraba el mensaje de las posibles rutas a abordar para llegar al destino.
- ✓ Errores ortográficos en varias interfaces.

Las no conformidades detectadas por el cliente fueron solucionadas en las iteraciones de pruebas. Además se probó la solución propuesta en varias ocasiones, formando el cliente una parte significativa de las pruebas y obteniéndose resultados satisfactorios. Los resultados finales de las iteraciones se resumen en el **grafico 2 de los Anexo V** de la presente investigación.

### 4.4 Impacto de la solución en el campo Ciencia, Tecnología y Sociedad (CTS)

La ciencia y la tecnología constituyen hoy un poderoso pilar del desarrollo cultural, social, económico y, en general, de la vida en la sociedad moderna. A tal punto llega su influencia que la vida actual se ha visto inundada en todos sus aspectos por una creciente avalancha de productos procedentes tanto de una esfera como de la otra, cuya utilización sistemática se ha impuesto como condición para el desarrollo en esta etapa histórica (46).

## Capítulo 4: Verificación y Validación de la solución propuesta

En el momento por el cual está atravesando la sociedad cubana respecto al violento progreso científico y tecnológico, obliga a preparar a las nuevas generaciones para orientarse y actuar en un mundo donde la ciencia y la tecnología se han convertido en un elemento vital de la actividad humana.

Por lo que hoy más que nunca el uso de la ciencia y las tecnologías deben llevarse de la mano, para ayudar al hombre a resolver problemas sociales que en ocasiones limitan su desarrollo. De ahí que la presente investigación juegue un papel primordial para el avance y el crecimiento de la población nacional.

El sistema de información para el transporte público en la capital de Cuba, es de vital importancia para el desarrollo del país. Ya que el mismo está enfocado a resolver un problema que afecta de manera directa a la comunidad, debido a esto se han alineado las tecnologías con el conocimiento y el desarrollo científico del equipo de trabajo, para de una forma u otra erradicar las deficiencias existentes en tan importante renglón, como lo es hoy el transporte para cualquier sociedad. Además La Habana en el año 2016, ha sido declarada como una de las ciudades maravillas del mundo, lo que trae consigo que los porcentajes de visitantes extranjeros tiendan a aumentar en los próximos años, por lo cual se hace necesario contar con una buena infraestructura visual, para que el viajero se sienta a placer en el país, ya que para nada es un secreto que es el sector del turismo un renglón sumamente importante para el desarrollo económico del país.

El sistema Ruta Habana se desarrolló en la UCI, universidad que marca el paso en la informatización de la sociedad cubana. Esta aplicación ha tenido un gran impacto en la población, ya que provee a los usuarios información sobre las rutas e itinerarios de los ómnibus en la capital. Apoyada en la tecnología móvil, Ruta Habana es el resultado de una investigación que abarca todo el campo CTS, el cual engloba tres renglones principales para que toda sociedad alcance un desarrollo sostenible. Por todo lo anteriormente planteado se puede concluir que el producto obtenido en el presente trabajo investigativo, es sin duda un gran aporte a la sociedad cubana.

### **Conclusiones del capítulo**

En este capítulo se documentaron las pruebas realizadas al sistema, para ello se concibió un análisis de los distintos tipos de pruebas existentes, seleccionando las Pruebas Unitarias y las Pruebas de Aceptación como métricas para reducir el número de errores, medir el desempeño, la eficiencia y aumentar la calidad del producto a entregar. Además dichas pruebas son las propuestas por la metodología de desarrollo de software seleccionada para desarrollar el sistema Ruta Habana. Una vez terminada la fase de pruebas se llegó a la conclusión que el sistema, no presenta errores, quedando listo para ser entregado en la fecha acordada con el cliente. Por último se analizó el impacto del sistema de información para el transporte público en el campo ciencia, tecnología y sociedad, quedando demostrado la importancia del mismo para la sociedad cubana.

# Conclusiones Generales

## **Conclusiones Generales**

El desarrollo del presente trabajo de diploma permitió llevar a cabo todas las tareas de la investigación a fin de dar cumplimiento al objetivo general logrando arribar a las siguientes conclusiones:

- ✓ La utilización de los métodos empíricos en especial el método de la encuesta permitió al equipo de desarrollo ratificar la existencia del problema de investigación.
- ✓ La elaboración del marco teórico de la investigación a través de la definición de conceptos fundamentales para su comprensión y el estudio del estado del arte realizado, sobre las soluciones informáticas existentes enfocadas al desarrollo de sistemas de información geográfica, específicamente para dispositivos que operan con Android, permitió al equipo de trabajo identificar posibles funcionalidades para el sistema desarrollado .
- ✓ La correcta selección de la metodología de desarrollo, el lenguaje de modelado y la herramienta CASE a utilizar propició la optimización del capital humano y la correcta representación del sistema a través de los diferentes modelos.
- ✓ Las etapas de Exploración y Planificación definidas por la metodología de desarrollo seleccionada permitieron al equipo de trabajo cumplir con el plan de entrega predefinido, además guiaron el desarrollo por iteraciones, logrando la integración de un producto final completamente funcional.
- ✓ El desarrollo de la aplicación Ruta Habana para la plataforma Android permitió al equipo de trabajo, la creación de un sistema de información para mejorar la comunicación visual del transporte público en La Habana, y con ello contribuir con la informatización de la sociedad cubana.
- ✓ La planificación y puesta en marcha de las pruebas realizadas al sistema permitieron entregar un producto con la calidad requerida, para así de esta forma satisfacer las necesidades de los usuarios.

Por todo lo expuesto, se concluye que el objetivo propuesto para el presente trabajo de diploma se ha cumplido satisfactoriamente, poniendo en práctica todas las tareas presentadas para el desarrollo del sistema de información para el transporte público de La Habana. Cuyo sistema se reconocerá por Ruta Habana en su versión para dispositivos que operan bajo la plataforma Android.

# Recomendaciones

## **Recomendaciones**

Desde el propio comienzo de la investigación se detectó la importancia de un Sistema de Información Geográfica para el transporte público de la capital para dispositivos móviles en este caso que operan bajo la plataforma Android, las metas planteadas con este trabajo se han cumplido y en base a los resultados obtenidos se recomienda:

- ✓ Extender la aplicación al resto del país para de esta forma disminuir la falta de información sobre las rutas del transporte público de cada una de las provincias del país.
- ✓ Agregar a la aplicación las distancias entre paradas y el tiempo promedio de los recorridos de cada ruta.
- ✓ Extender la aplicación Ruta Habana a otras plataformas como iOS y Window Phone, para así aumentar la portabilidad de la misma y cubrir el resto de la población que no cuenta con tecnología Android.

# Bibliografía

## **Bibliografía**

1. Alegsa, Leandro. Diccionario de Informática y Tecnología. [www.alegsa.com.ar](http://www.alegsa.com.ar). [En línea] [Citado el: 2015 de Noviembre de 18.] <http://www.alegsa.com.ar/Dic/software.php>.
2. Brand Speaker, S.L. TICbeat. [tecnologias/android-domina-mercado-mundial-smartphones/ticbeat.com](http://www.ticbeat.com). [En línea] [Citado el: 24 de Noviembre de 2015.] [http://www.ticbeat.com/tecnologias/android-domina-mercado-mundial-smartphones/.](http://www.ticbeat.com/tecnologias/android-domina-mercado-mundial-smartphones/)
3. Montés, Nicolás. Ranking de sistemas operativos mas usados para 2015. Blog de Ingeniería Informática. [En línea] Universidad CEU Cardenal Herrera, 25 de Febrero de 2015. [Citado el: 2 de Diciembre de 2015.] [https://blog.uchceu.es/informatica/ranking\\_de\\_sistemas\\_operativos\\_para\\_2015](https://blog.uchceu.es/informatica/ranking_de_sistemas_operativos_para_2015).
4. Cubadebate. [Online] Directivos del MINTRANS, Julio 22, 2013. [Cited: Septiembre 21, 2015.] <http://www.cubadebate.cu/especiales/2013/07/22/realidades-y-perspectivas-del-transporte-de-pasajeros-en-cuba/>.
5. Hernández León, Rolando Alfredo y Coello González, Sayda. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. La Habana : EDUNIV, 2002. pág. 114. ISBN: 959-16-0343-6.
6. E, Agazzi. El bien, el mal y la ciencia. . Madrid, España. : Tecnos S.A, ,(1996).
7. Alegsa, Leandro. Diccionario de informática y Tecnología. [www.alegsa.com.ar](http://www.alegsa.com.ar). [En línea] 22 de Junio de 2016. [Citado el: 24 de Junio de 2016.] <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>.
8. —. Diccionario de Informática y Tecnología. [www.alegsa.com.ar](http://www.alegsa.com.ar). [En línea] 31 de Julio de 2015. [Citado el: 24 de Febrero de 2016.] <http://www.alegsa.com.ar/Dic/smartphone.php>.
9. Miguel, Luis. [mx.search.yahoo.com](https://mx.search.yahoo.com). [En línea] Yahoo, 19 de Octubre de 2013. [Citado el: 19 de 12 de 2016.] <https://mx.search.yahoo.com/search?p=%C2%BFque+es+sistema+de+inforhttps://mx.answers.yahoo.com/question/index?qid=20091012151754AA2WyR5>.
10. Geoenseñanza. [saber.ula.ve](http://www.saber.ula.ve). [En línea] 1 de Enero de 2006. [Citado el: 13 de Febrero de 2016.] <http://www.saber.ula.ve/bitstream/123456789/27748/1/articulo9.pdf>.
11. Paredes. M.I.M.A.V. «Universidad del Valle - Bolivia,». [En línea] 24 de Junio de 2012. [Citado el: 2016 de Febrero de 17.] <http://www.univalle.edu/publicaciones/brujula/brujula19/pagina15.htm>.
12. Torres, Cristina. [la-verdadera-historia-de-android-nacimiento-del-sistema-operativo-2003/](http://www.androidsis.com). AndroidSIS. [En línea] AB Internet Networks 2008 S.L, 6 de Junio de 2014. [Citado el: 17 de Febrero de 2016.] <http://www.androidsis.com/la-verdadera-historia-de-android-nacimiento-del-sistema-operativo-2003/>.
13. Digital Millennium Copyright Act (DMCA). APPSAPK.Osmand. [En línea] [Citado el: 17 de Febrero de 2016.] <http://www.appsapk.com/osmand-maps-navigation/>.
14. Digital Millennium Copyright Act (DMCA). APPSAPK. Genius Maps – Offline GPS Navigation. [En línea] [Citado el: 2016 de Febrero de 17.] <http://www.appsapk.com/genius-maps-offline-gps-navigation/>.
15. Sygic. [apkpure.com](https://apkpure.com). [En línea] 12 de Febrero de 2016. [Citado el: 17 de Febrero de 2016.] <https://apkpure.com/gps-navigation-maps-sygic/com.sygic.aura>.

# Bibliografía

16. Google Inc. Transit App. [En línea] Google Play, 8 de Febrero de 2016. [Citado el: 17 de Febrero de 2016.] 223. <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>.
17. Cubadebate. [En línea] 30 de Enero de 2015. [Citado el: 12 de Enero de 2016.] <http://www.cubadebate.cu/noticias/2015/01/30/descargue-en-su-movil-el-andariego-un-servicio-de-localizacion-para-cuba/>.
18. Aplicaciones oficiales del metro de Madrid. Metro Madrid. [En línea] 7 de Abril de 2015. [Citado el: 22 de Febrero de 2016.] [https://www.metromadrid.es/es/viaja\\_en\\_metro/APP/index.html](https://www.metromadrid.es/es/viaja_en_metro/APP/index.html).
19. Laguna, Miguel Ángel. Requisitos. 2012.
20. Jacobson, Ivar y Rumbaugh., Grady Booch y James. El Proceso Unificado de Desarrollo de Software. . s.l. : Addison Wesley, 1999.
21. Marroquín, Mónica Lucía Tapia. Estudio y desarrollo de aplicaciones para dispositivos móviles android. Ibarra : s.n., 2013.
- 22 Pressman, Roger S. Pressman. 2007. Vol. VI.
24. Cáceres, Paloma. kybele.etsii.urjc.es. [En línea] 2010. [Citado el: 29 de Marzo de 2016.] [http://www.kybele.etsii.urjc.es/docencia/AIR\\_GIS\\_M/2010-2011/Material/%5BAIR-1011%5DTema4.pdf..](http://www.kybele.etsii.urjc.es/docencia/AIR_GIS_M/2010-2011/Material/%5BAIR-1011%5DTema4.pdf..)
25. Cruz Zapata, Belén . Android Studio Application. BIRMINGHAM - MUMBAI : Packt Publishing, 2013.
26. Amaro, José Enrique. El gran libro de programación avanzada con Android. México : Alfaomega Grupo Editor, S.A. de C.V, 2012. ISBN: 978-607-707-551-6.
27. Komatineni, Satya y MacLean, Dave . Expert Android.
28. Gargenta, Marko y Nakamura, Masumi. Learning Android, Second Edition. United States of America. : O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2014.
29. Wei, Meng Lee. Android™ Application Development. Indianapolis, Indiana : John Wiley & Sons, Inc., 2013. ISBN: 978-1-118-17767-9.
30. Burton, Michael y Felker, Donn. Android™ Application Development For Dummies 2ND EDITION. Hoboken, New Jersey : John Wiley & Sons, Inc., 2012 .
31. Gianluca, Pacchiella. Instant Android Fragmentation Management. BIRMINGHAM - MUMBAI : Packt Publishing, 2013 .
32. Friesen, Jeff. Learn Java for Android Development. New York : Springer Science+Business Media New York, 2013. ISBN 978-1-4302-5722-6.

