

Universidad de las Ciencias Informáticas
Facultad 3



Título: Aplicación móvil informativa de entidades que brindan servicios jurídicos en Cuba para dispositivos móviles con sistema operativo Android.

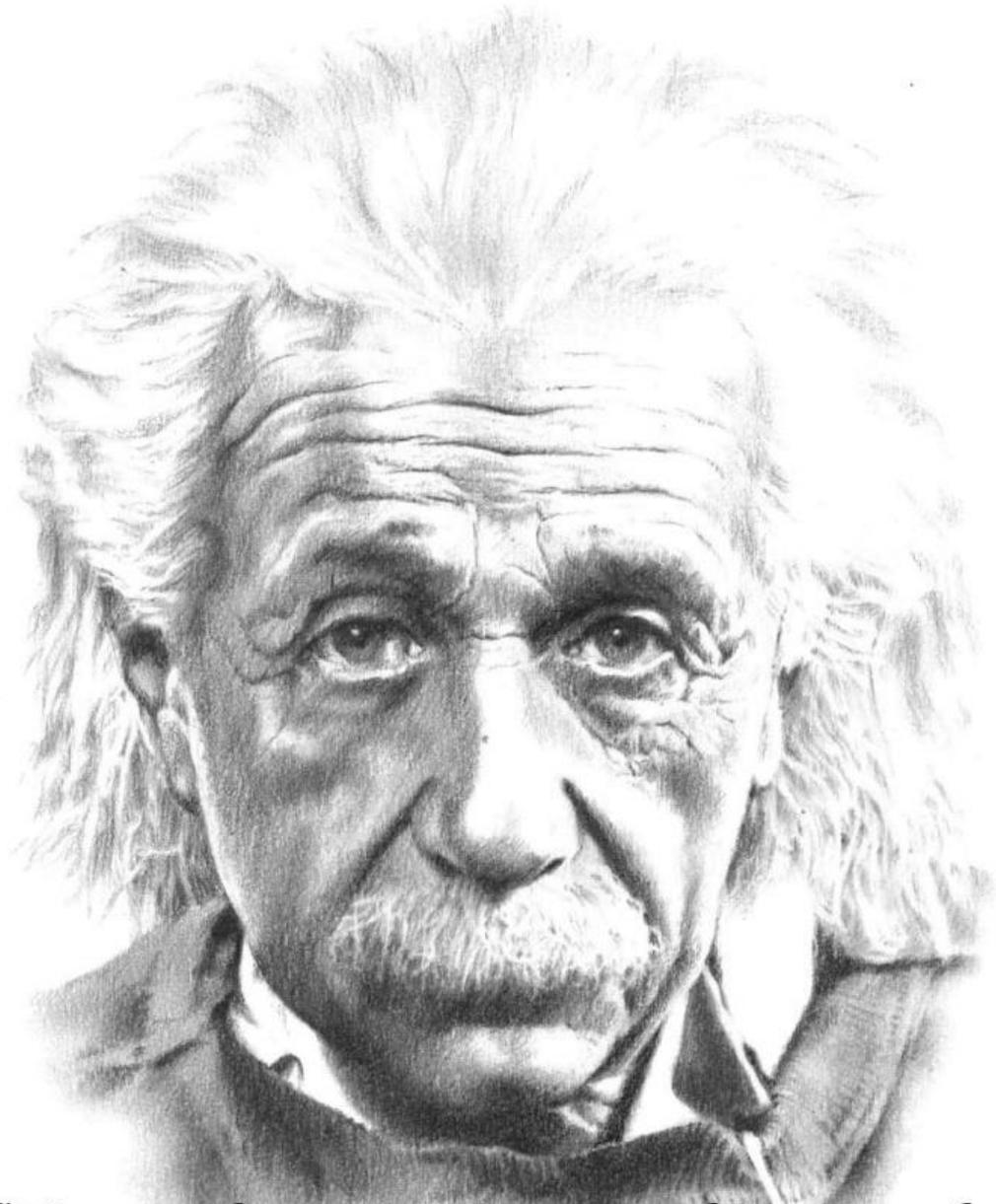
Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Adrian Pérez Zaldívar.

Tutor(es): MsC. Yarina Amoroso Fernández.

Co-tutor: Ing. Felix A. Marrero Pentón.

PENSAMIENTO



“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad”.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaración de autoría

Declaro ser 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 _____ del año _____.

Autor:

Adrian Pérez Zaldivar

Tutora:

MsC. Yarina Amoroso Fernández

Co-Tutor:

Ing. Felix A. Marrero Pentón.

DEDICATORIA

A mi madre principalmente, por el sacrificio de tantos años. Por haber puesto mi bienestar personal ante todas las cosas y por la educación ejemplar que me dio y por convertirme en el hombre que hoy soy.

AGRADECIMIENTOS

A mi madre, sé que, desde el cielo, sabes que te extraño mucho y que este sueño se ha hecho realidad gracias a ti.

A mi padre, por ser un ejemplo para mí. Por darme siempre su apoyo y confianza.

A mi hermano, porque, aunque la distancia se encargue de separarnos, has sido siempre mi amigo, mi guía y protector, te quiero mucho.

A Juan Ramón, por iniciarme en el mundo de la Informática, sin ti esto no hubiera sido posible.

A mi novia, por los momentos buenos, por los malos, por derramar tu amor y cariño sobre mí y haber forjado en mí una mejor persona. Gracias titi, te amo.

A mi madrastra, por estar siempre atenta a mis necesidades, por los consejos, por los regaños, por todo, gracias.

A mis familiares y amigos, por haber estado siempre presentes, en los buenos y malos momentos.

A la nueva familia que conocí gracias a mi novia, gracias por brindarme su apoyo incondicional en todo momento.

A mis amigos de la UCI, especialmente a los Vikingos: Chandy, Abelito, Samper, Ariel y Dayan, gracias por todo, ustedes han hecho que la estancia sea un poco más cómoda, gracias por largas noches, por las discusiones, por los juegos, en fin, por todo.

RESUMEN

Resumen

A nivel global, la demanda de servicios jurídicos procede tanto de empresas y organizaciones, como de personas naturales y jurídicas. Los servicios jurídicos son brindados en Cuba por diferentes entidades de aquellas que integran el llamado sector jurídico, a saber: Organización Nacional de Bufetes Colectivos, Consultorías Jurídicas, Oficinas Registrales y otras. Actualmente, para la consulta de información básica de estas entidades, es necesario acceder a sus respectivos sitios web o consultando las Páginas Amarillas de los Directorios Telefónicos, limitando el área de acceso a la información. El objetivo de la investigación es desarrollar una aplicación que garantice la centralización y portabilidad de la información referente a estas Entidades Jurídicas desde dispositivos móviles con sistema operativo Android. Para ello se realizó un análisis de las Entidades Jurídicas cubanas, identificando las características fundamentales que debía tener la herramienta. También se estudiaron aplicaciones Android que brindaban información asociada a las Entidades Jurídicas de diferentes países. Se identificaron y describieron las Historias de Usuario, el Plan de Iteraciones y el Plan de Entrega, proporcionando el punto de partida para las actividades de implementación. Posteriormente se implementó la aplicación posibilitando la portabilidad y centralización de la información referente a estas Entidades. Se realizaron pruebas para validar el correcto funcionamiento de la herramienta corrigiendo los errores detectados. Como resultado se obtuvo la aplicación Callejero Jurídico que permite el acceso a la información de las entidades cubanas que brindan servicios jurídicos y la localización de estas en mapas *offline* desde un dispositivo móvil con SO Android.

Palabras clave: acceso, Android, centralización, Entidades Jurídicas, información, Servicios Jurídicos

ABSTRACT

Abstract

At a global level, the demand for legal services comes from companies and organizations, as well as natural and legal persons. Legal services are provided in Cuba by different entities of those that make up the so-called legal sector, namely: National Organization of Collective Law Firms, Legal Consulting, Registry Offices and others. Currently, for the consultation of basic information of these entities, it is necessary to access their respective websites or by consulting the Yellow Pages of the Telephone Directories, limiting the area of access to information. The objective of the research is to develop an application that guarantees the centralization and portability of the information referring to these Legal Entities from mobile devices with Android operating system. For this, an analysis of the Cuban Legal Entities was carried out, identifying the fundamental characteristics that the tool should have. Android applications that provided information associated with the Legal Entities of different countries were also studied. The User Stories, the Iterations Plan and the Delivery Plan were identified and described, providing the starting point for the implementation activities. Subsequently, the application was implemented, making possible the portability and centralization of information regarding these Entities. Tests were performed to validate the correct functioning of the tool correcting the detected errors. As a result, the Legal Street Map application was obtained that allows access to the information of Cuban entities that provide legal services and the location of these in offline maps from a mobile device with Android SO.

Keywords: Android, centralization, Legal Entities, information, portability, Legal Services

ÍNDICE DE CONTENIDO

Índice de contenido	
Introducción.....	1
Capítulo 1: Fundamentación teórica.....	6
1.1 Introducción.....	6
1.2 Conceptos asociados a la investigación.....	6
1.2.1 Tipos de servicios jurídicos.....	7
1.3 Estudio del estado del arte.....	8
1.3.1 Soluciones internacionales.....	8
1.3.2 Soluciones nacionales.....	11
1.3.3 Análisis de las soluciones encontradas.....	12
1.4 Metodología de Desarrollo de software.....	12
1.4.1 SCRUM.....	13
1.4.2 AUP.....	13
1.4.3 Metodología XP.....	14
1.4.4 Metodología de desarrollo de software seleccionada.....	15
1.5 Sistema Operativo móvil.....	16
1.6 Mapas <i>offline</i>	18
1.7 Lenguajes, herramientas y tecnologías.....	19
1.7.1 Herramienta de modelado.....	19
1.7.2 Entorno de Desarrollo Integrado.....	20
1.7.3 Lenguajes de programación.....	21
1.7.4 SQLite.....	21
1.7.5 Android SDK.....	22
1.7.6 Gradle 3.3.....	22
1.7.7 Maven.....	22
1.7.8 Java SE JDK 1.8.....	23
1.7.9 JUnit 4.12.....	23

ÍNDICE DE CONTENIDO

1.8 Conclusiones del capítulo	23
Capítulo 2: Planificación y diseño del sistema	24
2.1 Introducción	24
2.2 Modelo de la propuesta de solución	24
2.3 Planeación	25
2.3.1 Historias de usuario.....	25
2.3.2 Características no funcionales del sistema.....	27
2.3.3 Tiempo de ejecución del proyecto	28
2.3.4 Iteraciones.....	29
2.3.5 Plan de entrega	30
2.4 Diseño de la propuesta de solución.....	31
2.4.1 Arquitectura de software.....	31
2.4.2 Patrones de diseño	34
2.4.3 Tarjetas Clase-Responsabilidad-Colaborador	38
2.5 Modelo de datos.....	39
2.6 Conclusiones del capítulo	40
Capítulo 3: Implementación y pruebas de la solución.....	41
3.1 Introducción	41
3.2 Estándares de codificación.....	41
3.2.1 Nomenclatura de los comentarios en las funciones	42
3.2.2 Nomenclatura de las variables en Java	42
3.2.3 Nomenclatura de las variables en XML	42
3.2.4 Nomenclatura de las variables referentes a la base de datos.....	42
3.2.5 Nomenclatura de los métodos	43
3.2.6 Nomenclatura de las clases.....	43
3.3 Tareas de ingeniería	43

ÍNDICE DE CONTENIDO

3.4 Validación de la solución propuesta	44
3.4.1 Pruebas unitarias	44
3.4.1.1 Resultados de las pruebas unitarias.....	45
3.4.2 Pruebas de aceptación.....	47
3.4.2.1 Resultados de las pruebas de aceptación.....	50
3.4.3 Pruebas de rendimiento	51
3.4.3.1 Resultados de las pruebas de rendimiento.....	51
3.5 Conclusiones del Capítulo.....	52
Conclusiones generales	53
Recomendaciones.....	54
Bibliografía	55

ÍNDICE DE TABLAS

Índice de tablas

Tabla 1: Historia de usuario Mostrar detalles de Entidad Jurídica	26
Tabla 2: Estimación del costo en tiempo por historia de usuario	29
Tabla 3: Plan estimado de duración de las iteraciones	30
Tabla 4: Plan de entrega por iteraciones	31
Tabla 5: Tarjeta CRC BufetesActivity	39
Tabla 6: Resumen de los errores de las pruebas unitarias en la Iteración 1	46
Tabla 7: Resumen de los errores de las pruebas unitarias en la Iteración 2	46
Tabla 8: Resumen de los errores de las pruebas unitarias en la iteración 3	47
Tabla 9: HU4_PA4 Mostrar detalles de una Entidad Jurídica	48
Tabla 10: HU6_PA6 Mostrar Entidad Jurídica en el mapa	48
Tabla 11: HU13_PA13 Obtener la ubicación geográfica del dispositivo mediante GPS	49
Tabla 12: HU14-PA14 Trazar rutas entre dos puntos en el mapa	50
Tabla 13: Muestra de dispositivos escogidos para las pruebas de rendimiento	51
Tabla 14: Resultados de las pruebas de rendimiento	52

ÍNDICE DE FIGURAS

Índice de figuras

Figura 1: Evolución de las líneas móviles en Cuba (2003-2018)	2
Figura 2: Cuota de Mercado Sistemas Operativos [2009-2018].....	17
Figura 3: Cuota de mercado de Sistemas Operativos para Móviles y Tabletas [2012-2018]	18
Figura 4: Modelo de la propuesta de solución	24
Figura 5: Patrón arquitectónico Modelo-Vista-Presentador.....	32
Figura 6: Diagrama de paquetes patrón arquitectónico MVP.....	33
Figura 7: Ejemplo patrón GRASP-Experto en la clase BufetesDbHelper.....	35
Figura 8: Relación de composición de la clase BufetesFragment con las clases BufetesDbHelper y la clase BufetesCursorAdapter	36
Figura 9: Patrón Alta cohesión en Callejero Jurídico	37
Figura 10: Esquema de aplicación del patrón Hilo-Esclavo en Callejero Jurídico	38
Figura 11: Diagrama Entidad-Relación Callejero Jurídico.....	40
Figura 12: Nomenclatura de los comentarios en las funciones	42
Figura 13: Nomenclatura de los nombres de las variables en Java	42
Figura 14: Nombres de las variables en XML	42
Figura 15: Nomenclatura de las variables referentes a la base de datos.....	43
Figura 16: Nomenclatura de los métodos	43
Figura 17: Nomenclatura de las clases.....	43
Figura 18: Implementar la funcionalidad mostrarBufete.....	44
Figura 19: Resultados de las pruebas unitarias en la iteración 1.	45
Figura 20: Resultados de las pruebas unitarias en la iteración 2.	46
Figura 21: Resultados de las pruebas unitarias en la iteración 3.	46
Figura 22: Resultados de las pruebas unitarias en la iteración 4.	47
Figura 23: Resultados de las pruebas de aceptación en cada iteración.	50

INTRODUCCIÓN

Introducción

A nivel global la demanda de servicios jurídicos procede de empresas y organizaciones, así como de ciudadanos particulares. Aquellas tienen necesidad de una corriente constante de asistencia jurídica, en tanto que los particulares sólo la necesitan esporádicamente y con frecuencia en situaciones de importancia crucial para sus vidas (divorcio, sucesión, compra-venta de inmuebles y causas penales) (Organización Mundial de Comercio, 1998).

Los servicios jurídicos son brindados en Cuba por diferentes entidades de aquellas que integran el llamado sector jurídico, a saber: Organización Nacional de Bufetes Colectivos, Consultorías Jurídicas, Oficinas Registrales tales como las del Registro Civil, Actos de última voluntad, Antecedentes Penales. También brindan servicios jurídicos otras entidades estatales como por ejemplo el Ministerio de Agricultura con el Registro Pecuario, el Ministerio del Interior con la expedición del Carnet de Identidad o la Licencia de Conducción. Por ello se puede decir que muchos servicios jurídicos, especialmente los servicios que se corresponden con la inscripción y solicitud de certificaciones hacen parte de los denominados trámites de la población.

