

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
ESCUELA DE SISTEMAS Y COMPUTACIÓN**



**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN SISTEMAS Y COM PUTACIÓN**

**TEMA**

**“IMPLEMENTACIÓN DEL PROTOTIPO DE UNA APLICACIÓN MÓVIL DE  
CONSULTA BIBLIOGRÁFICA PARA EL SISTEMA OPERATIVO IOS, DIRIGIDO  
A ESTUDIANTES DE LA PUCE.”**

**AUTORES**

**DANIEL SEBASTIÁN DÁVALOS GONZÁLEZ  
ESTÉFANO FABRICIO VALDEZ PACHARD**

**QUITO, 2016**

## **AGRADECIMIENTOS**

Todo el agradecimiento y dedicatoria de este trabajo va dirigido a mi familia y a mis amigos. A mis papás Guillermo y Marina por su inmenso cariño y apoyo incondicional. A mis hermanos/as Ricardo, Javier y Juli por ser un gran ejemplo en mi vida y a mi sobrino Samuel por rodearme siempre de buena energía. A mi amigo y compañero de tesis, Estéfano Valdez, por el gran trabajo en equipo que hizo posible el desarrollo de esta disertación. A todos mis amigos del colegio y la universidad, por ser parte de mi vida y de mi aprendizaje permanente.

**Sebastián Dávalos**

Primeramente, quisiera agradecer a mis padres Fabricio y Darcy y a mi hermana Raisa, quienes me apoyaron durante todo este proceso, brindándome la fuerza, apoyo y cariño necesario para llegar a la meta. A cada uno de los miembros de mi familia, mis abuelitos Flor, Yoli y Luis que han estado siempre en los momentos más importantes de mi vida. A mis amigos y compañeros por traer las alegrías y distracciones necesarias durante los momentos más duros de mis estudios. A mis compañeros de trabajo que con su experiencia, conocimiento y comprensión que me brindaron en cada momento que lo necesite. Y finalmente quisiera agradecer a Andrés y Sebastián con quienes recorrí este camino y con quienes forme una sincera amistad.

**Estéfano Valdez**

Finalmente, queremos agradecer a nuestro director Andrés Jiménez y a nuestras lectoras Susana Masapanta y Beatriz Campos por compartir su conocimiento con nosotros y guiarnos a lo largo del desarrollo de este trabajo.

## **RESUMEN**

En la presente disertación de grado se expone el desarrollo del prototipo de una aplicación móvil de consulta bibliográfica para el sistema operativo iOS. El objetivo de la aplicación es concentrar varios recursos que pone a disposición la PUCE para que los estudiantes puedan realizar sus tareas y trabajos universitarios, tales como el portal de Biblioteca General de la PUCE y las bibliotecas virtuales SpringerLink y ProQuest, de tal manera que se simplifique la búsqueda de documentos y la generación de citas bibliográficas en formatos estándar.

En este trabajo se explora el concepto de desarrollo de aplicaciones multiplataforma por medio de la herramienta Titanium, utilizando el lenguaje JavaScript del lado del cliente y se emplea la técnica web scraping para agrupar los resultados de los motores de búsqueda de varios portales web de consulta bibliográfica en un solo dispositivo. El modelo de computación en la nube seleccionado para el despliegue de la aplicación es plataforma como servicio, utilizando el lenguaje de programación Java del lado del servidor y un servidor de aplicaciones Tomcat brindado por OpenShift. Finalmente, la metodología que se emplea para el desarrollo de la aplicación es Kanban aplicado al software.

## TABLA DE CONTENIDOS

JUSTIFICACIÓN.....	1
CAPÍTULO 1 .....	2
ENTORNO DE APLICACIONES MÓVILES DE CONSULTA BIBLIOGRÁFICA...2	
Contexto del desarrollo de aplicaciones móviles .....	2
Principios generales de consulta bibliográfica .....	4
Principales librerías virtuales.....	4
Estándares internacionales de referencia bibliográfica .....	7
Principales aplicaciones móviles de consulta bibliográfica .....	9
CAPÍTULO 2 .....	11
ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	11
Metodología de desarrollo.....	11
Herramientas de desarrollo.....	15
Análisis de requerimientos .....	22
Diseño arquitectónico.....	23
Diseño detallado .....	24
Interfaz de usuario .....	25
Plan de pruebas de la aplicación.....	31
CAPÍTULO 3 .....	33
IMPLEMENTACIÓN DEL PROTOTIPO DE LA APLICACIÓN.....	33
Estándares de desarrollo .....	33
Integración de la librería SpringerLink .....	37
Integración de la librería eBrary.....	41
Integración de la Biblioteca General de la PUCE .....	44
Funcionalidades desarrolladas en la aplicación.....	48

CAPÍTULO 4 .....	56
PRUEBAS FUNCIONALES DE LA APLICACIÓN.....	56
Caso de prueba (C1) .....	56
Caso de prueba (C2) .....	59
Caso de prueba (C3) .....	62
Caso de prueba (C4) .....	65
Caso de prueba (C5) .....	68
Caso de prueba (C6) .....	71
CAPÍTULO 5 .....	73
CONCLUSIONES Y RECOMENDACIONES .....	73
Conclusiones.....	73
Recomendaciones .....	75
BIBLIOGRAFÍA .....	77
ANEXOS .....	80

## JUSTIFICACIÓN

Actualmente los estudiantes de la PUCE<sup>1</sup> no disponen de una herramienta que integre las diferentes fuentes de consulta bibliográfica que pone a disposición la universidad para realizar sus tareas y trabajos universitarios. Dichas fuentes poseen motores de búsqueda restringidos a los documentos propios de sus bases de datos, tales como el buscador de libros de la biblioteca de la universidad, la página de búsqueda de las bibliotecas o hemerotecas virtuales que tienen convenio con la universidad, como es el caso de SpringerLink o EBSCO, entre otras. Por esta razón, un estudiante debe recurrir a varios sitios para poder obtener el material bibliográfico necesario, siendo así una pérdida de tiempo para el estudiante al generar referencias de manera manual y un desperdicio de los recursos que pone a disposición la universidad para realizar consultas. Además, ninguna de estas plataformas se encuentra optimizada para dispositivos móviles, lo que dificulta su uso cuando un estudiante no tiene un computador a su alcance.

La necesidad de recibir información de manera más simple y rápida a través de aplicaciones móviles aumenta cada vez más en el entorno universitario, gracias a esto el uso de los dispositivos móviles se ha expandido y se ha convertido en una herramienta muy útil en el ámbito académico. Por lo tanto, se espera que el desarrollo de este prototipo para consultas de referencias bibliográficas incentive el uso de citas en formatos estándar para la presentación de los distintos trabajos estudiantiles, utilizando los recursos que forman parte de las bibliotecas virtuales y física de la PUCE y adicionalmente abra las puertas para el desarrollo de nuevas aplicaciones móviles académicas.

---

<sup>1</sup> PUCE. Pontificia Universidad Católica del Ecuador.

## CAPÍTULO 1

### ENTORNO DE APLICACIONES MÓVILES DE CONSULTA BIBLIOGRÁFICA

En el presente capítulo se va a analizar el entorno de aplicaciones móviles de consulta bibliográfica, examinando el contexto del desarrollo de aplicaciones móviles y determinando cuáles son los principios generales de consulta. Además se va a exponer las principales librerías virtuales que están a disposición para la comunidad académica, los estándares que serán utilizados para generar referencias bibliográficas y cuáles han sido las principales aplicaciones de consulta bibliográfica desarrolladas para el sistema iOS.

#### Contexto del desarrollo de aplicaciones móviles

En los últimos años el desarrollo de aplicaciones móviles ha crecido con gran rapidez, el uso de los teléfonos inteligentes es ahora tan común que ha llegado a convertirse en una parte necesaria de nuestra vida cotidiana, tanto personal como profesional. Según la consultora Gartner, el número de descargas de aplicaciones móviles ha crecido en un 181% en los últimos 3 años y se espera que para el 2017 la cantidad de descargas llegue a 268 billones<sup>2</sup>.

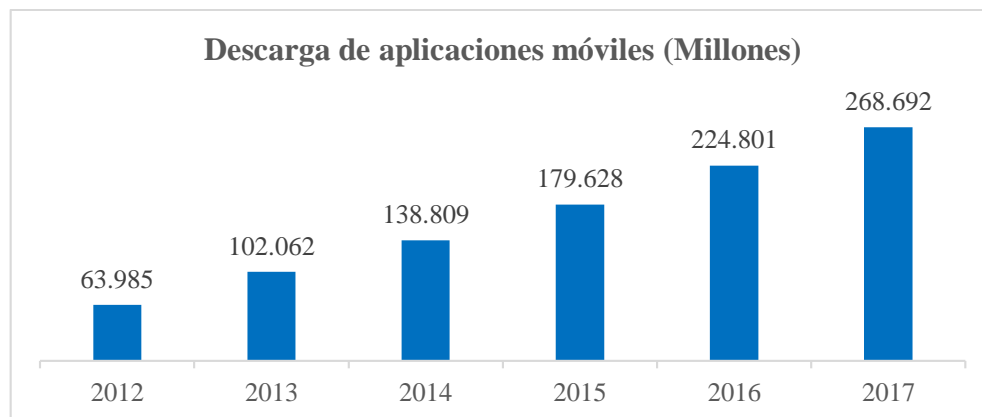


Figura 1: Número de descargas de aplicaciones móviles por año.

Fuente: Gartner, 2013.

---

<sup>2</sup> Gartner, 2013.