# Referencias Bibliográficas

## Referencias bibliográficas

1. Alegsa, Leandro. Diccionario de Informática y Tecnología. *www.alegsa.com.ar*. [En línea] [Citado el: 2015 de Noviembre de 18.] <http://www.alegsa.com.ar/Dic/software.php>.
2. Brand Speaker, S.L. TICbeat. *tecnologias/android-domina-mercado-mundial-smartphones/ticbeat.com*. [En línea] [Citado el: 24 de Noviembre de 2015.] [http://www.ticbeat.com/tecnologias/android-domina-mercado-mundial-smartphones/..](http://www.ticbeat.com/tecnologias/android-domina-mercado-mundial-smartphones/)
3. Montés, Nicolás. Ranking de sistemas operativos mas usados para 2015. *Blog de Ingeniería Informática*. [En línea] Universidad CEU Cardenal Herrera, 25 de Febrero de 2015. [Citado el: 2 de Diciembre de 2015.] [https://blog.uchceu.es/informatica/ranking\\_de\\_sistemas\\_operativos\\_para\\_2015](https://blog.uchceu.es/informatica/ranking_de_sistemas_operativos_para_2015).
4. Cubadebate. [Online] Directivos del MINTRANS, Julio 22, 2013. [Cited: Septiembre 21, 2015.] <http://www.cubadebate.cu/especiales/2013/07/22/realidades-y-perspectivas-del-transporte-de-pasajeros-en-cuba/>.
5. Hernández León, Rolando Alfredo y Coello González, Sayda. *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. La Habana : EDUNIV, 2002. pág. 114. ISBN: 959-16-0343-6.
6. E, Agazzi. *El bien, el mal y la ciencia*. Madrid, España. : Tecnos S.A. ,(1996).
7. Alegsa, Leandro. Diccionario de informática y Tecnología. *www.alegsa.com.ar*. [En línea] 22 de Junio de 2016. [Citado el: 24 de Junio de 2016.] <http://www.alegsa.com.ar/Dic/sistema%20informatico.php>.
8. —. Diccionario de Informática y Tecnología. *www.alegsa.com.ar*. [En línea] 31 de Julio de 2015. [Citado el: 24 de Febrero de 2016.] <http://www.alegsa.com.ar/Dic/smartphone.php>.
9. Miguel, Luis. *mx.search.yahoo.com*. [En línea] Yahoo, 19 de Octubre de 2013. [Citado el: 19 de 12 de 2016.] <https://mx.search.yahoo.com/search?p=%C2%BFque+es+sistema+de+inforhttps://mx.answers.yahoo.com/question/index?qid=20091012151754AA2WyR5>.
10. Geoenseñanza. *saber.ula.ve*. [En línea] 1 de Enero de 2006. [Citado el: 13 de Febrero de 2016.] <http://www.saber.ula.ve/bitstream/123456789/27748/1/articulo9.pdf>.
11. Paredes. M.I.M.A.V. «*Universidad del Valle - Bolivia*,». [En línea] 24 de Junio de 2012. [Citado el: 2016 de Febrero de 17.] <http://www.univalle.edu/publicaciones/brujula/brujula19/pagina15.htm>.
12. Torres, Cristina. *la-verdadera-historia-de-android-nacimiento-del-sistema-operativo-2003/AndroidSIS*. [En línea] AB Internet Networks 2008 S.L, 6 de Junio de 2014. [Citado el: 17 de Febrero de 2016.] <http://www.androidsis.com/la-verdadera-historia-de-android-nacimiento-del-sistema-operativo-2003/>.
13. Digital Millennium Copyright Act (DMCA). *APPSAPK.Osmand*. [En línea] [Citado el: 17 de Febrero de 2016.] <http://www.appsapk.com/osmand-maps-navigation/>.
14. Digital Millennium Copyright Act (DMCA). *APPSAPK. Genius Maps – Offline GPS Navigation*. [En línea] [Citado el: 2016 de Febrero de 17.] <http://www.appsapk.com/genius-maps-offline-gps-navigation/>.

## Referencias Bibliográficas

15. Sygic. apkpure.com. [En línea] 12 de Febrero de 2016. [Citado el: 17 de Febrero de 2016.] <https://apkpure.com/gps-navigation-maps-sygic/com.sygic.aura>.
16. Google Inc. *Transit App*. [En línea] Google Play, 8 de Febrero de 2016. [Citado el: 17 de Febrero de 2016.] 223. <https://play.google.com/store/apps/details?id=com.thetransitapp.droid>.
17. Cubadebate. [En línea] 30 de Enero de 2015. [Citado el: 12 de Enero de 2016.] <http://www.cubadebate.cu/noticias/2015/01/30/descargue-en-su-movil-el-andariego-un-servicio-de-localizacion-para-cuba/>.
18. Aplicaciones oficiales del metro de Madrid. *Metro Madrid*. [En línea] 7 de Abril de 2015. [Citado el: 22 de Febrero de 2016.] [https://www.metromadrid.es/es/viaja\\_en\\_metro/APP/index.html](https://www.metromadrid.es/es/viaja_en_metro/APP/index.html).
19. Google Play. *Metro de Madrid*. [En línea] 1.13, Google Inc, 7 de Abril de 2015. [Citado el: 22 de Febrero de 2016.] 5000. [https://play.google.com/store/apps/details?id=es.metromadrid.metroandroid&hl=es\\_419](https://play.google.com/store/apps/details?id=es.metromadrid.metroandroid&hl=es_419).
20. Visitar Paris en Metro. *RATP.FR*. [En línea] RATP. [Citado el: 22 de Febrero de 2016.] [http://www.ratp.fr/es/ratp/r\\_90744/visitar-paris-en-metro/](http://www.ratp.fr/es/ratp/r_90744/visitar-paris-en-metro/).
21. Google Play. *Visitar Paris en Metro*. [En línea] 1.6.12.4 , RATP, 16 de Febrero de 2016. [Citado el: 22 de Febrero de 2016.] 10608. <https://play.google.com/store/apps/details?id=net.ixxi.ratp.tourisme>.
22. *SIG-Rutas*. Ramirez Martín, Carlos Enrique, García Avila, Adrian y Pérez Vazquez, Maridalia. La Habana : s.n., 12 de diciembre de 2011, Serie Científica, pág. 12.
23. Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. *Metodologías Ágiles*. Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo. Trujillo : s.n., 2010. pág. 37.
24. *Change,Extreme Programming Explained*. Beck, Kent y Wesley, Addison. S.L : Embrace, 1999.
25. Gregory, S. Parallel. *Logic Programming in PARLOG. The language and its implementation*. s.l. : Addison Wesley,, 1987.
26. Alegsa, Leandro. Diccionario de informática y tecnologías. *alegsa.com.ar*. [En línea] 12 de Mayo de 2010. [Citado el: 22 de Febrero de 2016.] <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php>.
27. Digital Learning SL. Academia Android. [En línea] 11 de Diciembre de 2014. [Citado el: 3 de Marzo de 2016.] <http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>.
28. JMPergar. Androcode. *MapsForge: OpenStreetMap en Android*. [En línea] Diciembre de 2012. [Citado el: 15 de Febrero de 2016.] <http://code.google.com/p/mapsforge>.
29. Mapsforge.org. OpenStreetMap. [En línea] 25 de Noviembre de 2015. [Citado el: 17 de Febrero de 2016.] <https://wiki.openstreetmap.org/wiki/ES:Mapsforge>.
30. Silberschatz, Abraham , Korth, Henry F y Sudarshan, S. *Fundamentos de Base Datos* . [ed.] Concepción Fernández . Cuarta edición. Madrid : McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U., 2011. pág. 787.
31. Martín Maldonado, Daniel. Empresa&Economía. [En línea] 1 de Julio de 2008. [Citado el: 2 de Diciembre de 2015.] <http://empresayeconomia.republica.com/aplicaciones-para-empresas/sqlite-el-motor-de-base-de-datos-agil-y-robusto.html>.

# Referencias Bibliográficas

32. Visual Paradigm Copyright. UML, BPMN and Software Design Tools for Agile Teams . [En línea] [Citado el: 2015 de Diciembre de 2.] <https://www.visual-paradigm.com/>.
33. Pressman, Roger S. *Ingeniería del Software : Un Enfoque Práctico*. 6ta Edición. 1998.
34. Pressman, Roger S. *Ingeniería de Software*. 2001.
35. Larman, Craig. *UML y Patrones | 2da Edicion*. Murcia : Prentice Hall, 2010.
36. Pressman, Roger S. *Pressman*. 2007. Vols. IV-V.
37. Patricio Letelier, M<sup>a</sup> Carmen Penadés. *Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP)*. Valencia : Universidad Politécnica de Valencia : s.n.
38. Joskowicz, Ing. José. *Reglas y Prácticas en eXtreme Programming*. Virgo, España : s.n., 2008.
39. Márquez Avendaño, Bertha Mariel y Zulaic, José Manuel . Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español. [En línea] 5 de Marzo de 2014. [Citado el: 29 de Marzo de 2016.] [https://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/](https://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/).
40. Tedeschi, Nicolás. ¿Qué es un Patrón de Diseño? *msdn.microsoft.com*. [En línea] 2010. [Citado el: 19 de Mayo de 2016.] <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
41. Mora, Roberto Canales. AdictosAlTrabajo.com. *Patrones de GRASP*. [En línea] [Citado el: 5 de abril de 2016.] <https://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp..>
42. Martínez, Francisco Javier. *Guía de construcción de software en java con patrones de diseño*. Oviedo : s.n., 2011.
43. Gamma, Erich, et al. *Design Patterns. Elements of Reusable Software*. USA : Addison-Wesley, 1995. s.n.
44. Gracia, J. Ingeniero Software. *Patrones de Diseño*. [En línea] 2 de Agosto de 2010. [Citado el: 16 de Marzo de 2016.] <https://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
45. Calabria, Luis y Píriz, Pablo. *Metodología XP*. Uruguay : Universidad ORT Uruguay, 2003.
46. Cañedo Andalia, Rubén. *Ciencia y tecnología en la sociedad. Perspectiva histórico-conceptual*. La Habana : ACIMED, 2001.

# Anexos

## Anexos

### Anexos I: Historias de Usuario

Historia de usuario	
<b>Número:</b> HU-3	<b>Nombre de la historia de usuario:</b> Localizar parada más cercana a la posición del usuario.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Media	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice la parada más cercana a su posición.	
<b>Observaciones:</b>	
<b>Prototipo de Interfaz:</b>	
	

Tabla 20: HU-Localizar parada cercana a la posición de usuario.

Historia de usuario	
<b>Número:</b> HU-4	<b>Nombre de la historia de usuario:</b> Localizar parada más cercana al lugar de destino.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 0.4
<b>Riesgo en desarrollo:</b> Media	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice la parada más cercana al lugar de destino.	
<b>Observaciones:</b>	

# Anexos

Prototipo de Interfaz:



Tabla 21: HU-Localizar parada cercana al lugar de destino del usuario.

Historia de usuario	
<b>Número:</b> HU-5	<b>Nombre de la historia de usuario:</b> Mostrar rutas de enlace en una parada.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 0.6
<b>Riesgo en desarrollo:</b> Media	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice los puntos de enlaces que tiene una ruta en determinada parada, para que este conozca las rutas que pasan por determinada parada.	
<b>Observaciones:</b>	
Prototipo de Interfaz:	

Tabla 22: HU Mostrar rutas de enlace en una parada.

Historia de usuario	
<b>Número:</b> HU-6	<b>Nombre de la historia de usuario:</b> Sugerir posibles rutas a abordar.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 2

# Anexos

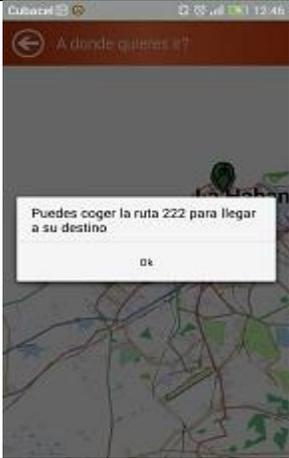
<b>Riesgo en desarrollo:</b> Alta
<b>Descripción:</b> El sistema debe mostrarle un mensaje al usuario con las posibles combinaciones de rutas que puede tomar para trasladarse desde su lugar de origen hacia su destino.
<b>Observaciones:</b>

<b>Prototipo de Interfaz:</b>

Tabla 23: HU-Sugerir posibles rutas a abordar.

Historia de usuario	
<b>Número:</b> HU-7	<b>Nombre de la historia de usuario:</b> Mostrar el recorrido de las rutas en el mapa.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> media	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice los puntos de enlaces que tiene una ruta en determinada parada, para que este conozca las rutas que pasan por determinada parada.	
<b>Observaciones:</b>	
	
<b>Prototipo de Interfaz:</b>	

Tabla 24: HU-Mostrar recorrido de las rutas en el mapa.

Historia de usuario	
<b>Número:</b> HU-8	<b>Nombre de la historia de usuario:</b> Mostrar ubicación de las terminales en el mapa.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 1

# Anexos

<b>Prioridad en el negocio:</b> media	<b>Puntos estimados:</b> 0.4
<b>Riesgo en desarrollo:</b> baja	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice la ubicación de las terminales de la provincia de La Habana.	
<b>Observaciones:</b>	
<p><b>Prototipo de Interfaz:</b></p> 	

Tabla 25: HU-Mostrar ubicación de las terminales en el mapa.

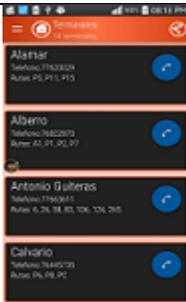
Historia de usuario	
<b>Número:</b> HU-9	<b>Nombre de la historia de usuario:</b> Realizar llamadas a las terminales.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 3
<b>Prioridad en el negocio:</b> baja	<b>Puntos estimados:</b> 0.4
<b>Riesgo en desarrollo:</b> baja	
<b>Descripción:</b> El sistema debe permitir que el usuario visualice los puntos de enlaces que tiene una ruta en determinada parada, para que este conozca las rutas que pasan por determinada parada.	
<b>Observaciones:</b>	
<p><b>Prototipo de Interfaz:</b></p> 	

Tabla 26: HU-Realizar llamadas a las terminales.