Actualmente, tanto las personas naturales como jurídicas hacen frecuente uso de los servicios que ofrecen estas entidades. En la presente investigación se centra la atención en buscar una solución para la gestión de la información relacionada a los servicios que se ofrecen en diversas entidades jurídicas cubanas. Por tal motivo se hace referencia a la Organización Nacional de Bufetes Colectivos, las Notarías y las Consultorías pertenecientes al Ministerio de Justicia y que son entidades cuyos servicios son constantemente solicitados por los nacionales y los extranjeros.

Actualmente, a través de sitios webs o directorios telefónicos les llega a los usuarios información básica de estas entidades, por ejemplo: su dirección, sus horarios de atención a la población, así como los servicios que ofrecen las mismas. Aun así, la constante demanda por parte de los ciudadanos hace necesario idear nuevas vías para dar a conocer los servicios que brindan estas entidades con el objetivo de lograr una mayor interacción con los usuarios. En este sentido las Tecnologías de la Información y las Comunicaciones (TIC) han revolucionado entre otras cosas la forma en la que las entidades dan a conocer sus servicios, siendo el campo de las tecnologías móviles uno de los que más impacto ha tenido en la sociedad. Es por ello que la información brindada a los usuarios por parte de las entidades jurídicas no debe limitarse al texto plano y en su lugar aprovechar las bondades que ofrecen las TICs para dinamizarla y promover la retroalimentación entre el usuario y la entidad.

INTRODUCCIÓN

En la actualidad, los teléfonos móviles gozan de una gran popularidad debido a que brindan servicios como: el acceso a Internet, la reproducción de música y videos, así como la posibilidad de tomar fotografías. Son utilizados además para la búsqueda de información asociada a la ubicación geográfica en la que se encuentre el usuario, esto es posible debido al receptor GPS¹ (Global Positioning System) incluidos en este tipo de dispositivos.

En Cuba, como en el mundo, las tecnologías móviles han tenido un gran auge y desarrollo debido a su desenfrenado progreso. Se conoce por datos ofrecidos por la propia Empresa de Telecomunicaciones de Cuba (ETECSA) al portal Cubadebate, en mayo del 2018 había ya 6,5 millones de líneas móviles contratadas por la población como muestra la Figura 1 (ETECSA, 2017).



Figura 1: Evolución de las líneas móviles en Cuba (2003-2018)

Sin embargo, en Cuba las condiciones de conectividad de las redes móviles de datos imposibilitan la utilización de aplicaciones que necesiten conectarse a Internet para su funcionamiento. Por lo

¹ El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación, basado en el espacio, que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita e ininterrumpidamente a usuarios civiles en todo el mundo. A todo el que cuente con un receptor del GPS, el sistema le proporcionará su localización y la hora exacta en cualesquiera condiciones atmosféricas, de día o de noche, en cualquier lugar del mundo y sin límite al número de usuarios simultáneos (Letham, 2001).

INTRODUCCIÓN

que el uso de aplicaciones móviles que permitan al usuario el trabajo *offline*² es esencial para la aceptación por parte del público en general.

Como parte de su misión de producir aplicaciones y servicios informáticos a partir de la vinculación estudio-trabajo la Universidad de las Ciencias Informáticas (UCI) se encarga en gran medida a la informatización de la nación. Para dar cumplimiento a lo anterior, la UCI organiza su actividad productiva en 15 centros de desarrollo de software, los cuales acogen proyectos de gran impacto nacional e internacional.

El Centro de Gobierno Electrónico (CEGEL por sus siglas en español) perteneciente a la Facultad 3 se encarga de desarrollar soluciones informáticas a las entidades del gobierno relacionadas con la rama jurídica. Debido a la gran demanda de productos informáticos que presenta este sector de la sociedad surge el proyecto LexCuba, conformado por un grupo de profesores, especialistas y estudiantes de dicha facultad. Este proyecto ya presenta resultados satisfactorios en aras de informatizar el sector jurídico con las aplicaciones para dispositivos móviles iLex-Leyes³ e iLex-Notario⁴. Dichas aplicaciones han causado un impacto positivo en la sociedad evidenciando así que las tecnologías móviles pueden resultar muy útiles y surgir como un método novedoso a la hora de dar a conocer el derecho a los ciudadanos.

La situación problemática antes descrita ha generado el siguiente **problema de investigación**: ¿cómo propiciar el *acceso a la información* de las entidades que brindan servicios jurídicos en el país desde un dispositivo móvil, a partir de la necesidad de lograr una mayor *centralización y portabilidad* de su información?

Se identifica como **objeto de estudio**: desarrollo de aplicaciones móviles de Informática Jurídica.

Para darle solución al problema planteado se traza como **objetivo general**: desarrollar una *aplicación móvil* para dispositivos con SO Android, que propicie una mayor *centralización y portabilidad* de la información asociada a las entidades que brindan servicios jurídicos en Cuba.

² **Offline o desconectado** (por su traducción al español) en términos informáticos se refiere a no estar conectado o controlado por un sistema central, o no conectado a Internet.

³ Aplicación móvil para dispositivos con sistema operativo Android destinada a dar a conocer las leyes.

⁴ Aplicación móvil para dispositivos con sistema operativo Android desarrollada con el objetivo de dar a conocer los servicios notariales de Cuba.

INTRODUCCIÓN

Centrándose en el **campo de acción**: centralización y portabilidad de la información de las entidades que brindan servicios jurídicos en Cuba, empleando dispositivos móviles con SO Android.

Objetivos específicos:

1. Construir el marco teórico de la investigación orientado al desarrollo de aplicaciones móviles destinadas a la informática jurídica.
2. Realizar la planificación y el diseño de la solución para obtener los componentes de software asociados al acceso de la información de entidades que brinden servicios jurídicos en nuestro país.
3. Realizar la implementación de la solución y valorar su nivel de aceptación y efectividad mediante la realización de pruebas de software.

Trazándose como posible resultado:

Obtención de una aplicación móvil que propicie una mayor centralización y portabilidad de la información asociada a las entidades que brindan servicios jurídicos en Cuba.

Para dar cumplimiento al objetivo propuesto, se utilizan diferentes métodos teóricos y empíricos. Estos métodos permiten la modelación del objeto de la investigación, el examen de la aplicación en su fundamentación y el arribo a consideraciones teóricas planteadas en el transcurso del proceso investigativo.

Métodos Teóricos:

El método **analítico-sintético**: se utilizó para simplificar todas las citas, apuntes y datos tomados al respecto del tema tratado en la presente investigación. La revisión bibliográfica se realizó sobre un conjunto de libros, publicaciones y documentos en soporte electrónico, que se encuentran situados en páginas web, artículos científicos, entre otros.

El método **histórico-lógico**: se utilizó para el trabajo recopilatorio sobre las características de los sistemas de acceso a información, los diferentes paradigmas en el desarrollo de aplicaciones móviles y los diferentes aspectos que conectan a ambos.

El método **modelación**: permitió realizar las actividades encaminadas a la adaptación o construcción de modelos a través de los cuales se representan parcial o totalmente las cualidades esenciales del fenómeno objeto de estudio.

INTRODUCCIÓN

Métodos Empíricos:

El método **entrevista**: brindó apoyo a la incorporación de conocimientos mediante las entrevistas no estructuradas efectuadas a los especialistas de áreas del sector jurídico cubano.

Estructura del Documento:

El presente ejercicio académico consta de la siguiente estructura:

Capítulo I: *Fundamentación teórica*, incluye los aspectos teóricos que soportan esta investigación y el estudio de las tendencias y paradigmas actuales, metodología de desarrollo, las herramientas y las tecnologías propuestas para el desarrollo de la aplicación de acceso a la información de las entidades que brindan servicios jurídicos en Cuba, para dispositivos móviles con sistema operativo Android (esta aplicación será referida en el contenido del presente trabajo como Callejero Jurídico).

Capítulo II: *Planificación y diseño del sistema*, en este capítulo se describen las características de la aplicación a desarrollar. Se realizan las especificaciones de las funcionalidades del sistema y se elabora el plan de entregas por iteraciones.

Capítulo III: *Implementación y pruebas del sistema*, en este capítulo se plantea la construcción de la solución propuesta en el Capítulo 2. Se presentan las principales tareas de ingeniería a realizar y los resultados de las pruebas realizadas al sistema, para comprobar la efectividad de la aplicación.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se define el marco teórico de la investigación. Se realiza un análisis del estado del arte de las principales aplicaciones del sistema operativo *Android* que puedan darle solución al problema planteado. Se analizan tecnologías y herramientas que apoyen el diseño y la implementación de la aplicación, así como la metodología de desarrollo de software a utilizar.

1.2 Conceptos asociados a la investigación

- **Dispositivos Móviles**

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. Entre los dispositivos móviles más utilizados encontramos las Tablet (‘‘Tablet’’) y los teléfonos inteligentes (‘‘Smartphone’’) (Nightingale, 2015).

- **Servicios Jurídicos:**

La palabra **servicio** define a la **actividad y consecuencia de servir** (un verbo que se emplea para dar nombre a la condición de alguien que está a disposición de otro para hacer lo que éste exige u ordena). En términos jurídicos se define como una serie de actividades que son realizadas por profesionales de la Administración Pública de una ciudad, región o país. Conjunto de acciones que tienen como objetivo satisfacer necesidades básicas y fundamentales de personas naturales y jurídicas. Los servicios jurídicos consisten en ejercicios de defensa, asesoramiento jurídico, representación en juicio, acusación pública, dirección jurídica, intervención ante un órgano no judicial de carácter administrativo, custodia de documentos, partición de herencias, preparación, así como la redacción y certificación de documentos jurídicos. La definición de cada uno de los servicios anteriormente mencionados se encuentra en el Sub-Epígrafe 1.2.1.

- **Geolocalización**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La geolocalización consiste en la localización de una persona, objeto, empresa, evento u otro sitio en un lugar geográfico exacto (normalmente representado por un mapa) determinado por unas coordenadas. Éstas provienen generalmente de satélites, aunque se pueden conseguir también por medio de otros dispositivos como las torres de telefonía móvil (RAE, 2018).

1.2.1 Tipos de servicios jurídicos

- **Servicios de asesoramiento y representación jurídicos en derecho penal.**

Los servicios jurídicos de asesoramiento y representación en juicio y de preparación de la documentación utilizada en asuntos penales. Generalmente consisten en la defensa de un cliente ante un órgano judicial por causa de una falta o de un delito penal. También pueden consistir en el ejercicio de la acusación pública en un juicio penal por parte de abogados contratados por el Estado. Se incluye tanto la intervención ante los tribunales como el trabajo de preparación de la misma (es decir, la búsqueda de documentación, el interrogatorio de testigos, el examen de los informes de la policía o de otro tipo) y las actuaciones posteriores a la sentencia penal (Organización Mundial de Comercio, 1998).

- **Servicios de asesoramiento y representación jurídicos en procedimientos judiciales relativos a otras esferas del derecho.**

Los servicios de asesoramiento jurídico y representación en juicio y de preparación de la documentación utilizada en asuntos no penales. Generalmente consisten en la dirección jurídica de la defensa o de la acusación particular por cuenta de un cliente. Se incluye tanto la intervención ante los tribunales como el trabajo de preparación de la misma (es decir, la búsqueda de documentación, el interrogatorio de testigos, el examen de informes de la policía o de otro tipo) y las actuaciones posteriores a la sentencia no penal (Organización Mundial de Comercio, 1998).

- **Servicios de asesoramiento y representación jurídicos en procedimientos administrativos de órganos no judiciales.**

Los servicios de asesoramiento y representación jurídica en un proceso administrativo y de preparación de la documentación utilizada en el mismo. Generalmente consisten tanto en la intervención ante un órgano no judicial de carácter administrativo como en el trabajo de preparación de la misma (es decir, la búsqueda de documentación, el interrogatorio de testigos, el examen de informes) y las actuaciones posteriores a la decisión administrativa (Organización Mundial de Comercio, 1998).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Servicios de documentación y certificación legales.**

Los servicios de preparación, redacción y certificación de documentos jurídicos, incluida la prestación de asesoramiento y la realización de las actividades necesarias para dicha redacción o certificación. Los documentos pueden ser testamentos, estipulaciones matrimoniales, contratos mercantiles, estatutos de sociedades, etc. (Organización Mundial de Comercio, 1998).

- **Otros servicios de asesoramiento e información jurídicos.**

Los servicios de asesoramiento al cliente sobre sus derechos y obligaciones jurídicas y de información sobre cuestiones legales no clasificadas en otra parte. También se incluyen los servicios de custodia de documentos y partición de herencias (Organización Mundial de Comercio, 1998).

1.3 Estudio del estado del arte

Para toda investigación resulta de trascendental importancia que se realice un estudio de los referentes de la misma tanto a nivel nacional como internacional. Por este motivo a continuación se exhiben distintas vías para dar a conocer información asociada a las entidades que brindan servicios jurídicos.

1.3.1 Soluciones internacionales

Poder Judicial

Esta aplicación es presentada por el Poder Judicial de Costa Rica como una nueva forma para promover el acceso a la justicia. En ella encontrará un conjunto de servicios tales como:

- Trámites judiciales.
- Solicitud de la hoja de delincuencia.
- Consulta de órdenes de apremio.
- Consulta de depósitos judiciales.
- Consulta de expedientes del Sistema de Gestión en Línea.
- Consulta de personas sentenciadas en fuga.
- Mapa de las sedes judiciales con números de teléfono y horarios.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

-Consulta de concursos y convocatorias para Jueces y Juezas.

-Información de cómo realizar diversos trámites judiciales (Poder Judicial, 2018).

Notarías Chile APK

Es una aplicación móvil para revisar información de las Notarías de Chile. Contiene información relevante de las notarías: teléfonos, horarios, dirección, sitio web y email. Todas las Notarías de Chile en un sólo lugar. Encontrará Notarías en Santiago Centro, Notarías en Concepción, Notarías en Viña, Notarías en Providencia, Notarías en Maipú, entre otras (Notarios y Conservadores, 2018).

e-Justicia Móvil Marruecos

En el marco de la puesta en marcha de los grandes objetivos estratégicos de la Carta de reforma del sistema judicial, en su faceta relativa a la modernización de la administración de justicia, de conformidad con los objetivos estratégicos de Marruecos y habida cuenta del valor añadido que suponen las nuevas tecnologías en concepto de rendimiento judicial, tanto a nivel de eficiencia y de transparencia, como en términos de mejora de la calidad de las prestaciones judiciales, considerando asimismo las aspiraciones del Ministerio de Justicia de alcanzar los más avanzados niveles de comunicación e interacción con los ciudadanos y usuarios y teniendo presente la importancia capital que reviste hoy el uso de aplicaciones móviles en la comunicación, así como en la difusión y explotación de la información.

El Ministerio de Justicia y Libertades ha desarrollado una aplicación relativa a los servicios judiciales electrónicos: e-Justicia Móvil. Se trata de una nueva concepción encaminada a acercar y facilitar el acceso de los ciudadanos y usuarios a la información judicial. Esta nueva aplicación ofrece varias funcionalidades técnicas y opciones que permiten beneficiarse de una serie de prestaciones judiciales disponibles las 24 horas y los 7 días de la semana. Esta aplicación permite también obtener en tiempo real la información actualizada contenida en los sistemas informáticos del Ministerio.