Dentro de la industria de dispositivos móviles, una de las empresas líder en el mercado es Apple, que ha revolucionado el consumo de teléfonos inteligentes desde el lanzamiento del iPhone en enero del 2007<sup>3</sup>, donde incluyó por primera vez control multitáctil, navegadores web modernos, mapas, entre otras funcionalidades nunca antes vistas en este tipo de dispositivos, siendo catalogado por Steve Jobs como “un producto mágico y revolucionario que está adelantado 5 años a cualquier otro teléfono móvil”. Desde entonces el sistema operativo iOS, desarrollado por esta empresa, ha sido uno de los más utilizados a nivel mundial, gracias a la experiencia e interfaz que brinda al usuario. En la actualidad este sistema operativo cuenta con una participación de mercado de aproximadamente 18,3%, siendo solo superada por Android, el sistema operativo líder en el mercado con una participación del 78%<sup>4</sup>.

Apple gestiona las aplicaciones móviles para los usuarios por medio de su tienda virtual AppStore. En la actualidad esta tienda cuenta con más de 1,4 millones de aplicaciones, de las cuales aproximadamente 20.000 son educativas<sup>5</sup>. La empresa estadounidense ha creado la plataforma educativa iTunes U, que es utilizada por cerca de 1000 universidades. Estas instituciones obtienen varios beneficios con el uso de iTunes U, entre los que destacan: cursos gratis de las mejores instituciones educativas, videos, libros, conferencias, entre otros recursos<sup>6</sup>.

A pesar de que existen varias alternativas de aplicaciones móviles (en tiendas como App Store o Google Play) para realizar consultas bibliográficas, uno de los principales problemas que enfrenta un estudiante universitario es que no tiene a su alcance una fuente que concentre varios de estos sitios y le permita generar citas para sus trabajos sin importar de qué biblioteca proviene el texto del cual ha tomado información.

Finalmente, es importante señalar que en el ámbito del entretenimiento las aplicaciones sociales se han encargado de llenar el deseo de recibir y compartir información personal; esto ha creado un espacio para el desarrollo de aplicaciones móviles para compartir información

---

<sup>3</sup> Apple, 2007.

<sup>4</sup> IDC, 2015.

<sup>5</sup> Apple, 2009.

<sup>6</sup> Apple, 2015.

académica y profesional, siendo este un ámbito que todavía no se ha explotado al máximo y del cual se pueden obtener muchos beneficios tanto para estudiantes como para las instituciones educativas y profesionales.

### **Principios generales de consulta bibliográfica**

Dentro del contexto académico, una bibliografía es el conjunto de documentos consultados durante la presentación de un trabajo. El uso de bibliografía aumenta la credibilidad del trabajo escrito y además protege la propiedad intelectual del autor original. La Ley de Propiedad Intelectual en Ecuador, publicada en el Registro Oficial 320, exige que se otorgue al autor original el derecho de ser reconocido como el titular de su creación y por consiguiente ser el beneficiario del mismo<sup>7</sup>, dentro de esta ley se establece que el derecho de autor es el encargado de proteger los derechos de los creadores sobre las obras, ya sean literarias o artísticas. En la práctica se deben incluir los documentos consultados en cualquier trabajo, incluso cuando no han sido citados textualmente.

Se pueden citar varios tipos de documentos entre los cuales destacan: libros, libros electrónicos, artículos de revista, artículos de revista electrónica, actas de congreso completas, sitios web, blogs, correos electrónicos, redes sociales, documentos gráficos, cartografía, bases de datos, entre otros. Para el desarrollo de la aplicación móvil en este trabajo se van a usar los formatos para libros y artículos de revista.

### **Principales librerías virtuales**

El desarrollo y la implementación de tecnologías de la información han hecho posible que instituciones educativas y librerías presenten sus documentos en espacios virtuales, proveyendo a la comunidad acceso a sus recursos de manera más simple y aumentando el alcance de personas a las que llega su contenido. Hasta ahora, las universidades han sido las pioneras en presentar sus recursos de manera digital, un ejemplo de esto es la universidad de

---

<sup>7</sup> Propiedad Intelectual del Ecuador, 2015.

Harvard que implementó un portal web de su biblioteca en junio del 2003<sup>8</sup>, actualmente cuenta con un buscador avanzado llamado HOOLIS+<sup>9</sup>, en donde la comunidad puede encontrar material bibliográfico digital de la librería de Harvard y de otros repositorios virtuales importantes como EBSCO<sup>10</sup> para realizar sus consultas y trabajos.

De esta manera, varias instituciones están concentrando sus recursos en grandes repositorios de datos, en donde se puedan consultar publicaciones periódicas académicas y científicas en artículos completos o en resúmenes. Dichos repositorios abarcan temas multidisciplinarios como: medicina, psicología, negocios, tecnología, humanidades, matemática, entre otros. Entre las bibliotecas virtuales destacan: EBSCOhost, EOLSS, Hinari, Dialnet y JSTORE.

Actualmente, la PUCE ofrece varias alternativas a los estudiantes para realizar consultas y citar textos en sus trabajos, en total en su portal web hay más de 20 librerías y hemerotecas virtuales<sup>11</sup>. Dentro de las principales librerías que pone a disposición la PUCE se encuentran: Biblioteca general de la PUCE, SpringerLink, eBrary (Pro Quest) y Digitalia. A continuación presentamos datos generales referentes a cada una de estas instituciones:

#### Biblioteca General de la PUCE:

Esta biblioteca está constituida como una unidad de información que apoya a la docencia, investigación y consultas bibliográficas de la comunidad universitaria de la PUCE y del público en general. Su misión es: “ofrecer a la Comunidad Universitaria y más usuarios, información procesada, actualizada y oportuna, a fin de satisfacer, con calidad, las necesidades de consulta, investigación y enseñanza-aprendizaje” y su visión está definida como: “Mantener optimizado el servicio de información de manera técnica, continua y oportuna. Enmarcarse en

---

<sup>8</sup> Harvard University, 2003.

<sup>9</sup> Harvard Library, 2014.

<sup>10</sup> Harvard University, 2015.

<sup>11</sup>PUCE, 2015.

estándares internacionales que le permitan ratificar su compromiso con la excelencia universitaria y trascendencia de sus servicios hacia afuera”<sup>12</sup>.

A julio del 2015, la Biblioteca General de la PUCE cuenta dentro de su colección con 120.676 volúmenes en diversas materias, 1.873 títulos de revistas, 300 carpetas de archivo vertical, 1.454 hojas topográficas, 2.710 diapositivas, depositarios del Fondo de Publicaciones de las Naciones Unidas y además es custodio de revistas culturales españolas. También fue la primera unidad de información universitaria que se automatizó en Ecuador, por lo que es importante generar nuevas herramientas digitales para obtener la mayor ventaja de estos recursos.

SpringerLink:

Esta biblioteca virtual busca proveer a las comunidades científicas y profesionales con información especializada en distintos campos académicos producida por autores de diferentes comunidades. Springer intenta promover la comunicación entre investigadores, profesionales y estudiantes, permitiéndoles trabajar de manera más eficiente, avanzando en el conocimiento y acelerando el aprendizaje.

A julio del 2015 Springer cuenta con una colección de más de 2.900 publicaciones científicas y 200.000 libros<sup>13</sup>.

eBrary (Pro Quest):

Fue creada en 1999 por Christopher Warnock, con el pensamiento de que la mayor cantidad de información debería estar disponible en línea para satisfacer las crecientes necesidades de las bibliotecas, investigadores y editores. El objetivo principal de eBrary es ayudar a librerías y otras organizaciones (académicas e industriales) a satisfacer sus necesidades

---

<sup>12</sup> PUCE, 2015.

<sup>13</sup> Springer, 2015.

de investigación reduciendo costos, brindando una nueva estrategia a las instituciones para la adquisición de libros electrónicos. Por medio de sus soluciones permite que el presupuesto que se designaba a material impreso ahora se diversifique en distintos modelos de adquisición por medio de recursos electrónicos.

En el 2011 eBrary fue adquirida por ProQuest, a julio del 2015 cuenta con más de 972.175 documentos dentro de su colección, con un promedio de 10.000 documentos nuevos de distintas materias añadidos cada mes<sup>14</sup>.

Digitalia:

Digitalia hispánica es una base de datos de documentos especializada en ciencias sociales y humanas. Ofrece una amplia cantidad de libros de distintas editoriales españolas y latinoamericanas, así como las revistas más relevantes en las diferentes materias. Dentro de sus colecciones destacan: artes, ciencias políticas, ciencias sociales, tecnología, medicina, literatura, filosofía, geografía, historia, entre otros.

A julio del 2015, esta biblioteca cuenta con más de 13.000 documentos almacenados en su base de datos, repartidos entre libros y revistas electrónicas.

### **Estándares internacionales de referencia bibliográfica**

Existen varios estándares para citar trabajos académicos, en la actualidad los principales son el estilo APA, IEE y MLA. A continuación detallamos las características de cada una de ellas:

**Bibliografía estilo APA:** Creado por la American Psychological Association (APA), es un estándar que los autores utilizan para hacer referencias bibliográficas para sus textos o documentos.

---

<sup>14</sup> Ebrary, 2015.

**Bibliografías estilo IEEE:** Este formato es comúnmente usado para artículos técnicos o de investigación, especialmente en ciencias de la computación.

**Bibliografías estilo MLA:** Creado por la Modern Language Association, se presenta como un estilo académico usado en Estados Unidos, Canadá y otros países, especialmente en el ámbito de humanidades y literatura.