Historia de usuario	
<b>Número:</b> HU-10	<b>Nombre de la historia de usuario:</b> Mostrar puntos de interés en el mapa.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 1
<b>Prioridad en el negocio:</b> Media	<b>Puntos estimados:</b> 0.6
<b>Riesgo en desarrollo:</b> Media	

# Anexos

**Descripción:** El sistema debe permitir que el usuario visualice los puntos de enlaces que tiene una ruta en determinada parada, para que este conozca las rutas que pasan por determinada parada.

**Observaciones:**



**Prototipo de Interfaz:**

Tabla 27: HU-Mostrar ubicación de los lugares de interés en el mapa.

Historia de usuario	
<b>Número:</b> HU-11	<b>Nombre de la historia de usuario:</b> Cambiar idioma.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lionys Ruiz Casanova, Pablo Miro Martínez	<b>Iteración Asignada:</b> 3
<b>Prioridad en el negocio:</b> Media	<b>Puntos estimados:</b> 0.4
<b>Riesgo en desarrollo:</b> baja	
<b>Descripción:</b> El sistema debe permitir que el usuario elija el idioma de su preferencia, para de esta forma lograr una mejor interacción sistema-usuario.	
<b>Observaciones:</b>	
<b>Prototipo de Interfaz:</b>	

Tabla 28: HU-Cambiar idioma de la aplicación.

## Anexo II: Tarjetas CRC

Nombre de la clase: Operador

## Anexos

Descripción: Define los tipos de operadores que el usuario puede seleccionar. Estos operadores no son más que los tipos de transporte existente en el transporte público de La Habana.	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Mostrar el tipo de transporte seleccionado por el usuario en un momento determinado (rutero, ferrocarril, metro bus u ómnibus).	Provincia Ruta

*Tabla 29: Tarjeta CRC de la clase Operador.*

Nombre de la clase: Ruta	
Descripción: Contiene toda la información referente a una ruta en específico.	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Describir la el origen, destino y la terminal a la cual pertenece una ruta.	Operador Terminal

*Tabla 30: Tarjeta CRC de la clase Ruta.*

Nombre de la clase: Provincia	
Descripción: Define la provincia a conocer	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Define la provincia para conocer sus operadores, terminales y puntos de interés.	Operador Terminal Categoría

*Tabla 31: Tarjeta CRC de la clase Provincia.*

Nombre de la clase: Recorrido	
Descripción: Define el recorrido que va a tener una determinada ruta.	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Define el recorrido de ida y regreso de una ruta seleccionada.	Parada

*Tabla 32: Tarjeta CRC de la clase Recorrido.*

Nombre de la clase: Lugar	
Descripción: Describe el nombre, la dirección, el teléfono y una breve reseña del lugar a visitar.	
<b>Responsabilidad</b>	<b>Colaboradores</b>
Describe los detalles de un lugar determinado.	Parada Categoría

*Tabla 33: Tarjeta CRC de la clase Lugar*

Nombre de la clase: Parada	
Descripción: Define los detalles de una parada.	

## Anexos

<b>Responsabilidad</b>	<b>Colaboradores</b>
Describe el nombre, la dirección y los puntos de enlace de una parada determinada.	Recorrido Ruta

*Tabla 34: Tarjeta CRC de la clase Parada.*

### **Anexo III: Tareas de Ingeniería**

<b>Tarea de Ingeniería</b>	
Número de tarea: 2	Nombre de la HU: Mostrar el recorrido de las rutas en el mapa.
Nombre de la tarea: Implementar la funcionalidad mostrar recorrido de las rutas en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/1
Fecha de inicio: 28/03/2016	Fecha fin: 01/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, realizar el seguimiento de los recorridos de las rutas disponibles en la aplicación apoyándose para ello en el mapa sin dificultad alguna. Se mostrará tanto la parada siguiente como la anterior del recorrido.	

*Tabla 35: TI # 2 Implementar la funcionalidad mostrar recorrido de las rutas en el mapa.*

<b>Tarea de Ingeniería</b>	
Número de tarea: 3	Nombre de la HU: Mostrar la ubicación de las terminales en el mapa.
Nombre de la tarea: Implementar la funcionalidad mostrar ubicación de las terminales en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4/1
Fecha de inicio: 04/04/2016	Fecha fin: 05/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, ubicar de manera dinámica las terminales en la provincia apoyándose para ello en el mapa sin dificultad alguna.	

*Tabla 36: TI # 3 Implementar la funcionalidad mostrar ubicación de las terminales en el mapa.*

<b>Tarea de Ingeniería</b>	
Número de tarea: 4	Nombre de la HU: Mostrar puntos de interés en el mapa.
Nombre de la tarea: Implementar la funcionalidad mostrar puntos de interés en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6/1
Fecha de inicio: 06/04/2016	Fecha fin: 08/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, ubicar lugares de interés en la provincia apoyándose para ello en el mapa sin dificultad alguna.	

*Tabla 37: TI # 4 Implementar la funcionalidad mostrar puntos de interés en el mapa.*

<b>Tarea de Ingeniería</b>
----------------------------

## Anexos

Número de tarea: 6	Nombre de la HU: Localizar parada más cercana a la posición del usuario
Nombre de la tarea: Implementar mostrar ubicación en el mapa.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/1
Fecha de inicio: 18/04/2016	Fecha fin: 22/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que permitan mostrar la parada más cercana a la posición del usuario, así como también, podrá conocer la parada más cercana a un lugar seleccionado en el mapa.	

*Tabla 38: TI # 6 Implementar localizar parada más cercana.*

<b>Tarea de Ingeniería</b>	
Número de tarea: 7	Nombre de la HU: Localizar parada más cercana al punto de destino.
Nombre de la tarea: Implementar la funcionalidad localizar parada más cercana al punto de destino.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4/1
Fecha de inicio: 25/04/2016	Fecha fin: 26/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, localizar de manera rápida y eficiente la parada más cercana al punto de destino apoyándose para ello en el mapa sin problema alguno.	

*Tabla 39: TI # 7 Implementar la funcionalidad localizar parada más cercana al punto de destino.*

<b>Tarea de Ingeniería</b>	
Número de tarea: 8	Nombre de la HU: Mostrar puntos de enlaces en una parada.
Nombre de la tarea: Implementar la funcionalidad mostrar puntos de enlaces en una parada.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6/1
Fecha de inicio: 27/04/2016	Fecha fin: 29/04/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, mostrar los puntos de enlaces de una parada determinada.	

*Tabla 40: TI # 8 Implementar la funcionalidad mostrar puntos de enlaces en una parada.*

<b>Tarea de Ingeniería</b>	
Número de tarea: 9	Nombre de la HU: Sugerir posibles rutas a abordar.
Nombre de la tarea: Implementar la funcionalidad sugerir posibles rutas a abordar.	
Tipo de tarea: Desarrollo	Puntos estimados: 2/2
Fecha de inicio: 02/05/2016	Fecha fin: 13/05/2016

# Anexos

Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación, sugerir al usuario las rutas que puede abordar para llegar a un destino determinado.

Tabla 41: TI # 9 Implementar la funcionalidad sugerir posibles rutas a abordar.

Tarea de Ingeniería	
Número de tarea: 11	Nombre de la HU: Cambiar Idioma.
Nombre de la tarea: Implementar cambiar idioma.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4/1
Fecha de inicio: 18/05/2016	Fecha fin: 19/05/2016
Programador(es): Pablo Ernesto Miró Martínez y Lionys Julio Ruiz Casanova.	
Descripción: Se implementarán las funcionalidades que le permitan al usuario, una vez puesto en marcha la aplicación seleccionar el idioma de su preferencia.	

Tabla 42: TI # 11 Implementar cambiar idioma.

## Anexo IV: Pruebas Unitarias

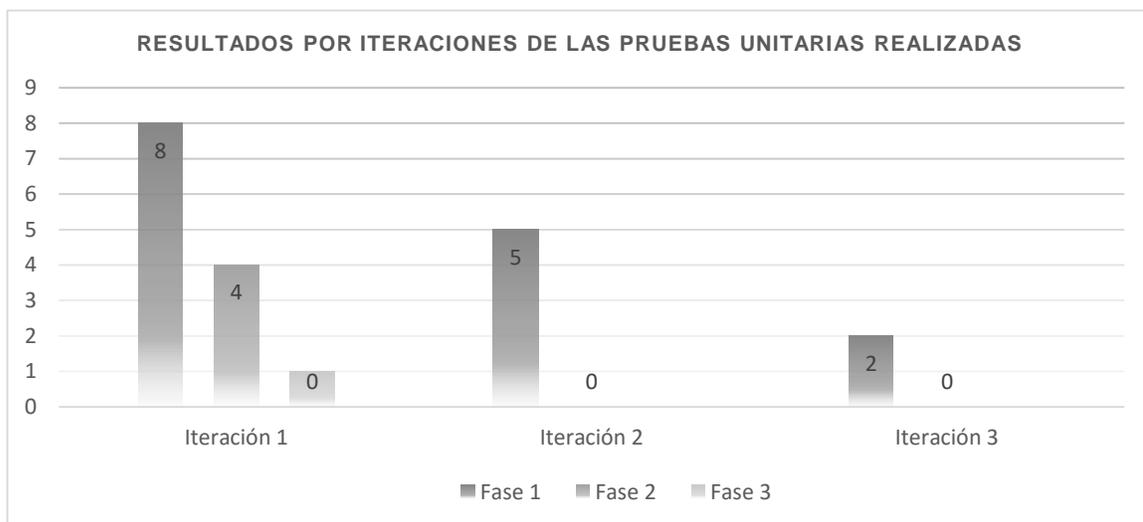


Gráfico 1: Resultado por iteraciones de las pruebas unitarias realizadas.

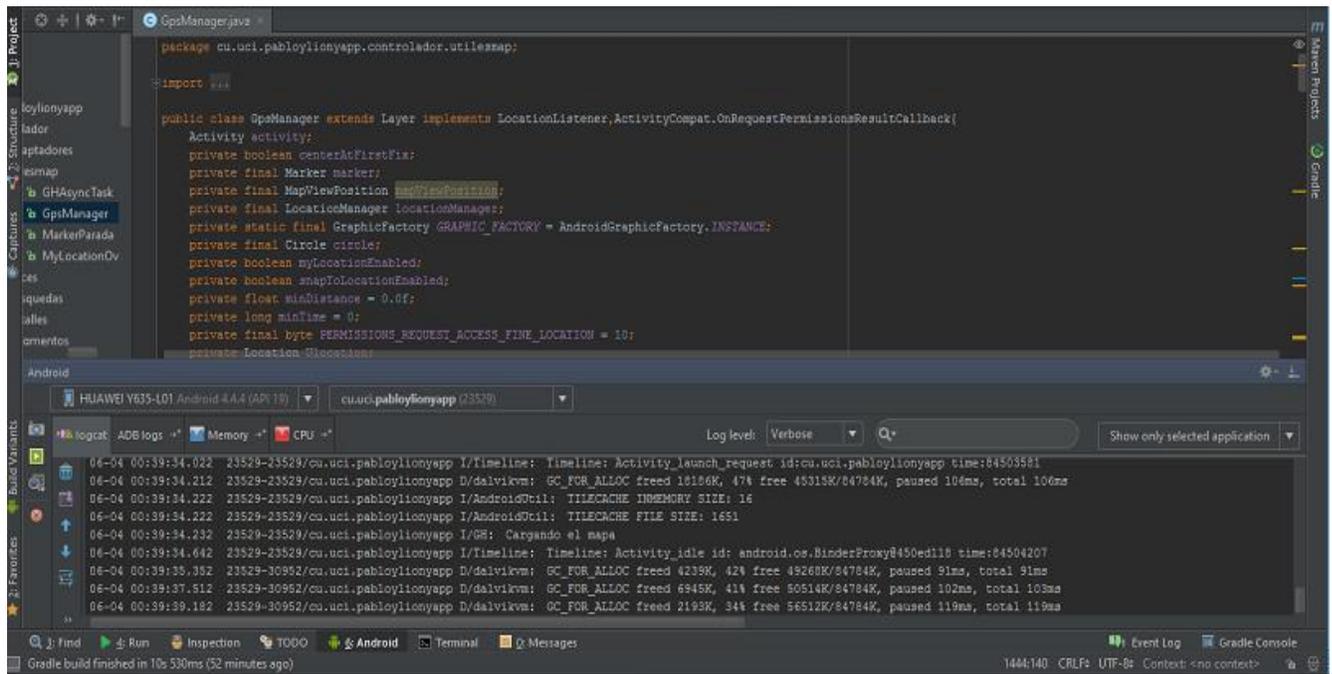


Figura 8: Prueba Unitaria a la funcionalidad Mostrando ubicación en el mapa.