La aplicación e-Justicia Móvil proporciona la siguiente gama de servicios:

- Servicio de seguimiento de los asuntos: permite consultar los datos y las medidas tomadas en los diferentes procesos civiles y penales a nivel de los juzgados de primera instancia y de los tribunales de apelación; en lo contencioso administrativo a nivel de los tribunales administrativos; y en asuntos comerciales a nivel de los juzgados de lo mercantil.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Servicio de consulta del estado de trámite de las solicitudes de certificado de los antecedentes penales: permite seguir las etapas de tratamiento de las solicitudes de certificado de los antecedentes penales efectuadas vía la página web correspondiente.
- Servicio de consulta de la lista de anuncios judiciales: permite consultar los diferentes anuncios relativos a la venta de bienes muebles e inmuebles que se colocan en los diferentes tribunales del Reino, sin necesidad de desplazarse.
- Servicio de registro mercantil: ofrece, de una parte, la posibilidad de conocer los documentos requeridos para la creación de empresas y los textos jurídicos de referencia que rigen dicha creación y, por otra parte, la posibilidad de informarse acerca de la existencia legal de empresas inscritas en los registros mercantiles, sean personas jurídicas o naturales.
- Servicio de mapa judicial: permite localizar el tribunal territorialmente competente para resolver un asunto, según la naturaleza de éste, y partiendo de la colectividad local donde está instalado. La aplicación ofrece asimismo información de contacto y las coordenadas geográficas de los diferentes tribunales del Reino, por medio de un sistema de mapas interactivos (Ministerio de Justicia de Marruecos, 2017).

Rama Judicial de Puerto Rico APK

Accede información de La Rama Judicial de Puerto Rico a través de tu dispositivo móvil de forma ágil y sencilla. Brinda los siguientes servicios:

- Consulta de Casos: permite realizar búsquedas de casos por nombre de participantes o entidades, número de caso o utilizando la búsqueda avanzada que permite búsquedas de texto por diferentes elementos del caso tales como: nombre de los participantes, región, año o materia para acceder información relevante de los casos. Los resultados se pueden filtrar por año, región o materia. Además, permite añadir los señalamientos al calendario del dispositivo. Permite seleccionar los "Casos Favoritos" para recibir alertas de señalamientos futuros (Utilizando el botón de favorito: ☆).
- Consulta de Calendario: permite visualizar el calendario de las diferentes salas y obtener las direcciones a través del GPS del dispositivo para llegar a las salas.
- Directorio de Abogados y Abogadas: como parte de un servicio público de la Rama Judicial podrá acceder a la base de datos que contiene todos los nombres y direcciones, incluyendo la dirección

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

electrónica, en la mayor parte de los casos, de todos los profesionales admitidos y que se encuentran activos en ejercicio de la abogacía en el País. La información que despliega el motor de búsqueda es la que consta en el Registro Único de Abogados y Abogadas (RUA). Mediante su uso los ciudadanos y las ciudadanas podrán optar por el contacto inicial con abogados y abogadas cercanos a su residencia, a la Región Judicial donde se verá la acción judicial y especializados en la materia que interesan. Igualmente, los miembros de la profesión tendrán el beneficio de la información certera para la comunicación entre colegas, esto entre algunos beneficios que podemos mencionar que trae consigo esta publicación. Este servicio permite además enviar correos electrónicos a otros abogados y abogadas o crea contactos nuevos en tu dispositivo móvil con la información de algún abogado o abogada.

- Directorio Telefónico: permite obtener información de contacto de los diferentes Tribunales, oficinas y dependencias de la Rama Judicial de Puerto Rico (Rama Judicial de Puerto Rico, 2014).

Busca Justicia APK

Busca Justicia es una aplicación que facilita la búsqueda de servicios de justicia en todo Chile, permitiendo a los ciudadanos acceder a ella en cualquier momento y desde diversos dispositivos móviles. Desarrollada por el Ministerio de Justicia, esta aplicación ha sido implementada en el marco del proyecto Justicia Responde, plataforma que busca solucionar los diferentes requerimientos judiciales de la ciudadanía, a través de la entrega de orientación, soluciones a los requerimientos de las personas además de la realización de trámites en línea (Ministerio de Justicia de Chile, 2018).

1.3.2 Soluciones nacionales

iLEX Notario APK

iLEX Notario es una aplicación que facilita la búsqueda de servicios notariales en toda Cuba, permitiendo a los ciudadanos acceder a ella en cualquier momento y desde diversos dispositivos móviles con sistema operativo Android. Desarrollada por el Centro de Gobierno Electrónico (CEGEL), esta aplicación ha sido implementada en el marco del proyecto iLEX, plataforma que busca solucionar los diferentes requerimientos judiciales de la ciudadanía, a través de la entrega de orientación para dar soluciones a los requerimientos de las personas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.3 Análisis de las soluciones encontradas

En el proceso de investigación del estado del arte se encontraron varias soluciones internacionales que brindan información de entidades que ofrecen servicios jurídicos, las mismas no proponen una solución completa al problema existente, ya que fueron desarrolladas para sistemas judiciales distintos al nuestro, en el caso de la aplicación iLEX Notario contiene información únicamente relacionada con las Notarías cubanas fundamentando la necesidad de desarrollar una aplicación destinada a brindar información de todas las entidades que brindan servicios jurídicos en Cuba, así como la posibilidad de localizar estas entidades mediante el uso de mapas *offline*.

1.4 Metodología de Desarrollo de software

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo de software. Letelier y Penadés (Letelier, y otros, 2006) las dividen entre las propuestas más tradicionales, centradas especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán, y las centradas en dimensiones tales como el factor humano o el producto software. La segunda variante corresponde a la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software en iteraciones cortas. Este enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad.

Los equipos de desarrollo de aplicaciones empresariales utilizan prácticas tradicionales para definir y desarrollar aplicaciones de escritorio; sin embargo, la mayoría no funcionan con el desarrollo de aplicaciones móviles, debido a la diversidad de dispositivos, conectividad de red y otras consideraciones específicas para móviles, (...) En cambio, los directores de desarrollo de aplicaciones deben utilizar pruebas funcionales, de rendimiento, de carga y de experiencia de usuario, así como las prácticas de desarrollo ágil (Gartner Inc., 2016).

Van L. Baker, vicepresidente de investigaciones de *Gartner Inc.* (Gartner Inc., 2016), plantea que existen varias razones por las que las prácticas de desarrollo tradicionales no tienen éxito en el desarrollo de aplicaciones móviles, a pesar de que han funcionado históricamente. En primer lugar, las aplicaciones móviles son una nueva categoría para la mayoría de los usuarios y, en segundo lugar, están limitadas por la naturaleza de la plataforma y el tamaño de la pantalla. Lo anterior sugiere que el uso de metodologías ágiles es el enfoque práctico más favorable para el desarrollo de este tipo de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

aplicaciones; sin embargo, existen diversas opciones posibles a la hora de optar por una metodología entre las que pertenecen a este grupo. A continuación, se ofrece una breve descripción de algunas de ellas.

1.4.1 SCRUM

SCRUM es una metodología especialmente desarrollada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir de la siguiente forma: el desarrollo de software se realiza mediante iteraciones denominadas *Sprint*⁵, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Al finalizar un sprint, las tareas que se han realizado, y con las que el cliente ha quedado conforme, no se vuelven a modificar. El equipo de desarrollo trata de seguir el orden de prioridad que marca el cliente, pero puede modificarlo si determina que es mejor hacerlo y cada miembro del equipo trabaja de forma individual. Esta metodología se basa más en la administración del proyecto que en la propia programación o creación del producto (Scrum BoK, 2016).

1.4.2 AUP

El Proceso Unificado Ágil de Scott Ambler, o *Agile Unified Process* (AUP) en inglés, es una versión simplificada del Proceso Unificado Racional (RUP). Este describe, de una manera simple y fácil de entender, la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (Ambler, 2005). AUP aplica un conjunto de técnicas ágiles entre las que se incluyen:

- Desarrollo Dirigido por Pruebas (*Test Driven Development* - TDD en inglés).
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de base de datos para mejorar la productividad.

En la Universidad de las Ciencias Informáticas se ha realizado una variación de esta metodología de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la institución (Universidad de las Ciencias Informáticas, 2016). En esta variación la metodología cuenta con 3 fases:

⁵ **Sprint:** carrera corta, a toda velocidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Inicio: se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se llevará a cabo el mismo.

Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase, el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.

Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

1.4.3 Metodología XP

XP⁶ es una metodología ágil centrada en potenciar las relaciones inter-personales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Letelier, y otros, 2006). Cuenta con cuatro actividades estructurales: planeación, diseño, codificación y pruebas, las cuales se definen a continuación:

Planeación: esta actividad comienza escuchando, es una actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieren. Escuchar lleva a la creación de algunas “historias” (también llamadas historias del usuario) que describen la salida necesaria, características y funcionalidad del software que se va a elaborar (Pressman, 2010).

⁶ **XP:** eXtreme Programming (Programación Extrema).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Diseño: el diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después. XP estimula el uso de las tarjetas CRC (clase-responsabilidad-colaborador) como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software (Pressman, 2010).

Implementación o codificación: después de que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso (incremento de software). Una vez creada la prueba unitaria, el desarrollador está mejor capacitado para centrarse en lo que debe implementarse para pasar la prueba. No se agrega nada extraño (MS). Una vez que el código está terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores (Pressman, 2010).

Pruebas: quedó planteado que la creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (de modo que puedan ejecutarse en repetidas veces y con facilidad). Esto estimula una estrategia de pruebas de regresión siempre que se modifique el código (lo que ocurre con frecuencia, dada la filosofía del rediseño en XP). Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software (Pressman, 2010).

1.4.4 Metodología de desarrollo de software seleccionada

Para el desarrollo de Callejero Jurídico se escoge XP como metodología que guiará dicho proceso, teniendo en cuenta los aspectos que se listan a continuación:

- El proyecto es de corta duración.
- No es necesaria la generación excesiva de artefactos y roles ya que el proyecto es pequeño y está diseñado para ser realizado en el menor tiempo posible.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Los requisitos suelen cambiar con frecuencia en la medida en que avanza el desarrollo del proyecto, así el cliente puede ir añadiendo Historias de Usuario, dividir las para agilizar el trabajo o eliminarlas simplemente. XP permite al equipo de trabajo modificar los planes de desarrollo conforme a lo anterior.
- Tanto cliente como desarrolladores forman parte del equipo de trabajo de forma tal que se logra una interacción constante entre ellos. Esto posibilita la retroalimentación, corrección de errores y finalmente la realización de un producto que cumpla las expectativas de las partes involucradas en el proceso.

1.5 Sistema Operativo móvil

A la hora de desarrollar software hay que prestarle vital atención al sistema operativo que va a soportar la aplicación a desarrollar. Un Sistema Operativo (SO) es un programa o conjunto de programas que realizan funciones básicas y permiten el desarrollo de otros programas (RAE, 2018). Entre los sistemas operativos más utilizados para dispositivos móviles se encuentran *Android*, *iOS*, *Windows Phone* y *BlackBerry OS*. Según datos de *StatCounter GlobalStats*⁷ el sistema operativo más usado en el mundo el día de hoy es Android superando a Windows y marcando una elevada tendencia de ascenso, como muestra la Figura 2 (Statcounter GlobalStats, 2018):

⁷ **StatCounter GlobalStats**: es una página de estadísticas que se basan en datos agregados, recopilados por StatCounter en una muestra que supera los 10 mil millones de visitas a la página por mes, recopiladas en toda la red de StatCounter de más de 2 millones de sitios web.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

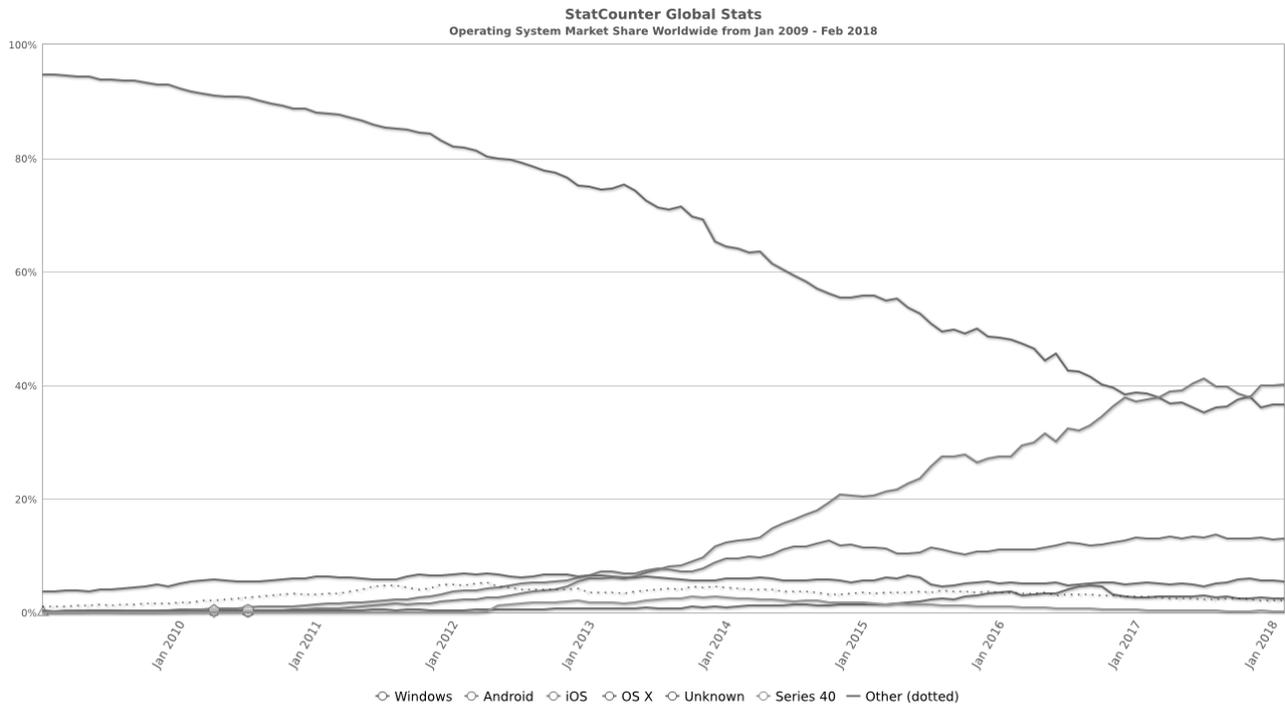


Figura 2: Cuota de Mercado Sistemas Operativos [2009-2018]

Pero si se muestra la comparación un poco más centrada a los dispositivos móviles y a nuestro país vemos como Android es el dueño casi total del mercado de los dispositivos móviles como se ve reflejado en la Figura 3:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

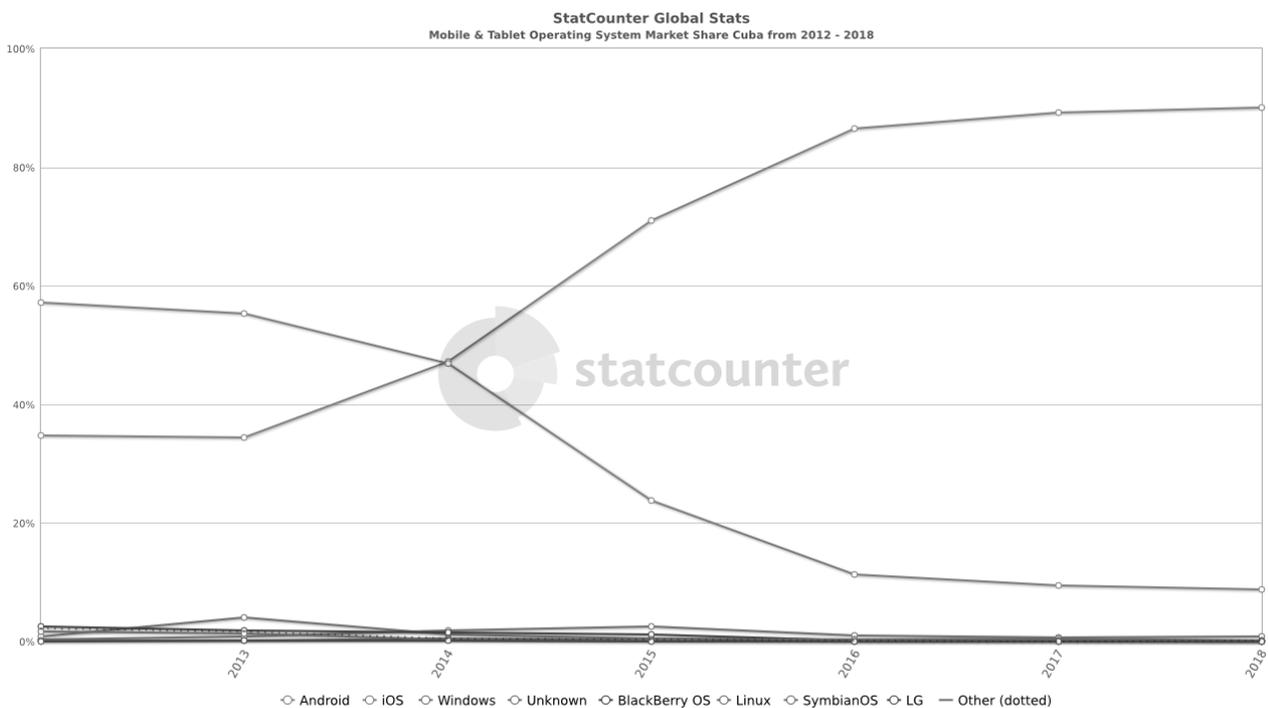


Figura 3: Cuota de mercado de Sistemas Operativos para Móviles y Tabletas [2012-2018]

Si bien iOS⁸ es el segundo SO⁹ móvil más utilizado, Android reina en el mercado cubano presente en el 90% de los dispositivos. Es posible encontrar tabletas (*tablets*), teléfonos móviles (celulares) y relojes equipados con Android, aunque el software también se usa en automóviles, televisores y otras máquinas. Además, al ser software libre es normal que muchos desarrolladores lo prefieran para realizar sus proyectos. Es por ello que se decide enfocar el presente proyecto en dispositivos que utilicen Android como sistema operativo, con el objetivo de que llegue a la mayor cantidad de personas posibles.