De acuerdo al laboratorio de escritura de la universidad de Purdue (OWL), el formato más utilizado para generar citas es el estilo APA, en especial en el área de ciencias sociales<sup>15</sup>. Este formato ha sido adoptado por la mayoría de estudiantes desde su creación, debido a su simplicidad y practicidad para hacer referencia a textos y publicaciones utilizados académicamente. En el 2009, la asociación psicológica americana (APA) publicó la sexta edición del manual de formato APA, en su libro llamado “The Publication Manual of the American Psychological Association”. En este texto describe las reglas de referencia bibliográfica según el estilo APA y además ofrece una guía detallada para redactar textos con simplicidad y precisión<sup>16</sup>. A continuación presentamos los estilos recomendados para libros y artículos de revistas por la PUCE en su guía para las disertaciones de grado<sup>17</sup>, basados en estándares internacionales.

**Libros:**

Apellidos y nombre del autor. Título. Lugar: Editorial, año de edición. ISBN (opcional).

**Artículos de revista:**

Apellidos y nombre del autor. "Título del artículo". Título de la revista, Número, Ciudad, año.

---

<sup>15</sup> Online Writing Lab Purdue University, 2015.

<sup>16</sup> American Psychological Association, 2015.

<sup>17</sup> Dirección General Académica PUCE, 2007.

## **Principales aplicaciones móviles de consulta bibliográfica**

Algunas bibliotecas y hemerotecas virtuales mencionadas anteriormente ya cuentan con aplicaciones móviles en el mercado para buscar material bibliográfico, tal es el caso de la aplicación de SpringerLink, que lleva el mismo nombre que la empresa, o EBSCO con su aplicación EBSCOhost. Estas aplicaciones se conectan con la base de datos de material bibliográfico propietaria de cada una de las empresas mencionadas; sin embargo, no integran otras fuentes de información.

Algunas de las principales aplicaciones de consulta bibliográfica publicadas en la tienda de AppStore son:

SpringerLink: Desarrollada por la empresa Springer ofrece una de las más largas colecciones de información científica. Actualmente se encuentra en su versión 3.0.1 y permite realizar búsquedas y obtener pequeños resúmenes de los resultados<sup>18</sup>.

EBSCOhost: Es una aplicación gratuita desarrollada por la empresa EBSCO provee a los usuarios de una base de datos con información de distintas instituciones alrededor del mundo. Actualmente en su versión 3.0, esta aplicación permite guardar archivos en formato PDF y mandar mails con los artículos guardados por cada usuario<sup>19</sup>.

EasyBib: Es una de las aplicaciones más conocidas en el mercado para generar citas bibliográficas, su descripción en la tienda AppStore es “La bibliografía es frustrante. No tiene que serlo”. Actualmente se encuentra en la versión 2.0 y es una de las aplicaciones mejor calificadas en AppStore dentro de su categoría<sup>20</sup>.

---

<sup>18</sup> Springer, 2015.

<sup>19</sup> EBSCO Publishing, 2015.

<sup>20</sup> Imagine Easy Solutions, 2015

iSource APA: Creada por Wesley Dyson es una aplicación que te permite almacenar tus citas y dar formato APA a la bibliografía que estás usando en tus trabajos académicos<sup>21</sup>.

Otras aplicaciones de consulta bibliográfica que han tomado popularidad en la tienda de Apple son: Mendely, Endnote, myBib, ReferenceMe y Quick Cite.

El siguiente cuadro comparativo presenta las características funcionales de las principales aplicaciones de consulta bibliográfica publicadas en AppStore.

	SpringerLink	EBSCOhost	Easybib	iSource APA
Búsqueda de texto en una librería	✓	✓		
Búsqueda de texto en varias librerías				
Habilidad para exportar a PDF		✓		
Envío de citas por correo		✓	✓	
Ordenar textos por relevancia o fechas de publicación		✓		
Guardar resultados para acceder offline	✓	✓		
Detalle del documento	✓			
Guardar documentos como favoritos	✓			
Compartir documentos				
Citaciones verificadas por expertos			✓	
Creación de citas			✓	✓
Escoger tipo de documentos			✓	✓
Escoger entre varios estilos de citas			✓	✓

Figura 2: Características funcionales de las principales aplicaciones.

Fuente: AppStore, 2015.

En el siguiente capítulo se realizará el diseño del prototipo de la aplicación de consulta bibliográfica tomando en cuenta las características funcionales de las principales aplicaciones de consulta bibliográfica que existen en el mercado. Para esto analizaremos los requerimientos iniciales y estableceremos el diseño arquitectónico y detallado del software que se va a desarrollar.

<sup>21</sup> Dyson, 2015

## CAPÍTULO 2

### ANÁLISIS Y DISEÑO DE LA APLICACIÓN

En el presente capítulo se va a realizar el análisis previo a la implementación de la aplicación. Para esto se va a presentar la metodología y herramientas de desarrollo, definiendo los requerimientos funcionales y el diseño arquitectónico y detallado del software. Finalmente, este capítulo terminará con el diseño de la interfaz de usuario con el que se va a construir la aplicación.

#### **Metodología de desarrollo**

Un buen proceso de desarrollo de software puede ayudar a las organizaciones a lograr mayor eficiencia en sus productos, por otro lado, un proceso no definido puede sobrecargar de trabajo a miembros del equipo y reducir la productividad. A lo largo del tiempo han surgido varias metodologías que han sido adoptadas por grandes empresas, dentro de los procesos tradicionales se encuentran PSP<sup>22</sup> y TSP<sup>23</sup> que ayudan a los desarrolladores a planificar, hacer seguimiento y analizar el trabajo realizado por una o varias personas, en proyectos de desarrollo de software a gran escala<sup>24</sup>; sin embargo, en los últimos años han surgido nuevas metodologías ágiles que aceleran el proceso en proyectos de menor magnitud, centrándose más en la construcción del producto que en la estimación de tiempo o planificación, como por ejemplo Scrum o Kanban. A continuación se presentan las principales características de estas metodologías, para de esta manera determinar la que mejor se adapte al proyecto.

TSP es un proceso diseñado para el desarrollo de software en equipo, el cual se enfoca en asegurar la calidad y productividad de cada uno de los integrantes. Esta metodología se desarrolló para ser aplicada en proyectos de gran escala (de miles a millones de líneas de código LOC<sup>25</sup>). El ciclo de TSP comienza con la fase de lanzamiento, después se establece la estrategia de desarrollo, se realiza la planificación del proyecto, se definen los requerimientos, se diseña e implementa el producto y finalmente se realizará las pruebas sobre el producto.

---

<sup>22</sup> PSP. Personal software process.

<sup>23</sup> TSP. Team software process.

<sup>24</sup> Humphrey, 1999.

<sup>25</sup> LOC. Lines of code.

Por otro lado, la metodología Scrum presenta un modelo iterativo e incremental por medio del cual se organizan los requerimientos y se les asigna una prioridad para crear un producto funcional, que puede ser probado por el cliente y sobre el cual se van a realizar mejoras en cada iteración. La idea de Scrum es organizar todos los pedidos del cliente, para así ir definiendo las características que debe tener el producto. Una vez que se han definido los requerimientos, el gerente del proyecto se encarga de organizar las actividades en etapas, de tal manera que en cada etapa se tenga un prototipo funcional listo para ser probado por el cliente.

Otra de las metodologías ágiles que ha tomado gran fuerza en los últimos años es Kanban, este proceso se enfoca en la repartición de tareas, controlando que los miembros del equipo no se sobrecarguen de trabajo; por medio de este proceso se crea una administración visual del avance del proyecto, potencializando la productividad del equipo. Esta metodología fue implementada por Toyota en 1953 para mantener el control logístico de su sistema de producción, con el principio de que se puedan entregar los productos a tiempo, administrando tareas y eliminando desperdicios e inconsistencias. Debido a su éxito en la industria de la manufactura, Kanban es ahora utilizado para el desarrollo de software, tomando los principios básicos del sistema original y poniendo énfasis en entregar el producto a tiempo (JIT<sup>26</sup>).

A continuación se presenta un cuadro comparativo con las principales características de las metodologías mencionadas anteriormente.

<b>Característica</b>	<b>TSP</b>	<b>Scrum</b>	<b>Kanban</b>
<b>Tamaño del proyecto</b>	Software de gran escala.	Todo tipo de proyectos.	Todo tipo de proyectos.
<b>Roles</b>	Líder, administrador de desarrollo, administrador de calidad, administrador de planificación y administrador de configuración.	Dueño del producto, equipo de desarrollo (desarrolladores y aseguradores de calidad), scrum master y cliente.	Libertad de adaptar los roles según el proyecto.

<sup>26</sup> JIT. Just in time.

<b>División de tareas</b>	De acuerdo al rol, cada integrante tiene tareas específicas.	El scrum master organiza los requerimientos y reparte las tareas para que las realice el equipo de desarrollo.	Los miembros del equipo dividen el proyecto en tareas y se asignan equitativamente.
<b>Limitación del trabajo en progreso</b>	El administrador de planificación limita las horas de trabajo de cada integrante.	El scrum master organiza las tareas en sprints, de tal manera que los integrantes no se sobrecarguen de trabajo.	Se define un número máximo de tareas en progreso por cada miembro del equipo.
<b>Reuniones</b>	Los miembros del equipo se deben reunir semanalmente.	Los miembros del equipo se deben reunir diariamente.	Los miembros se deben reunir semanalmente.
<b>Documentación</b>	Se genera documentación semanal con el avance del proyecto.	Se genera documentación según los requerimientos del cliente.	Se genera documentación con las tareas realizadas.