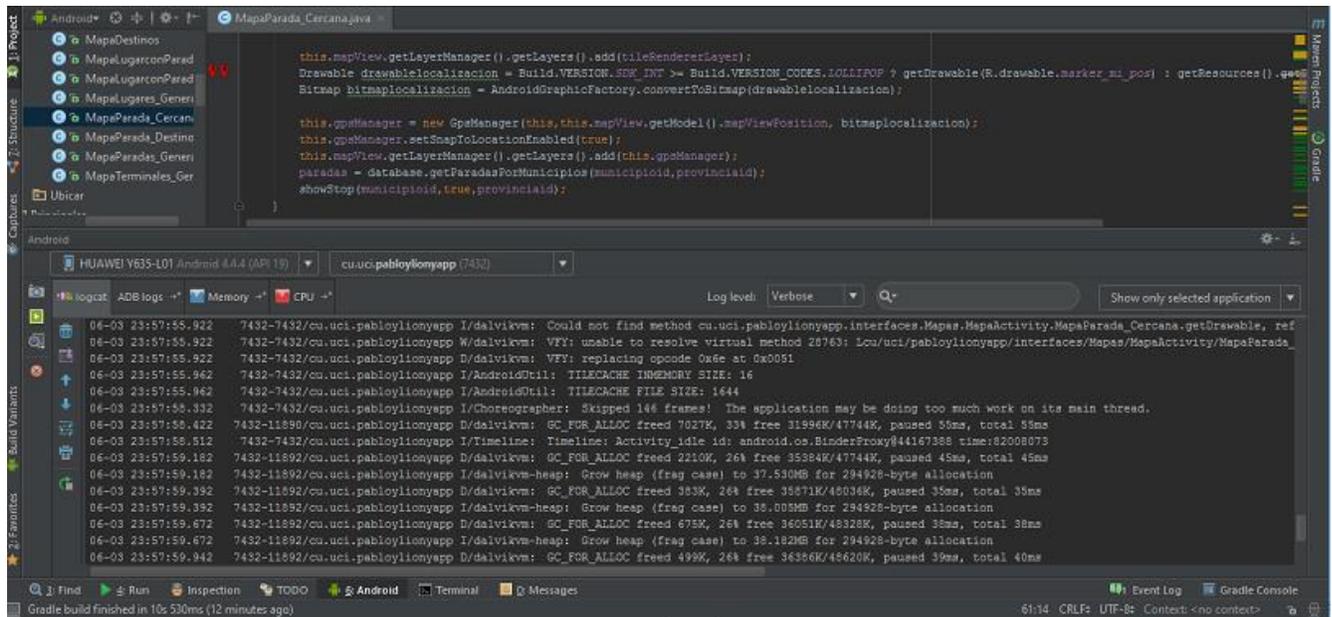


Figura 9: Prueba Unitaria a la funcionalidad mostrar parada más cercana a la posición origen.

# Anexos

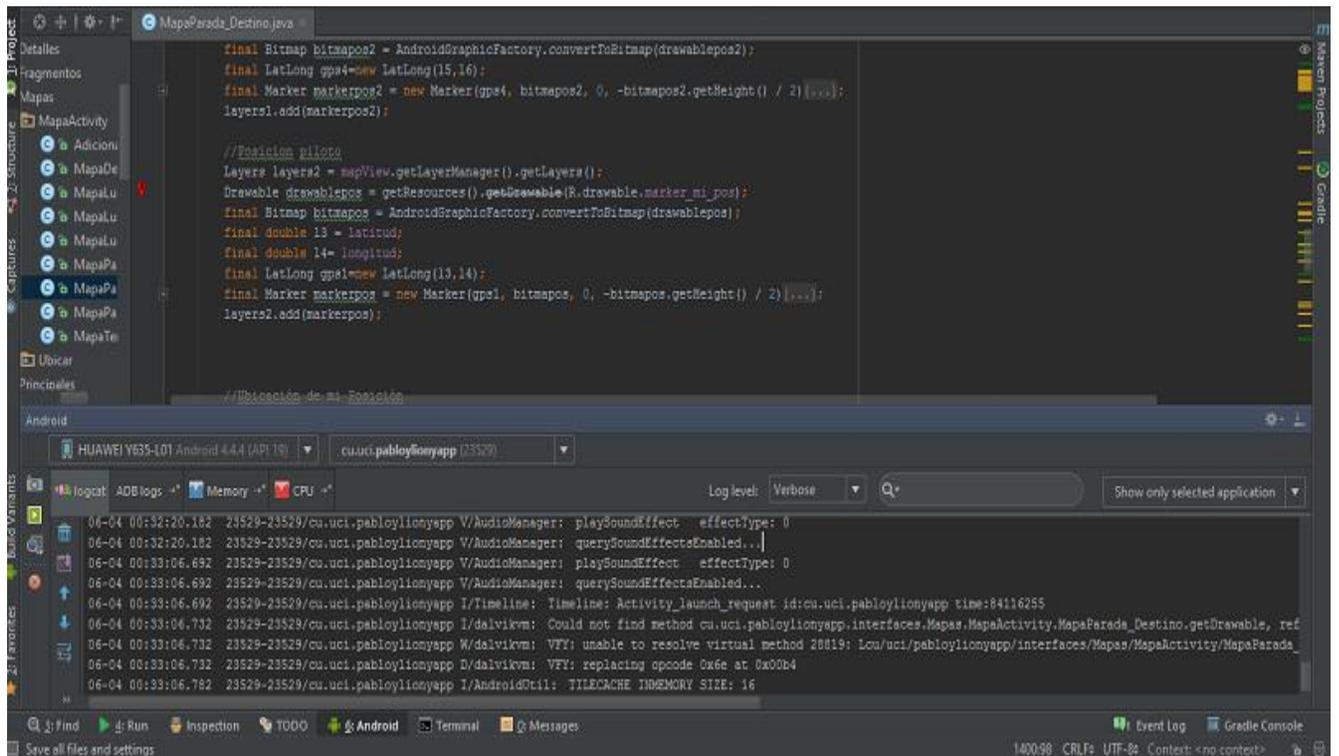


Figura 10: Prueba Unitaria a la funcionalidad mostrar parada más cercana al destino.

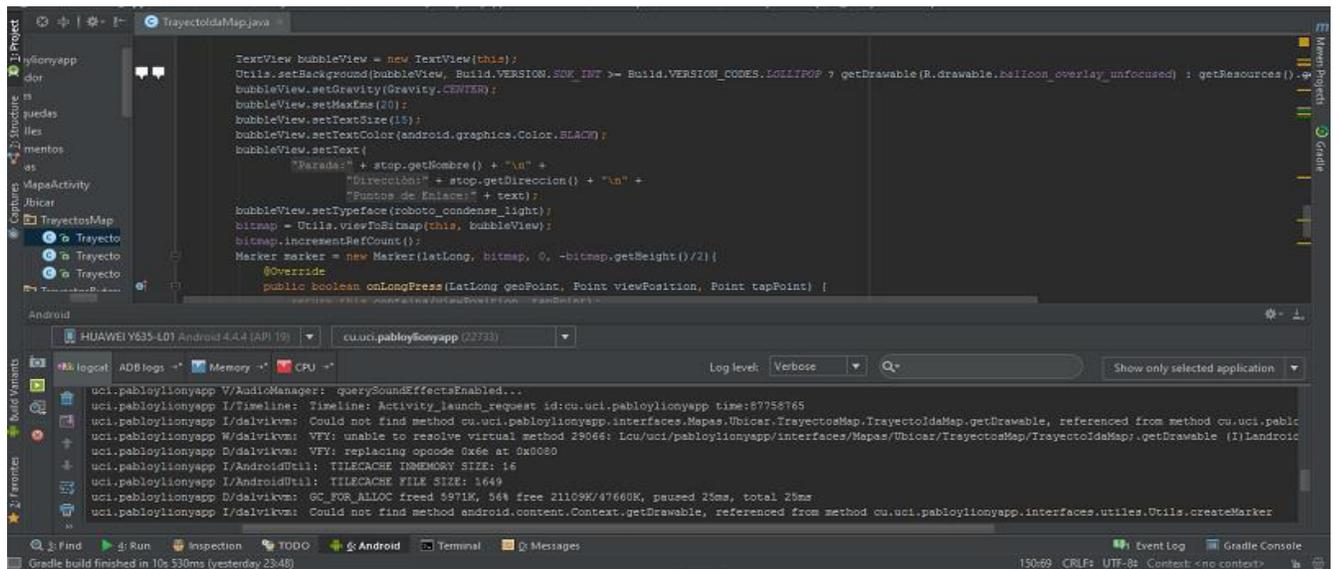


Figura 11: Prueba Unitaria a la funcionalidad mostrar puntos de enlaces en parada.

# Anexos

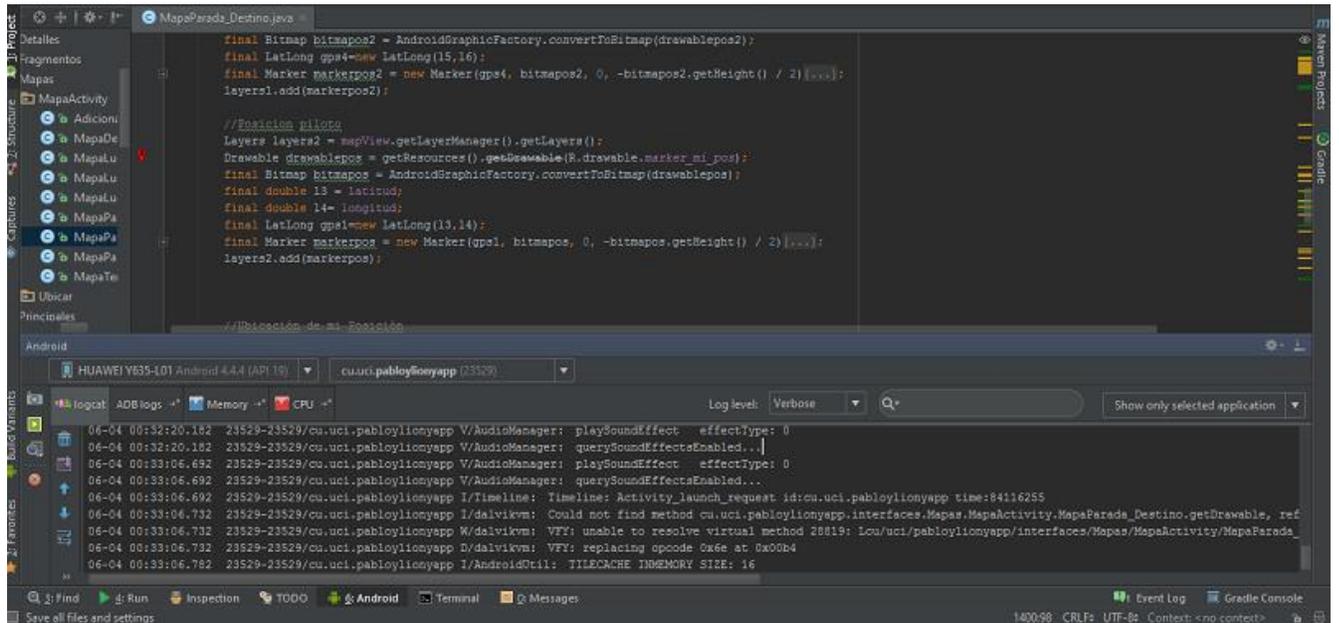


Figura 12: Prueba Unitaria a la funcionalidad sugerir ruta a abordar.

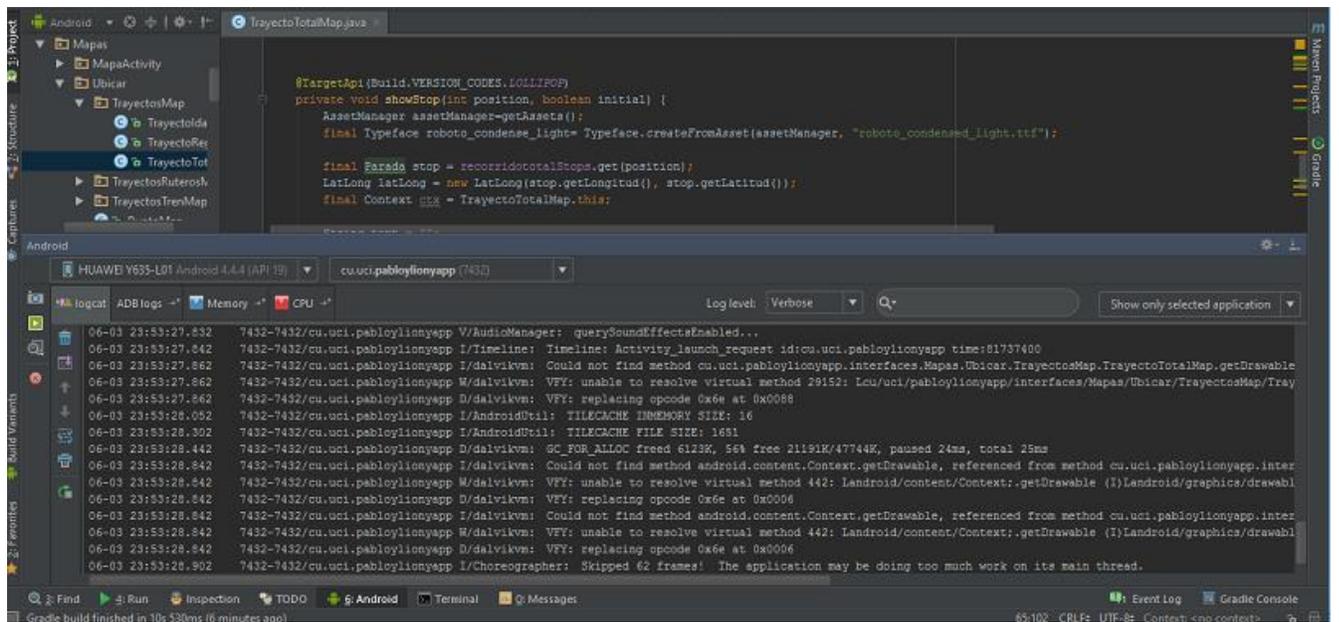


Figura 13: Prueba Unitaria a la funcionalidad mostrar recorrido.

