



Universidad de las Ciencias  
Informáticas

Centro de Identificación y Seguridad Digital

---

# **APLICACIÓN MÓVIL PARA LA AGENCIA ALQUILER DE VEHÍCULOS REX**

---

*Trabajo de diploma en opción al título  
de Ingeniero en Ciencias Informáticas*

Autor: Yobal Concepción Cabrera

Tutores: Ing. Yaima Oval Riverón

MsC. Hubert Viltres Sala

*La Habana, ... de junio del 2017*

*“Año 59 de la Revolución”*

## *Pensamiento*



*“Una persona predestinada es la que construye un sólido edificio, con los ladrillos que le van tirando los que quieren voltearlo.”*

*David Brinkley*

## *Declaración de autoría*

Yo Yobal Concepción Cabrera con CI: 92061924021 declaro ser el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de junio del 2017.

---

Firma del Autor  
Yobal Concepción Cabrera

---

Firma del Tutor  
Ing. Yaima Oval Riverón

---

Firma del Tutor  
MSc. Hubert Viltres Sala

## *Datos del Contacto*

**Tutor:**

**MsC.** Hubert Viltres Sala

**Correo:**

hviltres@uci.cu

**Tutora:**

**Ing.** Yaima Oval Riverón

**Correo:**

yoval@uci.cu

# *Agradecimientos*

*Quiero agradecerles:*

*Primeramente, a mi familia por todo el apoyo y el aliento recibido en el trayecto de toda mi carrera y aún mayor en este tramo final de la tesis.*

*A mis tutores, por haber dado el paso al frente pese a su carga de trabajo, en el momento que no pocos lo hubieran dado hacia atrás y por el apoyo, orientación, confianza y paciencia que me han dedicado en el trayecto de la tesis.*

*A mis amigos y compañeros (Dani, Friki, Jose, Yaidel, Yoel, Lolo, Osmel, Ortelio), por saberme soportar o por lo menos disimularlo pese a que puedo ser bastante insoportable en ocasiones.*

*A David por todos los consejos y ayuda ofrecida en cada uno de los momentos que lo he necesitado.*

*A los profesores del centro y de la facultad, por ayudarme en la aclaración de dudas y poderme atender en el tan poco tiempo libre del que disponen.*

*A todos, muchísimas gracias, estoy en deuda con ustedes.*

## *Dedicatoria*

*Al llegar a este punto de mi vida en el que me encuentro hoy, un punto de triunfo y metas cumplidas no puedo pasar por alto a las personas que más quiero en mi vida y que se merecen todo lo que está aconteciendo más que yo:*

*A mis dos madres, ustedes saben que no puedo diferenciar entre ustedes dos, son lo más grande que tengo en la vida y es por eso que les dedico este segundo gran triunfo de mi vida ya que el primero es tenerlas a mi lado.*

*A mi padre quien siempre ha estado a mi lado cuando lo necesito, ofreciendo su apoyo, luchando por que su hijo se gradué y quien ha sufrido más mi carrera que yo.*

*A mis abuelos Raimundo y Pastor, por todos los consejos y la gran preocupación que han tenido durante toda mi carrera para lograr que su nieto se haga ingeniero.*

*A mis tíos Enid y Ernes Camilo, por preocuparse por todo lo que iba aconteciendo, por el aliento ofrecido y por mostrar su apoyo incondicional siempre.*

*A Yosbelito (Tuti) a ti que no tienes conocimiento de que eres ni que vas a hacer con tu vida. Aunque tu primo siempre está distante siempre se acuerda de ti. Espero que seas una gran persona con un gran futuro.*

*A todos los presentes aquí en este local hoy por ayudarme a compartir que este mi sueño se haga realidad.*

*A toda mi familia en general, por siempre preocuparse por cómo iba marchando mi carrera.*

# Resumen

Una agencia de alquiler de vehículos es una compañía que ofrece autos de alquiler ya sea para cortos o largos períodos de tiempo. Cubre las necesidades primordialmente del turismo y el sector empresarial. Sus establecimientos están ubicados principalmente, en aeropuertos, estaciones de trenes y autobuses. Se complementan a menudo con una aplicación móvil permitiendo hacer reservas a través de *Internet*. Para la solución se tuvo en cuenta el estudio de las aplicaciones móviles utilizadas para la renta de autos, que permitió identificar los elementos y características distintivas de este tipo de negocio. La aplicación fue implementada para el sistema operativo móvil *Android*. El proceso de desarrollo fue guiado por la metodología de desarrollo *XP* y como *IDE* de programación *Android Studio*. Al concluir el desarrollo se sometió a pruebas para garantizar que la aplicación ofreciera la solución idónea, segura, mantuviera un funcionamiento estable y se integrara a sistemas desarrollados en *Phyton*. Se obtuvo como resultado una aplicación capaz de ofrecer los principales servicios que ofrece *REX* a sus clientes, así como las principales informaciones referentes a la agencia.

**Palabras Claves:** agencia de alquiler de vehículos, dispositivos móviles, aplicaciones móviles, sistema operativo *Android*.

# Índice de Contenido

Introducción .....	1
Capítulo 1: Las aplicaciones móviles para el alquiler de vehículos.....	6
1.1 Acceso a la información .....	6
1.2 Alquiler de autos.....	7
1.3 Aplicación móvil.....	7
1.4 Sistemas para la reservación de autos.....	9
1.5 Proceso de desarrollo de aplicaciones móviles .....	15
1.6 Metodología de desarrollo de software.....	16
1.7 Herramientas Utilizadas .....	19
1.8 Lenguajes Utilizados.....	21
1.9 Conclusiones parciales .....	22
Capítulo 2. Análisis y diseño de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.....	23
2.1 Propuesta de Solución .....	23
2.2 Modelo de dominio.....	23
2.3 Levantamiento de requisitos .....	25
2.4 Historias de usuario .....	27
2.5 Plan de entregas .....	33
2.6 Plan de iteraciones .....	34
2.7 Tarjetas CRC.....	34
2.8 Arquitectura del <i>software</i> .....	35
2.9 Modelo de datos.....	41
2.10 Diagrama de despliegue.....	42
2.11 Conclusiones parciales del capítulo.....	42
Capítulo 3: Implementación y pruebas a la aplicación para la Agencia de Alquiler de Vehículos REX .....	43
3.1 Estándares de codificación.....	43
3.2 Validación de la propuesta de solución .....	45

3.3 Pruebas Funcionales.....	45
3.4 Pruebas Unitarias.....	52
3.5 Pruebas de Integración .....	53
3.6 Pruebas de Rendimiento .....	54
3.7 Pruebas de Compatibilidad .....	55
3.7 Conclusiones del Capítulo .....	57
Conclusiones .....	58
Recomendaciones .....	59
Referencias Bibliográficas .....	60

# *Índice de tablas*

Tabla 1: Funcionalidades del Sistema.....	26
Tabla 2: Requisitos No Funcionales del sistema. ....	27
Tabla 3: HU_1.....	28
Tabla 4: HU_2.....	29
Tabla 5: HU_3.....	31
Tabla 6: Plan de entregas.....	33
Tabla 7: Plan de Iteraciones .....	34
Tabla 8: Tarjeta CRC "Reservación" .....	34
Tabla 9: Tarjeta CRC "Persona" .....	35
Tabla 10: Tarjeta CRC "Traslado" .....	35
Tabla 11: Estándares de Codificación de Java (Elaboración Propia). ....	44
Tabla 12: Caso de Prueba Realizar Reservación de Traslado (Elaboración Propia). ....	46
Tabla 13: Descripción de las variables del Caso de Prueba Realizar Reservación para traslado.....	46
Tabla 14: Caso de Prueba Realizar Reservación por Horas (Elaboración Propia). ....	47
Tabla 15: Descripción de las variables del Caso de Prueba Realizar Reservación por horas. ....	48
Tabla 16: Caso de Prueba Realizar Reservación con o sin Chofer (Elaboración Propia). ....	49
Tabla 17: Descripción de las variables del Caso de Prueba Realizar Reservación con o sin chofer. ....	50
Tabla 18: Prueba de Integración para la funcionalidad Realizar Reserva (Elaboración Propia). ....	54
Tabla 19: Resultados de las Pruebas de Rendimiento (Elaboración Propia).....	55
Tabla 20: Resultados de las Pruebas de Compatibilidad. ....	56

# *Índice de figuras*

Ilustración 1: Aplicación móvil <i>Sixt</i> ( <i>SIXT</i> , 2016). .....	11
Ilustración 2: Aplicación móvil <i>SocialCar</i> ( <i>SocialCar</i> , 2017). .....	12
Ilustración 3: Aplicación móvil <i>Car2go</i> ( <i>Car2go</i> , 2017) .....	13
Ilustración 4: Aplicación móvil <i>Drivy</i> ( <i>Drivy</i> , 2016). .....	14
Ilustración 5: Modelo de Dominio (Elaboración propia). .....	24
Ilustración 6: Modelo-Vista-Controlador en Android. ....	37
Ilustración 7: Modelo de datos (Elaboración propia).....	41
Ilustración 8: Diagrama de despliegue del sistema (Elaboración propia).....	42

## Introducción

Las nuevas tecnologías han avanzado a un ritmo acelerado convirtiendo varios procesos, que hasta el momento habían sido muy utilizados, en obsoletos. *Internet* ha revolucionado en poco tiempo el modo de comunicarse entre las personas y de acceder a la información, permitiendo enviar y recibir grandes cantidades de datos mediante la utilización de sitios *web* o el correo electrónico. Diariamente se genera una gran cantidad de información digital: existen más de 3,5 billones de usuarios a nivel mundial que acceden a la información disponible en los más de 1 billón de sitios *web* de *internet* (Stats, 2017).

El modo en que se produce la conexión a *internet* ha evolucionado en los últimos años gracias a la cuota de mercado alcanzada por los teléfonos inteligentes. De esta forma la conexión se realiza de manera más fácil, sencilla y los móviles se han convertido en uno de los principales dispositivos de conexión a la red. Por tanto, se hace imprescindible que los contenidos de la *web* sean visibles desde cualquier dispositivo móvil. Además, es necesario para la sociedad explotar de manera adecuada el acceso hacia los beneficios del *internet* móvil que surgen a partir del uso de todas las funciones integradas en los dispositivos inteligentes, accesibles a través de aplicaciones móviles. Es por esta causa que se ha comenzado a utilizar por las distintas entidades y empresas, este tipo de aplicaciones para establecerlas como herramientas publicitarias (Guiu, 2012).

Para los países que se encuentran en desarrollo, las nuevas tecnologías móviles tales como los servicios de datos, la asesoría agrícola y empresarial, la atención sanitaria y la transferencia de dinero basados en el teléfono móvil ofrecen beneficios para la economía y el desarrollo. La telefonía móvil cumple una función clave en la obtención de contratos de negocios y da accesibilidad a los usuarios. Contribuye a mejorar la productividad o suministra información importante a los empresarios por el simple hecho de usar el teléfono para comunicarse con clientes y proveedores. En el mundo hay un creciente interés por los servicios que van más allá de la comunicación vocal y los mensajes de texto. La oportunidad que ofrecen los servicios financieros móviles es enorme en países donde los usuarios no tienen cuenta bancaria y la propiedad de teléfonos móviles supera el uso de esas cuentas.

Actualmente en Cuba tienen la posibilidad de acceder a la información disponible en la *web* un número superior a los 4 millones de usuarios. De acuerdo a datos revelados por ETECSA (Empresa de Telecomunicaciones de Cuba), diariamente acceden a *internet* un aproximado de 250 mil usuarios desde los puntos de acceso inalámbricos. La adquisición de los teléfonos inteligentes y el acceso a *internet* a través

de redes *Wifi*<sup>1</sup>, ha ocasionado la aparición de pequeños y medianos negocios relacionados con la telefonía móvil, capaces de generar ingresos y servicios que incorporaran nuevas tecnologías y una fuente de empleo a un buen número de cubanos (Etecsa, 2016).

Para potenciar el desarrollo en cuanto a lo referido a la telefonía móvil, se han trazado políticas encaminadas a lograr resultados que impacten de forma decisiva en el desarrollo de la nación y contribuyan a la informatización de la sociedad cubana. Se trabaja en muchas direcciones para que las Tecnologías de la Información y la Comunicación (TIC) se conviertan en un sector estratégico para el país y potenciar una mejor economía del conocimiento (Juventud Rebelde, 2016). Además, en el séptimo congreso del Partido Comunista de Cuba se trazaron lineamientos para guiar la política económica del país en los próximos cinco años, vinculados a este importante tema, en los que se exhorta a las empresas cubanas a explotar esta tecnología en aras de lograr situarlas en lo más alto de la competencia a nivel mundial. (PCC, 2016).

Dentro del proceso de desarrollo tecnológico de Cuba, se encuentra inmersa la Universidad de las Ciencias Informáticas (UCI), que tiene entre sus principales objetivos producir *software* y servicios informáticos. Posee varios centros productivos que se encargan del desarrollo de soluciones informáticas dentro de los que se encuentra el Centro de Identificación y Seguridad Digital (CISED). Este centro tiene entre sus proyectos el Sistema de Gestión para la Agencia de Alquiler de Vehículos REX, el cual tiene como principal objetivo desarrollar un sistema informático que permita realizar los procesos que se desarrollan en esa entidad, haciendo uso de nuevas tecnologías.

REX es una agencia dedicada a la renta de autos perteneciente a la empresa de transporte para el turismo en Cuba Transtur. Posee una flota única que se encuentra distribuida por todo el país a través de sucursales de renta. Actualmente el sitio *web* que representa a la empresa presenta dificultades que afectan el acceso a la información, limitando que más personas contraten los servicios que se ofertan debido a que la misma se encuentra desorganizada y no permite ser visualizada de forma correcta. Los componentes del sitio no se ajustan cuando se accede desde dispositivos móviles pues no permite que su información sea visualizada desde estos (*responsive*<sup>2</sup>). El tiempo de respuesta a las peticiones de los usuarios es largo en comparación

---

<sup>1</sup> *Wifi*: es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con *wifi* pueden conectarse a *internet* a través de un punto de acceso de red inalámbrica.

<sup>2</sup> *Responsive*: es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas *web* al dispositivo que se esté utilizando para visitarlas.

con la media establecida (3 segundos) para este tipo de sistemas. Tiene un nivel de usabilidad deficiente, dado que se presta mayor atención a elementos relacionados con el interior del sistema como su rendimiento o confiabilidad, por lo que aspectos tan relevantes como realizar un diseño interactivo centrado en mantener la atención del receptor, que se adapte a las características específicas de cada usuario, han formado parte de un segundo plano. Además, fue desarrollado con tecnologías que surgieron hace más de 10 años como el *framework .NET* en su versión 1.0, lo cual de ser utilizadas nuevamente dificultarían seriamente el proceso de renovación del sitio web.

Una vez que se termine la implementación y puesta en línea del sistema de gestión, para poder acceder a este los usuarios tendrán que hacerlo a través de computadoras o dispositivos móviles. Son más las personas que tienen la posibilidad económica de poseer un teléfono inteligente u otro tipo de dispositivo móvil debido al alto costo de las computadoras. El acceso a la información desde estos se produce de manera más sencilla y directa que desde las computadoras. Además, es una tendencia actual que se desarrollen aplicaciones móviles relacionadas a los sitios *web* de las agencias para que los usuarios puedan acceder a sus servicios desde sus dispositivos móviles. Mediante ellas los clientes se sentirán familiarizados con los servicios y características de las empresas pues para acceder a los mismos solo será necesario conexión a *internet*.

A partir de la problemática antes mencionada se obtiene como **problema de investigación**: ¿Cómo mejorar el acceso a la información brindada por la Agencia de Alquiler de Vehículos REX para potenciar la gestión de los servicios?

Derivado del problema de investigación se llega al **objeto de estudio**: Proceso de creación de aplicaciones móviles; enmarcado en el **campo de acción**: Proceso de creación de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.

Según lo planteado anteriormente se busca dar cumplimiento al siguiente **objetivo general**: Desarrollar una aplicación móvil para la gestión de los servicios de la Agencia de Alquiler de Vehículos que contribuya a mejorar el acceso a la información de la empresa.