Figura 3: Cuadro comparativo metodologías de desarrollo.

Fuente: Anderson, Concas, Lunesu, 2010.

Cada proyecto tiene diferentes características y magnitud, por lo tanto se debe escoger la metodología que más se adapte al proyecto y equipo de desarrollo, de tal manera que el producto sea construido en el menor tiempo posible, con la menor cantidad de defectos y satisfaciendo las necesidades del cliente<sup>27</sup>. El proyecto que se va a realizar en este trabajo requiere una metodología ágil, que centre su atención en el desarrollo del producto, más que en la documentación generada; adicionalmente requiere un proceso que otorgue flexibilidad a los miembros del equipo en la repartición de tareas debido a que no es un equipo de muchos integrantes. Por estas razones la metodología que se va a usar para el desarrollo de esta aplicación es Kanban.

<sup>27</sup> Anderson, Concas, 2012

Los principios básicos de Kanban adaptado al software son:

1. Visualizar las actividades que deben ser realizadas y estado en el que se encuentran.
2. Limitar el número de tareas en progreso (WIP<sup>28</sup>).
3. Manejar el flujo de las tareas para poder medirlas, monitorearlas y optimizarlas.
4. Eliminar los requerimientos que no agreguen valor al cliente.
5. Ajustar los niveles de producción

El control de tareas se maneja a través de un tablero y tarjetas visuales que representan las actividades que se deben realizar para la construcción del producto. El tablero se divide en columnas que constituyen los estados de las actividades en el flujo de trabajo y pueden ser categorizados desde nuevos hasta finalizados. Uno de los principales beneficios de esta metodología es que permite a los miembros del equipo identificar de manera fácil las tareas completadas, las que se encuentran en progreso o las que todavía no han iniciado.

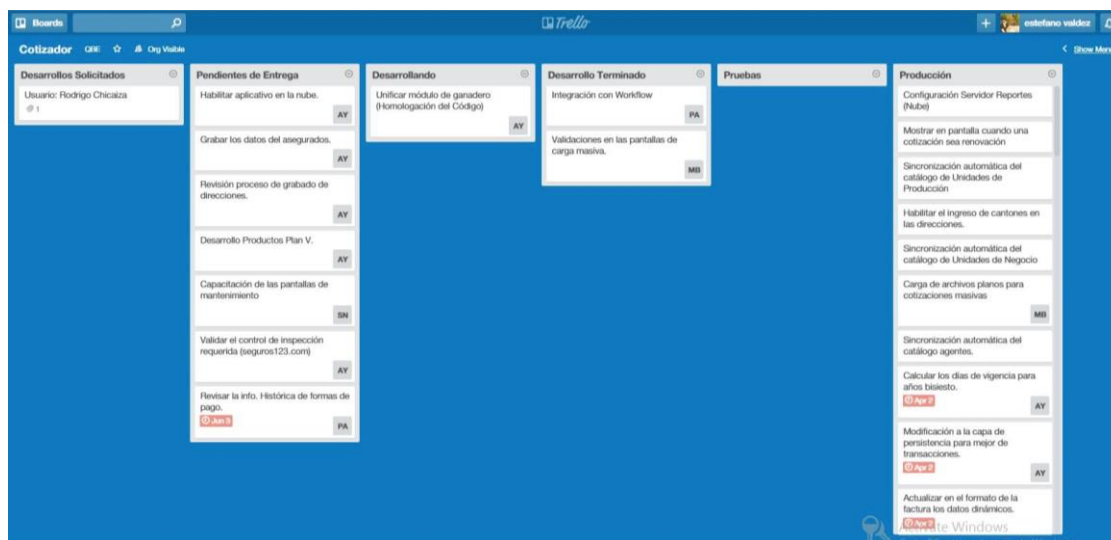


Figura 4: Tablero Kanban.

Fuente: Trello<sup>29</sup>, 2015.

<sup>28</sup> WIP. Work in progress.

<sup>29</sup> Trello. Herramienta Kanban para organizar tareas en equipo. <https://trello.com/>

A continuación se describen los pasos que se van a seguir para el desarrollo de la aplicación con base en la metodología escogida:

1. Listar todos los requerimientos como actividades en el tablero con el estado de pendientes.
2. Asignar las actividades a los miembros del equipo.
3. Cada miembro del equipo realiza las tareas, respetando los límites de tareas en progreso.
4. Las tareas finalizadas pasan al estado de completas.
5. Las tareas completas pasan a pendiente de prueba.

Consideraciones del tablero:

- Las categorías que se van a utilizar son: nuevas, en progreso, finalizadas y probadas.
- El análisis de requerimientos va a determinar las actividades a realizarse.
- El límite de tareas en progreso es 4.
- Cada semana se reúne el equipo para revisar el tablero y el estado de las tareas.

### **Herramientas de desarrollo**

Actualmente, para el desarrollo de aplicaciones móviles existen tres alternativas:

1. Desarrollo de software con programación nativa, basado en el SDK de cada plataforma.
2. Desarrollo multiplataforma basado en ambientes web creados sobre el entorno de desarrollo nativo, con la capacidad de exportar hacia varias plataformas.
3. Desarrollo de aplicaciones con HTML, JavaScript y CSS, donde la aplicación funciona como un navegador web.

Dependiendo de la industria y los requerimientos del negocio, se deben tomar en cuenta varios factores para identificar la alternativa adecuada para la implementación. Entre los aspectos más importantes a tomar en cuenta se encuentran: el propósito de la aplicación, quiénes serán los usuarios, qué funcionalidades se necesitan, el tiempo que se dispone para desarrollo,

el presupuesto, entre otras características. A continuación se presenta un cuadro que analiza las ventajas y desventajas de cada alternativa.

<b>Alternativa</b>	<b>Nativo</b>	<b>Multiplataforma</b>	<b>Web Móvil</b>
<b>Herramientas</b>	<ul style="list-style-type: none"> <li>• iPhone SDK</li> <li>• Android SDK</li> <li>• Windows SDK</li> <li>• Etc.</li> </ul>	<ul style="list-style-type: none"> <li>• RhoMobile</li> <li>• Titanium</li> <li>• PhoneGap</li> <li>• Xamarin</li> </ul>	<ul style="list-style-type: none"> <li>• HTML5</li> <li>• jQuery Mobile</li> <li>• Sencha</li> </ul>
<b>Lenguaje de programación</b>	<ul style="list-style-type: none"> <li>• Objective-C</li> <li>• Swift</li> <li>• Java</li> <li>• .NET</li> </ul>	<ul style="list-style-type: none"> <li>• Ruby on Rails</li> <li>• JavaScript</li> <li>• HTML</li> <li>• C#</li> </ul>	<ul style="list-style-type: none"> <li>• HTML</li> <li>• JavaScript</li> </ul>
<b>Distribución</b>	Tienda de aplicaciones	Tienda de aplicaciones	Internet
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>• Alto rendimiento</li> <li>• Uso de funcionalidades offline</li> <li>• Uso de componentes nativos</li> <li>• Optimizadas para el sistema operativo</li> <li>• Aprovecha completamente el hardware</li> <li>• Aprovecha completamente el software del sistema operativo</li> </ul>	<ul style="list-style-type: none"> <li>• Se puede exportar para varias plataformas</li> <li>• Solo se necesita conocer un entorno de desarrollo</li> <li>• Reutilización de código</li> <li>• Menor tiempo y costo de desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>• Fácil actualización</li> <li>• Puede ejecutarse en cualquier dispositivo con acceso a un explorador web</li> <li>• Menor tiempo y costo de desarrollo</li> <li>• Fácil implementación</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>• Conocimiento de un lenguaje para cada plataforma</li> <li>• Es más costoso y toma más tiempo</li> </ul>	<ul style="list-style-type: none"> <li>• No se tiene la misma experiencia de usuario</li> <li>• No existe soporte inmediato para nuevas versiones del sistema operativo</li> </ul>	<ul style="list-style-type: none"> <li>• Se necesita conexión a internet</li> <li>• No hay acceso a hardware del dispositivo</li> <li>• Depende del explorador web</li> </ul>

Figura 5: Cuadro alternativas de desarrollo de aplicaciones móviles

Fuente: RapidValue, 2012.

Según lo expuesto anteriormente, cada uno de estos ambientes tiene sus fortalezas y debilidades; debido a las características de este proyecto se ha seleccionado la alternativa multiplataforma ya que esto reduce significativamente el tiempo de desarrollo, tomando en cuenta que no se necesita que la aplicación se encuentre optimizada para un sistema operativo específico. Adicionalmente, esto permitiría exportar a otras plataformas sin tener que volver a construir el prototipo. Cabe recalcar también que de acuerdo a la consultora Gartner, se estima que para el 2016 más del 50% de las aplicaciones móviles serán híbridas<sup>30</sup>, por lo que es importante prestar atención a esta alternativa en el ámbito de desarrollo para dispositivos móviles.

### **Herramientas de desarrollo del lado del cliente**

El mercado de plataformas de desarrollo de aplicaciones móviles (MADP) ofrece distintas herramientas, tecnologías, componentes y servicios para que las empresas o emprendedores puedan diseñar, desarrollar y distribuir su portafolio de aplicaciones abarcando diversos requerimientos de sus clientes. La siguiente representación muestra el posicionamiento de los proveedores más destacados de MADP en julio del 2015, divididos en jugadores de nicho, visionarios, retadores y líderes (medidos por su visión de negocio y su habilidad de ejecución).