# Anexos

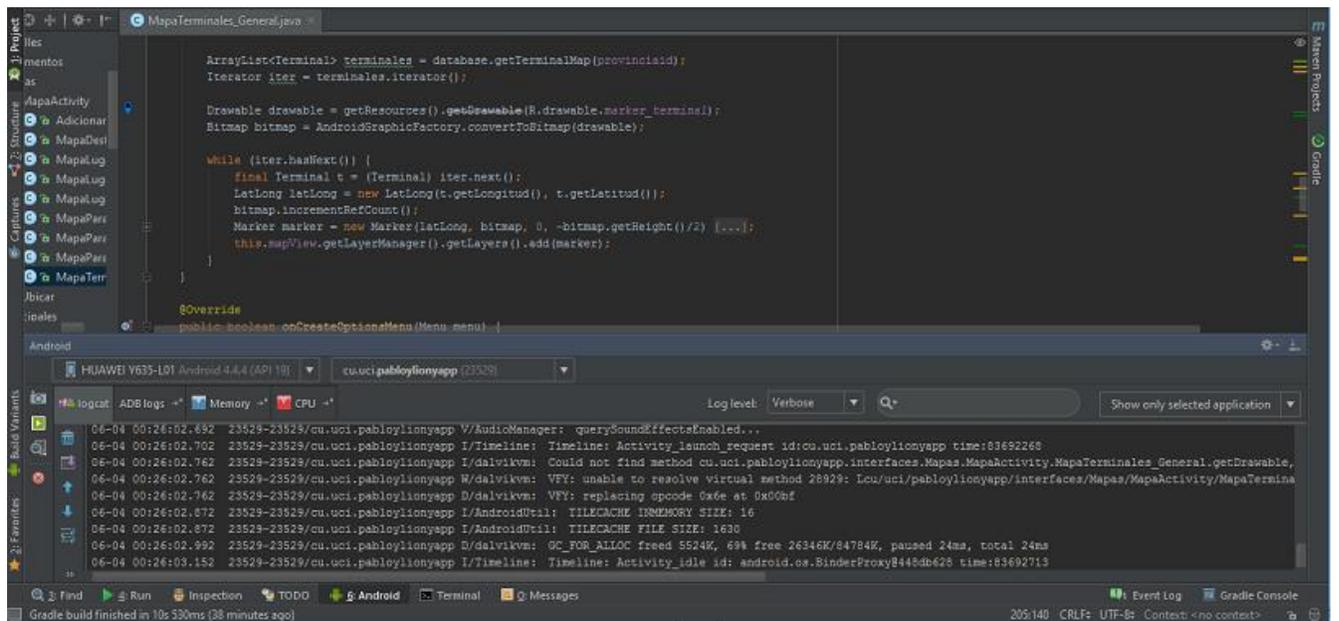


Figura 14: Prueba Unitaria a la funcionalidad mostrar ubicación de terminales en el mapa.

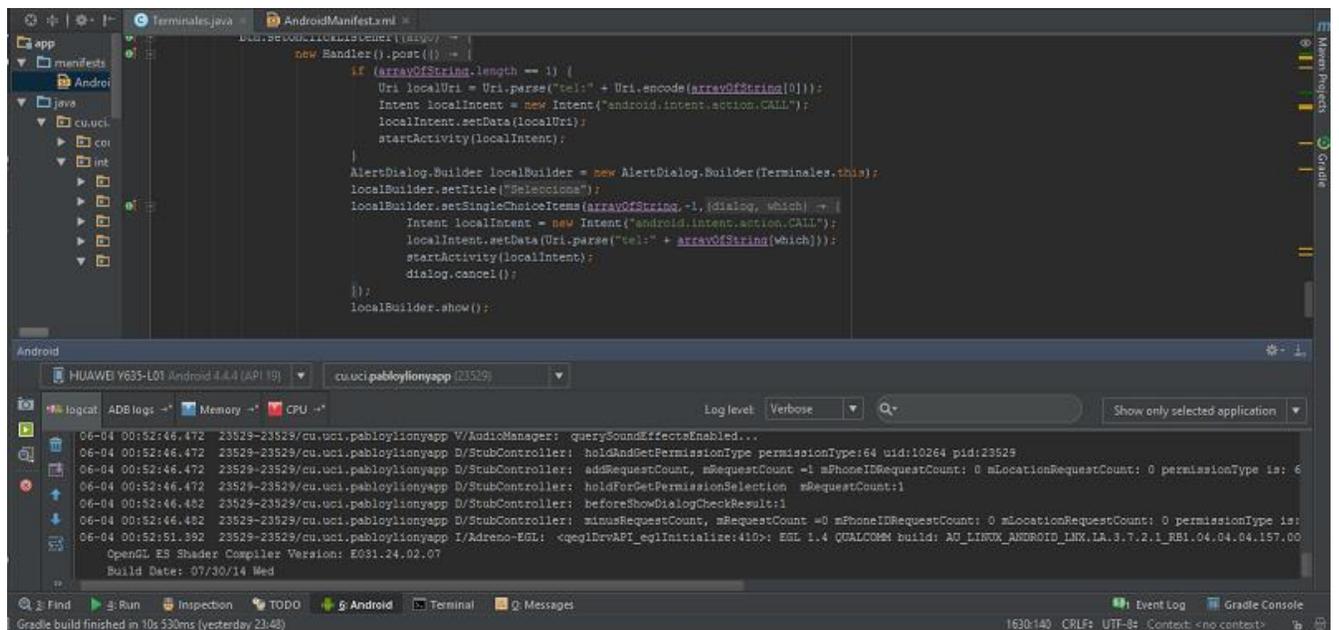


Figura 15: Prueba Unitaria a la funcionalidad realizar llamada a la terminal.

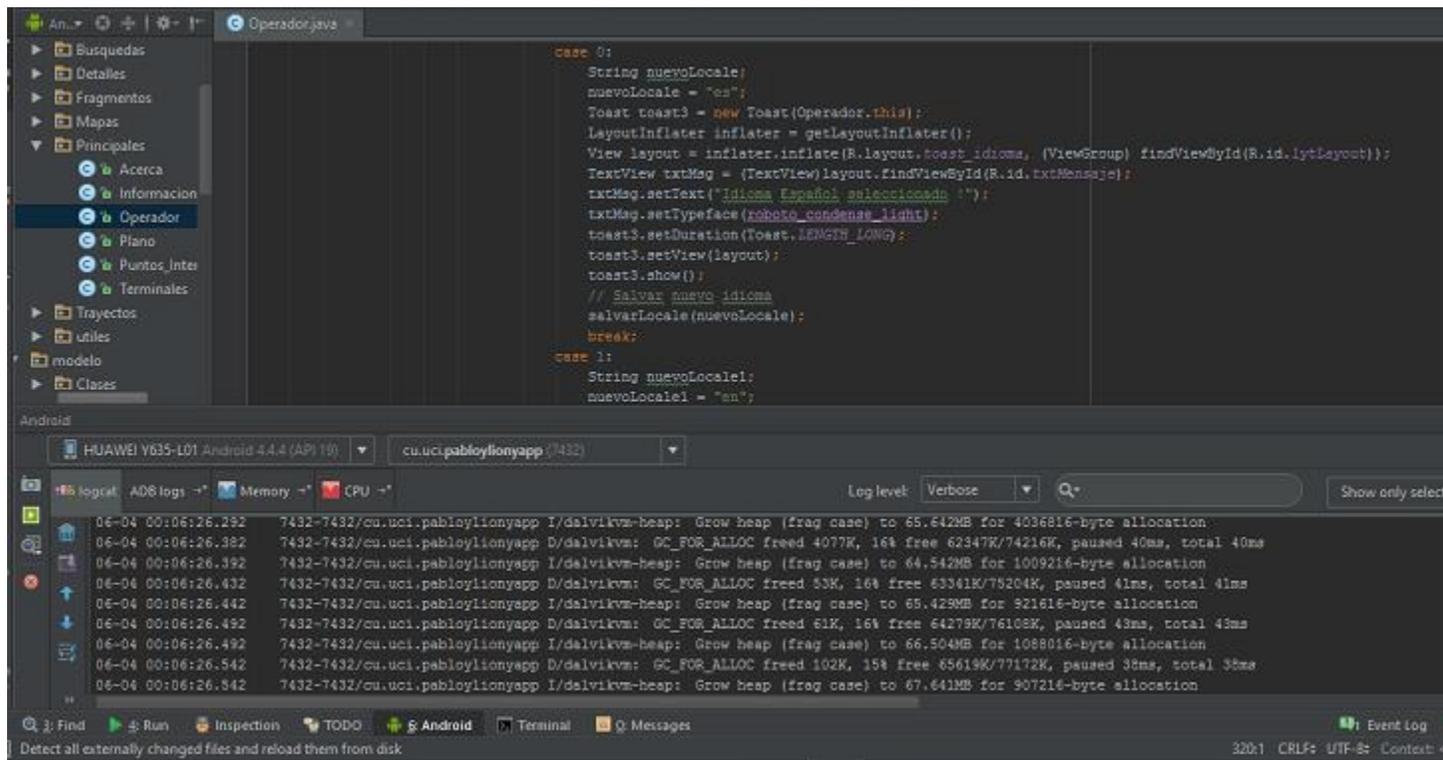


Figura 16: Prueba Unitaria a la funcionalidad cambiar idioma.

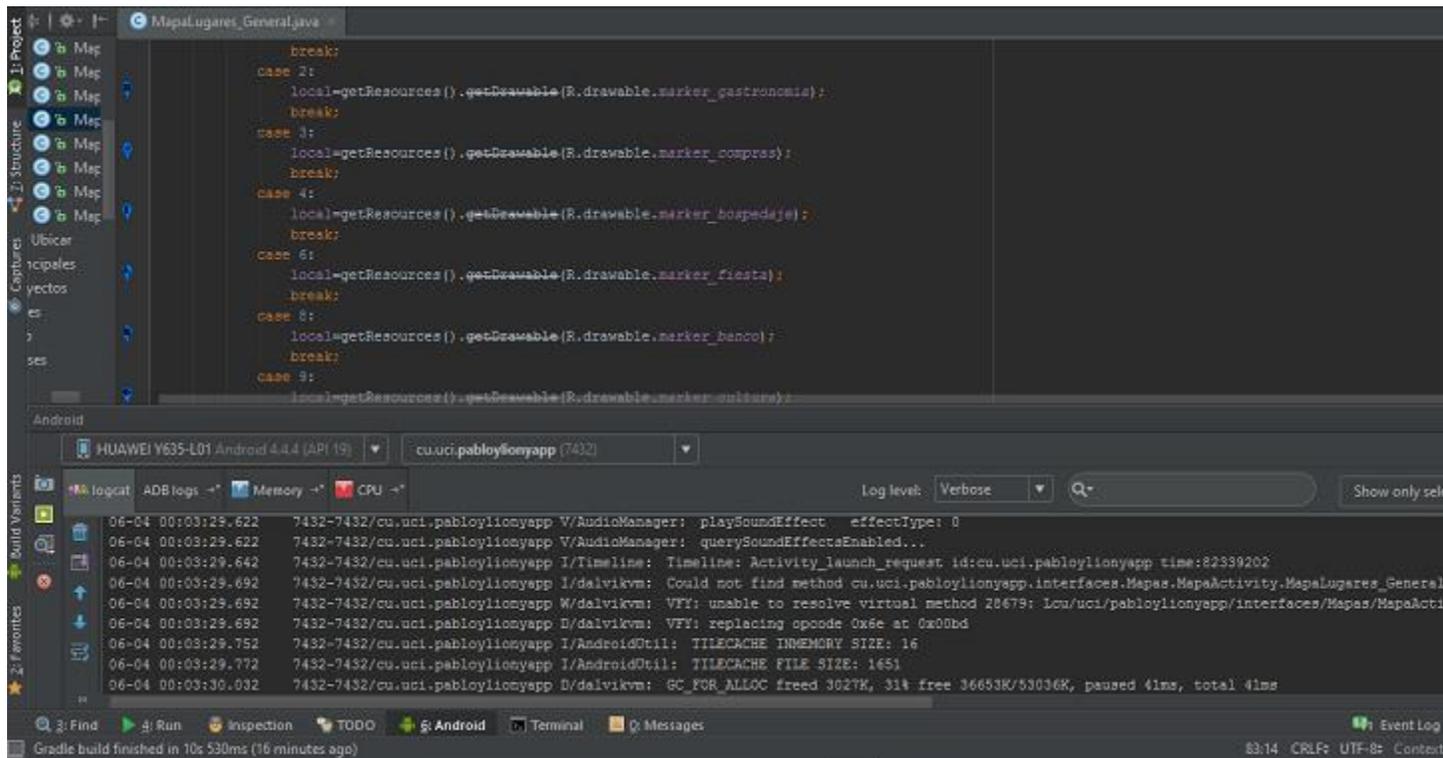


Figura 17: Prueba Unitaria a la funcionalidad mostrar POI en el mapa.

## Anexo V: Pruebas de Aceptación

# Anexos

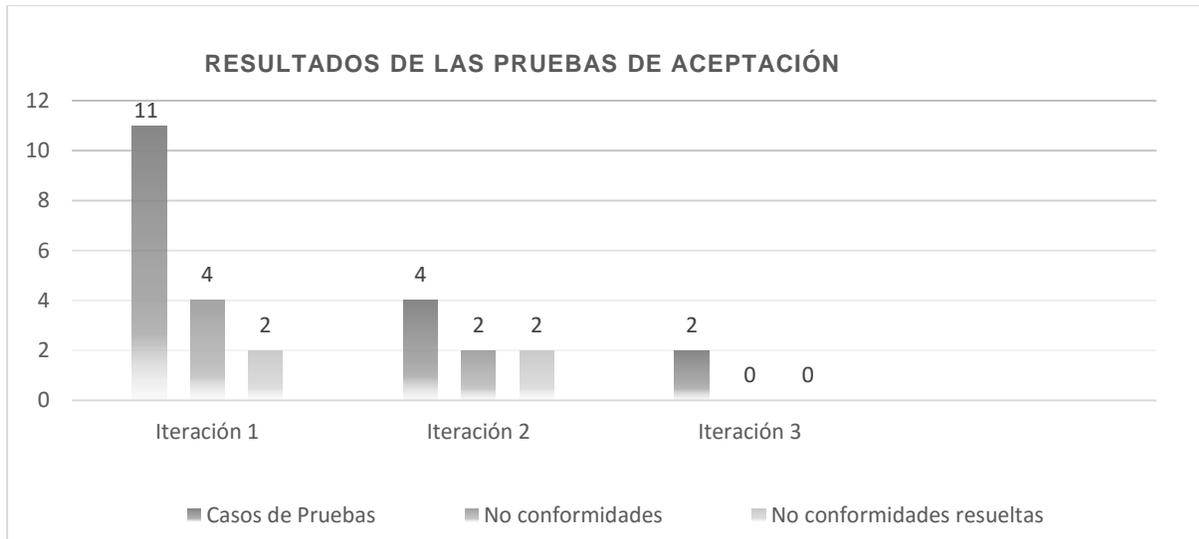


Gráfico 2: Resultados por iteraciones de las pruebas aplicadas.