Para darle cumplimiento al objetivo general planteado se responderán las siguientes **preguntas científicas**:

- ¿Cuáles son los presupuestos teóricos que sustentan la implementación de aplicaciones móviles?
- ¿Qué fundamentos presenta el análisis y diseño de aplicaciones móviles para la renta de vehículos?

- ¿Qué concepción debe tener la aplicación móvil para la Agencia de Alquiler de Vehículos REX para sistema operativo *Android*?
- ¿Cómo validar la contribución de la aplicación móvil para la Agencia de Alquiler de Vehículos REX?

Para darle cumplimiento a las preguntas científicas planteadas se definen las siguientes **tareas de la investigación**:

- Sistematización de los referentes teórico-metodológicos relacionados con la implementación de aplicaciones móviles.
- Definición de las herramientas a utilizar para el desarrollo de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.
- Realización del análisis y diseño de la solución para guiar el proceso de implementación de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.
- Implementación de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.
- Realización de pruebas de *software* a la aplicación móvil para la Agencia de Alquiler de Vehículos REX.

Durante la realización de la investigación se han empleado los métodos de investigación científica teóricos que a continuación se plantean:

- **Analítico-Sintético:** se ha utilizado para realizar la síntesis de los elementos más importantes de las fuentes bibliográficas analizadas sobre desarrollo de aplicaciones móviles, para que de esta manera puedan ser identificadas las tecnologías y herramientas que se utilizarán en el desarrollo de la aplicación.
- **Inductivo-Deductivo:** se ha empleado para generalizar el conocimiento adquirido para utilizarlo en el desarrollo de la aplicación, a través de conceptos básicos e independientes.
- **Modelación:** se ha utilizado para crear el proceso donde se elaboran los modelos referentes al ciclo de vida del *software* que ayudan a dar cumplimiento a las tareas de diseño de los procesos involucrados en la implementación de la aplicación.

Además, el siguiente método empírico:

- **Análisis documental:** se ha empleado para la revisión bibliográfica, la revisión de las fuentes básicas de información, el estudio de documentos y demás bibliografías que se han consultado para adquirir el conocimiento para una posterior implementación de la aplicación.

La investigación está estructurada de la siguiente manera:

**Capítulo 1: Las aplicaciones móviles para el alquiler de vehículos.** Se describe todo lo referente a la fundamentación teórica de la investigación, con lo que se busca justificar la necesidad de la realización de este trabajo. Se plantean los beneficios que serán obtenidos con su realización.

**Capítulo 2: Aplicación móvil para la Agencia de Alquiler de Vehículos REX.** Es efectuado el proceso de recopilación de todos los requisitos funcionales y no funcionales necesarios para un correcto funcionamiento de la aplicación móvil. Se realiza el análisis y diseño de la solución.

**Capítulo 3: Implementación y pruebas a la aplicación móvil para la Agencia de Alquiler de Vehículos REX.** Se realiza la implementación de la propuesta de solución y una vez terminada se le aplican todas las pruebas necesarias para verificar el correcto funcionamiento.

Finalmente se presentan las Conclusiones y Recomendaciones derivadas de la investigación, las Referencias Bibliográficas, así como los Anexos que apoyan la comprensión y dan información adicional sobre el trabajo realizado.

## Capítulo 1: Las aplicaciones móviles para el alquiler de vehículos

En este capítulo es formalizado todo lo referente a la etapa de conceptualización de la aplicación móvil para la Agencia de Alquiler de Vehículos REX. Es ejecutada la fundamentación teórica de la investigación, que trae consigo el conocimiento teórico de los procedimientos y tecnologías que se asocian a lograr el cumplimiento de la solución propuesta. Se intenta alcanzar la justificación de la necesidad de realizar el trabajo y los beneficios que traerá consigo. Es efectuada una investigación sobre la existencia de otras soluciones similares a la que se desea llegar con el objetivo de establecer una comparación para determinar la mejor propuesta de solución. Además, son mencionadas las herramientas que fueron utilizadas durante el proceso de realización de la solución.

Para la ejecución del proceso de desarrollo de una aplicación móvil para la renta de vehículos es necesario comprender una serie de conceptos que ayuden a garantizar un mayor entendimiento del mismo. Entre los cuales podemos encontrar:

### 1.1 Acceso a la información

Comprende diferentes temáticas que se refieren a la disponibilidad, la accesibilidad y la asequibilidad de la información. Estas están representadas por el multilingüismo, los metadatos, la interoperabilidad, los programas informáticos de fuente abierta, el contenido libre, las licencias *Creative Commons*<sup>3</sup>, así como las necesidades especiales de los discapacitados. El novedoso ambiente tecnológico y económico origina preocupaciones sobre la degradación del acceso a determinadas informaciones y conocimientos que antes se encontraban libremente. Al mismo tiempo existe *Internet*, que ofrece la oportunidad de compartir información, promover la diversidad de idiomas y rescatar idiomas que se estaban perdiendo. A partir del año 2003 la UNESCO<sup>4</sup> ha trazado una serie de políticas para respaldar los esfuerzos mundiales a favor de los programas gratuitos y de código fuente abierto. También buscando la utilización de las TIC para lograr un acceso equitativo a la información sobre todo para las personas con discapacidad (UNESCO, 2017).

---

<sup>3</sup> *Creative Commons* : organización cuya oficina central está ubicada, en el estado de California (Estados Unidos). Permite usar y compartir tanto la creatividad como el conocimiento a través de una serie de instrumentos jurídicos de carácter gratuito.

UNESCO: Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.

## 1.2 Alquiler de autos

Antes de alquilar un auto es ventajoso comparar los precios que ofrecen varios establecimientos, pues es un servicio en el que los precios pueden variar mucho de una empresa a otra. Los precios deben estar expuestos en una lista desglosada por vehículos y por la modalidad del alquiler ya sea por día, por semana, con kilometraje limitado o ilimitado. Este tipo de empresas exigen a sus clientes una edad mínima comprendida entre los 21 y 23 años y cierta antigüedad del permiso de conducción. Cuando se realiza el alquiler este debe ser validado por un contrato en el cual constarán los datos del cliente que lo efectúa, las del establecimiento, las del auto y las normas y condiciones del alquiler. Todas las infracciones, accidentes por negligencia y costos por combustible deberán ser costeados por los clientes. En caso de que ocurra algún tipo de conflicto este puede ser plasmado en las hojas oficiales de queja, reclamación, o denuncia de cada establecimiento (Consum, 2017).

Es una alternativa para cuando se quiere realizar un viaje o si necesita un automóvil para un tiempo determinado. Una empresa que se dedique a este tipo de negocios debe ofrecer a sus clientes la descripción exacta de las características del auto a alquilar, el precio total de la renta, el plazo de alquiler, posibles cambios de precio y gastos adicionales que se puedan producir. Debe cumplir con lo establecido en la Orden del 20 de julio de 1995 en la cual se formula el Reglamento de la Ley de Ordenación de los Transportes Terrestres en materia de arrendamientos de vehículos sin conductor.

## 1.3 Aplicación móvil

Una aplicación móvil es un programa diseñado para funcionar solamente en dispositivos móviles, tabletas o cualquier dispositivo inteligente. Pueden ser encontradas en varias plataformas que dependen del fabricante y desarrollador en cuestión (*Android, IOS*, entre otras plataformas). Su distribución se realiza a través de tiendas virtuales o por otras vías, donde pueden ser adquiridas de forma gratuita o a través del pago (Concepto, 2015).

La utilización de aplicaciones móviles brinda una serie de ventajas a sus usuarios entre las que se encuentran:

- Acceso rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso.
- Almacenamiento de datos personales de manera segura.
- Gran versatilidad respecto a su utilización o aplicación práctica.

- Posee funcionalidades específicas.
- Mejora la capacidad de conectividad y disponibilidad de servicios y productos.

El mercado de las aplicaciones móviles se encuentra en un continuo crecimiento. Son muchas las empresas y compañías que se dedican a crear aplicaciones que dan respuesta a distintas necesidades que puedan o no presentar determinada persona.

Existen tres tipos de aplicaciones dentro de los que se puede encontrar las aplicaciones nativas, las aplicaciones *web* y las aplicaciones híbridas. A continuación, se hará una descripción de cada uno de estos tipos.

### **Aplicación nativa**

Una **aplicación nativa** es la que se desarrolla de forma específica para un determinado sistema operativo móvil, llamado *Software Development Kit*<sup>5</sup> o *SDK*. Cada una de las plataformas, *Android*, *iOS* o *Windows Phone*, tienen un sistema diferente. Este tipo de aplicaciones corren de forma más eficiente sobre estos dispositivos al estar sus componentes diseñados de manera específica para este sistema operativo. Cuando se habla de desarrollo móvil casi siempre se refiere a aplicaciones nativas. La principal ventaja con respecto a los otros dos tipos, es la posibilidad de acceder a todas las características del *hardware* del móvil: cámara, *GPS (Global Positioning System)*, agenda, dispositivos de almacenamiento y otras muchas. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de aplicaciones (Geospatial, 2016) .

### **Web App**

Una **aplicación web o web app** es la desarrollada con lenguajes como *HTML*, *JavaScript* y *CSS*. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones. Las aplicaciones *web* se ejecutan dentro del propio navegador *web* del dispositivo a través de una *URL*. Por ejemplo, en *Safari*, si se trata de la plataforma *iOS*, el contenido se adapta a la pantalla y adquiere un aspecto de navegación *APP*. En realidad, la gran diferencia con una aplicación nativa es que no necesita instalación por lo que no pueden estar visibles en tienda de aplicaciones y la promoción y comercialización debe realizarse de forma independiente. De todas formas, se puede crear un acceso

---

<sup>5</sup> *Software Development Kit*: conjunto de herramientas de desarrollo de *software* que le permite al programador o desarrollador de *software* crear aplicaciones para un sistema concreto.

directo que sería muy parecido a instalar la aplicación en el dispositivo. Las *apps web* móviles son siempre una buena opción si el objetivo es adaptar la *web* a formato móvil (Lancetalent, 2014).

### **Web App nativa o Híbrida**

Una **aplicación híbrida** es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las *apps* híbridas se desarrollan con lenguajes propios de la *web app*, es decir, *HTML*, *JavaScript* y *CSS* por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del *hardware* del dispositivo. La principal ventaja es que, a pesar de estar desarrollada con *HTML*, *Java* o *CSS*, es posible agrupar los códigos y distribuirla en las tiendas de aplicaciones. Los otros dos grupos son más utilizados en cuanto al desarrollo de aplicaciones móviles se refiere. Un buen ejemplo de aplicaciones híbridas es *Facebook* (Geospatial, 2016). A simple vista podría parecer que las aplicaciones nativas tienen más ventajas. Pero lo cierto es que en dependencia del tipo de aplicación que se vaya a desarrollar, puede convenir utilizar uno u otro método.

Después de analizar las diferentes características que se vieron anteriormente, la propuesta de solución debe ser una aplicación móvil nativa. Pues es la opción más viable para lograr resolver la problemática existente en la Agencia de Alquiler de Vehículos REX porque posibilita la utilización de todos los elementos y recursos del *hardware* del teléfono. Ofrece una experiencia de usuario más completa al ser el tipo de aplicaciones que más se desarrolla a nivel mundial. Por lo tanto, posee mejor estética, diseño, utilidad, accesibilidad, rendimiento y usabilidad. No es necesario para su funcionamiento la conexión a *internet* y se puede actualizar de forma constante mediante las distintas tiendas de aplicaciones disponibles para la venta de aplicaciones móviles o a través de otras vías menos convencionales.

## **1.4 Sistemas para la reservación de autos**

Para lograr que las aplicaciones móviles cumplan con las características necesarias para su correcto funcionamiento en los diferentes dispositivos móviles es fundamental que se realice un estudio relacionado con las diferentes aplicaciones que sean utilizadas por agencias de renta de vehículos. El análisis se va a centrar en el objeto de estudio definido en la investigación pues las aplicaciones móviles de renta de autos encontradas necesitan de pago para poder ser utilizadas. Se tienen en consideración solo las características que presentan las aplicaciones móviles para renta de autos encontradas, pues en Cuba aún no se utiliza este tipo de aplicaciones para realizar el alquiler de vehículos.

## **Sixt**

Es la aplicación móvil nativa desarrollada para varias plataformas móviles que ofrece la empresa de alquiler de coches alemana con sede en *Pullach im Isartal*, un municipio de las afueras de *Múnich*, Alemania. La misma brinda acceso a todos los grupos de vehículos, incluidos camiones (*SIXT*, 2016).

Estas son algunas de las características que presenta en dependencia del sistema operativo en que se use:

### Android (*SIXT*, 2016):

- Acceso a todos los grupos de vehículos, incluidos camiones.
- Búsqueda sencilla de vehículo gracias a la función de filtro.
- Extras opcionales como seguros o un navegador.
- Reserva con tarifas estándar y corporativa.
- Acceso al servicio *Sixt Unlimited*.
- Búsqueda de estaciones de forma manual y a través de *GPS*.
- Vista y cancelación de reservas anteriores mediante el centro de clientes o consultas individuales.
- Permite escoger entre tarifas flexible y de prepago.
- Una funcionalidad exclusiva de la aplicación para *Sixt* es que posibilita seleccionar el coche que desea de la clase que ha reservado 30 minutos antes de recogerlo.

Es compatible para los sistemas operativos *Android* 4 o superior y se encuentra disponible en idioma alemán, árabe, checo, coreano, español, finlandés, francés y griego.

### BlackBerry OS (*SIXT*, 2016):

- Acceso a todos los grupos de vehículos, incluidos camiones.
- Extras opcionales como seguros o un navegador.
- Reserva con tarifas estándar y corporativa.
- Acceso al servicio *Sixt Unlimited*.
- Búsqueda de estaciones a través de *GPS*.
- Vista y cancelación de reservas anteriores mediante el centro de clientes.

Es compatible para los sistemas operativos *BB OS* 4.5.0 o superior, *BB OS* 10 o superior y se encuentra disponible en idioma alemán, checo, coreano, croata, danés, español, francés, griego, hebreo, holandés, húngaro, inglés, italiano, polaco, portugués, ruso, sueco y turco:

iOS (SIXT, 2016):

- Acceso a todos los grupos de vehículos, incluidos camiones.
- Búsqueda sencilla de vehículo gracias a la función de filtro.
- Extras opcionales como seguros o un navegador.
- Reserva con tarifas estándar y corporativa.
- Acceso al servicio *Sixt Unlimited*.
- Vista y cancelación de reservas anteriores mediante el centro de clientes o consultas individuales.
- Integración de *Passbook* (a partir de *iOS 7*).
- Una funcionalidad exclusiva de la aplicación para *Sixt* es que posibilita seleccionar el coche que desea de la clase que ha reservado 30 minutos antes de recogerlo.

Es compatible para los sistemas operativos *iOS 7* o superior, y se encuentra disponible en idioma alemán, checo, coreano, croata, chino, danés, español, francés, griego, hebreo, holandés, húngaro, inglés, italiano, noruego, polaco, portugués, ruso, sueco y turco.