1.6 Mapas *offline*

Los mapas *offline* o mapas sin conexión, permiten mostrar al usuario de la aplicación mapas con toda la información necesaria, y sin requerir conexión a internet, esto es posible gracias a la utilización de bibliotecas de código abierto y tecnologías integradas en este tipo de dispositivos.

Los archivos de mapas se obtienen de la web de *OpenStreetMap* (también conocido como **OSM**). *OpenStreetMap* es un proyecto colaborativo para crear mapas libres y editables. Los mapas se crean

⁸ iOS: es el sistema operativo diseñado por Apple para sus productos, iPhone, iPad, iPod Touch, y Apple TV.

⁹ SO: sistema operativo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías¹⁰ y otras fuentes libres. Esta cartografía, tanto las imágenes creadas como los datos vectoriales almacenados en su base de datos, se distribuye bajo licencia abierta Licencia Abierta de Bases de Datos (en inglés *ODbL*) (*OpenStreetMap*, 2018).

Para la visualización de los mapas en dispositivos con SO Android se emplea la biblioteca de código abierto *Mapsforge*, ya que es una biblioteca libre y gratuita que integra los mapas de *OpenStreetMap* en Android, sus características principales son:

- Formato de archivo ligero para renderizado rápido de datos *OpenStreetMap* en el dispositivo.
- De fácil uso, similar a la biblioteca de *Google Maps*.
- API de capas potente y flexible.
- Estilos de mapas personalizables mediante ficheros de configuración XML.
- Biblioteca de sólo 400KB de tamaño.

1.7 Lenguajes, herramientas y tecnologías.

A continuación, se definen las herramientas, lenguajes y tecnologías que apoyarán el desarrollo de la investigación.

1.7.1 Herramienta de modelado.

Las herramientas CASE¹¹ son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. El Lenguaje Unificado de Modelado (UML), permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra el desarrollo de un software (Sommerville, 2005).

Visual Paradigm for UML 8.0 Enterprise Edition:

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el

¹⁰ La **ortofotografía** es una presentación fotográfica de una zona de la superficie terrestre, en la que todos los elementos presentan la misma escala, libre de errores y deformaciones, con la misma validez de un plano cartográfico.

¹¹ CASE: *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

desarrollo orientado a objetos. Es una herramienta multiplataforma, por lo que tiene la capacidad de ejecutarse sobre diferentes Sistemas Operativos. Es fácil de usar y presenta una agradable interfaz para interactuar con el usuario (Sundara, 2013).

1.7.2 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado, del inglés *Integrated Development Environment (IDE)*, es un programa informático compuesto por un conjunto de herramientas de programación. Los IDE proveen un marco de trabajo para los lenguajes de programación (GlosarioIT, 2018).

Android Studio 2.3:

Es el IDE proporcionado por Google para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA¹². Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece funciones que aumentan la productividad durante la compilación de apps para Android, como las siguientes (Android Developers, 2018):

- Sistema de compilación flexible basado en Gradle¹³.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run, para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK¹⁴.
- Integración de plantillas de código y GitHub¹⁵, para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks¹⁶ de prueba.
- Herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.

¹² **IntelliJ IDEA:** es un ambiente de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ).

¹³ **Gradle:** sistema de automatización de compilación de código abierto.

¹⁴ **APK:** (*Android Application Package*, significado en español: *Aplicación empaquetada de Android*) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para teléfonos inteligentes y tabletas (Cataldi, y otros, 2013).

¹⁵ **GitHub:** GitHub es una plataforma de desarrollo inspirada en la forma en que trabajas. Desde el código abierto hasta el negocio, puede alojar y revisar códigos, administrar proyectos y crear software junto a millones de otros desarrolladores.

¹⁶ **Frameworks:** entornos de trabajo o marcos de trabajo, son un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Compatibilidad con C++ y NDK¹⁷.

1.7.3 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación (Sebesta, 1993).

Java como lenguaje nativo en Android

Java es un lenguaje de programación orientado a objetos que se incorporó al ámbito de la informática en los años noventa. La idea de Java es que pueda realizarse programas con la posibilidad de ejecutarse en cualquier contexto, en cualquier ambiente, siendo así su portabilidad uno de sus principales logros. Fue desarrollado por Sun Microsystems¹⁸, posteriormente adquirido por Oracle. En la actualidad puede utilizarse de modo gratuito, pudiéndose conseguir sin problemas un paquete para desarrolladores que oriente la actividad de programar en este lenguaje. Puede ser modificado por cualquiera, circunstancia que lo convierte en lo que comúnmente se denomina “código abierto”. En la actualidad, este lenguaje de programación ha ganado una enorme popularidad como consecuencias de su portabilidad, su simpleza y sus grandes posibilidades de utilización. Está especialmente extendido su uso en dispositivos móviles (Gosling, y otros, 2014).

1.7.4 SQLite

SQLite es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales *SQL*¹⁹ auto-contenido, sin servidor y sin configuración. El código de *SQLite* es de

¹⁷ **NDK:** es un conjunto de herramientas que le permite implementar partes de su aplicación en código nativo, usando lenguajes como C y C ++. Para ciertos tipos de aplicaciones, esto puede ayudarlo a reutilizar las bibliotecas de códigos escritas en esos idiomas.

¹⁸ **Sun Microsystems:** fue una empresa informática que se dedicaba a vender estaciones de trabajo, servidores, componentes informáticos, software (sistemas operativos) y servicios informáticos. Fue adquirida en el año 2010 por Oracle Corporation, actualmente se destaca como fabricante de semiconductores y software.

¹⁹ **SQL:** (Structured Query Language) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, gracias a la utilización del álgebra y de cálculos relacionales.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

dominio público y libre para cualquier uso, ya sea comercial o privado. Permite almacenar información en dispositivos de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser un teléfono celular. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel (Owens, y otros, 2010).

SQLite se esfuerza por proporcionar almacenamiento de datos local para aplicaciones y dispositivos individuales. También hace hincapié en la economía, fiabilidad, independencia y simplicidad. Android incorpora *SQLite* en sus API²⁰ por defecto para llevar a cabo de manera sencilla todas las tareas que se necesitan realizar en un proyecto (Owens, y otros, 2010).

1.7.5 Android SDK

Android SDK siglas en inglés de *Software Development Kit* (kit de desarrollo de software) es un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, en este caso para Android. Este paquete o kit de desarrollo incluye las API siglas en inglés de *Application Programming Interface* (Interfaz de Programación de Aplicaciones) y herramientas necesarias para desarrollar las aplicaciones, utilizando Java como lenguaje de programación (Developers, 2013).

1.7.6 Gradle 3.3

Es un sistema de compilación basado en JVM (Máquina Virtual de Java, del inglés Java Virtual Machine) a su vez es un plugin, lo que facilita su actualización y su exportación de un proyecto a otro. Gradle permite reutilizar código fácilmente, hace sencilla la tarea de configurar y personalizar la compilación, permite la distribución sencilla de código al resto del mundo y gestiona las dependencias de forma potente y cómoda ya que está basado en Maven. Esta herramienta emplea al *javac* (Compilador de Java o Java Compiler) para programar mediante “Scripting” el funcionamiento de la integración modular de la aplicación y permite compilar la aplicación mientras es desarrollada (Gradle, 2018).

1.7.7 Maven

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar tareas claramente definidas, como la compilación del código y su empaquetado. Es decir, hace posible la

²⁰ **API:** es el mecanismo más útil para conectar dos *softwares* entre sí para el intercambio de mensajes o datos en formato estándar.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

creación de *software* con dependencias incluidas dentro de la estructura del JAR. Es necesario definir todas las dependencias del proyecto (librerías externas utilizadas) en un fichero propio de todo proyecto *Maven*, el POM (*Project Object Model*). Este es un archivo en formato XML que contiene todo lo necesario para que cuando se quiera generar el fichero ejecutable de la aplicación, contenga todo lo que necesita para la ejecución (Apache Maven, 2018).

1.7.8 Java SE JDK 1.8

Es el conjunto de recursos Java, o plataforma, que conforman la denominada Java Standard Edition, fundamental para el desarrollo de aplicaciones Java y requerido por el entorno de desarrollo. Este conjunto de recursos no es propio de la plataforma Android, pero debe ser instalada en el ordenador para poder desarrollar aplicaciones (Oracle, 2018).

1.7.9 JUnit 4.12

JUnit es un *framework* que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

1.8 Conclusiones del capítulo

En este capítulo, la realización del estudio de los sistemas homólogos permitió determinar las principales características y funcionalidades de las soluciones que brindan información de entidades que ofrecen servicios jurídicos existentes a nivel tanto nacional como internacional. Además, la descripción de los principales conceptos posibilitó adquirir una mayor comprensión de los temas relacionados con el objeto de estudio y la necesidad de implementar una aplicación para dispositivos móviles que permita obtener información detallada de las entidades cubanas que brindan servicios jurídicos. Por otra parte, el análisis de diferentes herramientas y tecnologías permitió alcanzar los conocimientos necesarios para seleccionar las adecuadas para el desarrollo de la solución. Por lo que, se decide utilizar como entorno de desarrollo integrado Android Studio, como metodología a utilizar XP, y como lenguaje de programación Java para la implementación de la propuesta de solución.

Capítulo 2: Planificación y diseño del sistema

2.1 Introducción

En este capítulo se explican los principales aspectos de diseño de la aplicación móvil para dispositivos con SO Android Callejero Jurídico. Su contenido incluye la descripción del modelo de propuesta de solución, así como las fases de Planeación y Diseño correspondientes a la metodología de desarrollo de software seleccionada. También se define la arquitectura de software a utilizar y un conjunto de patrones que suministrarán un esquema para refinar componentes de software y la forma en que se relacionarán.

2.2 Modelo de la propuesta de solución

La solución propuesta está diseñada para todas las personas interesadas en las entidades jurídicas cubanas, y dispositivos móviles que operan con sistema operativo Android. Su objetivo es permitir la consulta de la información relacionada con las entidades cubanas que brindan servicios jurídicos, desde cualquier punto geográfico del país y sin necesidad de una conexión a la red, al tratarse de una aplicación offline. La solución debe permitir a sus usuarios conocer información asociada de estas entidades, así como la posibilidad de visualizarlos utilizando mapas interactivos.



Figura 4: Modelo de la propuesta de solución

Callejero Jurídico será capaz de mostrar de manera minuciosa, los datos de cada una de las entidades jurídicas cubanas; incluyendo su ubicación, dirección, teléfonos, correo electrónico, horarios de

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

atención a la población y servicios que brindan. Contará con la posibilidad de realizar búsquedas filtradas por nombre, dirección, provincia, municipio, así como la posibilidad de buscar solamente las entidades abiertas al momento de la búsqueda, permitirá también al usuario, realizar llamadas telefónicas desde la aplicación, el envío de correos electrónicos, añadir entidades a favoritos, ver la entidad deseada en un mapa interactivo, obtener la ubicación del dispositivo mediante GPS y trazar rutas desde la ubicación del dispositivo hasta una entidad o un punto determinado.

2.3 Planeación

En la etapa de Planeación comienza la interacción con el cliente para identificar los requerimientos del sistema y se identifican las iteraciones y ajustes necesarios a la metodología según las características de la solución. En la misma se describen tres elementos fundamentales, los cuales son: Historia de Usuario (HU), Tiempo de Ejecución del Proyecto y Plan de Entrega, en ellos se recogen las características principales del sistema a construir y se concreta el tiempo que se necesitará para implementarlas.

2.3.1 Historias de usuario

Kent Beck (Beck, 2000), define a las HU como una parte importante de XP. Constituyen el cimiento de los requisitos del sistema; son el artefacto primario utilizado para definir lo que el equipo de desarrollo estará implementando en una iteración dada. El cliente es el encargado de asignar una prioridad a cada HU y es el equipo de desarrollo el encargado de asignar un costo, que se traduce en las semanas que llevará el desarrollo de las mismas. En la tabla siguiente se muestra un ejemplo de las HU especificadas para el desarrollo de Callejero Jurídico. Las HU restantes se encuentran en el Anexo I Historias de Usuario.

Historia de usuario	
Número: HU-4	Nombre de la Historia de usuario: Mostrar detalles de una Entidad Jurídica.
Modificación de la historia de usuario: 1	
Usuario: Adrian Pérez Zaldivar	Iteración asignada: 1
Prioridad en el negocio: Alta	Puntos estimados: 1/2
Descripción: Se muestran los datos generales de una Entidad Jurídica, dígame, nombre, dirección, números de teléfono, horarios de atención a la población, foto de la entidad y servicios jurídicos que brinda.	

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

Observaciones:

Prototipo elemental de interfaz gráfica de usuario:



Tabla 1: Historia de usuario Mostrar detalles de Entidad Jurídica

Leyenda de las Historias de Usuario:

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

Nombre historia de usuario: es el nombre de la HU, sirve para identificarla fácilmente tanto para los desarrolladores como para los clientes.

Modificación de historia de usuario: cantidad de modificaciones que se le ha realizado a la HU (de no tener modificaciones se pone ninguna, si no la cantidad de veces que ha sido modificada).

Usuario: nombre del programador encargado de implementar la HU.

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

La prioridad en el negocio:

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

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

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

Puntos estimados: tiempo en semanas que se le asignará, donde un punto de estimación corresponde a una semana de 5 días con 8 horas de trabajo diarias.

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

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

Prototipo elemental de interfaz gráfica de usuario: se muestra una imagen de cómo quedaría la interfaz gráfica del requisito asociado a la HU.

De acuerdo a lo planteado por *Cohn y Paul* (Cohn, y otros, 2001), en XP no es necesario desarrollar una especificación formal de requisitos del sistema. En su lugar, se espera que un conjunto de HU informales realice esa función. Las HU no son un conjunto único, normalizado, vinculado, completo y consistente. Sin embargo, satisfacen las propiedades de ser abstractas, limitadas, modificables, trazables, comprobables, configurables y granulares. El concepto de HU de XP también satisface la mayor parte de los requisitos de categorización (Identificación, Prioridad, Criticidad, Viabilidad, Riesgo y Fuente) de la guía IEEE²¹.