Caso de Prueba de Aceptación	
<b>Código:</b> HU2-P1	<b>Historia de usuario:</b> Mostrar ubicación en el mapa.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice su posición actual en el mapa, a partir de las coordenadas obtenidas por GPS.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.	
<b>Entrada/Pasos de ejecución:</b> Verificar que el GPS está activado, si no está activo entonces pedirle al usuario que lo active.	
<b>Resultado esperado:</b> La aplicación muestra a través de un marcador la posición obtenida por el GPS.	
<b>Evaluación de la prueba:</b> No satisfactoria. No se visualiza el marcador en el mapa, por lo que se desconoce la posición actual del usuario.	

Tabla 43: Caso de Prueba de Aceptación HU2-P1.

Caso de Prueba de Aceptación	
<b>Código:</b> HU2-P2	<b>Historia de usuario:</b> Mostrar ubicación en el mapa.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice su posición actual en el mapa, a partir de las coordenadas obtenidas por GPS.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.	

## Anexos

<b>Entrada/Pasos de ejecución:</b> Verificar que el GPS está activado, si no está activo entonces pedirle al usuario que lo active.
<b>Resultado esperado:</b> La aplicación muestra a través de un marcador la posición obtenida por el GPS.
<b>Evaluación de la prueba:</b> Satisfactoria.

*Tabla 44: Caso de Prueba de Aceptación HU2-P2.*

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU3-P1	<b>Historia de usuario:</b> Localizar parada más cercana a la posición del usuario.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miro Martínez.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice la parada más cercana a su posición actual a partir de las coordenadas obtenidas por GPS o a una posición seleccionada en el mapa.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.	
<b>Entrada/Pasos de ejecución:</b> Verificar que el GPS está activado en caso de que el usuario necesite conocer su posición actual, si no está activo entonces pedirle al usuario que lo active.	
<b>Resultado esperado:</b> La aplicación muestra a través de un marcador la posición obtenida por el GPS o la posición marcada por el usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 45: Caso de Prueba de Aceptación HU3-P1.*

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4-P1	<b>Historia de usuario:</b> Localizar parada más cercana al punto de destino.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice la parada más cercana a una posición seleccionada en el mapa.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario presiona un punto en el mapa, se lanza un mensaje solicitando el lugar de destino, luego el usuario presiona un punto hacia dónde quiere llegar y el sistema le mostrará la parada más cerca a ese punto previamente seleccionado.	
<b>Resultado esperado:</b> La aplicación muestra a través de un marcador la parada más cercana al usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 46: Caso de Prueba de Aceptación HU4-P1.*

## Anexos

Caso de Prueba de Aceptación	
<b>Código:</b> HU5-P1	<b>Historia de usuario:</b> Mostrar rutas de enlaces en una parada.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice todas las rutas con las que se encuentra en una parada, el medio de transporte público en el cual está haciendo el recorrido el usuario.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona un medio de transporte (Rutero, Ferrocarril u Ómnibus), para realizar su viaje. A medida que se vaya trasladando, la aplicación le mostrará en cada parada las posibles rutas, con las que puede realizar trasbordos en dicha parada.	
<b>Resultado esperado:</b> La aplicación muestra a través de un mensaje, situado sobre cada parada, los diferentes puntos de enlace entre las otras rutas y la ruta en la que viaja el usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 47: Caso de Prueba de Aceptación HU5-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU6-P1	<b>Historia de usuario:</b> Sugerir posibles rutas a abordar.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miró Martínez.	
<b>Descripción:</b> El Sistema debe mostrar un mensaje, con las posibles combinaciones de rutas que debe abordar el usuario para llegar a su destino.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona su lugar de destino o lo obtiene por el GPS, luego selecciona un punto en el mapa hacia donde quiere dirigirse. Mostrándose el mensaje con las posibles rutas a abordar.	
<b>Resultado esperado:</b> La aplicación muestra a través de un mensaje las posibles combinaciones de rutas a abordar por el usuario para llegar desde un punto a otro seleccionado por el mismo.	
<b>Evaluación de la prueba:</b> No Satisfactoria. No se muestra ningún mensaje al usuario.	

*Tabla 48: Caso de Prueba de Aceptación HU6-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU6-P2	<b>Historia de usuario:</b> Sugerir posibles rutas a abordar.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miró Martínez.	

## Anexos

<b>Descripción:</b> El Sistema debe mostrar un mensaje, con las posibles combinaciones de rutas que debe abordar el usuario para llegar a su destino.
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona su lugar de destino o lo obtiene por el GPS, luego selecciona un punto en el mapa hacia donde quiere dirigirse. Mostrándose el mensaje con las posibles rutas a abordar.
<b>Resultado esperado:</b> La aplicación muestra a través de un mensaje las posibles combinaciones de rutas a abordar por el usuario para llegar desde un punto a otro seleccionado por el mismo.
<b>Evaluación de la prueba:</b> No Satisfactoria. Solo se muestra la combinación para llegar de forma directa.

*Tabla 49: Caso de Prueba de Aceptación HU6-P2.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU6-P3	<b>Historia de usuario:</b> Sugerir posibles rutas a abordar.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miró Martínez.	
<b>Descripción:</b> El Sistema debe mostrar un mensaje, con las posibles combinaciones de rutas que debe abordar el usuario para llegar a su destino.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente y el GPS del dispositivo tiene que haber estado activo.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona su lugar de destino o lo obtiene por el GPS, luego selecciona un punto en el mapa hacia donde quiere dirigirse. Mostrándose el mensaje con las posibles rutas a abordar.	
<b>Resultado esperado:</b> La aplicación muestra a través de un mensaje las posibles combinaciones de rutas a abordar por el usuario para llegar desde un punto a otro seleccionado por el mismo.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 50: Caso de Prueba de Aceptación HU6-P3.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU7-P1	<b>Historia de usuario:</b> Mostrar el recorrido de las rutas en el mapa.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miró Martínez.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice el recorrido completo de una ruta seleccionada, así como permitir que el usuario realice un seguimiento de los recorridos, tanto el de ida como el de regreso.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	

## Anexos

<b>Entrada/Pasos de ejecución:</b> El usuario selecciona una ruta, luego se le muestran todas las paradas que realiza dicha ruta y una vez seleccionada una de las paradas la aplicación visualizará la parada seleccionada y el resto del recorrido.
<b>Resultado esperado:</b> La aplicación mostrará el recorrido de ida y regreso de una ruta seleccionada por el usuario.
<b>Evaluación de la prueba:</b> No satisfactoria. No se muestra el recorrido completo de las rutas.

*Tabla 51: Caso de Prueba de Aceptación HU7-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU7-P2	<b>Historia de usuario:</b> Mostrar el recorrido de las rutas en el mapa.
<b>Nombre de la persona a cargo:</b> Pablo Ernesto Miró Martínez.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice el recorrido completo de una ruta seleccionada, así como permitir que el usuario realice un seguimiento de los recorridos, tanto el de ida como el de regreso.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona una ruta, luego se le muestran todas las paradas que realiza dicha ruta y una vez seleccionada una de las paradas la aplicación visualizará la parada seleccionada y el resto del recorrido.	
<b>Resultado esperado:</b> La aplicación mostrará el recorrido de ida y regreso de una ruta seleccionada por el usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 52: Caso de Pruebas de Aceptación HU7-P2.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU8-P1	<b>Historia de usuario:</b> Mostrar la ubicación de las terminales en el mapa.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice la ubicación de todas las terminales del transporte público de La Habana.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Terminales, después selecciona botón en el borde derecho superior, el cual visualiza un mapa. Mostrándose de esta forma al usuario la posición de todas las terminales en el mapa.	
<b>Resultado esperado:</b> La aplicación mostrará la posición de todas las terminales en el mapa.	

## Anexos

**Evaluación de la prueba:** Satisfactoria.

*Tabla 53: Caso de Pruebas de Aceptación HU8-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU9-P1	<b>Historia de usuario:</b> Realizar llamada a la terminal.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario realice llamadas a la terminal desde la misma aplicación.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Terminales, después presiona el botón situado a la derecha del nombre de la terminal, el cual visualiza un teléfono. Luego se le muestra al usuario los teléfonos existentes en la terminal seleccionada. Finalmente el usuario selecciona un número a marcar, iniciándose de esta forma una llamada a la terminal seleccionada.	
<b>Resultado esperado:</b> La aplicación permite que el usuario realice una llamada telefónica a la terminal seleccionada.	
<b>Evaluación de la prueba:</b> No satisfactoria. No se muestra el botón para realizar la llamada.	

*Tabla 54: Caso de Prueba de Aceptación HU9-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU9-P2	<b>Historia de usuario:</b> Realizar llamada a la terminal.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario realice llamadas a la terminal desde la misma aplicación.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Terminales, después presiona el botón situado a la derecha del nombre de la terminal, el cual visualiza un teléfono. Luego se le muestra al usuario los teléfonos existentes en la terminal seleccionada. Finalmente el usuario selecciona un número a marcar, iniciándose de esta forma una llamada a la terminal seleccionada.	
<b>Resultado esperado:</b> La aplicación permite que el usuario realice una llamada telefónica a la terminal seleccionada.	
<b>Evaluación de la prueba:</b> No satisfactoria. Al presionar el botón llamar la aplicación no muestra ningún listado de posibles números a llamar.	

*Tabla 55: Caso de Prueba de Aceptación HU9-P2.*

**Caso de Prueba de Aceptación**

## Anexos

<b>Código:</b> HU9-P3	<b>Historia de usuario:</b> Realizar llamada a la terminal.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario realice llamadas a la terminal desde la misma aplicación.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Terminales, después presiona el botón situado a la derecha del nombre de la terminal, el cual visualiza un teléfono. Luego se le muestra al usuario los teléfonos existentes en la terminal seleccionada. Finalmente el usuario selecciona un número a marcar, iniciándose de esta forma una llamada a la terminal seleccionada.	
<b>Resultado esperado:</b> La aplicación permite que el usuario realice una llamada telefónica a la terminal seleccionada.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 56: Caso de Prueba de Aceptación HU9-P3.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU10-P1	<b>Historia de usuario:</b> Mostrar puntos de interés en el mapa.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario visualice la ubicación de los puntos de interés registrados en el sistema.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	
<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Puntos de interés, después selecciona botón en el borde derecho superior, el cual visualiza un mapa. Mostrándose de esta forma al usuario la posición de todas los puntos de interés en el mapa.	
<b>Resultado esperado:</b> La aplicación mostrará la ubicación en el mapa de todos los puntos de interés registrados en la aplicación.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

*Tabla 57: Caso de Prueba de Aceptación HU10-P1.*

Caso de Prueba de Aceptación	
<b>Código:</b> HU11-P1	<b>Historia de usuario:</b> Cambiar Idioma.
<b>Nombre de la persona a cargo:</b> Lionys Julio Ruiz Casanova.	
<b>Descripción:</b> El Sistema debe permitir que el usuario elija el idioma de su preferencia.	
<b>Condiciones de ejecución:</b> La aplicación debe haber sido instalada e iniciada previamente.	

## Anexos

<b>Entrada/Pasos de ejecución:</b> El usuario selecciona en el menú izquierdo de la aplicación la opción Cambiar idioma, se le mostrará un diálogo con los idiomas con que cuenta la aplicación.
<b>Resultado esperado:</b> La aplicación se iniciará en el idioma seleccionado por el usuario.
<b>Evaluación de la prueba:</b> Satisfactoria.

*Tabla 58: Caso de Prueba de Aceptación HU11-P1*

# Anexos

## **Anexo VI: Modelo de la encuesta**

Estudio para proyecto de diploma de la Universidad de las Ciencias Informáticas, La Habana, Cuba:

### **Encuesta**

Buenos, estamos realizando una encuesta para evaluar la practicidad y éxito de un proyecto de diploma. Le agradecemos por brindarnos un minuto de su tiempo, por favor responda con sinceridad, señalando con una X su respuesta, una sola vez por cada pregunta:

**Sexo:** F ( ) M ( )

1. ¿Con qué frecuencia utiliza usted un ómnibus público como medio de transporte para trasladarse en la ciudad?  
Diariamente ( ) 2 a 3 veces por semana ( ) Casi nunca ( )
2. ¿Conoce usted los itinerarios completos de los ómnibus en los que acostumbra a viajar?  
Algunos ( ) No ( ) Si ( )
3. ¿Cree usted que las paradas de ómnibus poseen la información correcta o necesaria, para ubicar al medio de transporte que debe tomar?  
Sí ( ) No ( )
4. ¿El ómnibus en el que usted acostumbra a viajar, tiende a llegar en un tiempo específico?  
Sí ( ) No ( ) a veces ( )
5. ¿Alguna vez, al utilizar un ómnibus público como su medio de transporte, tomó el equivocado y tuvo inconvenientes por la falta de conocimiento e información?  
Sí ( ) No ( )
6. ¿Cuántas veces se ha equivocado?  
1 vez ( ) 4 veces ( ) más de 5 veces ( )
7. ¿Considera usted que la infraestructura y la comunicación visual (señalizaciones) de las paradas de ómnibus son?  
Suficiente ( ) insuficiente ( ) muy insuficiente ( )
8. ¿Es usted poseedor de un Smartphone, o llamado también teléfono inteligente?  
Sí ( ) No ( )
9. ¿Qué sistema operativo utiliza su dispositivo inteligente?  
Android ( ) iOS ( ) Window Phone ( ) Otros ( )
10. ¿Utilizaría usted una aplicación móvil; que le permita informarse sobre las rutas, tiempo de espera, costo y ubicación geográfica de las paradas de ómnibus más cercanas a su ubicación?  
Sí ( ) No ( )
11. ¿Por qué? \_\_\_\_\_.