**ILUSTRACIÓN 1:** APLICACIÓN MÓVIL SIXT (SIXT, 2016).

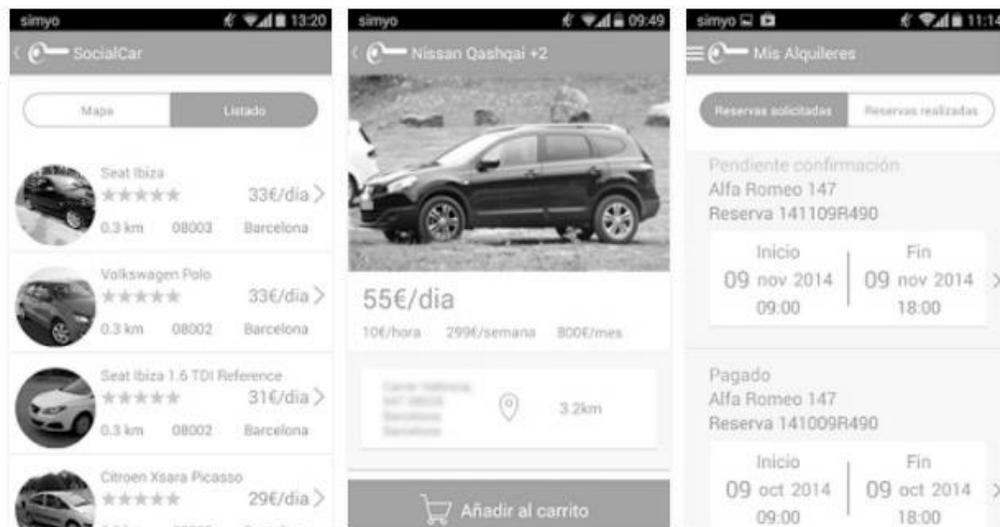
## **SocialCar**

Con esta aplicación nativa desarrollada para dispositivos móviles con sistema operativo *Android* proporcionada por la empresa *SocialCar* se puede alquilar coches y furgonetas particulares en toda España

a través del contacto directo con los propietarios. También permite a propietarios particulares alquilar sus vehículos por días o períodos más largos. Además, ofrece seguro a todo riesgo y asistencia personalizada vía correo electrónico o por teléfono.

A los usuarios les ofrece la funcionalidad de alquilar vehículo, incluido dentro de esta funcionalidad la búsqueda de autos, solicitud de reserva y pago del alquiler. Brinda acceso a mensajería para poder comunicarse con el propietario. El usuario puede registrarse en la plataforma o editar sus datos, cancelar reservas confirmadas y también ver su historial de reservas.

A los propietarios les permite recibir solicitudes de reserva, aceptar o rechazar las reservas pendientes. Les da acceso a mensajería para comunicarse con el conductor que alquiló su auto. Admite añadir vehículos en el sistema y editar sus datos. También se puede registrar en la plataforma y editar los datos de usuario, además de poder cancelar reservas confirmadas y ver el historial de las mismas. Requiere para su funcionamiento el sistema operativo *Android* 4.1 o versiones superiores a esta. La última versión de la aplicación disponible es 3.4.7 (SocialCar, 2017).



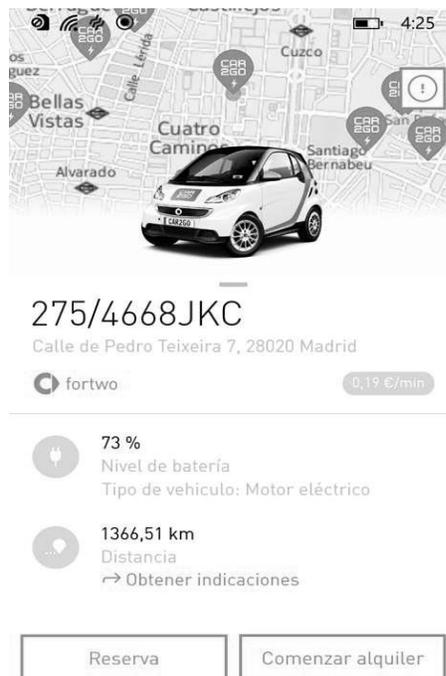
**ILUSTRACIÓN 2:** APLICACIÓN MÓVIL SOCIALCAR (SOCIALCAR, 2017).

### **Car2go**

Es la aplicación móvil nativa para sistema operativo *Android* de la empresa europea del mismo nombre que radica en ciudades tales como Madrid, Berlín, Viena, Roma o en otros países como Estados Unidos y China.

Esta muestra todos los coches disponibles en la ciudad en que se ubica el usuario o en todo el mundo, además la información importante relacionada con el auto como la ubicación aproximada, el nivel de llenado del tanque de combustible y el estado de la carga.

Posibilita reservar el vehículo hasta el rango comprendido entre los veinte y treinta minutos de antelación en dependencia de la ciudad en que se encuentre el usuario. Ofrece un servicio de guía hasta la sucursal de *Car2go* más cercana. Muestra todas las plazas de aparcamiento, zonas *Car2go* dentro de la ciudad, el transcurso del alquiler con datos sobre la dirección de salida, llegada y los gastos del viaje, además la función de radar con notificaciones de inserción cuando existan vehículos disponibles en los alrededores. Requiere para su funcionamiento la versión 4.0.3 de *Android* o superiores a esta. La última versión de la aplicación disponible es la 2.49.4 (*Car2go*, 2017).



**ILUSTRACIÓN 3:** APLICACIÓN MÓVIL *CAR2GO* (*CAR2GO*, 2017) .

### ***Drivy***

Esta aplicación móvil nativa desarrollada para los sistemas operativos móviles *iOS* y *Android* permite a los usuarios localizar el auto que necesita lo más cercano posible a su ubicación. Se puede contactar con los propietarios por teléfono o por correo electrónico, organizarlo todo y reservar el auto. Se recibe notificaciones *push* en tiempo real (confirmación del alquiler, nuevo mensaje, entre otras) además de que se puede solicitar

extensiones del alquiler con un *click*. A los propietarios les posibilita añadir su vehículo en menos de un minuto, gestionar su anuncio (actualizar su calendario de disponibilidad) u organizar sus alquileres (aceptar, rechazar o contactar con el conductor). Está disponible tanto para el sistema operativo *Android* y para *iOS* (*Drivy*, 2016) .



ILUSTRACIÓN 4: APLICACIÓN MÓVIL *DRIVY* (*DRIVY*, 2016).

### Resultados obtenidos del estudio de sistemas homólogos

Como resultado del estudio realizado se llega a la conclusión que no constituye una solución factible ninguna de las aplicaciones de reserva de autos anteriormente mencionadas. Estas fueron desarrolladas para empresas extranjeras privadas por lo que para su uso sería necesario realizar algún tipo de pago a las compañías propietarias. Se desenvuelven en entornos con características diferentes y bajo otras reglas de negocio que no se corresponden con la Agencia de Alquiler de Vehículos REX. No obstante, sí se identificaron características y funcionalidades que se pueden utilizar en el desarrollo de la solución propuesta como conocer la disponibilidad de autos y la reservación de autos listos para reservar, el catálogo de autos que se ofrece, así como ofrecer las informaciones generales sobre la agencia y tener opción para varios idiomas.

Una vez concluida la investigación sobre los sistemas homólogos existentes se decidió desarrollar una aplicación nativa que funcione en sistema operativo *Android* para la Agencia de Alquiler de Vehículos REX. Esta decisión está basada en que la mayoría de este tipo de aplicaciones a nivel mundial está desarrollada para un sistema operativo móvil específico. Además, en nuestro país hay una mayor tasa de usuarios que poseen teléfonos inteligentes que utilizan *Android* que con algún otro tipo de sistemas operativos móviles, debido a que todos los dispositivos que comercializa Etecsa que es la principal encargada de su venta y distribución en el país poseen este sistema operativo.

### 1.5 Proceso de desarrollo de aplicaciones móviles

Para el desarrollo de aplicaciones móviles se debe tener en consideración cada una de las limitaciones de los dispositivos para los cuales se desarrollen. Estos dispositivos móviles funcionan con batería, poseen varios tamaños de pantalla, así como distintas configuraciones.

El proceso de diseño y desarrollo de una aplicación móvil se puede dividir en cinco partes fundamentales las cuales son (Cuello, y otros, 2013):

**Conceptualización:** se parte de una idea que permita dar cumplimiento a una necesidad o ayudar a realizar una actividad determinada. Se debe responder a las expectativas reales y realizables, por lo que es imperiosa la necesidad de hacer un análisis de la viabilidad del concepto que se quiere realizar.

**Definición:** se procede a definir las funcionalidades de la aplicación en consonancia con el perfil de los usuarios y las especificaciones técnicas. Se establecen los parámetros de acceso al *hardware* del dispositivo, y para qué tipo de mercado específico se va a desarrollar. El dimensionado de todo ello permitirá determinar el alcance del proyecto, su duración, coste económico, complejidad del diseño y programación de la aplicación.

**Diseño:** en esta etapa se desarrollan los aspectos definidos de la etapa anterior. Se realiza, primeramente, un diseño representativo sin gráficos que será sometido a pruebas por varios usuarios. Una vez superada esta prueba inicial el diseño final es entregado al desarrollador en forma de archivos y pantallas estructuradas para que pueda ser añadido el código de programación. Los sistemas operativos ofrecen la posibilidad de interactuar con el usuario presentando en la pantalla los elementos necesarios para ello de forma distinta, lo cual debe tenerse en consideración por parte de los diseñadores pues el diseño de la aplicación influye de manera significativa tanto en el coste económico de la misma como en su desarrollo.

**Desarrollo:** el programador, en función del tipo de aplicación se encarga de desarrollar lo expuesto en los diseños, crear la estructura sobre la cual se apoyará el funcionamiento de la aplicación y el código funcional mediante un lenguaje de programación. Existen varios lenguajes de programación entre los que destacan: *Action Script, ADA, C, C#, C++, Java, ASP*, entre otros.

**Publicación:** una aplicación se publica después de realizarse una serie de pruebas para comprobar su correcto y estable funcionamiento, sin errores de usabilidad y diseño, y cumpliendo las políticas y requerimientos de las tiendas de aplicaciones. Durante toda la vida útil de la aplicación, es necesario evaluar el comportamiento y desempeño de la aplicación, detectar y corregir errores, y realizar mejoras o actualizaciones.

Una vez conocido el proceso de desarrollo de aplicaciones móviles es necesario establecer una guía de cómo se va desarrollar la aplicación móvil para la Agencia de Alquiler de Vehículos REX, para esto se debe definir una metodología de desarrollo de *software*.

## **1.6 Metodología de desarrollo de software**

Una metodología de desarrollo de *software* es un enfoque estructurado que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos. Las metodologías han evolucionado de manera significativa en las últimas décadas, tanto así, que pueden permitir el éxito o el fracaso de muchos de los sistemas desarrollados para distintas áreas (Cendejas Valdéz, 2014).

Dentro de las metodologías de investigación podemos encontrar:

Metodología Tradicional: impone una disciplina de trabajo sobre el proceso de desarrollo del *software*, con el objetivo de conseguir un *software* más eficiente y de mayor calidad. Se centra fundamentalmente en el control del proceso, a través de una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada (Ibañez Corrales, 2011). Otra de las características a destacar dentro de este enfoque son los altos costos que se producen al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil (Figuroa, y otros, 2015).

Entre las principales metodologías tradicionales tenemos *RUP* y *MSF*. Estas centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

Metodología Ágil: es aquella que posibilita adaptar la forma de trabajo a las condiciones que presenta el proyecto, para conseguir flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno (Merchán, y otros, 2011). Dentro de las metodologías ágiles más utilizadas se encuentran AUP (*Agil Unified Process*), Scrum y XP (*Extreme Programming*).

Para la implementación de la aplicación móvil para la Agencia de Alquiler de Vehículos se decidió seleccionar como metodología a utilizar *XP* la cual posee una tasa menor de errores al fomentar una comunicación continua entre los clientes y los desarrolladores. Ofrece una mayor satisfacción al programador, promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y garantiza un buen clima de trabajo. Utiliza una forma de trabajo donde la adaptación a nuevas circunstancias es fácil. Combina las prácticas de planificación, diseño, desarrollo y pruebas probadas para desarrollar *software* llevándolas al extremo. Además, reduce el costo del cambio en todas las etapas del ciclo de vida del sistema.

### ***XP (Extreme Programming)***

Es una metodología ágil diseñada para entornos dinámicos en los cuales se presentan equipos de desarrollo pequeños (hasta 10 programadores), orientada fuertemente hacia la codificación y programación de esta forma disminuye el tiempo de desarrollo de los proyectos. Hace énfasis en la comunicación informal y se necesita muy poca documentación en el desarrollo del sistema. La planificación de la misma es por entregas y posee las siguientes características (Letelier, 2005):

- Planificación por entregas (*releases*).
- Se priorizan aquellas historias de usuarios que el cliente selecciona porque son más importantes para el negocio
- Entregas:
  - ❖ Son lo más pequeñas posibles.
  - ❖ Se dividen en iteraciones (iteración con duración entre 2 o 3 semanas).
  - ❖ Están compuestas por historias de usuarios las cuales son la forma de especificar los requisitos.
  - ❖ A cada programador se le asigna una tarea de la historia de usuario.
  - ❖ La programación de tareas se realiza por parejas.
  - ❖ La pareja diseña, prueba, implementa e integra el código de la tarea.
  - ❖ Código dirigido por las pruebas.

- ❖ Código modular, para refactorizar<sup>6</sup> siempre que se pueda.

Ventajas de la metodología *XP* (Letelier, 2005) :

- Simplicidad: Propone el principio de hacer el desarrollo más simple y que pueda funcionar, en relación al proceso y la codificación.
- Comunicación: Algunos problemas en los proyectos tienen origen en que alguien no dijo algo importante en algún momento. *XP* hace casi nula la falta de comunicación.
- Realimentación: Debe ser concreta y frecuente entre el cliente, el equipo y los usuarios finales. Da una mayor oportunidad de dirigir el esfuerzo eficientemente.
- Coraje o valentía: Hay que ser valiente para confiar en que la programación individual beneficia la calidad del código sin repercutir negativamente en la productividad. La simplicidad es uno de los principios más difíciles de adoptar. Se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo mientras el cliente espera.

Cada vez es mayor la cifra de empresas que deciden apostar por las metodologías ágiles, aunque hay otros proyectos que necesitan ser desarrollados con metodologías tradicionales. En la actualidad la necesidad que más domina es efectuar procedimientos que viabilicen entregar productos de mayor calidad en el tiempo y el costo acordado inicialmente. Las metodologías tradicionales no logran adaptarse fácilmente a las nuevas expectativas de los usuarios y a las exigencias cambiantes del mercado por lo que se concluye que no son suficientes para alcanzar este objetivo. La tendencia es la metodología híbrida, la cual va a estar en dependencia de las características del proyecto que se quiera desarrollar. No obstante, las empresas que apuestan por utilizar una metodología ágil consiguen gestionar sus proyectos de forma más eficaz para lograr así reducir los costos e incrementar aún más su productividad (Definición de una metodología ágil, 2011).

---

<sup>6</sup> Refactorización: modificación del código fuente sin cambiar su comportamiento. Se realiza a menudo como parte del proceso de desarrollo del *software*: los desarrolladores alternan la inserción de nuevas funcionalidades y casos de prueba con la refactorización del código para mejorar su consistencia interna y su claridad.

A continuación, se describen las herramientas utilizadas para el diseño e implementación de la aplicación móvil para la Agencia de Alquiler de Vehículos REX.

## 1.7 Herramientas Utilizadas

### Herramienta para el modelado *Visual Paradigm 8.0*

Es una herramienta *CASE (Computer Aided Software Engineering)*. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, a través del paso por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas.

Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: *Enterprise, Professional, Community, Standard, Modeler* y *Personal*. Existe una alternativa libre y gratuita de este *software*, la versión *Visual Paradigm for UML 6.4 Community Edition*. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Esta herramienta permite aumentar la calidad del *software*, a través de la mejora de la productividad en el desarrollo y mantenimiento del *software*. Aumenta el conocimiento informático de una empresa para ayudar a la búsqueda de soluciones para los requisitos. También permite la reutilización del *software*, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de *Software* (Pressman, 2002).

### IDE para el desarrollo de aplicaciones móviles *Android Studio*

Un IDE (*Integrated Development Environment*) es una herramienta que ayuda a desarrollar de una manera amigable las aplicaciones. Brinda ayuda visual en la sintaxis, plantillas, *wizards*<sup>7</sup>, *plugins*<sup>8</sup> y sencillas opciones para probar y hacer un debug (*Mentoring*, 2012).

---

<sup>7</sup> Wizards: Cualquier programa intuitivo que guía paso a paso al usuario para realizar una tarea.

<sup>8</sup> Plugins: Aplicación o programa informático que se relaciona con otra para agregarle una función nueva y generalmente muy específica.