---

<sup>30</sup> Gartner, 2014



Figura 6: Cuadrante mágico de plataformas de desarrollo de aplicaciones móviles

Fuente: Gartner, 2015.

Dentro de las empresas más destacadas categorizadas como líderes o visionarias, que ofrecen un IDE<sup>31</sup> para los desarrolladores se encuentran Adobe, Xamarin y Appcelerator. A continuación se detallan las características de la plataforma que ponen a disposición estas empresas.

Appcelerator:

El producto que ofrece esta empresa es Titanium, desarrollado como un marco de desarrollo para aplicaciones móviles multiplataforma que incluye los siguientes sistemas operativos: iOS, Android, Windows Phone, Blackberry OS y Tizen. El kit de desarrollo de Titanium se encuentra bajo licencia Apache y el lenguaje de desarrollo con el que trabaja es JavaScript. Para poder utilizar esta herramienta es necesario contar con el SDK de cada plataforma.

<sup>31</sup> IDE: Ambiente de desarrollo integrado

Adobe:

El producto que ofrece esta empresa es PhoneGap, que permite construir proyectos web para exportarlos como aplicaciones nativas. De esta manera los desarrolladores pueden crear archivos HTML y CSS en un directorio local, como si se tratara de una página web, para posteriormente procesar estos archivos en PhoneGap y obtener como resultado una aplicación que se puede instalar en distintas plataformas.

Xamarin:

Xamarin es un producto basado en un ambiente .NET, que intenta mantener el diseño nativo de la interfaz de cada plataforma. Además de poner énfasis en la interfaz visual de cada sistema operativo, Xamarin ofrece un módulo para realizar pruebas, el cual puede mejorar el rendimiento de la aplicación antes de lanzarla al mercado.

Para desarrollar este prototipo se requiere una herramienta que facilite la creación de una aplicación con funcionalidades nativas, lo cual se puede aprovechar de mejor manera por medio de Xamarin y Appcelerator. A pesar de que estas dos plataformas ofrecen varias ventajas para el desarrollo multiplataforma, se ha determinado que la herramienta que se va a utilizar es Titanium por el uso de JavaScript y porque ofrece un alto porcentaje de reutilización de código para exportar a distintas plataformas, entre 60% y 90%<sup>32</sup>. Además es importante mencionar que ofrece una versión gratuita para los desarrolladores, que es de gran importancia en el desarrollo de aplicaciones que no tienen un uso comercial.

### **Herramientas de desarrollo del lado del servidor**

La computación en la nube es un modelo que permite entregar servicios tecnológicos a través de Internet, el proveedor de este servicio es el que se encargará de gestionar y mantener el almacenamiento, servidores, data centers, etc.

---

<sup>32</sup> Appcelerator, 2015

Las principales ventajas que el cliente percibe al utilizar computación en la nube son:

- El alto costo de adquisición y mantenimiento de una granja de servidores se transfiere al proveedor del servicio.
- El proveedor se encarga de manejar la escalabilidad de los servidores de acuerdo a los requerimientos del cliente.
- No se desperdicia en características innecesarias ya que estas se van solicitando en medida de las necesidades.
- El proveedor del servicio tiene personal amplio y capacitado para brindar soporte sobre cada aspecto de la infraestructura.

El modelo de plataforma como servicio (PaaS) fue diseñado para desarrolladores ofreciendo un ambiente listo para el despliegue de una solución que incluye bases de datos, servidores web, lenguajes de programación y sistema operativo. Así el cliente se encarga de subir su solución en un ambiente idóneo para ejecutar esta solución. A continuación se muestra el cuadrante mágico de los proveedores más destacados de PaaS hasta marzo del 2015.



Figura 7: Cuadrante mágico de aplicaciones empresariales de PaaS.

Fuente: Gartner, 2015.

El siguiente cuadro muestra una comparación de los servicios que ofrecen el lenguaje más popular (Java)<sup>33</sup> dentro de su solución de plataforma como servicio.

	<b>Amazon Beanstalk</b>	<b>Cloud Foundry</b>	<b>Google App Engine</b>	<b>Heroku para Java</b>	<b>OpenShift De RedHat</b>
<b>Tomcat</b>	Sí	Sí	No	No	Sí
<b>Java EE</b>	No	No	No	No	Sí
<b>Portabilidad de aplicaciones</b>	Alta	Moderado	Bajo	Bajo	Alta
<b>Integración con un IDE</b>	Sí	Sí	Sí	No	Sí
<b>Herramientas por línea de comando</b>	Sí	Sí	Sí	Sí	Sí
<b>Documentación</b>	Buena	Escasa	Buena	Buena	Buena
<b>Balancedador</b>	Sí	Sí	Sí	Sí	Sí
<b>Dominio personalizado</b>	Sí	No	Google Apps	Sí	Sí
<b>Escalabilidad automática de servidor de aplicaciones</b>	Sí	Planificado	Sí	No	Sí
<b>Escalabilidad automática de base de datos</b>	No	No	Sí	No	No
<b>Versión gratuita</b>	Sí	N/A	Si	Si	Si

Figura 8: Cuadro comparativo de PaaS que ofrecen Java

Fuente: Michael Yuan<sup>34</sup>, 2011.

<sup>33</sup> TIOBE, 2015.

<sup>34</sup> Yuan, Michael. A Java Developer's Guide to PaaS: [http://www.infoq.com/articles/paas\\_comparison](http://www.infoq.com/articles/paas_comparison)

Entre los servicios analizados, los que más destacan por las características que brindan son OpenShift y Amazon Beanstalk, de los cuales se escogió OpenShift ya que además de proveer de una versión gratuita, cumple con todos los requerimientos necesarios para implementar el proyecto como Tomcat, Java EE, escalabilidad, amplia documentación y la capacidad de integración con un IDE.

### **Análisis de requerimientos**

El objetivo principal del desarrollo de esta aplicación móvil de consulta bibliográfica es crear un motor de búsqueda de documentos de distintas bibliotecas virtuales. Para esto se va a utilizar la técnica web scraping<sup>35</sup>, de tal manera que se puedan concentrar varias fuentes disponibles y que los estudiantes de la PUCE puedan consultar y obtener información de manera fácil. Los siguientes requerimientos funcionales están basados en el análisis de las principales aplicaciones móviles realizado en el anterior capítulo, en donde se determinó que ninguna de estas aplicaciones existentes concentra documentos de varias librerías; gracias a esto se han tomado en cuenta concentrar las siguientes librerías en este prototipo: Biblioteca general de la PUCE, SpringerLink y eBrary (Pro Quest), de tal manera que se puedan a analizar a futuro la inclusión de nuevas librerías. Adicionalmente, una parte importante del desarrollo es que la aplicación tenga la funcionalidad de visualizar la descripción del documento y de generar referencias bibliográficas en formatos estándar, para que de esta forma se aprovechen de mejor manera los recursos que pone a disposición la universidad. De esta manera se han definido los siguientes requerimientos funcionales y no funcionales:

Requerimientos funcionales:

- Buscar documentos en la biblioteca de la PUCE.
- Buscar documentos en la biblioteca virtual SpringerLink.
- Buscar documentos en la biblioteca virtual eBrary.
- Visualizar descripción del documento y disponibilidad.

---

<sup>35</sup> Web scraping. Es una técnica para extraer información de sitios web, en la que un software simula la navegación de un humano.

- Generar citas bibliográficas.
- Enviar citas por correo electrónico.

Requerimientos no funcionales:

- Precisión: El motor de búsqueda deberá entregar resultados con exactitud y acorde a la biblioteca sobre la cual se está realizando la consulta.
- Desempeño: El tiempo de respuesta con la cual se presentan los resultados deberá estar dentro de un rango considerable.
- Usabilidad: La aplicación deberá ser intuitiva y fácil de usar para los usuarios finales.
- Portabilidad. El software deberá ser exportable a otros dispositivos móviles, con pocos cambios en el código fuente.

## **Diseño arquitectónico**

En esta sección se va a definir la estructura del sistema en el nivel más alto. Para el diseño de esta aplicación se definieron 3 capas:

1. Cliente: Dispositivos con sistema operativo iOS (iPhone o iPad), donde se instala la aplicación. Se encarga de realizar peticiones y esperar respuestas del servidor, mientras interactúa con usuarios finales mediante una interfaz gráfica.
2. Servidor: Se encarga de alojar al servidor web. Éste cumple dos funciones en el sistema, en primer lugar recibe las solicitudes por parte de los clientes para procesarlas y enviar respuestas y en segundo lugar recopila la información de las distintas páginas web por medio de la técnica web scraping.
3. Sitios web: Son las páginas de las bibliotecas virtuales que se van a encargar de proveer información al servidor web.