Muchas Gracias...!!!

# Anexos

## Anexo VII: Resultados de la entrevista

### Pregunta 1:

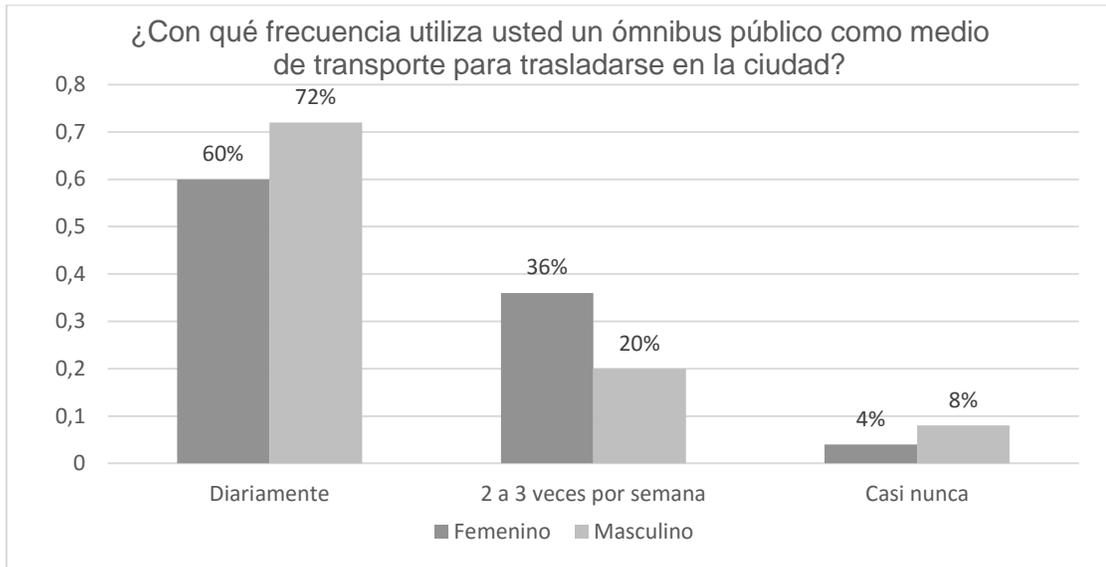


Gráfico 3: Pregunta 1 de la entrevista

### Pregunta 2:

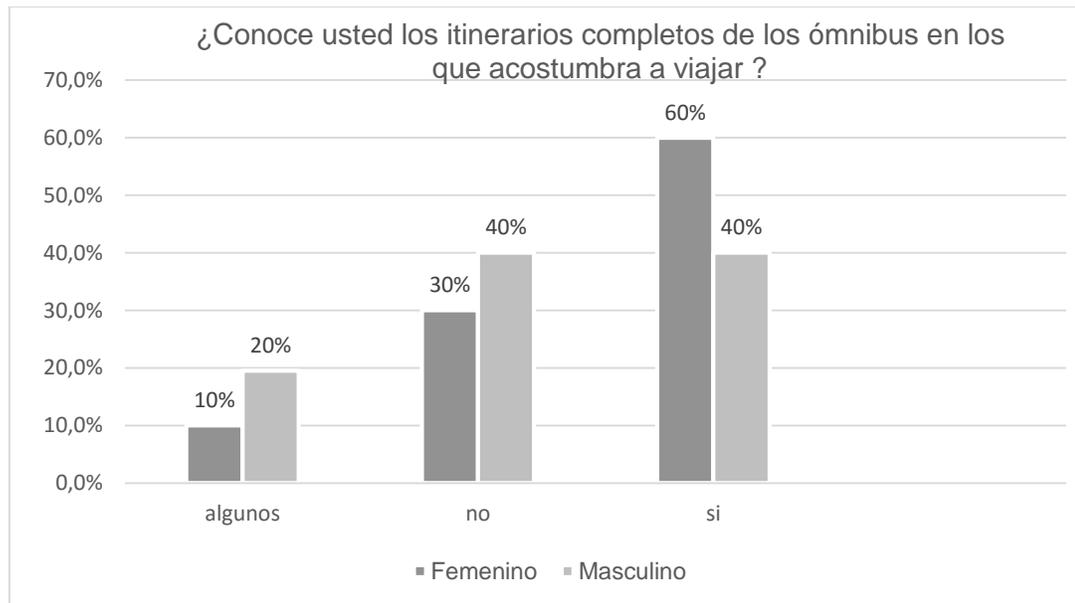


Gráfico 4: Pregunta 2 de la entrevista.

# Anexos

## Pregunta 3:

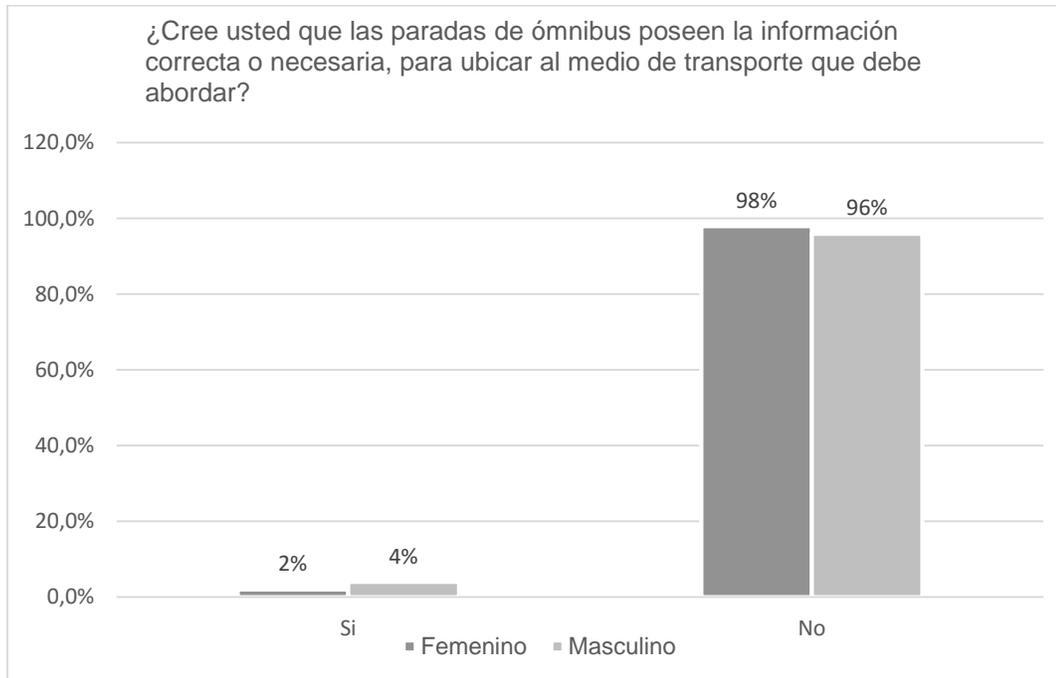


Gráfico 5: Pregunta 3 de la entrevista.

## Pregunta 4:

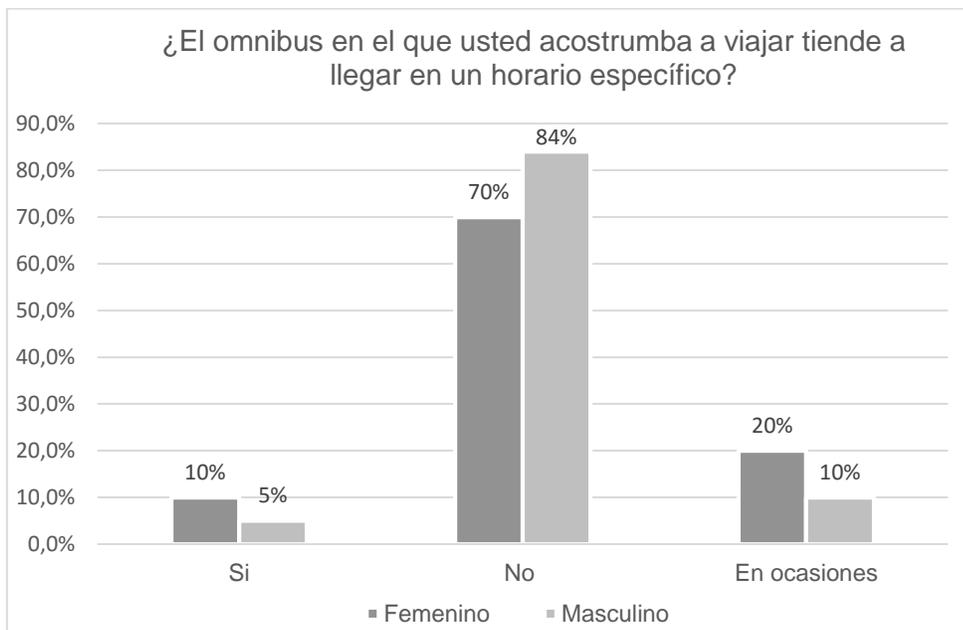


Gráfico 6: Pregunta 4 de la entrevista.

## Pregunta 5:

# Anexos

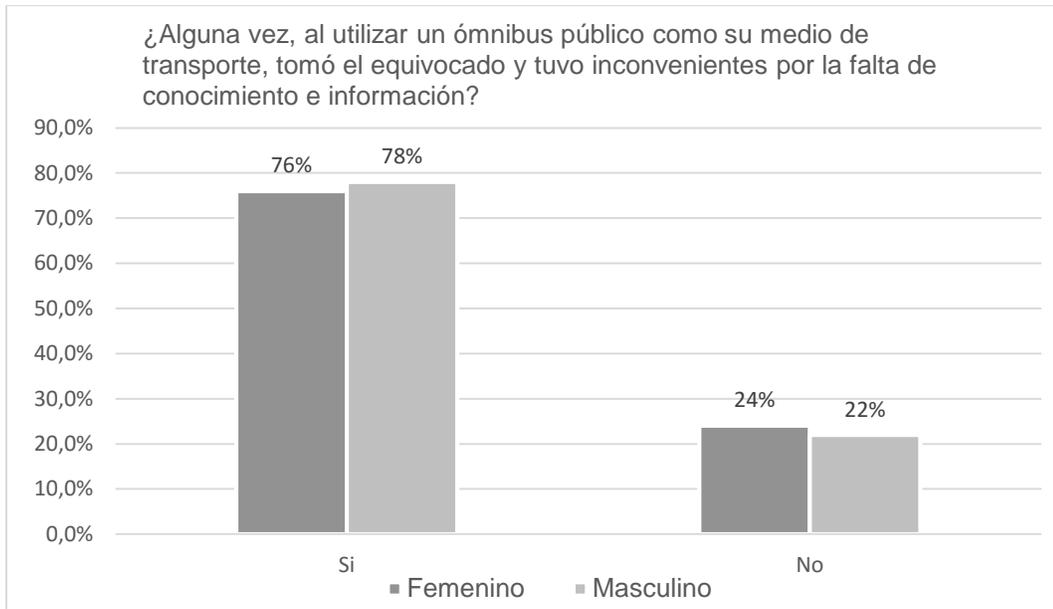


Gráfico 7: Pregunta 5 de la entrevista.

## Pregunta 6:

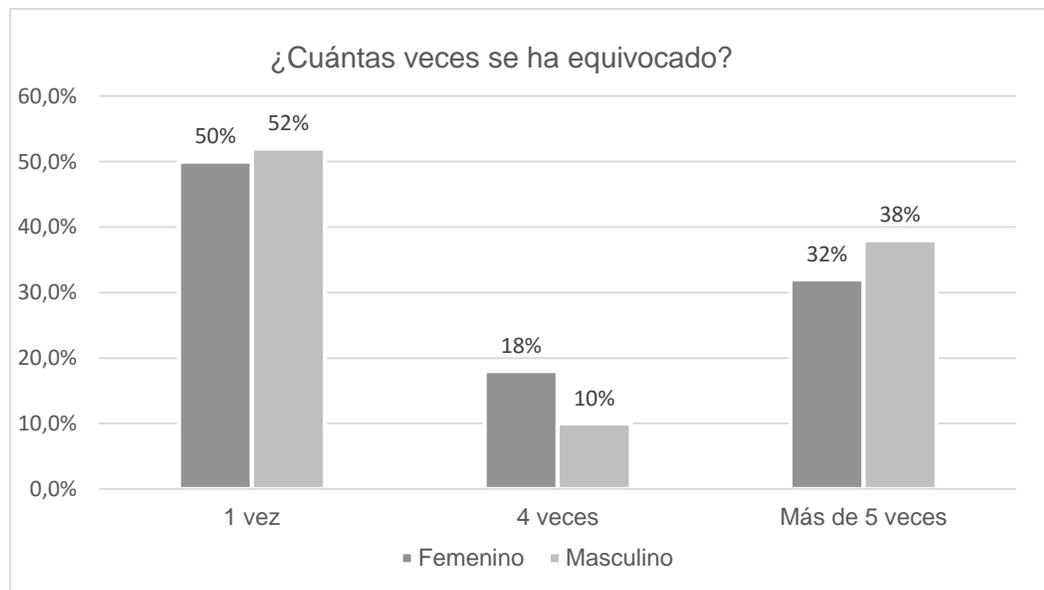


Gráfico 8: Pregunta 6 de la entrevista.