*Android Studio* es un entorno de desarrollo integrado para el sistema operativo *Android* lanzado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno *Eclipse*, es el *IDE* más utilizado. Permite ver los cambios que se realizan a las aplicaciones en tiempo real, además de comprobar cómo se visualiza en diferentes dispositivos con diferentes configuraciones y resoluciones. Posee un entorno de desarrollo claro y resistente. Proporciona asistentes y plantillas para los elementos más usuales de programación *Android*. Es muy completo al ofrecer varias herramientas extra para agilizar el desarrollo de las aplicaciones desarrolladas (Rosso, 2014).

### **Sistema gestor de base de datos SQLite 3**

“Un sistema gestor de bases de datos (*Data Base Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Cuando se define una base de datos se debe definir los tipos de datos, estructuras y restricciones de los datos que se almacenan en ella. Toda información que describe la base de datos se conoce como catálogo o diccionario de la base de datos (metadatos) (Caballero González, y otros, 2016).”

**SQLite:** es una herramienta de *software* libre lo cual posibilita no tener que pagar licencia alguna para su uso. Facilita almacenar información en dispositivos de una forma sencilla, eficaz, potente, rápida y en equipos que posean bajas capacidades de *hardware*, como puede ser un teléfono celular. Implementa el estándar *SQL92* y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que soporte tanto las consultas más sencillas como las más complejas del lenguaje *SQL* (*Structured Query Language*). Puede ser utilizado tanto en dispositivos móviles como en sistemas de escritorio a diferencia de los sistemas de gestión de bases de datos cliente-servidor, sin tener que realizar procesos complejos de importación y exportación de datos. Al existir una compatibilidad al 100% entre las diversas plataformas disponibles, hace que la portabilidad entre dispositivos y plataformas sea transparente (Ponsoda Montiel, 2008).

Algunas de las principales características de *SQLite* son (SQLite, 2015):

- Las transacciones son atómicas, consistentes, aisladas y duraderas aún después de que el sistema se bloquee u ocurra alguna falla en el suministro eléctrico.
- Toda la base de datos se almacena en un único archivo.
- Soporta tamaños del orden de los *gigabytes* o de los *terabytes*.
- Simple, fácil de usar de tamaño pequeño ya totalmente configurada.

- El motor de SQLite no es un proceso independiente con el que el programa principal se comunica.
- No posee dependencias externas.
- De código abierto y multiplataforma y (compatible con sistemas operativos como: *Android, BSD, iOS, Linux, Mac, Solaris, VxWorks* y *Windows*).

## 1.8 Lenguajes Utilizados

### Lenguaje de Modelado UML9 2.0

Es un lenguaje de modelado de sistemas de *software* que está respaldado y certificado como un estándar por *OMG (Object Management Group)*. Es un lenguaje gráfico creado para construir, documentar, especificar y visualizar un sistema de *software*. Mediante su utilización se ve disminuido el ámbito del problema de estudio al apuntar solo un aspecto a la vez. Es visual y a través de su sintaxis se modelan varios aspectos de situaciones reales, que posibilitan una mayor interpretación y entendimiento. Unifica varias técnicas de modelado en una sola. Posee una sintaxis y una semántica con reglas que tratan sobre agrupar los elementos del lenguaje y el significado de esta agrupación. Brinda un estándar para representar modelos del sistema, el cual incluye aspectos conceptuales como funcionalidades del sistema, procesos del negocio y aspectos específicos tales como esquemas de bases de datos, lenguajes de programación y componentes de *software* reutilizables (Peró Carbonell, y otros, 2013).

Algunas de las ventajas que ofrece la utilización de este lenguaje de modelado son (López, y otros, 2011):

- Es estándar pues facilita la comunicación.
- Es utilizado para realizar la modelación de un *software* en distintos dominios: sistemas críticos y de tiempo real, sistemas *web*, entre otros.
- Está basado en una notación gráfica breve y de fácil aprendizaje y entendimiento.
- Posee una semántica bien definida.
- Es fácilmente extensible.

---

<sup>9</sup> UML: Lenguaje Unificado de Modelado o *Unified Modeling Language*.

## Lenguaje de programación Java 8.0

Es un lenguaje muy valorado porque los programas desarrollados se pueden ejecutar en diversas plataformas con sistemas operativos como *Windows*, *Mac OS*, *Android*, *Linux* o *Solaris*. Los programas Java son portables, es decir, independientes de la plataforma, porque pueden ejecutarse en cualquier ordenador o dispositivo móvil, independientemente del sistema operativo que tengan instalado: Un programa Java puede ejecutarse en un ordenador de mesa, un ordenador portátil, una tableta, un teléfono, un reproductor de música o en cualquier otro dispositivo móvil con cualquier sistema operativo (Guevara, 2012).

### 1.9 Conclusiones parciales

Al concluir el capítulo se pueden arribar a las siguientes conclusiones:

- El estudio de todos los conceptos expuestos, en conjunto con el análisis al negocio de la Agencia de Alquiler de Vehículos REX permitió obtener los principales elementos teóricos que sustentan la elaboración de una aplicación móvil para la renta de vehículos.
- El estudio de los sistemas homólogos existentes permitió identificar que ninguna de estas aplicaciones da solución al problema planteado, por lo que queda evidenciada la necesidad de implementar una aplicación móvil.
- El estudio de homólogos permitió identificar funcionalidades que se deben tener presentes a la hora de implementar una aplicación móvil para la renta de vehículos tales como: conocer la disponibilidad de autos para reservar y la realización de reservas de autos. Brindar los principales temas referentes a la agencia, el catálogo de autos que posee, así como ofrecer la posibilidad de que la aplicación posea varios idiomas.
- El estudio de las herramientas, tecnologías y metodología de desarrollo de *software* permitió crear la base tecnológica adecuada para el desarrollo de este tipo de aplicación: *XP* como metodología a utilizar apoyándose en la herramienta *CASE Visual Paradigm* para el modelado, para el almacenamiento de los datos se utiliza el Sistema Gestor de Bases de Datos *SQLite* y como *IDE* de desarrollo el *Android Studio*.

## Capítulo 2. Análisis y diseño de la aplicación móvil para la Agencia de Alquiler de Vehículos REX

En el presente capítulo son mencionados los resultados que se alcanzan en las fases de conceptualización y diseño de la aplicación móvil para la Agencia de Alquiler de Vehículos REX. Una vez que se realiza una descripción del entorno a través de un modelo de dominio en el que se analiza los conceptos, las entidades y sus relaciones, se realiza la definición de las principales características a las cuales debe dar cumplimiento la aplicación en cuestión de requisitos funcionales y no funcionales. También se describen las historias de usuarios que posteriormente darán paso a las funcionalidades de la aplicación y se presentan los diagramas de diseño para poder alcanzar una visión generalizada de las funcionalidades de la aplicación.

### 2.1 Propuesta de Solución

La aplicación móvil para el sistema de alquiler de vehículos REX debe estar orientada principalmente a dos direcciones, una primera parte que se encargue de mostrar la información relevante relacionada con el alquiler y disponibilidad de autos en las distintas oficinas de rentas de todo el país. Por otra parte, la aplicación debe permitir la reserva de automóviles en cualquiera de las sucursales del país para los distintos tipos de rentas que existen: renta con o sin chofer, renta lineal, traslado y renta por horas. La aplicación debe admitir el trabajo sin conexión, así como con conectividad con acceso a *Internet* para las reservaciones y consultas de disponibilidad. Además, debe hacer uso de una base de datos en SQLite 3 donde serán almacenados en el dispositivo los datos de las reservas realizadas por el usuario y la disponibilidad general del sistema de reservas REX.

### 2.2 Modelo de dominio

El modelo del dominio muestra clases conceptuales significativas en un dominio de problema; es un artefacto de la disciplina de análisis construido durante la fase de inicio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de *software* sino de la propia realidad física. Es una representación de las clases conceptuales del mundo real, no de componentes de *software*. No se trata de un conjunto de diagramas que describen clases *software*, u objetos *software* con responsabilidades (Larman, 2003).

Como conclusión se plantea que el modelo de dominio es aquel que se utiliza para realizar la modelación de los conceptos que participan en el negocio. En este se plasman los conceptos fundamentales para el

entorno de la problemática a resolver, para intentar así establecerlo como un punto de partida para el diseño de la misma. Representa una vía para lograr la comprensión del sector de negocios del sistema y posibilita identificar las interrelaciones entre las clases de dominio.

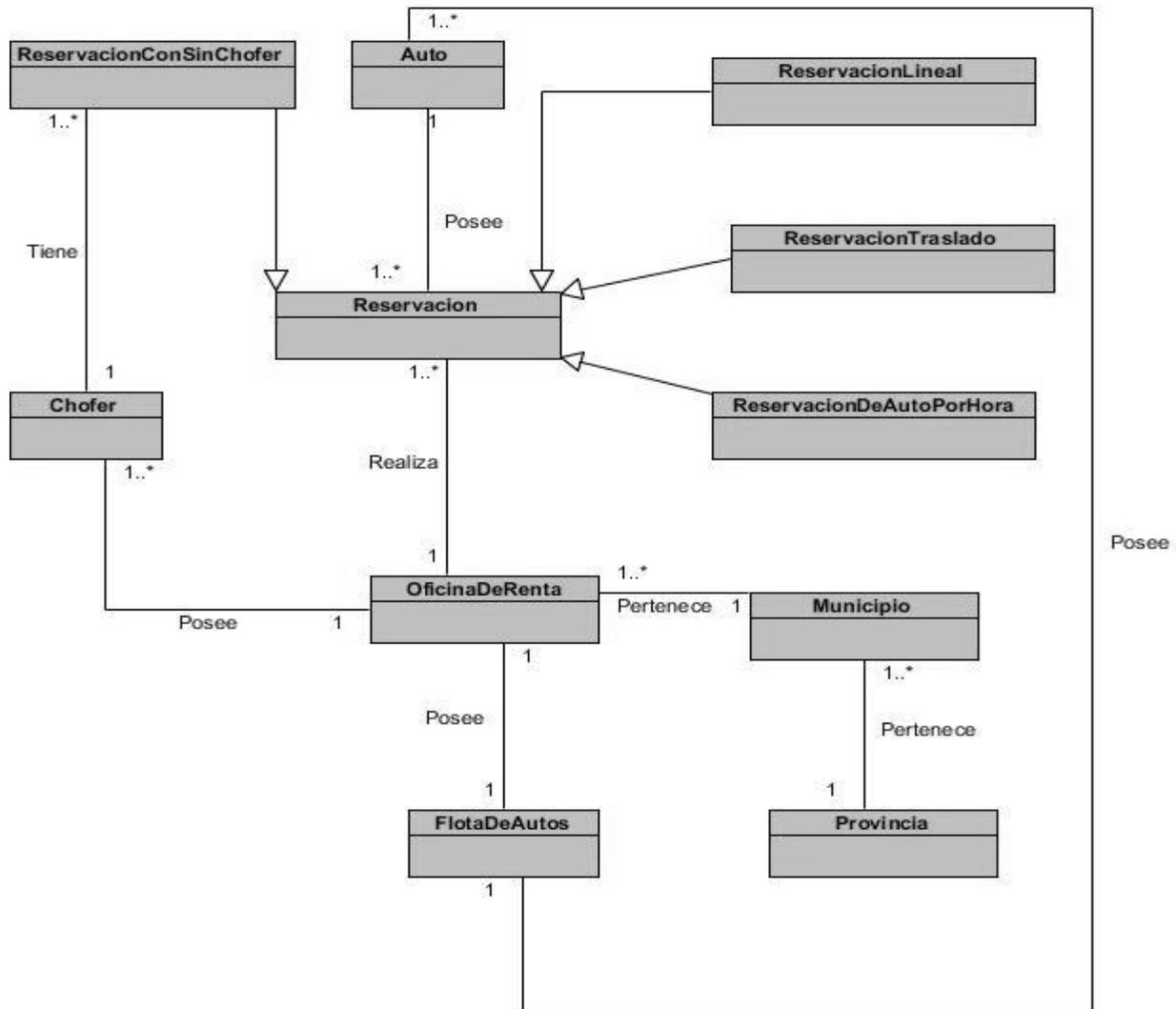


ILUSTRACIÓN 5: MODELO DE DOMINIO (ELABORACIÓN PROPIA).

### Descripción del Modelo de Dominio

**Reservación:** Solicitud de un usuario para rentar un auto después de pagar una determinada suma de dinero.

**Oficina De Renta:** Local donde el usuario puede acceder a recoger el auto después de haber efectuado la reservación a través de la aplicación móvil o por *internet*. También se puede efectuar la renta de forma directa en este local.

**Flota De Autos:** Cantidad de autos disponibles para ser rentados.

**Reservación Traslado:** Reservación en la que el usuario solicita ser traslado por los sitios de principal interés turístico en nuestro país.

**Reservación De Autos Por Hora:** Reservación en la que el usuario solicita la renta de un auto solo por un número determinado de horas.

**Municipio:** Municipio donde está ubicada la oficina de renta.

**Provincia:** Provincia donde está ubicada la oficina de renta.

**Auto:** Vehículo disponible para ser rentado por un determinado usuario.

**Chofer:** Conductor profesional disponible para casos en que el usuario lo solicite.

**Reservación Con Chofer:** Reservación en la que se solicita la presencia de un conductor profesional.

**Reservación Sin Chofer:** Reservación en la que no se solicita la presencia de un conductor profesional.

**Reservación Lineal:** Reservación que ofrece soluciones exclusivas de transportación para compañías y hombres de negocios por un período de tiempo desde tres hasta treinta y seis meses.

### **2.3 Levantamiento de requisitos**

El proceso de levantamiento de requisitos comienza cuando se interactúa con el cliente. En este proceso se hace una vinculación entre las necesidades que tiene el usuario y la definición de lo que debe hacer el sistema. Se plantean las condiciones que son necesarias para que los requisitos funcionales sean cumplidos o como se conocen comúnmente los requisitos no funcionales. Se especifican los usuarios que van a interactuar con el sistema y los niveles de acceso o permisos que tendrán en el sistema (Turner, 2005).

### **Funcionalidades del sistema**

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. La aplicación móvil para la Agencia de Alquiler de Vehículos posee las siguientes funcionalidades:

**TABLA 1:** FUNCIONALIDADES DEL SISTEMA.

Funcionalidad	
<b>F1:</b>	Buscar disponibilidad de autos para rentar.
<b>F2:</b>	Reservar autos con o sin chofer.
<b>F3:</b>	Reservar autos por hora.
<b>F4:</b>	Reservar auto para traslados.
<b>F5:</b>	Modificar reservación de auto con o sin chofer.
<b>F8:</b>	Modificar reservación de auto para traslados.
<b>F9:</b>	Modificar reservación de auto por hora.
<b>F10:</b>	Cancelar reservación de auto por hora.
<b>F11:</b>	Cancelar reservación de auto para traslados.
<b>F12:</b>	Cancelar reservación de auto con o sin chofer.
<b>F13:</b>	Mostrar informaciones importantes para la renta de autos.
<b>F14:</b>	Mostrar términos y condiciones para la renta de autos.
<b>F15:</b>	Mostrar información acerca de la renta lineal.
<b>F16:</b>	Mostrar información acerca de las oficinas de renta en todo el país.
<b>F17:</b>	Mostrar flota de vehículos.
<b>F18:</b>	Internacionalización de la aplicación.

### Requisitos No Funcionales

Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación. Son aquellos que no hacen mención directa a las funciones específicas del sistema, sino a las propiedades emergentes que posee este como la capacidad de almacenamiento, tiempo de respuesta o la fiabilidad. Definen las restricciones del sistema como la representación de los datos, la capacidad de los dispositivos de entrada/salida, así como la disponibilidad, la protección entre otras propiedades emergentes (Olivera, y otros, 2010).

**TABLA 2:** REQUISITOS NO FUNCIONALES DEL SISTEMA.

<b>Requisitos No Funcionales</b>	
<b>Requerimiento de estándares</b>	
<b>NF1:</b>	Todas las clases y funciones deben quedar bien documentadas.
<b>Requerimiento de fiabilidad</b>	
<b>NF2:</b>	Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad.
<b>Requerimiento de usabilidad</b>	
<b>NF3:</b>	Internacionalización de la aplicación en idiomas español e inglés.