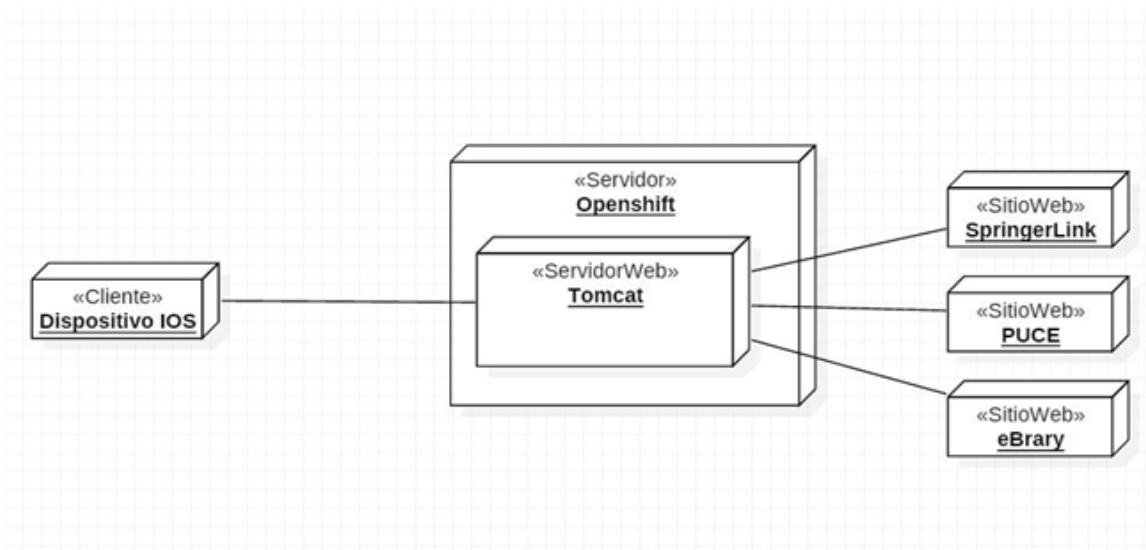


Figura 9: Diseño arquitectónico.

Fuente: Estefano Valdez, Sebastián Dávalos. Star UML, 2015

## Diseño detallado

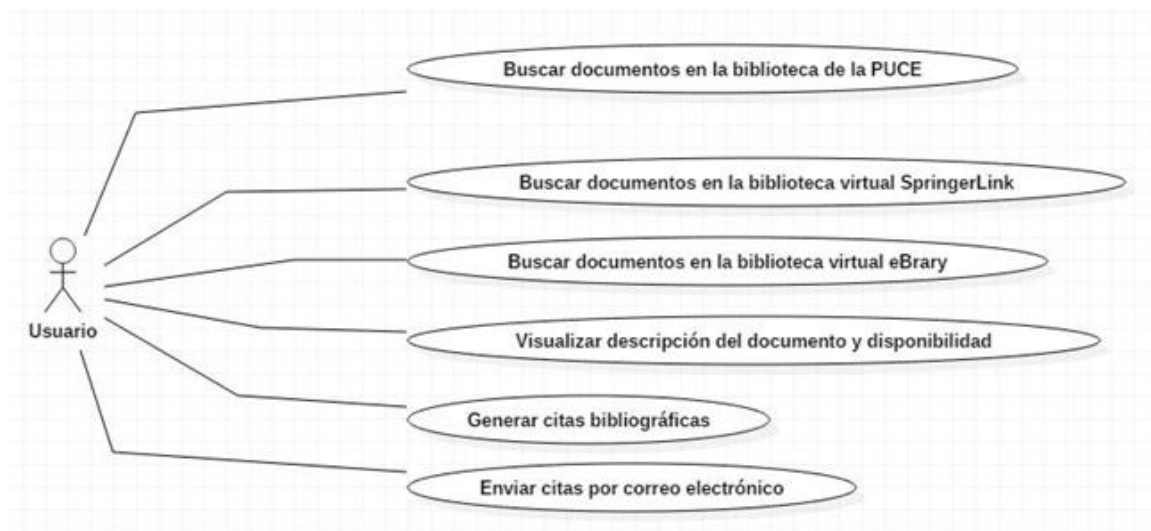


Figura 10: Casos de uso a nivel general.

Fuente: Estefano Valdez, Sebastián Dávalos. Star UML, 2015.

## Interfaz de usuario

Uno de los aspectos más importantes del desarrollo de aplicaciones móviles es la interfaz de usuario (UI<sup>36</sup>). A pesar de que no existe una regla que asegure un diseño ideal, una buena interfaz gráfica ayuda a tener una buena impresión con los usuarios para que ellos se sientan cómodos y familiarizados con cada uno de los elementos implementados en la pantalla de su dispositivo.

Además de proveer un buen diseño, es importante que una aplicación sea simple y fácil de usar. Según Steve Krug, en su libro “Don’t make me think”, la usabilidad es asegurarse de que la plataforma funcione de manera correcta y que un usuario con conocimiento promedio sea capaz de utilizarlo. La interfaz de usuario de la aplicación en este desarrollo se va a armar en base a los elementos que provee iOS y están detallados en la guía oficial de Apple para diseño de aplicaciones<sup>37</sup>. A continuación se describen algunos de estos elementos.

### Barra de status

Despliega información importante acerca del dispositivo. Generalmente es transparente y se ubica en la parte superior de la pantalla.



Figura 11: Barra de status (iOS).

Fuente: Apple, 2015.

---

<sup>36</sup> UI. User Interface

<sup>37</sup> Apple, 2015

## Barra de navegación

Permite la navegación entre ventanas de la aplicación dependiendo de la jerarquía que tengan dentro de la estructura. Se encuentra en la parte inferior de la barra de status.



Figura 12: Barra de navegación (iOS),  
Fuente: Apple, 2015.

## Barra de búsqueda

La barra de búsqueda recibe texto del usuario que puede ser usado como datos de entrada para realizar una búsqueda.

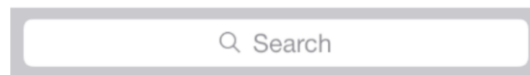



Figura 13: Barra de búsqueda (iOS)  
Fuente: Apple, 2015.

## Íconos

El uso de íconos estándar ayuda a los usuarios en la navegación de la aplicación, ya que su función es fácilmente reconocible.

Icono	Título	Acción
	Adelante	Permite obtener detalle sobre un elemento o ir hacia la ventana siguiente




	Atrás	Permite regresar a la anterior pantalla desplegada en la aplicación
	Correo	Activa la opción de envío de correo con la información sobre un elemento de la aplicación
	Copiar	Copia un texto al portapapeles del dispositivo
	Buscar	Permite al usuario encontrar el campo donde puede ingresar un texto de búsqueda

Figura 14: Íconos estándar.

Fuente: Apple, 2015.

En base a los elementos detallados anteriormente, se han diseñado las pantallas que serán implementadas en el desarrollo de la aplicación, expuestas a continuación.

<b>Código</b>	<b>Descripción</b>
UI1	Pantalla inicial, el usuario puede ingresar el texto que desee buscar.
UI2	El usuario ingresa el texto que desea buscar, la aplicación comienza a cargar los resultados.
UI3	La aplicación muestra los resultados de la biblioteca de la PUCE. El usuario puede seleccionar las otras bibliotecas.
UI4	La aplicación muestra los resultados de la biblioteca virtual Springer. El usuario puede seleccionar las otras bibliotecas.
UI5	La aplicación muestra los resultados de la biblioteca virtual eBrary. El usuario puede seleccionar las otras bibliotecas.
UI6	Pantalla que muestra el detalle de un documento, el usuario puede generar una cita bibliográfica y enviarla por mail.

Figura 15: Cuadro de interfaces.

Fuente: Estefano Valdez, Sebastián Dávalos, 2015.



Figura 16: UI1

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.

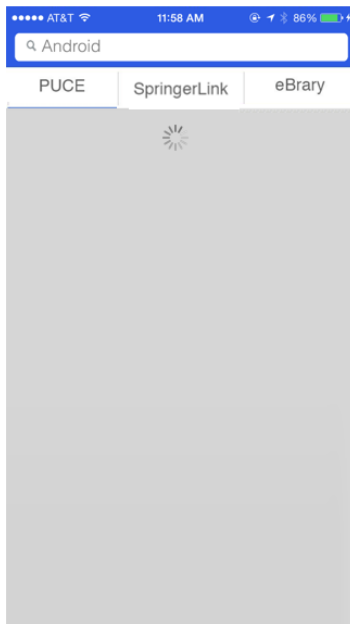


Figura 17: UI2

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.



Figura 18: UI3

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.

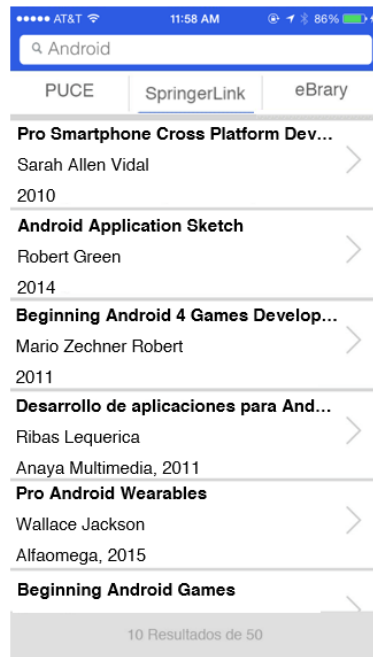


Figura 19: UI4

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.



Figura 20: UI5

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.



Figura 21: UI6

Fuente: Estefano Valdez, Sebastián Dávalos. Photoshop, 2015.

## Plan de pruebas de la aplicación

A continuación se muestra el plan de pruebas de la aplicación, con el cual se verificará la funcionalidad del sistema después de la implementación.