2.3.2 Características no funcionales del sistema

Aun cuando XP confía en las HU como principal artefacto para la especificación de los requisitos del sistema, Eberlein y Leite plantean en (Eberlein, y otros, 2002) que, si se consideran los aspectos no funcionales sólo en el nivel de implementación, como se entiende que es el caso de las metodologías ágiles, entonces los problemas serán aún mayores que cuando se tienen en cuenta sólo en el nivel de

²¹ IEEE: Instituto de Ingeniería Eléctrica y Electrónica, asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

diseño. Para evitar esto, se realiza una especificación formal de las características no funcionales de la solución propuesta.

Versión de Android: el dispositivo donde se instalará la aplicación debe tener instalado una versión de Android igual o posterior a la versión 4.0. Esta es la versión mínima con soporte completo en el SDK utilizado por el equipo de desarrollo.

Soporte multilingaje: en su primera versión, la aplicación Callejero Jurídico incluirá los lenguajes inglés y español, con el primero como lenguaje por defecto.

Interfaces: las interfaces gráficas implementadas por el sistema deben concebirse con un ambiente sencillo y de navegación fácil para el usuario, debido a que la aplicación podrá ser usada por usuarios con conocimientos básicos de dispositivos móviles.

Tipo de dispositivo: no es preciso que sea un teléfono móvil, también se pueden emplear tabletas o usar una computadora haciendo uso de un emulador de Android.

Hardware: para un correcto funcionamiento de la aplicación el dispositivo donde se encuentre instalada debe contar con los requerimientos de hardware que se describen a continuación. Procesador ARMv6, ARMv7, ARM o ARM64 a 800 Mhz o superior, 512 Mb mínimos de memoria RAM, contar con mínimo 512 Mb de memoria ROM y 80 Mb de espacio libre de almacenamiento.

2.3.3 Tiempo de ejecución del proyecto

El tiempo de ejecución del proyecto es un elemento que le permite al equipo de desarrollo completar las historias de usuario en una determinada iteración. Este elemento se calcula sumando los puntos estimados de las historias de usuario realizadas en una iteración. En este sentido un punto estimado equivale a una semana ideal de programación donde se trabaje sin interrupciones, ocho horas diarias durante cinco días. Dado que la planificación se realiza basándose en el tiempo de implementación de una historia de usuario, el tiempo de ejecución se utiliza para declarar cuántas historias de usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de las mismas. La tabla siguiente muestra la estimación del costo en tiempo realizada a partir de la especificación de las HU, la cual provee una idea de lo que tardará el desarrollo de la solución propuesta en el presente trabajo.

Número	Nombre de la historia de usuario	Puntos estimados
--------	----------------------------------	------------------

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

HU-1	Crear BD SQLite con información asociada a todas las entidades del país que brinden servicios jurídicos.	1
HU-2	Mostrar Menú Lateral.	1
HU-3	Mostrar listado de Entidades Jurídicas.	½
HU-4	Mostrar detalles de una Entidad Jurídica.	½
HU-5	Mostrar Mapa.	1
HU-6	Mostrar Entidad Jurídica en el mapa.	½
HU-7	Llamar a Entidad Jurídica.	¼
HU-8	Enviar correo a Entidad Jurídica.	½
HU-9	Mostrar información de contacto.	¼
HU-10	Capturar Pantalla.	½
HU-11	Mostrar Entidades Jurídicas por Provincia.	1
HU-12	Mostrar Diccionario de Términos Jurídicos.	1
HU-13	Obtener ubicación mediante GPS.	1
HU-14	Trazar rutas en el mapa.	1
HU-15	Compartir Aplicación.	1
HU-16	Mostrar Pantalla Inicial.	1

Tabla 2: Estimación del costo en tiempo por historia de usuario

La estimación realizada da como resultado una duración aproximada de tres meses para el desarrollo de la solución propuesta. Este resultado permite dividir el proceso de desarrollo en etapas para facilitar su realización. Letelier y Penadés (Letelier, y otros, 2006) describen estas etapas en la tercera fase de XP y las denominan iteraciones.

2.3.4 Iteraciones

Para cada iteración, se define un número determinado de historias de usuario a implementar y al final de cada iteración se obtiene como resultado la entrega de las funcionalidades correspondientes. La duración de cada iteración no debe ser superior a tres semanas. Esto ocurre conjuntamente con la aceptación, por parte del cliente, de los requerimientos solucionados. Idealmente, al final de la última iteración el sistema estará listo para entrar en producción. La tabla siguiente muestra las iteraciones en que estará dividido el proceso de desarrollo de la solución propuesta en el presente trabajo, así como la distribución de las HU a desarrollar por cada iteración definida.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

Iteración	Título de la historia de usuario	Duración total
1	<ol style="list-style-type: none">1. Crear Base de Datos SQLite con información asociada a todas las entidades del país que brinden servicios jurídicos.2. Mostrar Menú Lateral.3. Mostrar listado de Entidades Jurídicas.4. Mostrar detalles de una Entidad Jurídica.	3
2	<ol style="list-style-type: none">5. Mostrar Mapa.6. Mostrar Entidad Jurídica en el mapa.7. Llamar a Entidad Jurídica.8. Enviar correo a Entidad Jurídica.9. Mostrar información de contacto.10. Capturar Pantalla.	3
3	<ol style="list-style-type: none">11. Mostrar Entidades Jurídicas por Provincia.12. Mostrar Diccionario de Términos Jurídicos.13. Obtener ubicación mediante GPS.	3
4	<ol style="list-style-type: none">14. Trazar rutas en el mapa.15. Compartir Aplicación.16. Mostrar Tutorial Inicial.	3

Tabla 3: Plan estimado de duración de las iteraciones

Los resultados anteriores permiten estimar la duración de cada iteración, utilizando para ello los puntos estimados de las HU correspondientes. Esto permite elaborar un plan de entrega de funcionalidades implementadas al final de cada iteración.

2.3.5 Plan de entrega

El plan de entrega consiste en establecer, de manera conjunta entre el equipo de desarrollo y el cliente, la duración y fecha de entrega de cada iteración hasta lograr el producto final. En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento, hasta lograr el producto final en la fecha establecida. En la siguiente tabla se desglosa el plan de entrega para las funcionalidades implementadas en cada iteración del proceso de desarrollo de Callejero Jurídico.

Iteraciones	Fecha de entrega
-------------	------------------

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

1	8 de abril de 2018.
2	29 de abril de 2018.
3	20 de mayo de 2018.
4	13 de junio de 2018.

Tabla 4: Plan de entrega por iteraciones

2.4 Diseño de la propuesta de solución

En esta sección se realiza la descripción de la arquitectura de software, los patrones de diseño aplicados a la propuesta de solución y la descripción de las principales clases de la aplicación. Estos elementos muestran una vista técnica y lógica del software a desarrollar, describen la arquitectura utilizada, basada en las necesidades de la solución, así como las clases y sus relaciones; además, son el punto de partida para la posterior codificación.

2.4.1 Arquitectura de software

La arquitectura de software de un sistema o programa de computación es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de estos elementos, y las relaciones entre ellos (Len Bass, y otros, 2003).

Patrón Modelo-Vista-Presentador (MVP) en Android

MVP es un patrón de diseño que surge para ayudar a realizar pruebas automáticas de la interfaz gráfica, para ello la idea es codificar la interfaz de usuario lo más simple posible, teniendo el menor código posible, de forma que no merezca la pena probarla. En su lugar, toda la lógica de la interfaz de usuario, se hace en una clase separada (que se conoce como Presentador), que no dependa en absoluto de los componentes de la interfaz gráfica y que, por tanto, es más fácil de realizar pruebas. Idealmente el patrón MVP permitiría conseguir que una misma lógica pudiera tener vistas totalmente diferentes e intercambiables (Diéguez, 2015).

A continuación, en la Figura 4 se presenta el modelo de arquitectura uniforme para un proyecto Android basado en el Modelo Vista Presentador (MVP por sus siglas en inglés *Model View Presenter*) familia del patrón conocido como Modelo Vista Controladora (MVC por sus siglas en inglés *Model View Controller*).

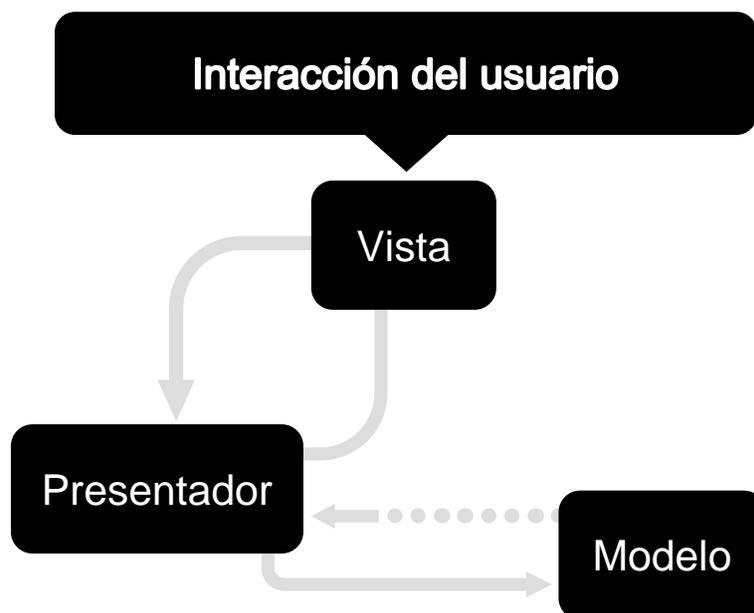


Figura 5: Patrón arquitectónico Modelo-Vista-Presentador

En la imagen se pueden ver la siguiente separación de capas basada en MVP:

Capa Vista: Incluye las interfaces de usuario (Actividades, recursos, layouts, entre otros), y aquellos artefactos de Android sin interfaz gráfica (utilidades visuales). Las clases actividades contenidas en esta capa son las responsables de la gestión de los componentes visuales, `BufetesActivity.java`, `Ajustes.java` y `VerFavoritos.java`. En los formularios (layouts), se incluyen los archivos XML `activity_bufetes.xml`, `activity_ajustes.xml` y `ver_favoritos.xml` y como clase de utilidad visual la clase `ActivityMain`.

Capa Presentador o Capa de Negocio: Incluye las clases del negocio, que hacen de puente entre la vista y el modelo, incluyendo clases con métodos que alberguen la lógica de la herramienta. Siempre los artefactos de la capa Vista invocarán a las clases albergadas en esta capa, nunca accederán al modelo directamente. Las clases presentadoras contenidas en esta capa de negocio son las que albergan el control y gestión sobre todo el contenido referente a cada caso de uso, `BufetesCursorAdapter`, `BufetesDbHelper` y `BufetesDetalleActivity`.

Capa Modelo: Dos partes importantes:

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

- **Clases de Entidad:** incluyen la definición lógica del modelo de datos a utilizar. Estas clases serán manejadas por la capa de negocio. Contiene las clases Bufete y BufeteContract, encargadas de controlar los elementos de los bufetes y los datos de estas entidades.
- **Capa de Infraestructura:** Abstrae la complejidad tecnológica de la gestión del almacenamiento físico. En el caso de la herramienta propuesta, contiene la base de datos local SQLite “entidades_jurídicas” (Diéguez, 2015).

El diagrama de paquetes de la figura No. 6 muestra el uso de la arquitectura MVP en la aplicación.

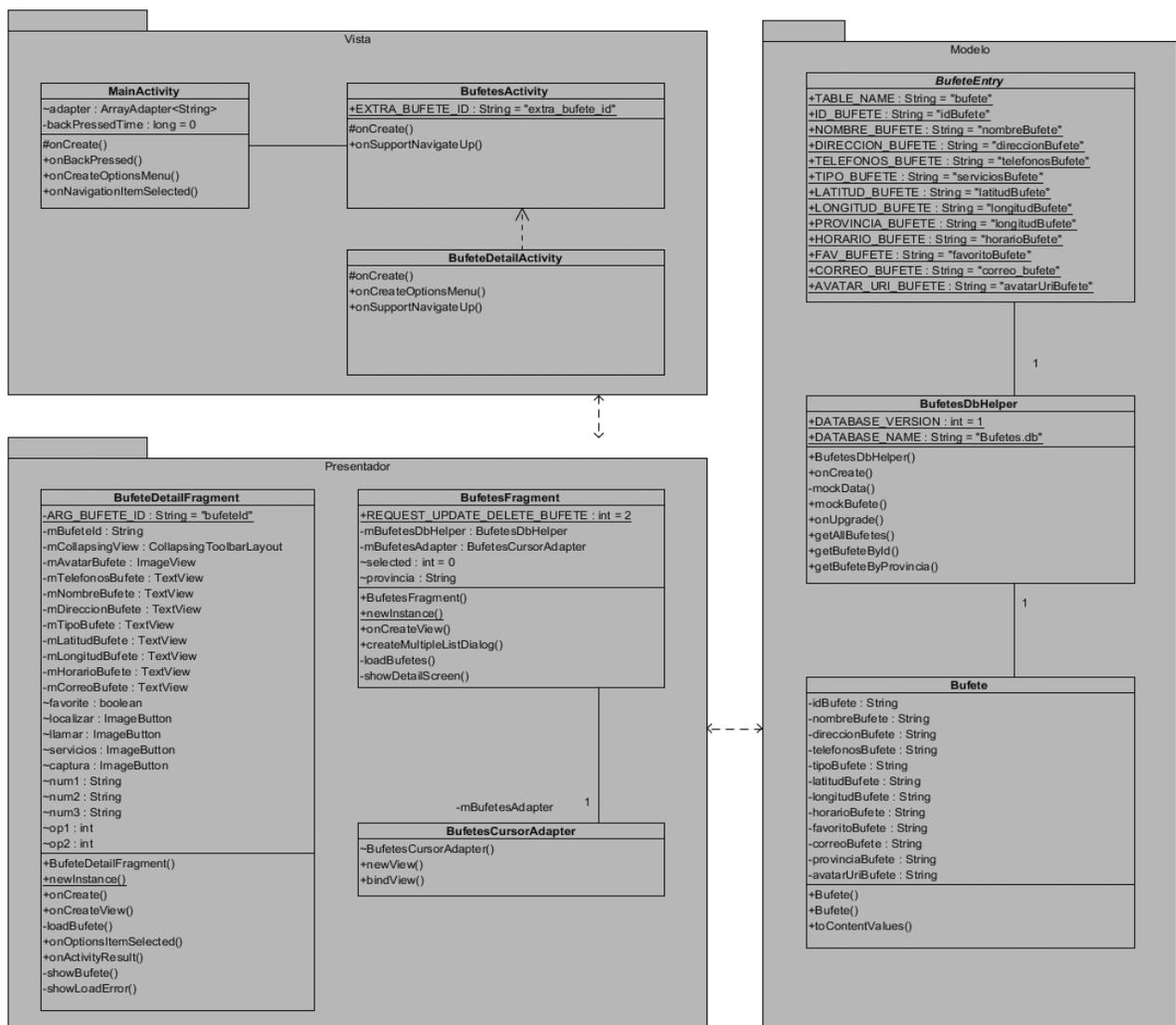


Figura 6: Diagrama de paquetes patrón arquitectónico MVP.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

2.4.2 Patrones de diseño

Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. De manera más simple, un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos (Larman, 2003).

Puede asumirse que un patrón describe una estructura de comunicación de componentes, generalmente recurrente, que resuelve un problema de diseño general dentro de un contexto particular. Por tanto, un patrón de diseño es una solución demostrada, eficiente y reutilizable en diferentes contextos y escenarios de desarrollo de software.

Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns) son un conjunto de patrones de diseño de software que:

Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Larman, 2003).

Los patrones pertenecientes a este conjunto son: Experto, Controlador, Creador, Alta Cohesión y Bajo Acoplamiento. En la siguiente sección se explica el uso de algunos de estos patrones en el diseño de Callejero Jurídico.