## 2.4 Historias de usuario

La metodología de desarrollo ágil de *software XP* tiene como uno de sus artefactos a las historias de usuario (HU), la cual consiste en una técnica para representar y especificar los requisitos del *software* mediante un conjunto de tablas en las que se hace una descripción breve de las características que debe tener el sistema. Estas se descomponen en tareas de programación las cuales serán implementadas durante una iteración (Canós, y otros, 2003).

En la presente investigación se generaron un total de 10 HU, a continuación, se mostrarán 3 y las restantes en el Anexo #3.

TABLA 3: HU\_1

<b>Historia de Usuario</b>	
<b>Número: 01</b>	<b>Nombre del requisito:</b> Gestionar reservación de traslado.
<b>Programador:</b> Yobal Concepción Cabrera	<b>Iteración Asignada: 1</b>
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 4 semanas
<b>Riesgo en Desarrollo:</b> No aplica	<b>Tiempo Real:</b> 4 semanas
<p><b>Descripción:</b></p> <p>Permite realizar una reservación para traslado en el sistema.</p> <p>Para realizar una reservación de traslado es necesario llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>• Nombre (Campo de texto de carácter obligatorio)</li> <li>• Apellidos (Campo de texto de carácter obligatorio)</li> <li>• Correo electrónico (Campo alfanumérico de carácter obligatorio)</li> <li>• Observaciones (Campo de texto de carácter no obligatorio)</li> <li>• Modelo de auto (Campo seleccionable de carácter obligatorio)</li> <li>• Fecha y hora de recogida (Valor seleccionable de un calendario de carácter obligatorio)</li> <li>• País de procedencia (Campo seleccionable de carácter obligatorio.)</li> </ul> <p>Según el negocio los accesorios pueden ser:</p> <ul style="list-style-type: none"> <li>• Silla para niños de [0-1] año.</li> <li>• Silla para niños de [1-3] años.</li> </ul>	
<p><b>Observaciones:</b></p> <p>El valor del correo debe ser un correo electrónico válido. Ejemplo: yconcepcionc@estudiantes.uci.cu</p>	
<b>Prototipo elemental de interfaz gráfica de usuario:</b>	

TABLA 4: HU\_2

Historia de Usuario	
<b>Número: 02</b>	<b>Nombre del requisito:</b> Gestionar reservación por horas.
<b>Programador:</b> Yobal Concepción Cabrera	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 semanas
<b>Riesgo en Desarrollo:</b> No aplica	<b>Tiempo Real:</b> 3 semanas
<b>Descripción:</b> Permite realizar una reservación por horas en el sistema. Para realizar una reservación por horas es necesario llenar los siguientes campos:	

- Nombre (Campo de texto de carácter obligatorio)
- Apellidos (Campo de texto de carácter obligatorio)
- Correo electrónico (Campo alfanumérico de carácter obligatorio)
- Observaciones (Campo de texto de carácter no obligatorio)
- Modelo de auto (Campo seleccionable de carácter obligatorio)
- Fecha y hora de recogida (Campo seleccionable de un calendario de carácter obligatorio)
- País de procedencia (Campo seleccionable de carácter obligatorio.)
- Cantidad de horas (Campo numérico de carácter obligatorio)
- Lugar de recogida (Campo seleccionable de carácter obligatorio)
- Lugar de entrega (Campo seleccionable de carácter obligatorio)

Según el negocio los accesorios pueden ser:

- Silla para niños de [0-1] año.
- Silla para niños de [1-3] años.

**Observaciones:**

El valor del correo debe ser un correo electrónico válido. Ejemplo: yconcepcionc@estudiantes.uci.cu

**Prototipo elemental de interfaz gráfica de usuario:**

**Renta Por Horas**

Nombre  Apellidos

Correo  País

Categoría de auto  Fecha

Accesorios

Silla para niños [0-1] año  Silla para niños [1-3] años

Cantidad de horas

Lugar de recogida  Lugar de entrega

Observaciones

Aceptar Cancelar

TABLA 5: HU\_3

Historia de Usuario	
<b>Número: 03</b>	<b>Nombre del requisito:</b> Gestionar reservación con o sin chofer
<b>Programador:</b> Yobal Concepción Cabrera	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 4 semanas
<b>Riesgo en Desarrollo:</b> No aplica	<b>Tiempo Real:</b> 4 semanas
<b>Descripción:</b> Permite realizar una reservación por horas en el sistema.	

Para realizar una reservación por horas es necesario llenar los siguientes campos:

- Nombre (Campo de texto de carácter obligatorio)
- Apellidos (Campo de texto de carácter obligatorio)
- Correo electrónico (Campo alfanumérico de carácter obligatorio)
- Observaciones (Campo de texto de carácter no obligatorio)
- Modelo de auto (Campo seleccionable de carácter obligatorio)
- Fecha y hora de recogida (Campo seleccionable de un calendario de carácter obligatorio)
- País de procedencia (Campo seleccionable de carácter obligatorio.)
- Lugar de recogida (Campo seleccionable de carácter obligatorio)
- Lugar de entrega (Campo seleccionable de carácter obligatorio)
- Fecha de recogida (Campo seleccionable de carácter obligatorio)
- Fecha de entrega (Campo seleccionable de carácter obligatorio)
- Chofer (Campo seleccionable de carácter no obligatorio)

Según el negocio los accesorios pueden ser:

- Silla para niños de [0-1] año.
- Silla para niños de [1-3] años.

**Observaciones:**

El valor del correo debe ser un correo electrónico válido. Ejemplo: yconcepcionc@estudiantes.uci.cu

**Prototipo elemental de interfaz gráfica de usuario:**

The screenshot shows a 'Reserva' (Reservation) form with the following elements:

- Nombre** (Name) and **Apellidos** (Surnames): Text input fields.
- Correo** (Email): Text input field.
- País** (Country): Dropdown menu.
- Categoría de auto** (Car Category): Dropdown menu.
- Fecha** (Date): Text input field.
- Fecha de recogida** (Pick-up Date) and **Fecha de entrega** (Delivery Date): Text input fields.
- Chofer** (Driver): Radio button.
- Accesorios** (Accessories):
  - Silla para niños [0-1] años
  - Silla para niños[1-3] años
- Lugar de recogida** (Pick-up Location) and **Lugar de entrega** (Delivery Location): Dropdown menus.
- Observaciones** (Observations): Large text area.
- Aceptar** (Accept) and **Cancelar** (Cancel): Buttons at the bottom right.

## 2.5 Plan de entregas

**TABLA 6:** PLAN DE ENTREGAS

Entregable	Duración (semanas)	Fecha de entrega
Gestión de reservaciones de autos con o sin chofer, por hora y de traslados.	11	Abril 2017
Servicio de información al cliente e internacionalización de la aplicación.	9	Mayo 2017

## 2.6 Plan de iteraciones

Es cuando se genera una planificación en la que los desarrolladores junto al cliente delimitan los tiempos correctos para la implementación de las historias de usuarios, se establece la prioridad y cuáles de estas serán implementadas en cada versión del programa (Carballo, 2014).

**TABLA 7:** PLAN DE ITERACIONES

Iteración	Duración (semanas)	Historias de Usuario
1	11	1 - 5
2	9	6 - 10

## 2.7 Tarjetas CRC

La metodología *XP* no utiliza diagramas de clases, en su lugar utiliza varias técnicas. Una de ellas son las Tarjetas *CRC* (*Class Responsibility Collaboration*). Estas garantizan la identificación de clases y asociaciones que participan del diseño del sistema. Posibilitan la obtención de las responsabilidades que debe cumplir cada clase. Realizan el establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades. Están compuestas por tres partes fundamentales: el nombre, las responsabilidades y los colaboradores. La descripción de una clase puede realizarse como cualquier objeto o clase por sus atributos y métodos. Las responsabilidades describen a alto nivel el propósito de la existencia de la clase y los colaboradores son aquellas clases que ayudan a ejecutar una responsabilidad.

A continuación, se muestran 3 Tarjetas CRC y las restantes se encuentran en el Anexo #2.

**TABLA 8:** TARJETA CRC "RESERVACIÓN"

Reservación	
Responsabilidades	Colaboración
Modelo que representa una reservación de automóvil en el sistema.	Oficina, Reserva, Auto, País, Persona, Traslado, Por Horas, Accesorio, Precio, Lugar

**TABLA 9:** TARJETA CRC "PERSONA"

Persona	
Responsabilidades	Colaboración
Modelo que representa a una persona en la aplicación.	Reservación

**TABLA 10:** TARJETA CRC "TRASLADO"

Traslado	
Responsabilidades	Colaboración
Modelo que representa a una reservación de traslado en el sistema.	Reservación

## 2.8 Arquitectura del software

Para el desarrollo de una aplicación móvil es necesario definir la arquitectura de software a utilizar que se puede definir como: la estructura de un sistema o bien la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos (Pressman, 2011).

La arquitectura representa la clave para comprender, organizar y comunicar un sistema, además, facilita la evolución de la solución. Es diseñada para satisfacer los requerimientos funcionales y no funcionales establecidos por los usuarios, clientes y proveedores del sistema. Un software que no posee un correcto diseño arquitectónico puede funcionar de forma muy deficiente o simplemente no funcionar generando consecuencias para la organización que sirve. Para las empresas que dependen de los sistemas de información las arquitecturas de software son fundamentales para el logro de sus objetivos organizacionales, lo que incluye el poder evolucionar rápidamente según las condiciones altamente cambiantes de los mercados actuales (Villegas, 2015).

Para definir la arquitectura de la aplicación móvil se debe tener presente los patrones asociados.

### **Patrones de arquitectura**

Ofrecen soluciones a problemas de determinada arquitectura de *software*. Facilitan una descripción de los elementos y el tipo de relación que poseen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico formula un esquema de organización estructural esencial para un sistema de *software*, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (Pressman, 2011). Algunos de estos patrones son: Programación por capas, Tres niveles, Pipeline, Invocación implícita, Arquitectura dirigida por eventos, Arquitectura orientada a servicios, Modelo Vista Controlador, entre otros patrones. Para el desarrollo de la aplicación móvil para la Agencia de Alquiler de Vehículos REX se seleccionó el patrón Modelo Vista Controlador.

### **Patrón Modelo-Vista-Controlador**

Este modelo fue diseñado para disminuir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características fundamentales vienen determinadas por el hecho de que el Modelo, las Vistas y los Controladores se tratan como entidades separadas. Realiza la separación de los componentes de un programa, lo cual posibilita su implementación de forma individual. Posee una interfaz de programación de aplicaciones muy bien definida. La conexión entre las Vistas y los Modelos se establece de forma dinámica en tiempo de ejecución y no en tiempo de compilación. Posibilita que las piezas de un programa se puedan construir por separado y luego integrarlas en tiempo de ejecución (Fernández, y otros, 2012).

Después de lo anteriormente mencionado se puede desglosar el modelo de arquitectura en sus tres componentes:

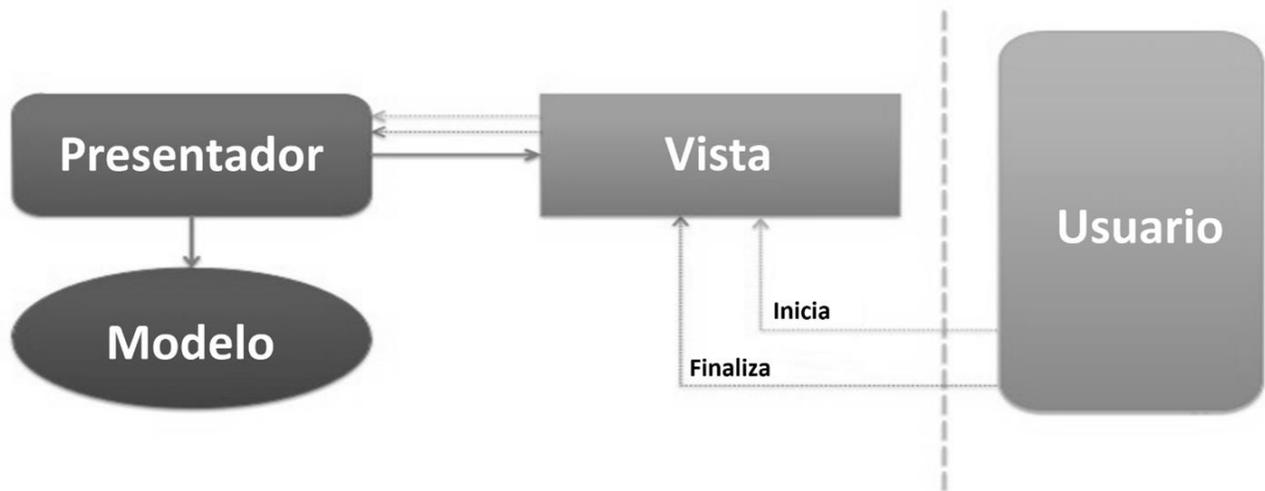


ILUSTRACIÓN 6: MODELO-VISTA-CONTROLADOR EN ANDROID.

### Modelo-Vista-Controlador en Android

En *Android* existe la problemática de que las actividades están fuertemente acopladas tanto con la interfaz como con las mecánicas de acceso a datos, hasta tal punto que existen clases que mezclan los adaptadores, que son parte de la vista, con los cursores, algo que debería estar relegado a lo más profundo de la capa de acceso a datos. Para que una aplicación sea fácilmente extensible y de fácil mantenimiento necesita tener bien separadas sus capas. Este modelo independiza la vista de la forma de conseguir esos datos, divide la aplicación en al menos tres capas distintas, realizándole pruebas a cada una de ellas de forma independiente. Realizar la independización de la interfaz de la lógica no es para nada fácil en *Android*, pero con este modelo es más sencillo evitar que las actividades se transformen en clases demasiado acopladas de muchas líneas de código. En aplicaciones de gran tamaño es de vital importancia tener correctamente organizado el código para garantizar que el mantenimiento o ampliación de la aplicación no se vuelva demasiado complicado. Existen muchas variaciones de MVC y cada una se puede ajustar en dependencia a la idea del patrón según las necesidades existentes. El patrón varía fundamentalmente en función de la cantidad de responsabilidades que sean delegadas en el presentador (Leiva Gordillo, 2014).

### **El presentador**

Actúa de intermediario entre la vista y el modelo. Recupera los datos del modelo los devuelve a la vista en un formato reconocible por esta. A diferencia del MVC típico, también decide lo que sucederá cuando se realiza la interacción con la vista.

### **La vista**

Es implementada por una *Activity* que a su vez contiene una referencia al presentador. La única función que realizará la vista será efectuar la llamada a un método del presentador cada vez que se realice una acción sobre la interfaz.

### **El modelo**

Simplemente es el proveedor de los datos que se quieren mostrar en la vista. Representa el tratamiento de los datos en la aplicación.

### **Patrones de diseño:**

Son valorados debido a que imponen reglas sobre la arquitectura y expresan esquemas para solucionar problemas de un mismo tipo que pueden presentarse durante el desarrollo de la aplicación. Constituyen la base para realizar la búsqueda de soluciones a problemas que se presentan en el desarrollo de *software* y otros marcos del diseño de interacción. Una solución es considerada un patrón de diseño cuando posee ciertas características entre las cuales se encuentran: su efectividad debe haberse comprobado resolviendo problemas similares en otras ocasiones, debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Pressman, 2011).

### **Patrones GRASP (*Responsability Assignment Software Patterns*)**

Patrones que se basan en la concesión de responsabilidades a objetos. Se considera una buena práctica para el desarrollo eficaz de la Programación Orientada a Objetos (Grosso, 2011).

### **Creador:**

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. Este patrón es el encargado de asignar a las clases la responsabilidad de instanciar otra clase, si esta asignación se realiza correctamente se puede garantizar que el diseño pueda soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización (Informáticos, 2005).