ID	Caso de prueba	Entradas	Resultado esperado	Interfaz
C1	Ingreso a la aplicación	El usuario abre la aplicación.	La aplicación se abre mostrando la opción de búsqueda de documentos.	UI1
C2	Búsqueda de documentos	El usuario ingresa el texto de referencia para buscar en las librerías.	La aplicación muestra todos los documentos que tengan coincidencia con la referencia ingresada, en las librerías PUCE, eBrary y Springer. Los campos que se presentan son: título, autor y año.	UI3, UI4, UI5
C3	No se encuentran resultados	El usuario ingresa un texto que no devuelve resultados en la búsqueda.	La aplicación genera un mensaje anunciando que no se encontraron resultados.	UI1
C4	Mostrar detalle del documento	El usuario selecciona un documento para visualizar el detalle.	La aplicación muestra el detalle del documento: título, autor, editorial, edición, año, ciudad, disponibilidad, ISBN y campos adicionales.	UI6
C5	Generar cita bibliográfica	El usuario selecciona la opción de generar cita.	La aplicación genera la cita en formato estándar y la copia al portapapeles.	UI6
C6	Enviar cita por mail	El usuario selecciona la opción de enviar cita por correo electrónico.	La aplicación llama al cliente de correo del dispositivo con la cita generada en el cuerpo del correo.	UI6

Figura 22: Casos de prueba.

Fuente: Estefano Valdez, Sebastián Dávalos, 2015.

Pruebas de compatibilidad en distintos dispositivos:

La aplicación será probada en los siguientes dispositivos físicos:

- iPhone 6 con sistema operativo iOS 8.1
- iPad 2 con sistema operativo iOS 8.1

Y en los siguientes dispositivos virtuales:

- iPhone 4s con sistema operativo iOS 8.1
- iPhone 5 con sistema operativo iOS 8.1
- iPad Air con sistema operativo iOS 8.1

En el siguiente capítulo se va a realizar la codificación de la aplicación, analizando la integración de cada librería y utilizando la metodología y las herramientas seleccionadas en este capítulo.

## CAPÍTULO 3

### IMPLEMENTACIÓN DEL PROTOTIPO DE LA APLICACIÓN

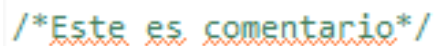
En el presente capítulo se va a documentar el desarrollo de la aplicación. Para esto se van a describir los estándares de desarrollo que se van a utilizar y las funcionalidades implementadas en la aplicación, con especial énfasis en el análisis de la integración con cada una de las bibliotecas y sus particularidades.

### Estándares de desarrollo

#### Estándares JavaScript

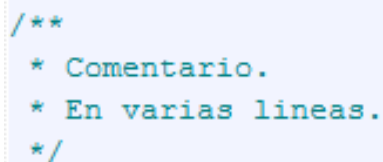
Los estándares JavaScript que se van a utilizar son los definidos por Google<sup>38</sup>, entre los cuales se destacan los siguientes:

- Los comentarios en JavaScript se escriben entre asteriscos. A continuación se exponen los comentarios de una y varias líneas.



```
/*Este es comentario*/
```

Figura 23: Comentario de una línea en JavaScript.  
Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.



```
/**  
 * Comentario.  
 * En varias lineas.  
 */
```

Figura 24: Comentario de varias líneas en JavaScript.  
Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

---

<sup>38</sup> <http://google.github.io/styleguide/javascriptguide.xml>

- Los nombres de las funciones empiezan con letra minúscula y se utiliza mayúscula a partir del inicio de cada palabra.

```
function miFuncion(uno, dos)
{
    ...
}
```

Figura 25: Función en JavaScript.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Debe existir un espacio entre palabras clave y paréntesis que utiliza la sentencia. Esto se utiliza en sentencias switch, if, for, while, etc.

```
if (condicion)
{
    ...
}
```

Figura 26: Espacio después de palabras clave en JavaScript.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Cada sentencia debe estar en una línea nueva, si utiliza llaves deben ir cada una en una línea nueva.

```
if (condicion) return;      /* NO SE DEBE USAR */
if (condicion)              /* SE DEBE USAR */
{
    return;
}
```

Figura 27: Sentencias en JavaScript.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Las llaves deben estar en todas las sentencias tipo: if, else, do, switch, etc. Además deben ocupar su propia línea.

```
if (condicion)
{
    ...
}
else
{
    ...
}
```

Figura 28: Uso de llaves en JavaScript.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

Entre otras consideraciones importantes cuando se escribe código en JavaScript destacan:

- Las constantes deben estar definidas en mayúsculas.
- La longitud de cada línea de código no debe superar los 80 caracteres de ser posible.
- No debe existir un espacio entre la declaración de la función y los parámetros.
- Debe existir un espacio después de una coma.

## Estándares Java

Los estándares Java que se van a utilizar también son los definidos por Google<sup>39</sup>, entre estos destacan los siguientes.

- Los comentarios se escriben con slash (/) y asterisco (\*) cuando son de más de una línea. En caso de ser sólo una línea se utiliza doble slash (//).

---

<sup>39</sup> <https://google.github.io/styleguide/javaguide.html>, 2015

```
// Comentario de una línea
```

Figura 29: Comentario de una línea en Java.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

```
/*  
 * Comentario de varias líneas  
 * okay.  
*/
```

Figura 30: Comentario de varias líneas en Java.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Los nombres de las funciones empiezan con letra minúscula y se utiliza mayúscula a partir del inicio de cada palabra.

```
public boolean esAutor(String str)  
{  
    return str.matches("[A-zÀ-ÿ]+\\s?([A-zÀ-ÿ]+)?, (\\s?[A-zÀ-ÿ])+" );  
}
```

Figura 31: Función en Java.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Se utilizan llaves para sentencias del tipo if, else, for, do y while, incluso si dentro de estas no existan sentencias o exista solo una. Adicionalmente, debe existir un espacio entre palabras clave y paréntesis. La llave va al final de la línea que contiene la sentencia que la utiliza, separada por un espacio.

```
if (txt.indexOf("Piso") > -1) {  
    retorno = true;  
}
```

Figura 32: Uso de llaves en Java.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

- Los nombres de las clases deben tener mayúsculas en el inicio de cada palabra, y debe encontrarse en un archivo con extensión java del mismo nombre.

```
public class EbraryController
```



Figura 33: Nombre de clases en Java.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

Entre otras consideraciones importantes para Java destacan:

- Las constantes deben estar definidas en mayúsculas.
- Los archivos java deben encontrarse en codificación UTF-8
- Debe existir un espacio después de una coma.
- Después de cada punto y coma va un salto de línea.

## Integración de la librería SpringerLink

La librería SpringerLink cuenta con una barra de búsqueda en su sitio web<sup>40</sup> por medio de la cual los usuarios pueden ingresar el texto para buscar distintos documentos. Una vez que el usuario realiza la búsqueda, el sitio web muestra los resultados con un máximo de 20 documentos por página. Esto plantea una dificultad en la integración de la librería en la aplicación, ya que utilizando esta opción se deberían recorrer todas las páginas de resultado de la búsqueda, utilizando una gran cantidad de procesamiento por parte del servidor y habría una dificultad en el envío de datos del servidor al celular, lo cual afectaría de manera importante el rendimiento y la experiencia que tendría el usuario con la aplicación.

---

<sup>40</sup> Sitio oficial Springer. <http://link.springer.com/>, 2015

The screenshot shows the SpringerLink search results for the query 'android'. At the top, the SpringerLink logo is visible along with navigation links for 'Sign up / Log in', 'English', and 'Academic edition'. A search bar contains the text 'android' and a 'New Search' button. Below the search bar, there are links for 'Home' and 'Contact Us'. The main search results area displays '12,687 Result(s) for 'android''. On the left side, there are several filter sections: 'Include Preview-Only content' (checked), 'Refine Your Search' with a table for 'Content Type', 'Discipline', 'Subdiscipline', and 'Language'. The 'Content Type' table lists: Chapter (9,270), Article (3,221), Reference Work Entry (100), Book (80), and Protocol (16). The 'Discipline' table lists: Computer Science (9,052), Engineering (2,236), Medicine (1,885), Business & Management (1,408), and Social Sciences (494). The 'Subdiscipline' table lists: Database Management & Information Retrieval (3,260), SWE (2,805), Communication Networks (2,613), AI (2,349), and HCI (2,152). The 'Language' table lists: English (11,542). The main results area shows three items: a chapter titled 'Android' with a description and a 'Look Inside' link; another chapter titled 'Android' with a description and a 'Look Inside' link; and a book titled 'Android Application Sketch Book' by Dean Kaplan (2011) with a book cover image. Below it, another book titled 'Beginning Android 4 Games Development' is partially visible.

Figura 34: Página de resultados de SpringerLink.

Fuente: SpringerLink, 2015.

Debido a esto se buscó otra alternativa para obtener los resultados de la librería de Springer. Además de proveer varias páginas con los resultados, el portal de SpringerLink pone a disposición la opción de descargar un archivo de texto (en formato csv) que contiene todos los resultados de la búsqueda, el cual facilita el procesamiento de los datos y permite realizar una sola llamada del servidor a la página y una sola llamada de la aplicación al servidor, gracias a esta alternativa se reduce el tiempo consulta de textos en este portal.

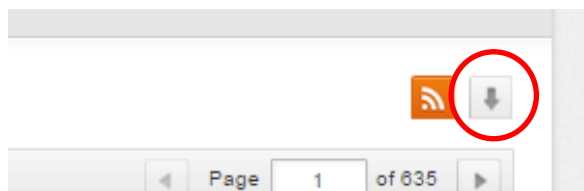


Figura 35: Opción de descarga de resultados de SpringerLink.

Fuente: SpringerLink, 2015.