Experto

El patrón Experto en Información establece el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto, o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El comportamiento se distribuye entre las clases que cuentan con la información requerida (Larman, 2003). Un ejemplo de este patrón se evidencia en la clase *BufetesDBHelper*, la cual posee la información asociada a cada Bufete Colectivo, por tanto, es la encargada de brindar información de cada Bufete mediante las funcionalidades *getAllBufetes()*, *getBufetesPorId()*, *getBufetesPorProvincia()*, entre otras.

```
//Obtener todos los Bufetes
public Cursor getAllBufetes() {
    return getReadableDatabase()
        .query(
            BufeteEntry.TABLE_NAME,
            null,
            null,
            null,
            null,
            null,
            null);
}

//Obtener Bufetes dado su Id
public Cursor getBufeteById(String bufeteId) {
    Cursor c = getReadableDatabase().query(
        BufeteEntry.TABLE_NAME,
        null,
        BufeteEntry.ID_BUFETE + " LIKE ?",
        new String[]{bufeteId},
        null,
        null,
        null);
    return c;
}
```

Figura 7: Ejemplo patrón GRASP-Experto en la clase *BufetesDbHelper*

Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización (Larman, 2003).

La Figura 7 muestra un ejemplo de la aplicación de este patrón en la clase *BufetesFragment*, perteneciente al paquete `cu.uci.cegel.cj.pbufetes`.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

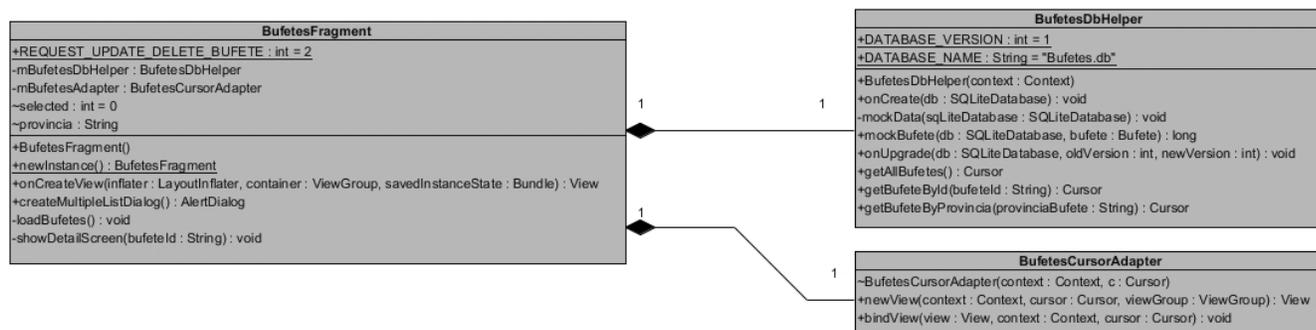


Figura 8: Relación de composición de la clase *BufetesFragment* con las clases *BufetesDbHelper* y la clase *BufetesCursorAdapter*

La clase *BufetesFragment* es la única responsable de la creación de objetos de la clase *BufetesDbHelper* y de la clase *BufetesCursorAdapter* respectivamente. Entre ambas clases se establece una relación de composición con cardinalidad uno a uno.

Alta Cohesión

La alta cohesión indica que los datos y responsabilidades de una entidad están fuertemente ligados a la misma en un sentido lógico. La información que maneja una entidad de software tiene que estar conectada lógicamente con esta, no deben existir entidades con atributos que describan comportamientos que en realidad no le corresponden. Este patrón permite la obtención de clases fácilmente actualizables y reutilizables durante el proceso de desarrollo de software. Según se plantea en (Larman, 2003), se considera que un diseño de clases provee una alta cohesión cuando:

Una clase tiene una responsabilidad moderada en un área funcional y colabora con otras clases para llevar a cabo las tareas (Larman, 2003).

Un diseño con alta cohesión puede observarse en la relación de las clases del paquete *cu.uci.cegel.cj.pbufetes.DatosBufetes*, donde se maneja la capa de acceso a datos locales de los Bufetes Colectivos en Callejero Jurídico. La clase *BufetesContract* contiene la definición estructural de la base de datos local de la aplicación Callejero Jurídico. Colabora con las clases *BufetesDBHelper*, responsable de la creación y actualización del esquema de la base de datos, y *Bufete*, clase que implementa los métodos necesarios para realizar las operaciones sobre los datos locales.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

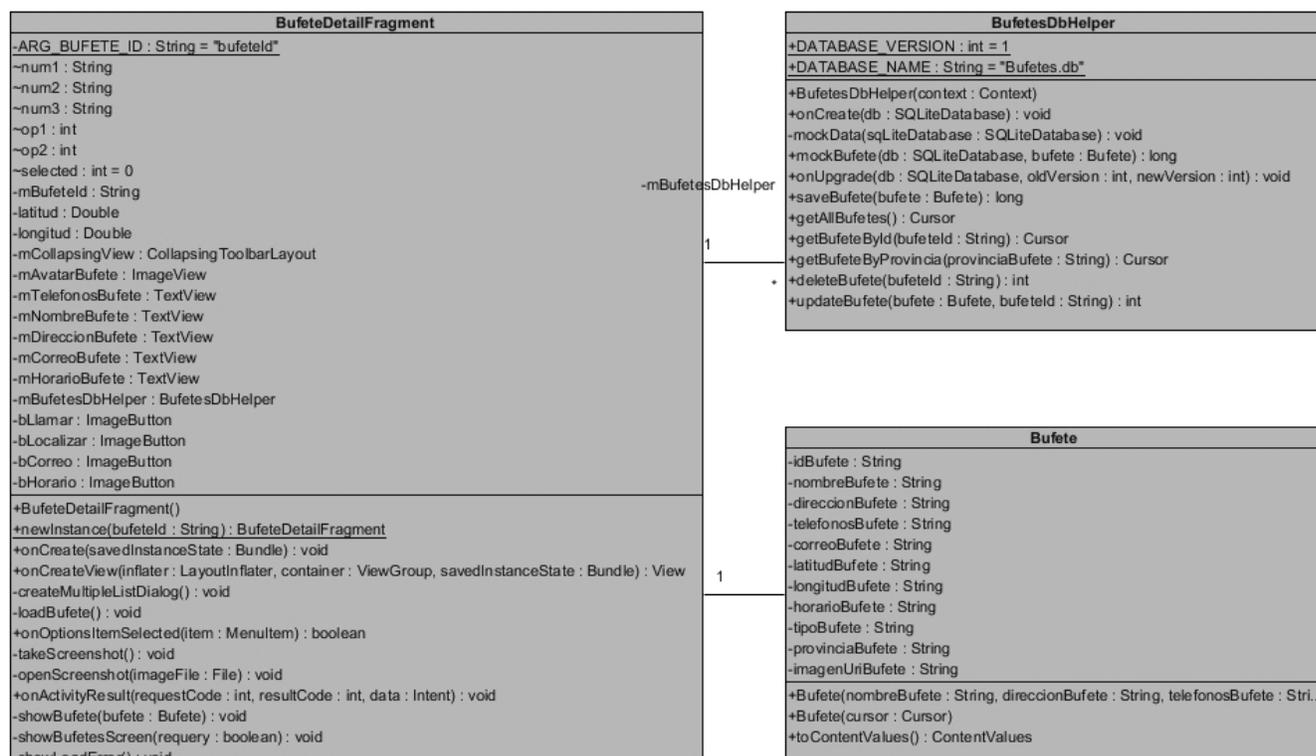


Figura 9: Patrón Alta cohesión en Callejero Jurídico

Patrones Gang Of Four o Pandilla de los Cuatro

Los patrones Gang Of Four (GOF) son un conjunto de patrones de diseño muy útiles durante el diseño de objetos. Estos se clasifican según el propósito para el que han sido definidos como: Patrones creacionales, estructurales y de comportamiento.

Patrón Maestro/Esclavo

El patrón de diseño Maestro/Esclavo es usado cuando se tienen dos o más procesos que necesitan ejecutarse simultánea y continuamente, pero a diferentes velocidades. National Instruments (National Instruments Corporation, 2015) lo describe como un patrón de diseño que consiste en múltiples ciclos paralelos. Cada ciclo puede ejecutar tareas a velocidades distintas. De estos, uno actúa como el maestro y los otros como esclavos. El ciclo maestro controla todos los ciclos esclavos y se comunica con ellos utilizando arquitecturas de mensajería.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

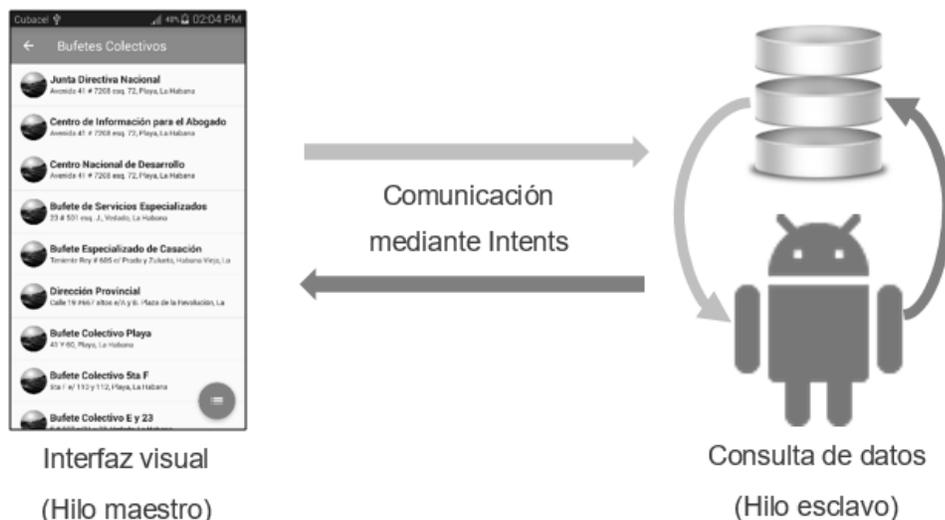


Figura 10: Esquema de aplicación del patrón Hilo-Esclavo en Callejero Jurídico

En el diseño de Callejero Jurídico, existen dos hilos de ejecución que se ejecutan simultáneamente en el proceso principal de la aplicación. Uno de estos hilos es el de la interfaz visual, que maneja toda la lógica de presentación de los datos al usuario. El segundo hilo es el de consulta de los datos, que se encarga de la manipulación de los datos locales. El hilo maestro es el de interfaz visual, ya que este es el que controla la ejecución del hilo de consulta mediante las acciones del usuario. La comunicación entre ambos hilos se establece mediante el uso de *Intents*²².

2.4.3 Tarjetas Clase-Responsabilidad-Colaborador

Scott Ambler (Ambler, 2002), define las tarjetas Clase-Responsabilidad-Colaborador (CRC) como tarjetas de índice estándar que han sido divididas en tres secciones, una que indica el nombre de la clase que la tarjeta representa, un inventario de las responsabilidades de la clase, y la tercera con los nombres de las otras clases con las que ésta colabora para cumplir sus responsabilidades.

En el Manifiesto para el Desarrollo Ágil de Software (Beck, 2001), se recomienda el uso de este artefacto para inventariar las clases que son relevantes para el incremento actual del software. La siguiente tabla muestra una de las tarjetas CRC elaboradas para el modelo de la solución propuesta en el presente trabajo. Las tarjetas CRC restantes se encuentran en el Anexo II Tarjetas CRC.

²² **Intents:** es un objeto de acción que puedes usar para solicitar una acción de otro componente de la aplicación. Aunque los *intents* facilitan la comunicación entre los componentes de muchas maneras, existen tres casos de uso fundamentales: para comenzar una actividad, para iniciar un servicio y para entregar un mensaje.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

Nombre de la clase: BufetesActivity

Responsabilidades de la clase: Se encarga de manejar las operaciones relacionadas con los Bufetes Colectivos. Esta clase sabe qué actividad mostrar para que el usuario pueda ver la lista de Bufetes Colectivos y posteriormente ver en detalle cada entidad de este tipo.

Colaboradores de la clase:

- *MainActivity.*
- *R.*
- *BufetesFragment.*
- *Toolbar.*
- *AppCompatActivity.*
- *Bundle.*

Tabla 5: Tarjeta CRC BufetesActivity

2.5 Modelo de datos

Para el almacenamiento de los datos se utilizó una base de datos *SQLite* con el objetivo de almacenar el contenido en la aplicación de manera persistente, es decir, que no se pierdan los datos cuando cerramos la aplicación. Dicha base de datos se compone por siete tablas que contienen información relacionada con cada tipo de Entidad Jurídica, cada tabla contiene once tuplas: "ID" de tipo string como "llave primaria", "NOMBRE_ENTIDAD", de tipo string, donde se almacena el nombre de Entidad, "TELEFONOS_ENTIDAD", de tipo long, contiene los números telefónicos, "CORREO_ENTIDAD", de tipo string, almacena la dirección de correo electrónico, "DIRECCION_ENTIDAD", de tipo string, contiene la dirección asociada, "HORARIO_ENTIDAD", de tipo string, contiene los horarios de atención de esta Entidad, "LATITUD_ENTIDAD" y "LONGITUD_ENTIDAD", de tipo float, contienen las coordenadas geográficas de la Entidad, "PROVINCIA_ENTIDAD", de tipo string guarda la provincia a la que pertenece la Entidad, "IMAGEN_URI", de tipo string contiene la ruta de la dirección de la imagen perteneciente y "SERVICIOS_ENTIDAD", de tipo string, contiene los diferentes servicios brindados por la Entidad en cuestión. En la Figura 11 se muestra el modelo de datos de la propuesta de solución.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DEL SISTEMA

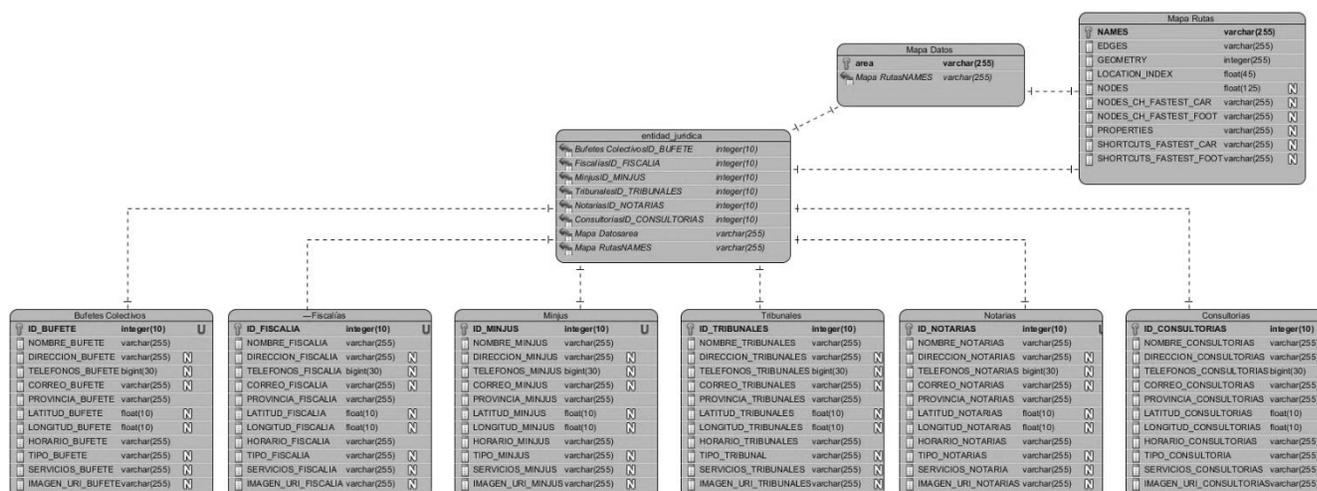


Figura 11: Diagrama Entidad-Relación Callejero Jurídico

En un principio se manejaba la idea de crear la base de datos externa a la aplicación. De esta manera el usuario al instalar la aplicación tendría que buscar la ruta a la base de datos que se encontraría en la memoria interna o externa del dispositivo. Esta idea se mantuvo para los archivos de mapas y se desechó para los datos informativos de las Entidades, pues sería muy fácil para una persona acceder a los datos, modificarlos y luego difundir la aplicación con información falsa. Es por ello que se optó por la opción de introducir la base de datos internamente en la aplicación pues de esta manera se hace más difícil modificar su contenido y luego compartir la aplicación a otras personas con información incorrecta.