Este patrón se manifiesta en el siguiente fragmento de código, al crear un objeto como instancia de la clase *ProgressDialog* y *VerticalSteperFormLayout*.

```
private ProgressDialog progressDialog;  
private VerticalStepperFormLayout verticalStepperForm;
```

**Experto en información:** es cuando se otorga una responsabilidad determinada al experto en información, es decir a la clase que posee toda la información que se necesita para desarrollar una responsabilidad (Informáticos, 2005).

A continuación, se presenta un ejemplo del patrón experto en información en la aplicación:

```
public String _ReservaToJSON() {  
    JSONObject jsonObject = new JSONObject();  
  
    try {  
        jsonObject.put(STATE_NAME,getName());  
        jsonObject.put(STATE_APELLIDOS,getAPELLIDOS());  
        jsonObject.put(STATE_CORREO,getCORREO());  
        jsonObject.put(STATE_PAIS, getPAIS());  
        jsonObject.put(STATE_MODELO, getMODELO());  
        jsonObject.put(STATE_FECHA,getFECHA());  
        jsonObject.put(STATE_SILLA13,getSILLA13());  
        jsonObject.put(STATE_SILLA36,getSILLA36());  
        jsonObject.put(STATE_FECHA_ENTRE,getFECHA_ENTREGA());  
        jsonObject.put(STATE_FECHA_RECOJ,getFECHA_RECOJIDA());  
        jsonObject.put(STATE_CHOFER,getCHOFER());  
        jsonObject.put(STATE_LUGAR_RECJ,getLUGAR_RE());  
        jsonObject.put(STATE_LUGAR_ENTREG,getLUGAR_EN());  
        jsonObject.put(STATE_OBS,getOBSERVACIONES());  
        jsonObject.put(STATE_TIPO,getTIPO());  
  
        return jsonObject.toString();  
    } catch (JSONException e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

Esta función se encarga de validar los datos al adicionar o editar los campos de una reserva. Este patrón se evidencia en este método al darle la responsabilidad a la clase Reserva de validar sus campos, al ser ella la que contiene la información necesaria para la gestión de las reservas.

**Controlador:** Da la responsabilidad de gestionar un mensaje de un determinado evento del sistema a una clase controladora. Es el intermediario entre una determinada interfaz y el algoritmo que la implementa, toma los datos del usuario y los remite a las distintas clases en dependencia del método al que se llame.

El patrón controlador es utilizado en el siguiente fragmento de código, donde se llama a la función “*createFechaStep()*” para obtener la fecha que se entra en la reservación y guardarla en la base de:

```
private View createFechaStep() {
    LayoutInflater inflater = getLayoutInflater();
    View v = inflater.inflate(R.layout.custom_layout_for_fecha, null);
    FechaTextView = (TextView) v.findViewById(R.id.et_fecha);
    Button BAddFecha = (Button) v.findViewById(R.id.bt_add_fecha);

    BAddFecha.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showDialogPickDate();
        }
    });

    return v;
}
```

### **Patrones de Diseño GoF (Gang of Four)**

Son patrones que constituyen soluciones técnicas que se basan en la Programación Orientada a Objetos que benefician la reutilización de código.

**Decorador:** Es un patrón que amplía la funcionalidad de un objeto dinámicamente, y utiliza para esto una instancia de la subclase de la clase original que delega las operaciones al objeto original. Ofrece una alternativa muy flexible para sumar una funcionalidad a una determinada clase (Peralta, y otros, 2014).

En el desarrollo de la aplicación móvil para el Sistema de alquiler de vehículos REX se utilizó el siguiente decorador:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_vertical_stepper_form);
}
```

```

initializeActivity();
}
    
```

## 2.9 Modelo de datos

Es un grupo de herramientas conceptuales que son utilizadas para representar determinada información en datos. Conjunto de convenciones, reglas y conceptos que posibilitan la especificación de los datos, la semántica asociada a estos, las restricciones de integridad y las relaciones existentes entre ellos (El Modelo Relacional, 2011).

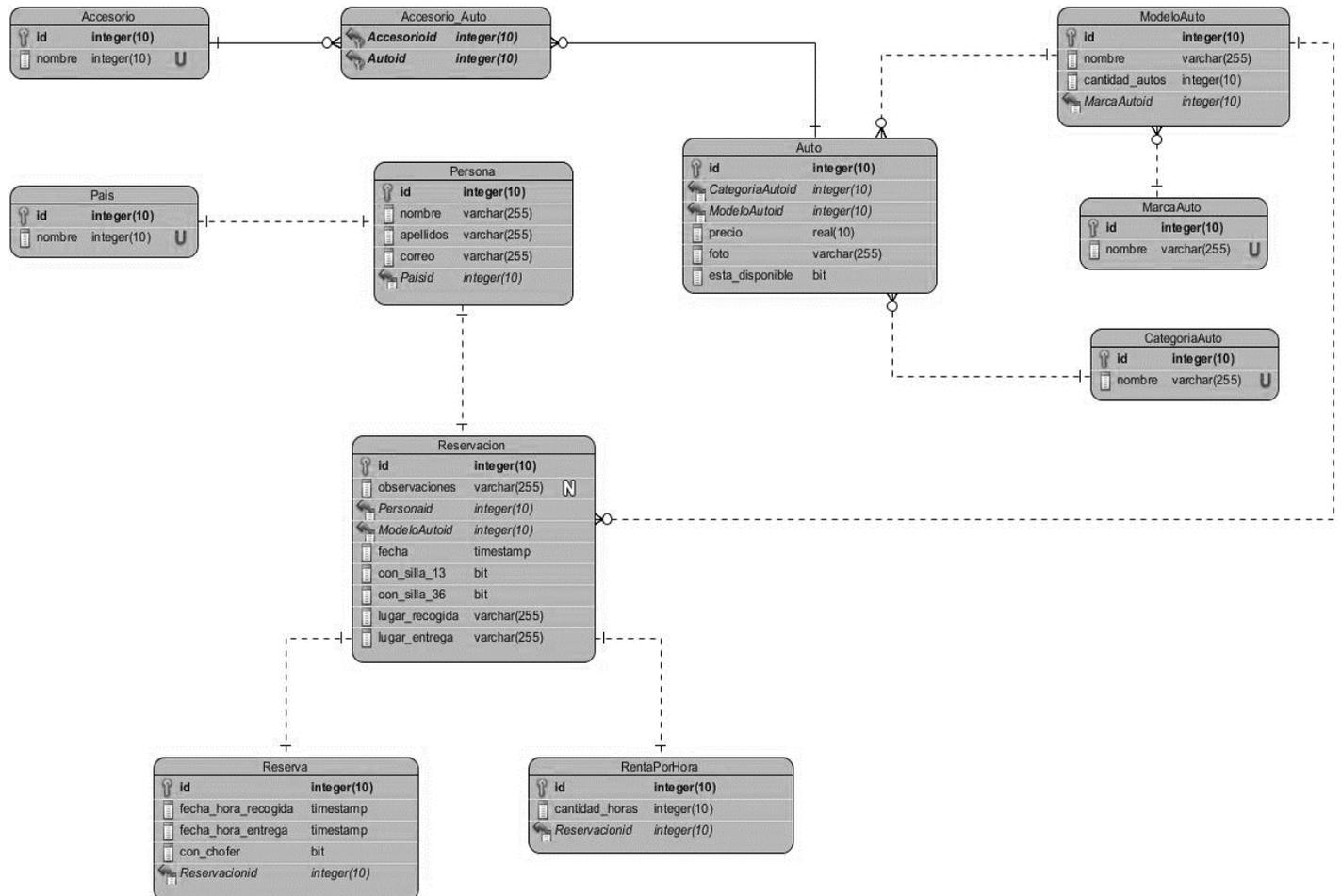


ILUSTRACIÓN 7: MODELO DE DATOS (ELABORACIÓN PROPIA)

## 2.10 Diagrama de despliegue

No es más que en un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue o distribución de los artefactos del *software* en los destinos en que se propone el mismo. Define a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son los archivos ejecutables, bibliotecas, archivos, esquemas de BD y archivos de configuración. Cuando se hace mención de destino de despliegue no es más que un nodo el cual es un dispositivo de *hardware* o un entorno de ejecución de *software* (Sarmiento, 2016).

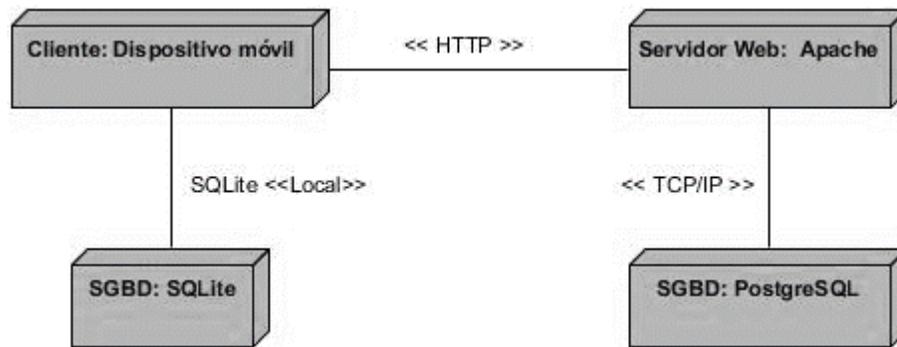


ILUSTRACIÓN 8: DIAGRAMA DE DESPLIEGUE DEL SISTEMA (ELABORACIÓN PROPIA).

## 2.11 Conclusiones parciales del capítulo

Una vez culminado este capítulo se puede llegar a las siguientes conclusiones:

- Con los requerimientos funcionales obtenidos y sus historias de usuario, además de los requisitos no funcionales posibilitaron trazar una propuesta de solución.
- Gracias a la arquitectura y demás artefactos generados se logró diseñar una propuesta de solución, la cual utiliza varios patrones de diseño *GRASP* y *GOF* mediante los cuales en la implementación se conseguirá una adecuada reutilización del código para su uso en posteriores versiones.
- Mediante el modelo de despliegue se pudo modelar una vista de las topologías del *hardware* sobre el que se ejecuta el sistema.

## Capítulo 3: Implementación y pruebas a la aplicación para la Agencia de Alquiler de Vehículos REX

El presente capítulo describe la implementación y las pruebas realizadas a la aplicación, lo cual está incluido en las fases de desarrollo y publicación del proceso de creación de aplicaciones móviles. A través de los estándares de codificación se muestran las pautas que se deben seguir para una correcta implementación de la aplicación. Según la metodología seleccionada serán ejecutadas pruebas a la apk<sup>10</sup> desarrollada para verificar su correcto funcionamiento, con el objetivo de detectar y corregir las posibles no conformidades en los futuros usuarios.

### 3.1 Estándares de codificación

En la propuesta de solución se utilizan los estándares de codificación los cuales son muy importantes para cualquier proyecto de desarrollo pues establecen la estructura del lenguaje, el orden y normas de estilos. Contribuyen a asegurar que el código contenga menos errores, pueda ser mantenido fácilmente y posea una alta calidad. Todos los programadores tienen su forma personalizada de programar y puede que no se parezca a la de ningún otro, pero de ellos si depende que otros programadores entiendan con facilidad, comprendan el código y lo reutilicen. Es de ahí de donde parte la necesidad de la utilización de los estilos de programación, conocidos también como estándares o convenciones de código que definen un grupo de reglas para escribir código fuente en varios lenguajes de programación.

En la propuesta de solución se utilizan los

A continuación, se hará mención de algunos estándares que posee *Java* que fueron utilizados en la implementación de la aplicación:

---

<sup>10</sup> *APK* (Aplicación empaquetada de Android): es un paquete para el sistema operativo Android. Este formato se usa para distribuir e instalar componentes empaquetados para la plataforma Android para teléfonos inteligentes, tabletas y algunas distribuciones enfocadas a su uso en ordenadores personales de escritorio y portátiles.

**TABLA 11: ESTÁNDARES DE CODIFICACIÓN DE JAVA (ELABORACIÓN PROPIA).**

Estándar	Descripción
<b>Nomenclatura</b>	Ofrece sentido funcional al nombre de clases, variables y constantes. Es una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada en este caso español.
<b>Longitud de la línea</b>	Evitar las líneas de más de 80 caracteres, pues estas no son muy bien manejadas por muchas terminales y herramientas.
<b>Indentación</b>	Consiste en dejar espacios en blanco o tabuladores en determinadas líneas del código para facilitar su mejor comprensión. <i>Java</i> utiliza cuatro espacios como unidad de indentación.
<b>Sentencias</b>	Cada línea debe contener como mucho una sentencia. Todo bloque de sentencias debe estar entre llaves.
<b>Comentarios</b>	Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase. Por tanto, se debe evitar referencias al diseño funcional de la misma y el uso de caracteres especiales.
<b>Espacios en blanco</b>	Las líneas en blanco facilitan la lectura separando bloques de código que están lógicamente relacionadas.
<b>Nombres de clases</b>	Los nombres de clases deben poseer letras mayúsculas y minúsculas, con la primera letra de

	cada palabra en mayúsculas. Se debe mantener los nombres de clases simples y lo más descriptivos posibles.
--	--

### 3.2 Validación de la propuesta de solución

Uno de los principales aspectos de la metodología *XP* son las pruebas de *software*, las cuales se realizan en la fase de publicación en el proceso de creación de aplicaciones móviles. Esta metodología propone realizar el mayor número de pruebas que sea posible garantizando así la mayor calidad posible de los sistemas desarrollados, disminuyendo el número de errores no detectados y su tiempo de solución. También disminuye el riesgo de la aparición de efectos colaterales no deseados cuando se le realizan modificaciones y reestructuraciones a la aplicación.

La metodología *XP* separa las pruebas que se realizan al sistema en dos grupos: pruebas de aceptación o pruebas funcionales, que se encargan de la evaluación de si se llega al cumplimiento o no de una funcionalidad específica y las pruebas unitarias, destinadas a verificar el correcto funcionamiento y estructuración del código (Rodríguez Corbea, y otros, 2007).

### 3.3 Pruebas Funcionales

Se realizan al sistema persiguiendo el objetivo determinar cuando el funcionamiento sometido a pruebas del *software* cumple o no con sus especificaciones. Este tipo de pruebas se realizan desde el punto de vista del usuario. Las funciones se prueban entrando juegos de datos y analizando las salidas (Perez, 2007).

#### Casos de Pruebas

Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato.

**Caso de Prueba 1:** Realizar reservación de traslado.

**Descripción general:** Prueba a la funcionalidad realizar reservación de traslado.

**Condiciones de ejecución:** El dispositivo móvil debe estar conectado a la red.

**TABLA 12:** CASO DE PRUEBA REALIZAR RESERVACIÓN DE TRASLADO (ELABORACIÓN PROPIA).

Escenario	Descripción	Reservación	Respuesta del sistema	Flujo central
<b>EC 1.1 Realizar reservación con todos los campos llenados correctamente.</b>	El sistema realiza la reservación de manera correcta.	V	El sistema realiza la reserva y le muestra al usuario una notificación informando que le reserva se ha hecho.	El usuario llena los campos de la reservación y oprime el botón "Enviar".
		Reservación realizada.		
<b>EC 1.2 Realizar reservación con campos llenados incorrectamente.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	I	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario introduce el valor en el campo y oprime el botón "Continuar".
		Reservación no realizada.		
<b>EC 1.3 Realizar reservación con campos vacíos.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	I	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario deja vacío el valor en el campo y oprime el botón "Continuar".
		Reservación no realizada.		

### Descripción de las variables

**TABLA 13:** DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA REALIZAR RESERVACIÓN PARA TRASLADO.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Permite solo letras.

2	Apellidos	Campo de texto	No	Permite solo letras.
3	Correo electrónico	Campo de texto	No	Permite todos los caracteres.
4	País de procedencia	Campo seleccionable	No	Permite seleccionar el país de procedencia de una lista de países.
5	Modelo de autos.	Campo seleccionable	No	Permite seleccionar el modelo de auto de una lista de modelos.
6	Fecha	Campo seleccionable	No	Permite seleccionar la fecha de la reserva.
7	Silla para niños	Campo seleccionable	Si	Permite seleccionar sillas para niños que estén entre las edades de 0-1 año o de 1-3 años.
8	Lugar de recogida	Campo de texto	No	Permite todos los caracteres.
9	Lugar de entrega	Campo de texto	No	Permite todos los caracteres.