## Pregunta 7:

## Anexos

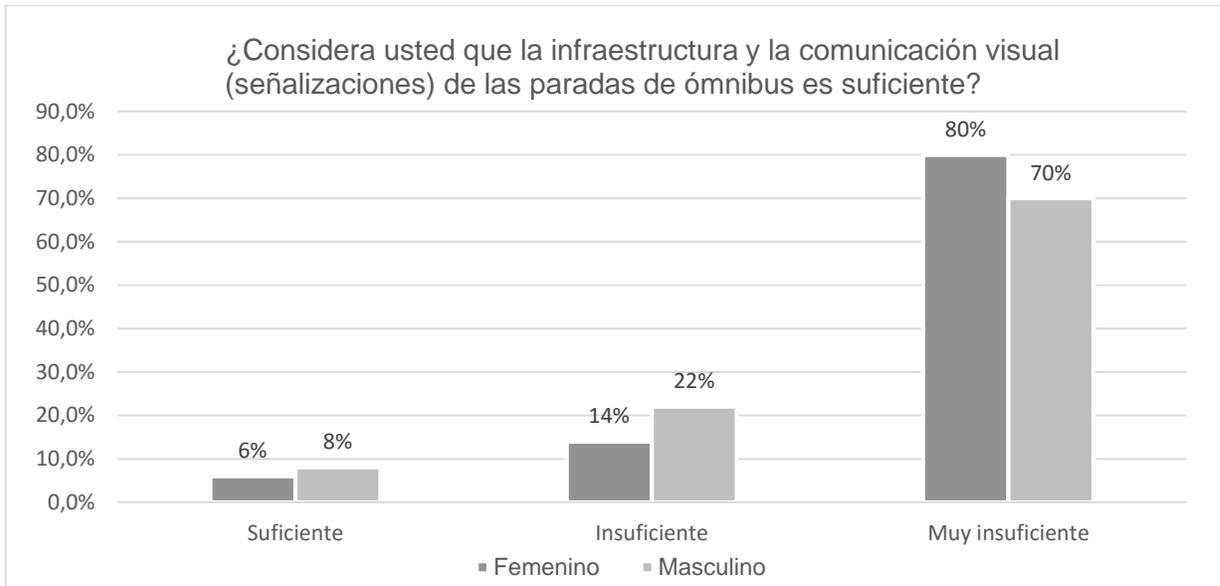


Gráfico 9: Pregunta 7 de la entrevista.

### Pregunta 8:

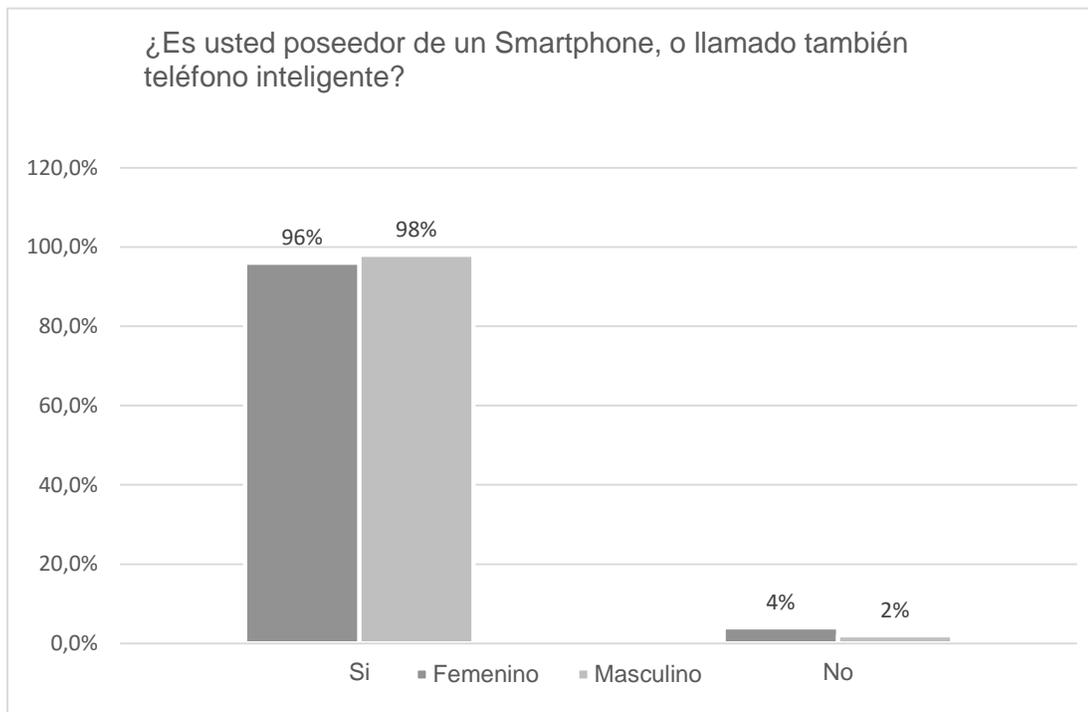


Gráfico 10: Pregunta 8 de la entrevista.

### Pregunta 9:

# Anexos

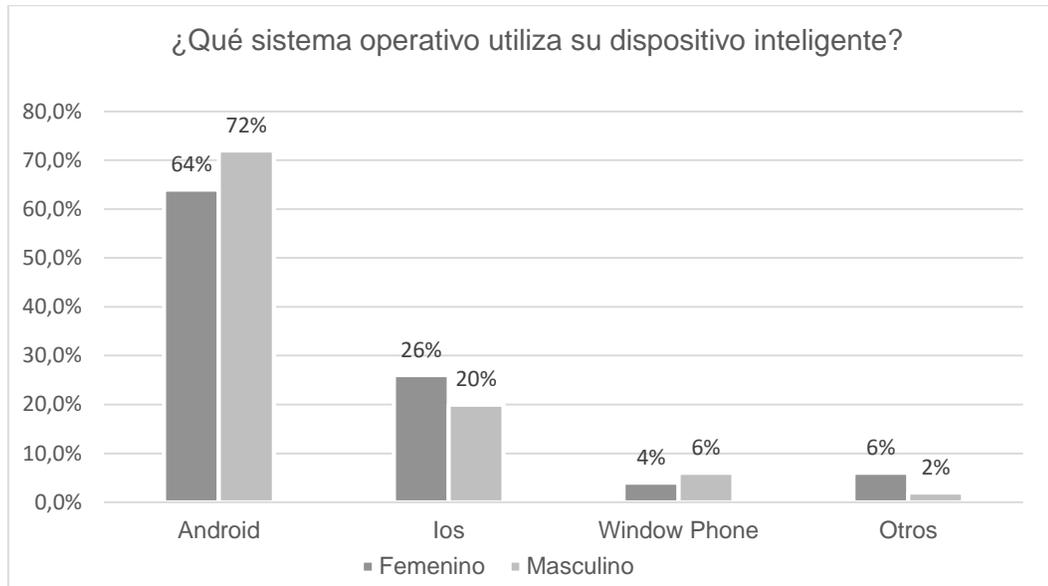


Gráfico 11: Pregunta 9 de la entrevista.

## Pregunta 10:

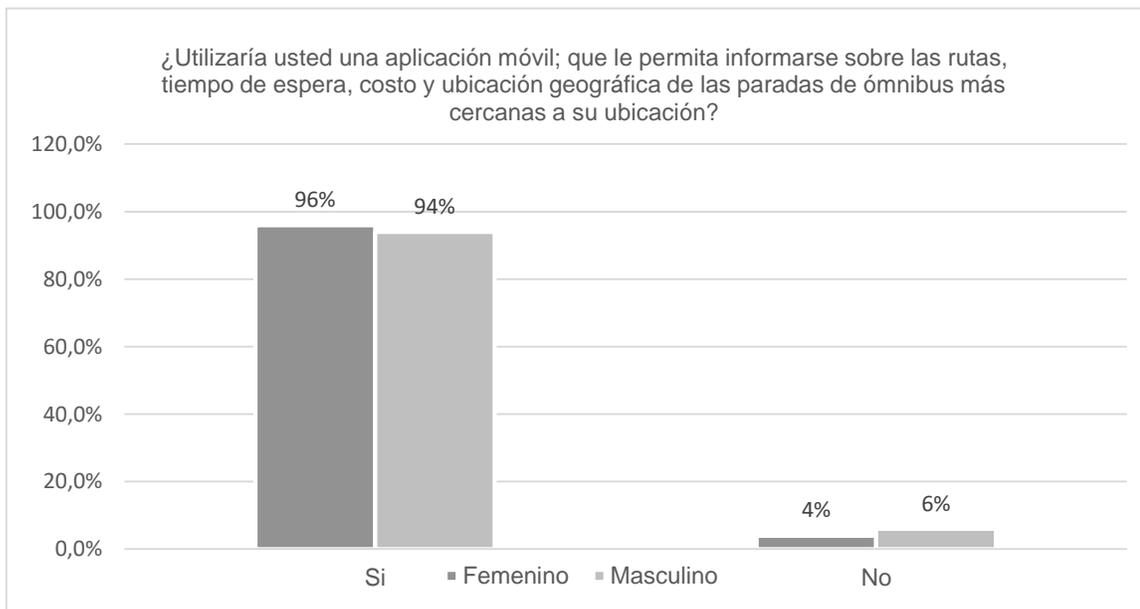


Gráfico 12: Pregunta 10 de la entrevista.

# Anexos

## Anexo VIII: Ejemplificación de los patrones de diseño utilizados

Figura 18: Utilización del patrón Experto.

```
private List<OperadorObj> ope = new ArrayList<>();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    cargarLocale();
    setContentView(R.layout.operador);

    carousel = (Carousel) findViewById(R.id.carousel);
    ope.add(new OperadorObj(R.drawable.guaguas, getResources().getString(R.string.Metrobus)));
    ope.add(new OperadorObj(R.drawable.dieciocho, getResources().getString(R.string.Omnibus)));
    ope.add(new OperadorObj(R.drawable.informacion_horarios, getResources().getString(R.string.Informacion)));
    ope.add(new OperadorObj(R.drawable.aaaa, getResources().getString(R.string.Rutero)));
    ope.add(new OperadorObj(R.drawable.tren, getResources().getString(R.string.Tren)));
}
```

Figura 19: Utilización del patrón Creador.

```
public class BusquedaAdapter extends ArrayAdapter<Recorrido> {
    private int layout;
    Context contexto;
    ArrayList<Recorrido> recorridos;

    public BusquedaAdapter(Context context, int resource, ArrayList<Recorrido> recorridos) {

    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
    }
}
```

# Anexos

Figura 20: Utilización del patrón Bajo Acoplamiento

```
public class FullScreenImage extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.full_screen);  
  
        int imagen = getIntent().getIntExtra("Image", 1);  
        TouchImageView imgDisplay;  
        imgDisplay = (TouchImageView) findViewById(R.id.imgDisplay);  
  
        if (imagen == 0){  
            imgDisplay.setImageResource(R.drawable.mapa2);  
        }  
        if (imagen == 1) {  
            imgDisplay.setImageResource(R.drawable.mapa3act);  
        }  
        if (imagen == 2){  
            imgDisplay.setImageResource(R.drawable.map4);  
        }  
    }  
}
```

Figura 21: Utilización del patrón Alta Cohesión.

```
public class Poi {  
    private int id;  
    private Bitmap foto;  
    private String nombre;  
    private String descripcion;  
  
    public Poi(int id, String nombre, String descripcion, String sobre, Bitmap foto) {  
  
    public Bitmap getFoto() {  
  
    public void setFoto(Bitmap foto) {  
  
    public int getId() {  
  
    public void setId(int id) {  
  
    public String getNombre() {  
  
    public void setNombre(String nombre) {  
  
    public String getDescripcion() {  
  
    public void setDescripcion(String descripcion) {  
  
}
```

# Anexos

Figura 22: Utilización del patrón Controlador.

```
public class Parada {
    private int id;
    private String nombre = "";
    private String direccion = "";
    private double longitud = 0;
    private double latitud = 0;
    private ArrayList<String> rutas = null;
    private String calles = "";

    public Parada(String nombre, String direccion, double latitud, double longitud) {
        this.nombre = nombre;
        this.direccion = direccion;
        this.longitud = longitud;
        this.latitud = latitud;
    }

    public int getId() {

    }

    public void setId(int id) {

    }

    public String getNombre() {

    }

    public void setNombre(String nombre) {

    }

    public String getDireccion() {

    }

    public void setDireccion(String direccion) {

    }

    public double getLongitud() {

    }

    public void setLongitud(double longitud) {

    }
}
```

Figura 23: Utilización del patrón Solitario

```
public class GpsManager extends Layer implements LocationListener, ActivityCompat.OnRequestPermissionsResultCallback {
    Activity activity;
    private boolean centerAtFirstFix;
    private final Marker marker;
    private final MapViewPosition mapViewPosition;
    private final LocationManager locationManager;
    private static final GraphicFactory GRAPHIC_FACTORY = AndroidGraphicFactory.INSTANCE;
    private final Circle circle;
    private boolean myLocationEnabled;
    private boolean snapToLocationEnabled;
    private float minDistance = 0.0f;
    private long minTime = 0;
    private final byte PERMISSIONS_REQUEST_ACCESS_FINE_LOCATION = 10;
    private Location Ulocation;

    public static LatLong locationToLatLong(Location location) {
        return new LatLong(location.getLatitude(), location.getLongitude(), true);
    }

    private static Paint getDefaultCircleFill() {
        return getPaint(GRAPHIC_FACTORY.createColor(60, 0, 0, 255), 0, Style.FILL);
    }

    private static Paint getDefaultCircleStroke() {
        return getPaint(GRAPHIC_FACTORY.createColor(160, 0, 0, 255), 2, Style.STROKE);
    }

    private static Paint getPaint(int color, int strokeWidth, Style style) {
        Paint paint = GRAPHIC_FACTORY.createPaint();
        paint.setColor(color);
        paint.setStrokeWidth(strokeWidth);
        paint.setStyle(style);
        return paint;
    }
}
```

# Anexos

Figura 24: Utilización del patrón Decorador.

```
public class OperadorObj {  
    private int id;  
    private Drawable img;  
    private String nombre;  
    private String name;  
    private Double precio;  
  
    public OperadorObj(int id, Drawable img, String nombre, String name, Double precio) {  
        this.id = id;  
        this.img = img;  
        this.nombre = nombre;  
        this.name = name;  
        this.precio = precio;  
    }  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public Drawable getImg() {  
        return img;  
    }  
    public void setImg(Drawable img) {  
  
    }  
    public String getNombre() {  
  
    }  
    public void setNombre(String nombre) {  
  
    }  
}
```