2.6 Conclusiones del capítulo

En este capítulo se elaboraron los artefactos correspondientes a la etapa de planeación de la metodología XP, determinando como resultado que el proceso de desarrollo durará un total de 12 semanas, divididas en cuatro iteraciones de 3 semanas cada una, de forma tal que el equipo de desarrollo y el cliente obtendrán un producto funcional al concluir cada iteración, posibilitando a este último plantear su conformidad o no con la aplicación obtenida.

Se especificó el patrón arquitectónico y los patrones de diseño, lo que permitió el diseño de una aplicación que podrá ser mantenida, auditada y actualizada con relativa facilidad, por parte de los desarrolladores en futuras versiones. Se confeccionaron además las tarjetas CRC, las cuales permitieron establecer el marco de relaciones para las clases del diseño. Estos artefactos ayudarán a una mejor comprensión de la manera en que se encuentra estructurada la aplicación, lo cual facilita su análisis y, de ser necesario, su modificación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Capítulo 3: Implementación y pruebas de la solución

3.1 Introducción

En el presente capítulo se detallan los aspectos fundamentales de los procesos de implementación y pruebas de la solución propuesta en el Capítulo 2. Se describen los estándares de codificación utilizados durante la implementación de la aplicación Callejero Jurídico, el diagrama de despliegue de la solución, así como los artefactos generados durante la etapa de pruebas.

3.2 Estándares de codificación

Los estándares de codificación establecen las pautas de programación enfocadas en la estructura y apariencia física del código para facilitar la legibilidad, comprensión y mantenimiento del mismo. Kent Beck (Beck, 2000) recomienda el uso de estándares de codificación, como una práctica de XP, en la que los programadores escriben todo el código de conformidad con normas que enfatizan la comunicación a través del código. Dichos estándares definen la nomenclatura de las variables, objetos, métodos y funciones.

Java CamelCase²³

La notación CamelCase es la práctica de escribir palabras compuestas, o frases, tales que cada palabra o abreviatura comienza con una letra mayúscula y omite guiones. CamelCase puede comenzar con una letra mayúscula, estilo que es también conocido como UpperCamelCase²⁴ y que es usado sobre todo para nombrar clases, y lowerCamelCase²⁵ con una letra minúscula para nombrar variables y métodos. Es un estilo recomendado por varias comunidades y empresas desarrolladoras de Java, tales como Oracle Corporation (Oracle Corporation, 1999); además, este estilo de codificación está integrado directamente en el sistema de auto completamiento de código de Android Studio, lo cual acelera el proceso de escritura del código manteniendo su coherencia y legibilidad.

²³ Estándar de codificación que define el comienzo con una letra minúscula o mayúscula y las siguientes iniciales de palabras con mayúsculas.

²⁴ Estándar de codificación CamelCase en el que todas las palabras comienzan con letra mayúscula.

²⁵ Estándar de codificación CamelCase que comienza con letra minúscula y el resto de las palabras las comienzan con mayúscula.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

3.2.1 Nomenclatura de los comentarios en las funciones

Las funciones deben tener un comentario antes de ser declaradas, explicando el funcionamiento de las mismas.

```
//Funcionalidad para obtener una Fiscalía dado su Id  
public Cursor getFiscaliaById(String fiscaliaId) {
```

Figura 12: Nomenclatura de los comentarios en las funciones

3.2.2 Nomenclatura de las variables en Java

Los nombres de variables emplean el estándar de codificación lowerCamelCase, deben ser cortos y descriptivos. Al leer el nombre el programador debe ser capaz de identificar para que se emplea. Pueden tener solo caracteres alfanuméricos.

```
private String idFiscalia;  
private String nombreFiscalia;  
private String direccionFiscalia;  
private String telefonosFiscalia;
```

Figura 13: Nomenclatura de los nombres de las variables en Java

3.2.3 Nomenclatura de las variables en XML

Los nombres de las variables en los archivos XML se definen en minúsculas todas las letras, comienzan con las iniciales del nombre del objeto seguido por underscore (“_”) y luego por un nombre corto y descriptivo.

```
android:id="@+id/tv_telefonos"
```

Figura 14: Nombres de las variables en XML

3.2.4 Nomenclatura de las variables referentes a la base de datos

Los nombres de las variables se definen en mayúsculas seguido de underscore (“_”) y luego por el nombre de la entidad a la que hace referencia el atributo en mayúscula.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

```
public static final String ID_BUFETE = "idBufete";
public static final String NOMBRE_BUFETE = "nombreBufete";
public static final String DIRECCION_BUFETE = "direccionBufete";
```

Figura 15: Nomenclatura de las variables referentes a la base de datos

3.2.5 Nomenclatura de los métodos

Los nombres de los métodos se definen con el estándar de codificación lowerCamelCase. Pueden tener solo caracteres alfanuméricos.

```
public long insertarFiscalia(SQLiteDatabase db, Fiscalia fiscalia) {
```

Figura 16: Nomenclatura de los métodos

3.2.6 Nomenclatura de las clases

En los nombres de las clases se emplea el estándar de codificación UpperCamelCase.

```
public class FiscaliasActivity extends AppCompatActivity {
```

Figura 17: Nomenclatura de las clases

3.3 Tareas de ingeniería

Las tareas de ingeniería son definidas por Kent Beck (Beck, 2000) como: *algo que el programador sabe que el sistema debe hacer*. Las mismas deben ser estimables dentro de uno a tres días de programación ideales y se derivarán, en su mayoría, de historias de usuario. La siguiente tabla representa la especificación una tarea de ingeniería definida en la primera iteración del desarrollo de la solución propuesta:

Tareas de ingeniería		
Número: 3	H.U: Mostrar Detalles de Entidad Jurídica.	Iteración: 1
Nombre: Implementar la funcionalidad <i>mostrarBufete</i>		
Tipo: Desarrollo	Puntos estimados: 2/5	
Fecha inicio: 21 de marzo, 2018	Fecha fin: 22 de marzo, 2018	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Responsable: Adrian Pérez Zaldivar

Descripción: Se implementará una funcionalidad que permitirá mostrar una interfaz de detalle del Bufete Colectivo seleccionado.

Figura 18: Implementar la funcionalidad mostrarBufete

Letelier y Penadés (Letelier, y otros, 2006), plantean que las tareas de ingeniería expresan todo el trabajo realizado durante una iteración. Aunque cada una de ellas son asignadas a un solo programador, son llevadas a cabo por parejas de programadores. El resto de las tareas de ingeniería definidas para cada iteración del proceso de desarrollo de la solución propuesta se encuentran en el Anexo III Tareas de ingeniería.

3.4 Validación de la solución propuesta

Uno de los pilares de la metodología XP es el proceso de pruebas. XP anima a realizar pruebas constantemente, tanto como sea posible. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones (Corbea, y otros, 2007).

XP divide las pruebas del sistema en dos grupos: pruebas unitarias (encargadas de verificar el código y diseñada por los programadores) y pruebas de aceptación o pruebas funcionales (destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final) (Corbea, y otros, 2007).

3.4.1 Pruebas unitarias

Uno de los métodos utilizados para realizar pruebas de software en la metodología XP son las pruebas unitarias. La base de este método es hacer pruebas en pequeños fragmentos del código de la aplicación, estos fragmentos deben ser unidades estructurales del programa, encargados de una tarea específica. En programación orientada a objetos se puede afirmar que estas unidades son los métodos o las funciones que se tienen definidos. El objetivo de estas pruebas es el aislamiento de partes del código y la demostración de que no contienen errores. Estas no generan artefactos y no son directamente palpables para el cliente (Corbea, y otros, 2007).

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

3.4.1.1 Resultados de las pruebas unitarias

Las pruebas unitarias fueron ejecutadas sistemáticamente, cada vez que se terminaba de implementar cada una de las iteraciones. Dichas pruebas fueron desarrolladas utilizando la herramienta Android JUnit. Esta herramienta permite probar componentes específicos mediante clases de casos de prueba. Estas clases proporcionan métodos auxiliares para la creación de objetos de imitación y métodos que ayudan a controlar el ciclo de vida de una aplicación. A continuación, se muestran los resultados de las pruebas realizadas a las funcionalidades en cada una de las iteraciones:



Figura 19: Resultados de las pruebas unitarias en la iteración 1.

Clase	Funcionalidad	Error	Estado
<i>BufetesDbHelper</i>	<i>getBufeteByProvincia</i>	Tipo de dato incorrecto en la salida del método.	Resuelto
<i>BufeteDetailFragment</i>	<i>deleteBufete</i>	Atributo de entrada del método incorrecto.	Resuelto
<i>MapaActivity</i>	<i>onCreate</i>	Id de <i>layout_mapa.xml</i> incorrecto.	Resuelto
<i>BufetesDbHelper</i>	<i>getBufeteById</i>	Tipo de dato incorrecto en la salida del método.	Resuelto
<i>BufeteDbHelper</i>	<i>onCreate</i>	Mala práctica de la programación. Para modificar una variable dentro del constructor de la propia clase, la variable necesita ser declarada de tipo <i>final</i> .	Pendiente
<i>MapaActivity</i>	Localizar	El método no contaba con los permisos para acceder al <i>GPS</i> del dispositivo.	Resuelto

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Tabla 6: Resumen de los errores de las pruebas unitarias en la Iteración 1.



Figura 20: Resultados de las pruebas unitarias en la iteración 2.

Clase	Funcionalidad	Error	Estado
<i>BufeteDetailFragment</i>	<i>deleteBufete</i>	Atributo de entrada del método incorrecto.	Resuelto
<i>MapaActivity</i>	<i>mostrarDialogRuta</i>	Versión de la clase <i>Dialog</i> no soportada para las versiones de Android definidas en el <i>Gradle</i> .	Resuelto
<i>BufeteDbHelper</i>	<i>onCreate</i>	Mala práctica de la programación. Para modificar una variable dentro del constructor de la propia clase, la variable necesita ser declarada de tipo <i>final</i> .	Resuelto

Tabla 7: Resumen de los errores de las pruebas unitarias en la Iteración 2.



Figura 21: Resultados de las pruebas unitarias en la Iteración 3.

Clase	Funcionalidad	Error	Estado
<i>BufetesDbHelper</i>	<i>getBufeteById</i>	Salida incorrecta del método.	Resuelto

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Tabla 8: Resumen de los errores de las pruebas unitarias en la iteración 3.



Figura 22: Resultados de las pruebas unitarias en la iteración 4.

3.4.2 Pruebas de aceptación

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación.

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones, se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a probar, cuando una historia de usuario ha sido correctamente implementada (Corbea, y otros, 2007).

Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas (Corbea, y otros, 2007).

A continuación, se muestran cuatro de los casos de prueba de aceptación correspondientes a una historia de usuario de cada iteración, el resto se encuentran expuestas en el Anexo IV del presente documento.

Caso de prueba de aceptación	
Código: HU4_PA4	Historia de usuario: Mostrar detalles de una Entidad Jurídica.
Nombre: Mostrar detalles de una Entidad Jurídica.	
Descripción: Se prueba la funcionalidad de mostrar la vista de detalle de una Entidad Jurídica.	
Condiciones de ejecución: El usuario debe encontrarse en la lista de Entidades Jurídicas.	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> • Se selecciona una Entidad Jurídica en el Menú Lateral. • Luego en la lista obtenida, se selecciona la Entidad que desee observar detalladamente.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se muestra una interfaz de detalle de la Entidad seleccionada.
<p>Evaluación: Satisfactoria.</p>

Tabla 9: HU4_PA4 Mostrar detalles de una Entidad Jurídica

Caso de prueba de aceptación	
<p>Código: HU6_PA6</p>	<p>Historia de usuario: Mostrar Entidad Jurídica en el mapa.</p>
<p>Nombre: Mostrar Entidad Jurídica en el mapa.</p>	
<p>Descripción: prueba la funcionalidad de mostrar al usuario la ubicación en el mapa de la entidad jurídica seleccionada.</p>	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario deberá encontrarse en la vista de detalle de la Entidad Jurídica. • El usuario deberá tener cargado inicialmente en la memoria de su dispositivo el fichero de mapas. • El usuario debe seleccionar la opción “Mostrar en mapa”. 	
<p>Entrada/Pasos de ejecución: el usuario selecciona una Entidad Jurídica determinada en el menú de la aplicación, la aplicación consulta la base de datos y muestra el resultado al usuario en una actividad, luego en dicha actividad el usuario selecciona en un listado, una entidad jurídica para ver su detalle, la aplicación consulta la base de datos y muestra el resultado en un Fragmento, posteriormente el usuario selecciona la opción “Mostrar en mapa”, la aplicación consulta la base de datos y muestra en una actividad el resultado.</p>	
<p>Resultado esperado: una interfaz con las siguientes características:</p> <ul style="list-style-type: none"> • Una interfaz con un mapa y un marcador indicando la posición geográfica de la Entidad Jurídica seleccionada. 	
<p>Evaluación: Satisfactoria.</p>	

Tabla 10: HU6_PA6 Mostrar Entidad Jurídica en el mapa.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Caso de prueba de aceptación	
Código: HU13_PA13	Historia de usuario: Obtener ubicación mediante GPS.
Nombre: Obtener la ubicación geográfica del dispositivo mediante GPS.	
Descripción: prueba la funcionalidad de mostrar al usuario la ubicación geográfica en la que se encuentra mediante el GPS del dispositivo.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario deberá haber seleccionado la opción Mostrar Mapa en el Menú Inicial.• El usuario deberá tener cargado inicialmente en la memoria de su dispositivo el fichero de mapas.• El usuario deberá tener activado el GPS o Ubicación del dispositivo.• El usuario deberá seleccionar la opción “Obtener ubicación”.	
Entrada/Pasos de ejecución: el usuario selecciona Mostrar Mapa, la aplicación comprueba el fichero de mapas y muestra el resultado al usuario en una actividad, luego el usuario selecciona la opción “Obtener ubicación”, la aplicación muestra el resultado al usuario.	
Resultado esperado: una interfaz con las siguientes características: <ul style="list-style-type: none">• Un mapa con un marcador indicando la posición geográfica en la que se encuentra el usuario.	
Evaluación: Satisfactoria.	

Tabla 11: HU13_PA13 Obtener la ubicación geográfica del dispositivo mediante GPS.

Caso de prueba de aceptación	
Código: HU14_PA14	Historia de usuario: Trazar rutas en el mapa.
Nombre: Trazar rutas entre dos puntos en el mapa.	
Descripción: prueba la funcionalidad de dibujar una ruta en el mapa.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario deberá encontrarse en la interfaz del mapa.• El usuario deberá seleccionar el punto de origen.• El usuario deberá seleccionar el punto de destino.	
Entrada/Pasos de ejecución: el usuario selecciona la Ver Mapa o Mostrar Entidad Jurídica en el mapa, la aplicación muestra una actividad con un mapa, el usuario selecciona los puntos de	

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

origen y destino que desee, la aplicación muestra la ruta más rápida para llegar desde el punto de origen hasta el punto de destino.

Resultado esperado:

- Se dibuja una ruta en el mapa desde el punto de origen hasta el punto de destino.

Evaluación: Satisfactoria.

Tabla 12: HU14-PA14 Trazar rutas entre dos puntos en el mapa.

3.4.2.1 Resultados de las pruebas de aceptación

Las pruebas aplicadas contribuyeron a mejorar la calidad y el funcionamiento del sistema, detectándose en la primera iteración un total de seis no conformidades significativas, las cuales fueron resueltas y cuatro recomendaciones. En la segunda iteración se detectaron tres no conformidades las cuales fueron resueltas y dos recomendaciones. En la tercera iteración no se detectaron no conformidades, sin embargo, se encontró una recomendación, ver Figura 20.