**Caso de Prueba 2:** Realizar reservación por horas.

**Descripción general:** Prueba a la funcionalidad realizar reservación por horas.

**Condiciones de ejecución:** El dispositivo móvil debe estar conectado a la red.

**TABLA 14:** CASO DE PRUEBA REALIZAR RESERVACIÓN POR HORAS (ELABORACIÓN PROPIA).

Escenario	Descripción	Reservación	Respuesta del sistema	Flujo central
<b>EC 1.1 Realizar reservación con todos los campos llenados correctamente.</b>	El sistema realiza la reservación de manera correcta.	V	El sistema realiza la reserva y le muestra al usuario una notificación informando que le reserva se ha hecho.	El usuario llena los campos de la reservación y oprime el botón "Enviar".
		Reservación realizada.		
		I		

<b>EC 1.2 Realizar reservación con campos llenados incorrectamente.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	Reservación no realizada.	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario introduce el valor en el campo y oprime el botón "Continuar".
<b>EC 1.3 Realizar reservación con campos vacíos.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	I Reservación no realizada.	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario deja vacío el valor en el campo y oprime el botón "Continuar".

### Descripción de las variables

**TABLA 15:** DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA REALIZAR RESERVACIÓN POR HORAS.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Permite solo letras.
2	Apellidos	Campo de texto	No	Permite solo letras.
3	Correo electrónico	Campo de texto	No	Permite todos los caracteres.
4	País de procedencia	Campo seleccionable	No	Permite seleccionar el país de procedencia de una lista de países.
5	Modelo de autos.	Campo seleccionable	No	Permite seleccionar el modelo de auto de una lista de modelos.
6	Fecha	Campo seleccionable	No	Permite seleccionar la fecha de la reserva.

7	Silla para niños	Campo seleccionable	Si	Permite seleccionar sillas para niños que estén entre las edades de 0-1 año o de 1-3 años.
8	Cantidad de horas	Campo seleccionable	No	Permite solo números.
9	Lugar de recogida	Campo de texto	No	Permite todos los caracteres.
10	Lugar de entrega	Campo de texto	No	Permite todos los caracteres.

**Caso de Prueba 3:** Realizar reservación con o sin chofer.

**Descripción general:** Prueba a la funcionalidad realizar reservación con o sin chofer.

**Condiciones de ejecución:** El dispositivo móvil debe estar conectado a la red.

**TABLA 16:** CASO DE PRUEBA REALIZAR RESERVACIÓN CON O SIN CHOFER (ELABORACIÓN PROPIA).

Escenario	Descripción	Reservación	Respuesta del sistema	Flujo central
<b>EC 1.1 Realizar reservación con todos los campos llenados correctamente.</b>	El sistema realiza la reservación de manera correcta.	V	El sistema realiza la reserva y le muestra al usuario una notificación informando que le reserva se ha hecho.	El usuario llena los campos de la reservación y oprime el botón "Enviar".
		Reservación realizada.		
<b>EC 1.2 Realizar reservación con campos llenados incorrectamente.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	I	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario introduce el valor en el campo y oprime el botón "Continuar".
		Reservación no realizada.		

<b>EC 1.3 Realizar reservación con campos vacíos.</b>	El sistema no deja continuar llenando los demás campos de la reserva.	I	El sistema no deja continuar llenando los demás campos de la reserva hasta que se llene correctamente el campo actual.	El usuario deja vacío el valor en el campo y oprime el botón "Continuar".
		Reservación no realizada.		

### Descripción de las variables

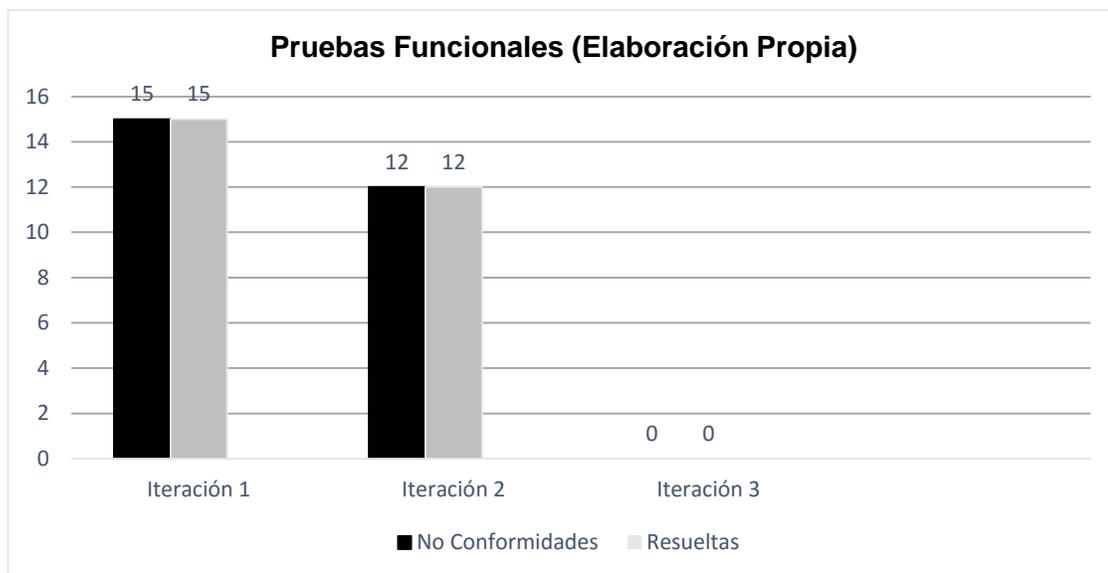
**TABLA 17:** DESCRIPCIÓN DE LAS VARIABLES DEL CASO DE PRUEBA REALIZAR RESERVACIÓN CON O SIN CHOFER.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Permite solo letras.
2	Apellidos	Campo de texto	No	Permite solo letras.
3	Correo electrónico	Campo de texto	No	Permite todos los caracteres.
4	País de procedencia	Campo seleccionable	No	Permite seleccionar el país de procedencia de una lista de países.
5	Modelo de auto.	Campo seleccionable	No	Permite seleccionar el modelo de auto de una lista de modelos.
6	Fecha	Campo seleccionable	No	Permite seleccionar la fecha de la reserva.
7	Fecha de recogida	Campo seleccionable	No	Permite seleccionar la fecha de comienzo de la reserva.
8	Fecha de entrega	Campo seleccionable	No	Permite seleccionar la fecha donde termina la reserva.
9	Chofer	Campo seleccionable	Si	Permite seleccionar la presencia de un chofer profesional en la reserva.

10	Silla para niños	Campo seleccionable	Si	Permite seleccionar sillas para niños que estén entre las edades de 0-1 año o de 1-3 años.
11	Lugar de recogida	Campo de texto	No	Permite todos los caracteres.
12	Lugar de entrega	Campo de texto	No	Permite todos los caracteres.

### Resultados de las pruebas funcionales

Fueron probadas todas las funcionalidades del sistema que se corresponden con los casos de prueba de la aplicación. Para lograr la validación de los requisitos funcionales se efectuaron dos iteraciones donde se encontraron en general 27 no conformidades siendo en su mayor parte errores de faltas de ortografía, funcionalidades y textos escritos en inglés. En general fueron identificadas 15 no conformidades en la primera iteración realizada y 12 en la segunda iteración, al realizar una tercera iteración no fue encontrada ninguna no conformidad, una vez terminadas todas las iteraciones fueron solucionadas las no conformidades en su totalidad. En la siguiente gráfica se observan los resultados de cada una de las iteraciones de pruebas que se efectuaron a la aplicación móvil para la Agencia de Alquiler de Vehículos REX.



### 3.4 Pruebas Unitarias

Las pruebas unitarias son la comprobación de una unidad específica de código de la aplicación. Se aplica a componentes representados en el modelo de implementación para comprobar que los flujos de control y de datos funcionen correctamente. Las pruebas de unidad están orientadas a caja blanca y le permiten al programador conocer si cierta funcionalidad se puede agregar al sistema sin afectar el funcionamiento del mismo.

*XP* plantea ejecutar las HU en pequeñas iteraciones para incrementar la calidad del sistema y reducir la cantidad de errores no detectados aminorando el tiempo entre la aparición de un error y su descubrimiento. Por lo que se decidió dividir el proceso de desarrollo de la aplicación móvil para la Agencia de alquiler de vehículos en tres iteraciones, efectuándose, al terminar cada iteración, pruebas al código de las HU realizadas y comprobar así su correcto funcionamiento. Para la realización de estas pruebas se utilizó el framework *JUnit 4.0* que está integrado con el *SDK* de *Android*. Este *framework* reconoce realizar la elaboración de clases *Java* de manera controlada, para poder valorar si el funcionamiento de cada uno de los métodos de la clase se comporta de forma adecuada. Por lo que el código en su totalidad debe haber sido sometido a pruebas unitarias y debe pasarlas antes de ser entregada la aplicación (Rodríguez Corbea, y otros, 2007).

#### Resultados de las pruebas unitarias

Fueron ejecutadas de forma sistemática una vez que se terminaba de implementar una de las iteraciones. Estas pruebas se ejecutaron haciendo uso de la herramienta de *Android JUnit*. Esta herramienta posibilita la realización de pruebas a componentes específicos a través de casos de prueba. Estas clases proporcionan métodos auxiliares para la creación de objetos de imitación de métodos que ayudan a controlar el ciclo de vida de la aplicación. A continuación, se muestra los resultados de las pruebas realizadas a las funcionalidades en cada una de las iteraciones:

#### Primera iteración de las Pruebas Unitarias

Al terminar la realización de las HU de la primera iteración fueron hechas pruebas al código de la aplicación y se detectaron varias no conformidades:

- Los campos del formulario de realizar reservación para traslado no ofrecían la posibilidad de ser llenados.

- No se guardaban los datos de las reservas en la base de datos.
- Los campos de los formularios no se encontraban correctamente validados.
- No se visualizaba la información de las oficinas de renta de la forma más adecuada.

Las no conformidades identificadas en esta primera iteración fueron resueltas en iteraciones posteriores eliminando las consecuencias que este tipo de errores traen para el funcionamiento de la aplicación en general.

### **Segunda Iteración de las Pruebas Unitarias**

Fueron hechas pruebas al código de las HU de la segunda iteración y se hallaron las siguientes no conformidades:

- No se mostraba la lista de los autos disponibles para rentar.
- No se enviaban los datos de las reservaciones realizadas al servidor.

Luego de solucionar las no conformidades encontradas en esta iteración se ejecutaron varias iteraciones de pruebas automáticas y se obtuvieron resultados satisfactorios.

### **Tercera Iteración de las Pruebas Unitarias**

- No se podían modificar los datos de las reservas hechas con anterioridad.
- La aplicación era incompatible para versiones inferiores de *Android* a la 4.2.2.

Las no conformidades encontradas durante las iteraciones fueron resueltas en su totalidad y en las siguientes etapas de la tercera iteración la aplicación logró un correcto funcionamiento tal y como era esperado.

## **3.5 Pruebas de Integración**

Las pruebas de integración están diseñadas para probar la interacción entre los distintos componentes de un sistema, también pueden certificar el funcionamiento o integración entre dos o más sistemas. Su objetivo es tomar los componentes probados en unidad y construir una estructura acorde a la que propone el diseño (US, 2013).

En el caso de la aplicación desarrollada se integró con un sistema desarrollado en lenguaje *Python* y mantuvo un adecuado funcionamiento realizando peticiones y mostrando información proveniente del sistema. Para probar la correcta integración con el sistema se realizaron 25 consultas de disponibilidad de

autos listos para reservar, 15 reservaciones de cada uno de los tipos, se modificaron estas 15 reservas y se eliminaron 10 reservas realizadas con anterioridad. En cada uno de los casos el sistema recibe correctamente los datos de las reservaciones y los añade a su base de datos y la aplicación puede actualizar los datos de los autos disponibles para rentar y modificar las reservaciones hechas. Por lo que se puede constatar que la aplicación está lista para integrarse al sistema y acceder a los servicios ofrecidos por REX desde un dispositivo móvil con sistema operativo *Android*.

**TABLA 18:** PRUEBA DE INTEGRACIÓN PARA LA FUNCIONALIDAD REALIZAR RESERVA (ELABORACIÓN PROPIA).

Escenario	Descripción	Manual de Configuración	Valor Esperado	Respuesta de la Aplicación
<b>Realizar Reservación</b>	El usuario realiza una reservación y se envía la información al sistema de REX.	Deben estar llenos todos los campos de selección obligatoria para poder enviar la reservación.	Se muestra una notificación donde se informa al usuario que la reserva se ha realizado con éxito.	El sistema verifica que sean válidos todos los datos de la reserva, de ser así esta es almacenada en la base de datos.

### 3.6 Pruebas de Rendimiento

Son imprescindibles para que el producto final del desarrollo de la aplicación cause el mayor nivel de satisfacción en los usuarios. Es un tipo de prueba a la cual se somete el *software* o aplicación desarrollada que ayuda a garantizar la compatibilidad de esta con sistemas operativos, usuarios, plataformas de *hardware* y navegadores *web*. Ayudan a determinar el desenvolvimiento de un sistema en determinado ambiente, el cual puede incluir *hardware*, red, sistema operativo u otro *software*.

Para la realización de estas pruebas se utilizaron los siguientes dispositivos:

Samsung Galaxy S4 i337 versión de *Android*: 4.4.2 CPU: 4 cores @1.9GHz RAM: 2GB.

Alcatel One Touch 4015X versión del *Android*: 4.2.2 CPU: 1 core @1.0GHz RAM: 512MB.

Para establecer la comparación se tuvo en cuenta la utilización de los siguientes recursos: batería, memoria RAM, tiempo de ejecución y microprocesador.

### Resultados de las Pruebas de Rendimiento

De acuerdo a las características de la aplicación móvil para la Agencia de Alquiler de Vehículos REX fue instalada y probado su funcionamiento en dispositivos móviles de características funcionales distintas y mantuvo un correcto funcionamiento en todos los dispositivos probados. Así mismo se ajustó correctamente y mantuvo una adecuada visualización a distintas resoluciones de dispositivos que poseen una versión superior de *Android* a la 4.2.2. A continuación se muestran los resultados obtenidos:

**TABLA 19: RESULTADOS DE LAS PRUEBAS DE RENDIMIENTO (ELABORACIÓN PROPIA).**

	<i>Samsung Galaxy S4 i337</i>	<i>Alcatel One Touch 4015X</i>
<b>Microprocesador (%)</b>	4,59	12,66
<b>Memoria RAM (MB)</b>	27,46	28,23
<b>Tiempo de ejecución (segundos)</b>	2,3	3,8
<b>Batería (%)</b>	1	3

### 3.7 Pruebas de Compatibilidad

Las pruebas de compatibilidad son las encargadas de comprobar el funcionamiento del sistema en varios escenarios de ejecución. Se ejecutan en aplicaciones y *software*, para comprobar que funcionan correctamente en versiones distintas de sistemas operativos, navegadores o dispositivos con distintas características de *hardware*. Se realizan principalmente porque el *software* o aplicación puede presentar errores en dependencia del entorno donde se desarrolle funcionales o errores estéticos en dependencia del entorno donde se desarrolle (Pressman, 2011).

Para efectuar estas pruebas fue usado el emulador de *Android Studio*, *Android Virtual Device* para emular el funcionamiento de la aplicación móvil en diferentes tipos de dispositivos móviles, con diferentes versiones

de sistema operativo y resolución de pantalla. Se comprobará la compatibilidad de la aplicación para cada nivel de *API*<sup>11</sup> superior al valor especificado en *minSdkVersion*<sup>12</sup>.

Con el objetivo de comprobar el comportamiento de la aplicación en versiones diferentes de *Android* y dispositivos con distintas características de *hardware* y resolución se realizaron cuatro iteraciones y se obtuvieron 26 no conformidades relacionadas al comportamiento visual de la aplicación, las cuales fueron resueltas en su totalidad. Para ver como se muestra la vista Renta Lineal en distintos dispositivos de sistema operativo *Android* vea Anexo 4.

En la tabla y en el gráfico que se muestran a continuación, se exponen los resultados obtenidos en cada una de las iteraciones, así como la corrección de cada una de las no conformidades.

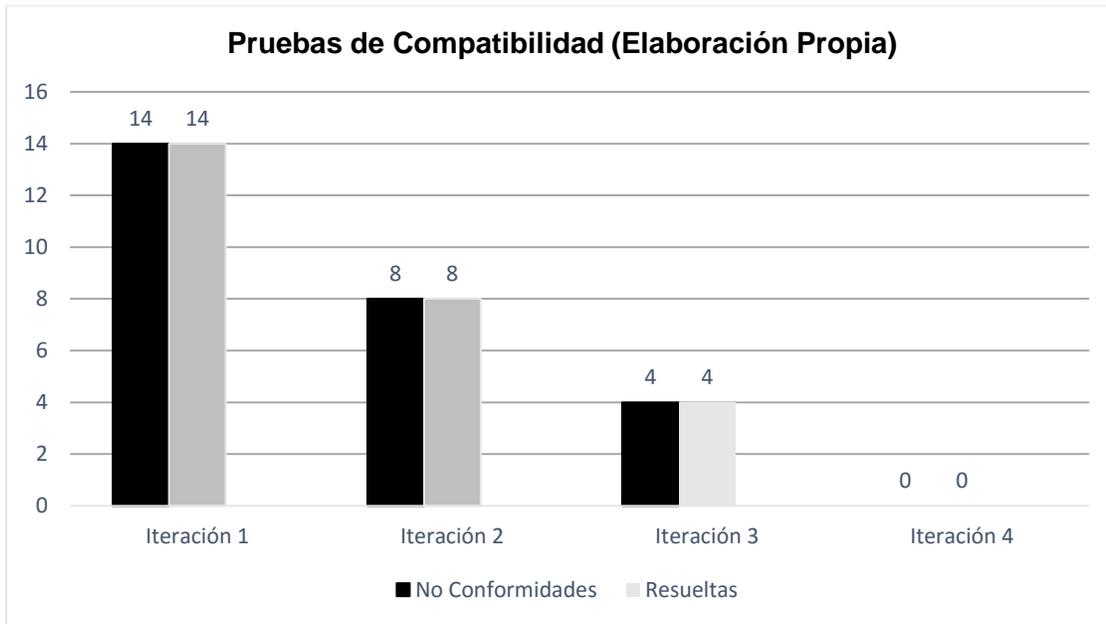
**TABLA 20:** RESULTADOS DE LAS PRUEBAS DE COMPATIBILIDAD.

No conformidades	Iteración 1	Iteración 2	Iteración 3	Iteración 4
<b>Detectadas</b>	14	8	4	0
<b>Resueltas</b>	14	8	4	0

---

<sup>11</sup>*API*: La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

<sup>12</sup> *minSdkVersion*: Valor entero que designa el nivel de *API* mínimo que se requiere para ejecutar la aplicación.



Después de realizar las tres iteraciones y haber corregido cada una de las no conformidades se realizó una última iteración en la cual no se obtuvo ninguna no conformidad al no detectar ningún problema de compatibilidad de la aplicación.

### 3.7 Conclusiones del Capítulo

- La utilización de estándares de codificación de código permite la utilización de una estructura homogénea que facilita la comunicación y asegura la calidad, menos errores y fácil mantenimiento.
- La ejecución de las pruebas unitarias al código de la aplicación, así como de las pruebas funcionales, permitieron presentar los resultados arrojados en cada iteración logrando finalmente una aplicación que responde al conjunto de funcionalidades identificadas.

## Conclusiones

Una vez culminada el proceso de desarrollo de la aplicación móvil para la Agencia de Alquiler de Vehículos REX se puede afirmar que se dio cumplimiento al objetivo general planteado, arribando así a las siguientes conclusiones:

- El estudio de los principales aspectos relacionados con las aplicaciones móviles para la renta de vehículos permitió establecer las bases para el posterior desarrollo de la aplicación móvil para la Agencia de Alquiler de Vehículos REX, permitiendo escoger una arquitectura que proporcionará una adecuada estructura y facilitando la elección de las herramientas necesarias para dicho proceso.
- A partir del análisis y diseño de la aplicación móvil para la Agencia de Alquiler de Vehículos REX se obtuvieron los artefactos necesarios para guiar el proceso de desarrollo y lograr un mejor entendimiento de la misma.
- Se implementó una aplicación móvil para sistema operativo *Android* que permite a los usuarios consultar la disponibilidad, hacer reservas y mostrar las informaciones más relevantes de REX para contribuir a la divulgación y promoción de esta información a través de tecnologías móviles.
- Las pruebas realizadas a la aplicación móvil para la Agencia de Alquiler de Vehículos REX, evidenciaron que la aplicación es una solución idónea y posee un correcto funcionamiento, que se integra sin ningún tipo de problema al servidor que utiliza, con un rendimiento estable y segura.

## Recomendaciones

- Desarrollar la aplicación móvil para la Agencia de reserva de Vehículos REX para otras plataformas móviles.
- Una vez que la infraestructura tecnológica del país lo permita, implementar otras funcionalidades en la aplicación móvil que utilicen la ubicación GPS para que la Agencia de Alquiler de Vehículos REX brinde un servicio más completo a sus clientes.

## *Referencias Bibliográficas*

*El Modelo Relacional*. **Zorrilla Pantaleón, M.E y Duque Medina, R. 2011**. Cantabria : Universidad de Cantabria, 2011, Bases de Datos.

**Caballero González, C y Montoya Cordero, R. 2016**. Almacenamiento de la información e introducción a SGBD. [aut. libro] Carlos Caballero González y Raúl Montoya Cordero. *Almacenamiento de la información e introducción a SGBD*. Madrid. España. : Ediciones Nobel S.A., 2016.

**Canós, J H, Leterier, P y Maria del Carmen. 2003**. *Métodologías Ágiles en el Desarrollo de Software*. s.l. : No Quality Inside, 2003.

**Car2go. 2017**. Google Play. [En línea] Car2go, 30 de 1 de 2017. [Citado el: 21 de 2 de 2017.] <https://play.google.com/store/apps/details?id=com.car2go&hl=es>.

**Carballo, Y A. 2014**. *Sistemas de Objetos de Aprendizaje Interactivos y Experimentales para la Matemática Discreta*. La Habana, Cuba : Universidad de las Ciencias Informaticas, 2014. Trabajo de Diploma en opción al título de Ingeniero en Ciencias Informáticas.

**Cendejas Valdéz, J.L. 2014**. Ingeniería de Software. *Modelos y metodologías para el desarrollo de software*. [En línea] 2014. [Citado el: 25 de 11 de 2016.] <http://www.eumed.net/tesisdoctorales/2014/jlcv/software.html>.

**Concepto. 2015**. Concepto.de. *Concepto de Aplicación*. [En línea] 2015. [Citado el: 21 de 2 de 2017.] <http://concepto.de/aplicacion/>.

**Consum, Agencia Catalana del. 2017**. Alquiler de vehículos. *Generalitat de Catalunya*. [En línea] Agencia Catalana del Consum, 5 de 4 de 2017. [Citado el: 12 de 5 de 2017.] [http://consum.gencat.cat/temes\\_de\\_consum/lloguer\\_de\\_vehicles/index\\_es.html](http://consum.gencat.cat/temes_de_consum/lloguer_de_vehicles/index_es.html).

**Cuello, Javier y Vittone, Jose. 2013**. Diseñando Apps Para Moviles. *Diseñando Apps Para Moviles*. 2013.  
*Definición de una metodología ágil*. **Merchán, L, Urrea, A y Rebollar, R. 2011**. No 1, Cali, Colombia : Revista Científica Guillermo de Ockham, marzo de 2011, Vol. 6. ISSN: 1794-1924.

**Drivy. 2016.** Drivy. *Las aplicaciones móviles de Drivy*. [En línea] Drivy, 2016. [Citado el: 21 de 2 de 2017.] <https://www.drivy.es/mobile>.

**Etecsa. 2016.** Estadísticas de la Internet en Cuba. *Cuba 2.0*. [En línea] 23 de 11 de 2016. [Citado el: 6 de 4 de 2017.] <http://cubaenred.cubava.cu/2016/11/23/estadisticas-de-la-internet-en-cuba/?repeat=w3tc>.

**Fernández, Y y Díaz, Y. 2012.** *Patrón Modelo-Vista-Controlador*. La Habana : CUJAE, 2012. págs. p. 47-57. ISSN 1729.

**Figueroa, R G, Solís, C J y Cabrera, A A. 2015.** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2015.

**Geospatial. 2016.** Tipos de aplicaciones móviles. *Geospatial*. [En línea] 2016. [Citado el: 22 de 2 de 2017.] <http://geospatialtraininges.com/recursos-gratuitos/tipos-de-aplicaciones-moviles/>.

**Grosso, A. 2011.** [En línea] 2011. [Citado el: 3 de 3 de 3027.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.

**Guiu, D. 2012.** La importancia de las apps móviles. *SocialEtic*. [En línea] SocialEtic, 2012. [Citado el: 3 de 4 de 2017.] <http://www.socialetic.com/la-importancia-de-las-apps-moviles.html>.

**Ibañez Corrales, F J. 2011.** *Metodología Investigativa y Tradicional*. 2011. pág. 8. ISSN: 1988-6047.

**Informáticos, D. d. 2005.** Patrones de asignación de responsabilidades (GRASP). s.l. : Universidad de Sevilla., 2005.

**Juventud Rebelde. 2016.** Las bases estratégicas de la informatización cubana . *República de Cuba. Ministerio de las Comunicaciones*. [En línea] Ministerio de las Comunicaciones, 2016. [Citado el: 3 de 4 de 2017.] <http://www.mincom.gob.cu/?q=node/803>.

**Lancetalent. 2014.** Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes. *Lancetalent*. [En línea] 20 de 2 de 2014. [Citado el: 9 de 12 de 2016.] <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>.

**Larman, C. 2003.** *UML y Patrones*. s.l. : 2da Edición, 2003.

**Leiva Gordillo, A. 2014.** MVP en Android: cómo organizar la capa de presentación. *Lime Creative Labs*. [En línea] 2 de 4 de 2014. [Citado el: 5 de 4 de 2017.] <http://www.limecreativelabs.com/mvp-android/>.

**Letelier, P. 2005.** *Metodologías Ágiles y XP*. Valencia : Departamento de Sistemas Informáticos y Computación, 2005.

**López, P y Ruiz, F. 2011.** *Lenguaje Unificado de Modelado - UML*. 2011.

**Mentoring, Global. 2012.** Global Mentoring. *¿Qué es un IDE?* [En línea] 26 de 1 de 2012. [Citado el: 22 de 2 de 2017.] <http://globalmentoring.com.mx/cursos-java/java-fundamentos/que-es-un-ide/>.

**Merchán, Luis, Urrea, Alba y Rebollar, Rubén. 2011.** *Definición de una metodología ágil*. Cali, Colombia : Revista Científica Guillermo de Ockham, 2011. ISSN: 1794-1924.

**Olivera, A.G y & Novelo, C. 2010.** *Reporte de instalación de apache*. Escárcega. 2010.

**PCC. 2016.** Lineamientos de la Política Económica y Social del Partido y la Revolución para el período 2016-2021. *Actualización de los Lineamientos de la Política Económica y Social del Partido y la Revolución para el período 2016-2021 aprobados en el 7mo. Congreso del Partido en abril de 2016 y por la Asamblea Nacional del Poder Popular en julio de 2016*. 2016.

**Peralta, C y Durán, D. 2014.** *Módulos de edición de plantillas y recepción de órdenes de impresión para el Sistema de Personalización de Documentos de Identidad basado en tecnologías libres*. Universidad de las Ciencias Informáticas. La Habana. Cuba : s.n., 2014. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Perez, B. 2007.** *Gestión de las pruebas funcionales*. Montevideo : Universidad de la República, Montevideo, Uruguay., 2007.

**Peró Carbonell, D, Hernández Carballé, A y Cabrera González, L. 2013.** *Extensión de la herramienta CASE "Visual Paradigm for UML" para la estimación de proyectos*. La Habana. : Universidad de Ciencias Informáticas., 2013.

**Ponsoda Montiel, D. 2008.** *Introducción a SQLite*. s.l. : s.l. : Creative Commons, 2008.

**Pressman, R. 2011.** *Ingeniería de Software, un enfoque práctico*. New York. EEUU. : McGraw-Hill., 2011. ISBN 9780071267823.

—. 2011. *Ingeniería de Software: Un Enfoque Práctico*. Quinta Edición. New York, EEUU. : McGraw-Hill., 2011.

**Pressman, R. S. 2002.** *Ingeniería de Software, un enfoque práctico*. Quinta Edición. s.l. : McGraw-Hill Companies, 2002. ISBN: 8448132149.

**Rodríguez Corbea, M y Ordoñez Pérez, M. 2007.** *La Metodología XP Aplicable al Desarrollo del Software*. La Habana, Cuba. : Universidad de las Ciencias Informáticas., 2007.

**Rosso, R. 2014.** Android Studio. *UptoDown*. [En línea] Android Studio, 12 de 2014. [Citado el: 31 de 3 de 2017.] <https://android-studio.uptodown.com/windows>.

**Sarmiento, J. 2016.** *UML: Diagrama de despliegue*. s.l. : Visión general de los diagrama de despliegue, 2016.

**SIXT. 2016.** SIXT rent a car. [En línea] SIXT, 2016. [Citado el: 21 de 2 de 2017.] <https://www.sixt.com/es/servicio-de-alquiler/reserva-del-telefonía-movil/android/>.

—. 2016. SIXT rent a car. [En línea] SIXT, 8 de 12 de 2016. [Citado el: 21 de 2 de 2017.] <https://www.sixt.com/es/servicio-de-alquiler/reserva-del-telefonía-movil/blackberry/>.

—. 2016. SIXT rent a car. [En línea] 8 de 12 de 2016. [Citado el: 31 de 3 de 2017.] <https://www.sixt.es/servicio-de-alquiler/reserve-por-movil/android/>.

—. 2016. SIXT rent a car. [En línea] SIXT, 2016. [Citado el: 21 de 2 de 2017.] <https://www.sixt.com/es/servicio-de-alquiler/reserva-del-telefonía-movil/iphone-ipad/>.

**SocialCar. 2017.** Google Play. *SocialCar - Alquiler de coches*. [En línea] SocialCar, 3 de 2 de 2017. [Citado el: 21 de 2 de 2017.] <https://play.google.com/store/apps/details?id=com.socialcar&hl=es>.

**SQLite. 2015.** Features Of SQLite. *SQLite*. [En línea] SQLite, 2015. [Citado el: 3 de 3 de 2017.] <https://www.sqlite.org/features.html>.

**Stats, Internet Live. 2017.** *Internet Live Stats*. [En línea] 2017. [Citado el: 6 de 4 de 2017.] <http://www.internetlivestats.com/>.

**Turner, R. 2005.** *The Foundations of Specification*. s.l. : s.l. : Journal of Logic and Computation, 2005. Vol. Vol. 15.

**UNESCO. 2017.** Comunicación e Información. *Organización de las Naciones Unidas para la Educación la Ciencia y la Cultura*. [En línea] UNESCO, 2017. [Citado el: 3 de 4 de 2017.] <http://www.unesco.org/new/es/communication-and-information/intergovernmental-programmes/information-for-all-programme-ifap/priorities/information-accessibility/#topPage>.

**US. 2013.** *Pruebas de integración*. Sevilla. : Universidad de Sevilla., 2013.

**Villegas, N. 2015.** Importancia de las arquitecturas de software en las organizaciones. *Web Unicesi. Plataforma de medios de comunicación de la Universidad Icesi*. [En línea] Universidad Icesi., 11 de 6 de 2015. [Citado el: 13 de 5 de 2017.] <http://www.icesi.edu.co/unicesi/2015/06/11/importancia-de-la-arquitectura-de-software-en-las-organizaciones/>.