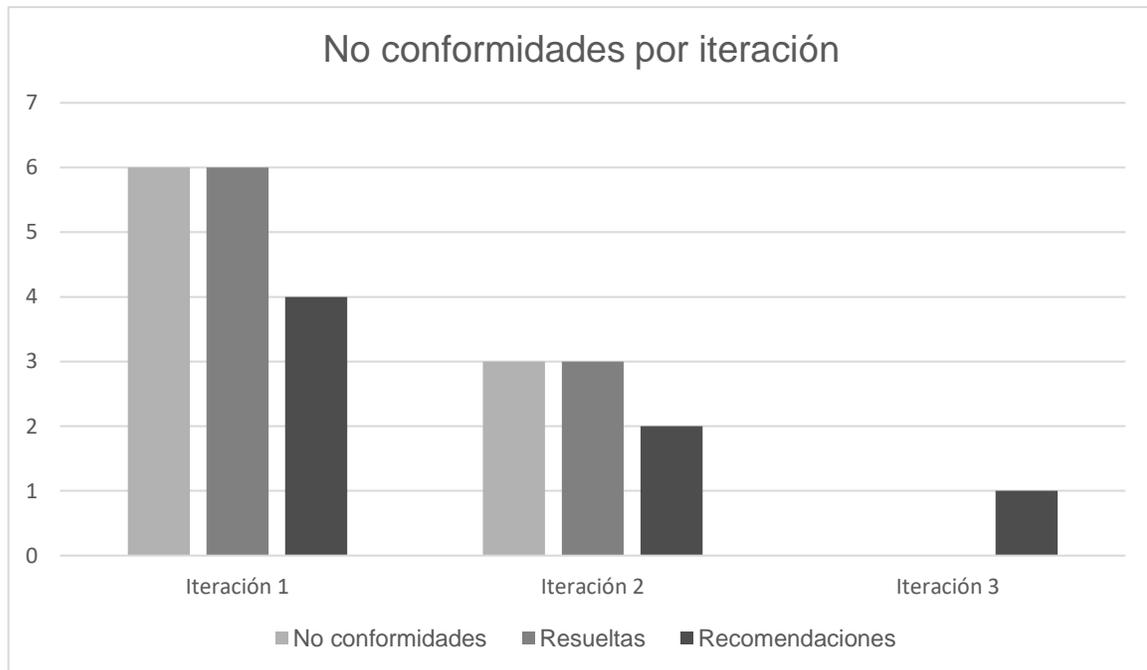


Figura 23: Resultados de las pruebas de aceptación en cada iteración.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

3.4.3 Pruebas de rendimiento

De acuerdo a las características de la aplicación se decide realizar pruebas de rendimiento para comprobar el desempeño de la solución bajo distintos entornos de trabajo. Para llevar a cabo estas pruebas se utilizaron los siguientes dispositivos móviles:

Dispositivo	Versión de Android	CPU	RAM	Pantalla
Alcatel One Touch 4030A	4.1.1	1 core @ 1.0 GHz	512 MB	3.5", 165 dpi
Samsung Galaxy S4 i337	4.4.2	4 cores @ 1.9 GHz	2 GB	4.7", 420 dpi
Samsung Galaxy S5 G900A	6.0.1	4 cores @ 2.4 GHz	2 GB	5.2", 480 dpi
Huawei Ascend XT2 H1711	6.0.1	8 cores @ 1.4 GHz	2 GB	5.5", 420 dpi
BLU Studio 5.0 II	4.2.2	2 cores @ 1.3 GHz	512 MB	5.0", 220 dpi
Samsung Galaxy J3 2016	5.1.1	4 cores @ 1.5 GHz	1.5 GB	5.0", 370 dpi
Samsung Galaxy Tab SIII	4.1.2	2 cores @ 1.2 GHz	2 GB	7.0", 170 dpi

Tabla 13: Muestra de dispositivos escogidos para las pruebas de rendimiento.

En la comparación se tuvo en cuenta el uso de los siguientes recursos: tiempo de uso, batería, microprocesador, memoria RAM, GPS.

3.4.3.1 Resultados de las pruebas de rendimiento

En esta etapa de las pruebas se pudo analizar el comportamiento de los dispositivos móviles en la ejecución de la aplicación. A continuación, se muestran los resultados obtenidos:

	Alcatel One Touch	Samsung Galaxy S4	Samsung Galaxy S5	Huawei Ascend XT2	BLU Studio 5.0 II	Samsung Galaxy J3 2016	Samsung Galaxy Tab SIII
Tiempo de uso	35 minutos	30 minutos	1 hora y 12 minutos	17 minutos	40 minutos	11 minutos	1 hora 17 minutos
Consumo de batería (%)	12%	9%	7%	3%	11%	6%	22%

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN

Microprocesador (%)	19,13	2,33	0,98	2,98	9,35	6,12	11,13
Memoria RAM (en MB)	49,33	19,90	11,21	21,12	40,84	23,32	42,45
GPS	Sí	Sí	No	No	No	Sí	Sí

Tabla 14: Resultados de las pruebas de rendimiento.

Después de hacer pruebas de rendimiento en los dispositivos tomados como muestra, se comprobó que la solución es factible para dispositivos tanto de altas prestaciones, como para dispositivos con pocos recursos de hardware y con un software desactualizado. Se comprobó que los elementos en la pantalla se distribuyen de manera uniforme, independientemente del tamaño y la resolución de pantalla del dispositivo usado. El consumo de memoria RAM, CPU y batería es el adecuado para una aplicación de este tipo, ya que el servicio de ubicación o más conocido como GPS, se caracteriza por ser un alto consumidor de recursos de hardware.

3.5 Conclusiones del Capítulo

En el presente capítulo se expuso el proceso de implementación de la solución propuesta, el cual dio lugar a un producto que cumple con las pautas definidas en el modelo descrito en el Capítulo 2. Se describieron las pruebas diseñadas para validar la implementación de Callejero Jurídico, que sirvieron como elemento clave para validar que la solución cumple con las expectativas del cliente.

CONCLUSIONES

Conclusiones generales

El desarrollo del presente trabajo de diploma permitió que se llevaran a cabo todos los objetivos específicos, a fin de dar cumplimiento al objetivo general, logrando arribar a las siguientes conclusiones:

- La elaboración del marco teórico de la investigación a través de la definición de conceptos fundamentales para su comprensión y el estudio del estado del arte realizado, de aplicaciones informativas de entidades que brindan servicios jurídicos, permitió al equipo de desarrollo establecer un punto de partida de la posición actual del mundo respecto a la gestión de información desde dispositivos móviles. Esto permitió validar la premisa inicial de la presente investigación.
- Las etapas de planificación y diseño, definidas por la metodología de desarrollo de software seleccionada, facilitaron el cumplimiento del plan de entrega predefinido con el cliente, permitiendo el desarrollo de la aplicación por iteraciones y la integración de un producto final completamente funcional.
- La implementación de la aplicación Callejero Jurídico, luego de ser validada a partir de las pruebas de rendimiento y las definidas por la metodología XP, permitió el acceso a la información de las Entidades Jurídicas cubanas en los dispositivos móviles con SO Android.

RECOMENDACIONES

Recomendaciones

Luego de la experiencia adquirida durante la realización de la investigación, se recomienda:

- Incorporar a la aplicación la lista de abogados pertenecientes a cada Bufete Colectivo, con su información de contacto, para propiciar una mayor facilidad por parte de la población a la hora de realizar trámites jurídicos.
- Realizar una versión de la aplicación compatible con el sistema operativo para dispositivos móviles iOS, con el objetivo de que el producto llegue a un mayor número de personas.
- Implementar un servicio de evaluación del desempeño del personal en una Entidad Jurídica, ya sea cualitativamente o cuantitativamente, para a la hora de elegir a una Entidad Jurídica, escoger la mejor evaluada.

BIBLIOGRAFÍA

Bibliografía

Ambler, . 2005. Ambysoft Inc. [Online] Vol 14, 2005. [Cited: 11 2 2018.] <http://www.agilealliance.hu/materials/books/SWA-AUP.pdf>.

Android Developers. 2018. Conoce Android Studio | Android Studio. [Online] 2018. [Cited: 2018 3 16.] <https://developer.android.com/studio/intro/index.html>.

Apache Maven. 2018. Apache Maven Project. [En línea] 2018. [Citado el: 23 de 2 de 2018.] <https://maven.apache.org/>.

BBVA. 2016. BBVA Open4U. *Qué es una API y qué puede hacer por mi negocio.* [En línea] 2016. [Citado el: 25 de 1 de 2018.] <https://bbvaopen4u.com/es/actualidad/que-es-una-api-y-que-puede-hacer-por-mi-negocio>.

Cadavid, Andrés Navarro, Martínez, Juan Daniel Fernández y Vélez, Jonathan Morales. 2013. *Revisión de metodologías ágiles para el desarrollo de software.* 2013. Vol 2.

Cambridge University Press. 2018. Cambridge Advanced Learner's Dictionary & Thesaurus. [Online] 2018. [Cited: 2 2 2018.] <https://dictionary.cambridge.org/es/diccionario/ingles/offline>.

Cataldi, Zulma y Lage, Fernando. 2013. *Entornos personalizados de aprendizaje (EPA) para dispositivos móviles: situaciones de aprendizaje y evaluación.* s.l. : EDMETIC, 2013. págs. 117-148. Vols. vol. 2, no 1.

Developers, . 2013. Android Developers. [Online] 2013. [Cited: 2018 3 10.] <http://developer.android.com/sdk/ndk/index>.

Diéguez, Diego Sánchez. 2015. *Patrón Modelo-Vista-Presentador (MVP)* . Sevilla, España : Moleqla, 2015. ISSN 2173-0903 .

ETECSA. 2017. Cubadebate. [En línea] mayo de 2017. [Citado el: 16 de 1 de 2018.] <http://www.cubadebate.cu/especiales/2017/05/11/etecsa-tiene-la-palabra-infografia-y-video/>.

Figuroa, Roberth, Solís, Camilo y Cabrera, Armando. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] [Citado el: 26 de 2 de 2018.] <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>.

Fowler, Martin. 2004. *UML distilled: a brief guide to the standard object modeling language.* s.l. : Addison-Wesley Professional, 2004.

Gartner Inc. 2016. Gartner Inc. [Online] 2016. [Cited: 26 6 2018.] <http://www.gartner.com/newsroom/id/2823619>.

GlosarioIT. 2018. GlosarioIT. Glosario Informático. [Online] 2018. [Cited: 1 2 2018.] <http://www.glosarioit.com/#!IDE>.

Gosling, J., y otros. 2014. *The Java Language Specification, Java SE 8 Edition.* s.l. : Java Series, 2014.

Gradle. 2018. Gradle Build Tool. [Online] 2018. [Cited: 1 2 2018.] <https://gradle.org/>.

BIBLIOGRAFÍA

JetBrains. 2018. IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains. [Online] 2018. [Cited: 22 3 2018.] <https://www.jetbrains.com/idea/>.

Larman, Craig. 2003. *UML y Patrones*. Madrid : Pearson Education, 2003.

Letelier, and Penadés Ma., . 2006. *Métodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP)*. Buenos Aires : Técnica Administrativa, 2006. ISSN: 1666 1680.

Letelier, Patricio y Penadés, Ma. Carmen. 2006. [En línea] 2006. [Citado el: 14 de 2 de 2018.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.

Letham, . 2001. *GPS fácil. Uso del sistema de posicionamiento global*. s.l. : Editorial Paidotribo, 2001. 978-84-8019-591-1.

Metodología de Desarrollo para la Actividad Productiva de la UCI. Universidad de las Ciencias Informáticas. La Habana : s.n.

Ministerio de Justicia de Marruecos. 2017. Aplicaciones en Google Play. [Online] 3 2 2017. [Cited: 28 3 2018.] <https://play.google.com/store/apps/details?id=com.justice.service>.

Nightingale, . 2015. Universidad de California. *UCR Today*. [Online] 3 Diciembre 2015. [Cited: 12 2 2018.] <https://ucrtoday.ucr.edu/33508>.

Notarios y Conservadores. 2018. Asociación de Notarios, Conservadores y Archiveros Judiciales de Chile. [Online] 2018. [Cited: 23 3 2018.] <http://www.notariosyconservadores.cl/contenidos/noticias/0q-aplicacion-que-permite-sacar-numero-de-atencion-en-notarias>.

Oracle. 2018. Oracle Technology Network. [Online] 2018. [Cited: 13 3 2018.] <http://www.oracle.com/technetwork/java/javase/jdk-8-readme-2095712.html>.

—. **2018.** Oracle: Integrated Cloud Applications and Platform Services. [En línea] 2018. [Citado el: 12 de 1 de 2018.] <https://www.oracle.com/index.html>.

Organización Mundial de Comercio. 1998. Consejo del Comercio de Servicios. [En línea] 1998. [Citado el: 15 de marzo de 2018.] https://www.wto.org/spanish/tratop_s/serv_s/w43.doc.

Owens, Mike y Allen, Grant. 2010. *SQLite*. s.l. : Apress LP, 2010.

Palacio, Juan. 2014. *Gestión de proyectos Scrum Manager*. s.l. : Scrum Manager, 2014. págs. 29-34.

Plasencia Prado, Carlos Eduardo. 2017. DevCode. *¿Qué es y por qué aprender SQL?* [En línea] 2017. [Citado el: 26 de 2 de 2018.] <https://devcode.la/blog/que-es-sql/>.

Poder Judicial. 2018. Poder Judicial Costa Rica. [Online] 2018. [Cited: 21 3 2018.] <http://www.poderjudicialapp.com/>.

Pressman, Roger S. 2010. *Ingeniería de Software. Un enfoque práctico*. 7ma Edición. s.l. : McGraw-Hill Interamericana de España S.L., 2010. 978-607-15-0314-5.

BIBLIOGRAFÍA

RAE. 2018. [Online] 2018. [Cited: 2018 2 20.] <http://dle.rae.es/srv/fetch?id=Y2AFX5s#5JCEafg>.

Rahimian, V. y Ramsin, R. 2008. *Designing an agile methodology for mobile software development: A hybrid method engineering approach.* IEEE Xplore. 2008. págs. 337-342.

Rama Judicial de Puerto Rico. 2014. La Rama Judicial de Puerto Rico. [Online] 2014. [Cited: 21 3 2018.] <http://www.ramajudicial.pr/Prensa/comunicados/2014/04-12-14b.htm>.

Rodríguez T., Eduardo A. 2011. *Importancia de las pruebas de software.* s.l. : Cinvestav-Tamaulipas, 2011.

Schwaber, Ken. 1997. Scrum development process. *Business object design and implementation.* s.l. : Springer, 1997.

Schwaber, Ken y Beedle, Mike. 2002. *Agile software development with Scrum.* s.l. : Upper Saddle River: Prentice Hall, 2002.

Scrum BoK. 2016. ScrumManager. [Online] 2016. [Cited: 12 3 2018.] <http://www.scrummanager.net/bok/index.php?title=Artefactos>.

Sebesta, Robert W. 1993. *Concepts of programming languages.* s.l. : Pearson Education India, 1993.

Sommerville, Ian. 2005. *Ingeniería de Software.* s.l. : Pearson Education, 2005.

Sundara, Oscar. 2013. *Visual Paradigm for Uml.* s.l. : International Book Market Service Limited, 2013. 978-613-9-16653-4.

Universidad Politécnica de Madrid. 2013. *Patrones del "Grang of Four".* España - Madrid : s.n., 2013.

Vique, Robert Ramírez. 2012. *Métodos para el desarrollo de aplicaciones móviles.* Barcelona : s.n., 2012.

Visconti, Marcello y Astudillo, Hernán. 2012. *Fundamentos de Ingeniería de Software.* 2012.