El archivo que se genera con la consulta de textos en SpringerLink contiene las siguientes columnas: título del ítem, título de la publicación, título de la serie, volumen, tema, DOI (Identificador de objeto digital), autores, año de publicación, URL y tipo de documento. Con este archivo se obtiene todos los datos necesarios para generar citas de cualquiera de los documentos que devuelve la búsqueda.

Item Title	Publication Title	Book Series Title	Journal Volume	Journal Issue	Item DOI	Authors	Publication Year	URL	Content Type
Android	Pro Smartphone Cross-Platform Development				10.1007/978-Sarah AllenVid		2010	http://l	Chapter
Android	Pro Web Gadgets				10.1007/978-1-4302-2552-2_		2009	http://l	Chapter
Android Application Sketch Book					10.1007/978-Dean Kaplan		2011	http://l	Book
Beginning Android 4 Games Development					10.1007/978-Mario Zechnerl		2011	http://l	Book
Pro Android Wearables					10.1007/978-Wallace Jackso		2015	http://l	Book
Beginning Android Games					10.1007/978-Mario Zechner		2011	http://l	Book
Decompiling Android					10.1007/978-Godfrey Nolan		2012	http://l	Book
Android Apps for Absolute Beginners					10.1007/978-Wallace Jackso		2014	http://l	Book
The Business of Android Apps Development					10.1007/978-Mark Rollins		2011	http://l	Book
Beginning Android 3D Game Development					10.1007/978-Robert Chin		2014	http://l	Book
Pro Android UI					10.1007/978-Wallace Jackso		2014	http://l	Book
Learn Android App Development					10.1007/978-Wallace Jackso		2013	http://l	Book
Android Apps Security					10.1007/978-Sheran Gunase		2012	http://l	Book
Taking Your Android Tablets to the Max					10.1007/978-Russell Holly		2012	http://l	Book
Learn HTML5 and JavaScript for Android					10.1007/978-Gavin Williams		2012	http://l	Book
Android Fragments					10.1007/978-Dave MacLeanS		2014	http://l	Book
Learn Android Studio					10.1007/978-Adam GerberCl		2015	http://l	Book
Beginning Android 2					10.1007/978-Mark L. Murphy		2010	http://l	Book
Android Arcade Game App					10.1007/978-J. F. DiMarzio		2012	http://l	Book
Learn Java for Android Development					10.1007/978-Jeff Friesen		2014	http://l	Book
Migrating to Swift from Android					10.1007/978-Sean Liao		2014	http://l	Book
Android Essentials					10.1007/978-Chris Haseaman		2008	http://l	Book
Pro Android					10.1007/978-Sayed Y. Hashir		2009	http://l	Book
Pro Android Augmented Reality					10.1007/978-Raghav Sood		2012	http://l	Book
Pro Android Python with SL4A					10.1007/978-Paul Ferrill		2011	http://l	Book
Android Application Development for the Intel® Platform					10.1007/978-Ryan CohenTac		2014	http://l	Book
Android Apps for Absolute Beginners					10.1007/978-Wallace Jackso		2011	http://l	Book
Beginning Android C++ Game Development					10.1007/978-Bruce Sutherlai		2013	http://l	Book
Pro Android Media					10.1007/978-Shawn Van Eve		2009	http://l	Book
Pro Android 2					10.1007/978-Sayed Y. Hashir		2010	http://l	Book
Practical Android Projects					10.1007/978-Lucas JordanDit		2011	http://l	Book

Figura 36: Archivo de resultados de SpringerLink.

Fuente: SpringerLink, 2015.

Por medio de los siguientes comandos en Java, el servidor es capaz de ingresar al sitio web, ingresar el texto de búsqueda y descargar el archivo que contiene los resultados:

```
URL url = new URL("http://link.springer.com/search/");
Document doc = Jsoup.parse(url, 300000);
Elements buscar = doc.select("search");
buscar.val("android");
Elements botonBuscar = doc.select("submit");
Elements linkDescarga = doc.select("a[href$=csv]");
File csv=new File("temp.csv");
URL urlDescarga=new URL (linkDescarga.val());
FileUtils.copyURLToFile(urlDescarga, csv);
```

Figura 37: Código para búsqueda en SpringerLink

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

De esta manera, el proceso que se realiza para consultas en SpringerLink tiene el siguiente orden:

1. El usuario consulta en la aplicación.
2. El texto es enviado al servidor.
3. El servidor se conecta a la página de consulta de SpringerLink.
4. El servidor realiza la consulta en el sitio web de SpringerLink.
5. El servidor obtiene el archivo con resultados de la consulta.
6. El servidor procesa el archivo con los resultados.
7. El servidor devuelve los datos de la consulta a la aplicación en formato JSON.
8. La aplicación procesa y presenta los resultados.

## Integración de la librería eBrary

El sitio web de Ebrary<sup>41</sup> también brinda un campo de búsqueda en su página principal, por medio de este campo se puede ingresar cualquier texto que se desee buscar en esta librería. Una vez que se realiza la búsqueda, la página muestra los resultados que coincidan con el texto ingresado, al igual que el caso anterior se muestra un número limitado de resultados por página (en este caso se presentan 15) por lo que también se buscó una alternativa que permita obtener una mayor cantidad de resultados mediante una sola llamada.

The screenshot shows the eBrary website interface. At the top, there is a navigation bar with 'Home', 'Search', 'QuickView', and 'Bookshelf'. A search bar contains the text 'android'. Below the search bar, there are tabs for 'TITLE RESULTS', 'CHAPTER RESULTS', and 'SEARCHES'. A filter section allows users to focus their search using 'Any of the selected subjects' or 'All of the selected subjects', with various subject categories like 'Android (Electronic Resource)', 'Tablet Computers', 'Security Measures', etc. The main content area displays search results for 'android', showing 15 of 14,649 documents. Three results are visible: 'Expert Android' by Komatinen, Satya, MacLean, Dave, and Kanakala, Phani; 'Android Cookbook' by Darwin, Ian E.; and 'Android Receptury' by Helton. Each result includes a small image, title, author, publisher, date published, and subjects. To the right of each result, there is a box containing document ID, pISBN, eISBN, single-user price, and loan information. At the bottom of the page, there are options to 'Save Search' and 'Export Search Results', and a 'View' dropdown menu set to 'Short'.

Figura 38: Pagina de resultados de búsqueda de eBrary.

Fuente: eBrary, 2015.

En este caso también se pudo obtener un archivo con los primeros 300 documentos que coincidan con la búsqueda, cantidad suficiente para presentar resultados en la aplicación.

<sup>41</sup> Sitio oficial eBrary. <http://www.ebrary.com/>, 2015.

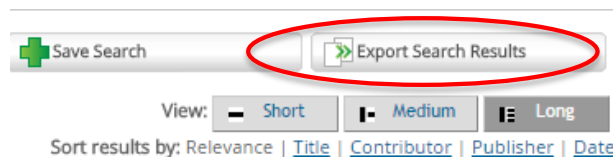


Figura 39: Opción de descarga de resultados de eBrary.

Fuente: eBrary, 2015.

El archivo de resultados que se genera en el sitio web de esta librería contiene las siguientes columnas que se utilizarán dentro de la aplicación: autor, editorial, año de publicación y URL, estos datos serán utilizados para generar las citas bibliográficas.

Doc ID	pISBN	eISBN	Title	Author	Publisher	Published year	LC Call Number	Single-User-Price	Three-User-Price	Multi-User-Price	URL
5007507	9,7819E+12		Phillip K. D Butler, Andre Pocket Essential			2001	P53554.I3 -- Z8 2000eb				http://site.e
5007501	9,7819E+12		Ridley Scd Robb, Brian J. Pocket Essential			2001	PN1998.3.S393 -- R633 2001eb				http://site.e
5007097	9,78031E+12	9,78031E+12	Transreali Broderick, Da Greenwood Pres			2000	PN3433.5 -- .B76 2(\$145.00			\$217.50	http://site.e
5006360	9,78052E+12	9,78051E+12	Beyond Ai Carroll, Noel Cambridge Univ			2001	BH39 -- .C3752 200.\$195.00				http://site.e
5006092	9,78042E+12		Philosoph Falzon, Christ Routledge			2002	BD21 -- .F34 2002e(\$113.00		\$141.25		http://site.e
5005534	9,78188E+12		Explanatic Louch, Alfred Cybereditions Ci			2000	H61 -- .L67 2000eb \$15.95	\$19.94	\$23.92		http://site.e
5004180	9,78186E+12	9,7802E+12	Theorizing Adam, Barbar UCL Press			1995	HM101 -- .T4743 19 \$131.95	\$164.94			http://site.e
5004050	9,78185E+12	9,78185E+12	Self Help Matthews, Jo Jessica Kingsley			2000	RJ506.A9 -- .M38 2(\$29.95	\$37.44	\$44.92		http://site.e
5004048	9,78185E+12	9,78185E+12	Learning t Stanton, Mike Jessica Kingsley			2000	RC553.A88 -- S83 2(\$24.95	\$31.19	\$37.42		http://site.e
5004020	9,78185E+12	9,78185E+12	Key Ideas Stuart-Hamilt Jessica Kingsley			1996	BF121 -- .S883 1995 \$29.95	\$37.44	\$44.92		http://site.e
5003998	9,78031E+12	9,78031E+12	100 More McDougald, D Greenwood Pres			1999	Z710 -- .M19 1999e \$48.95		\$73.42		http://site.e
5003991	9,7809E+12	9,78031E+12	Tropics of West, Alan Greenwood Pres			1997	PQ7378 -- .W4 1997 \$122.00		\$183.00		http://site.e
5003800	9,78042E+12	9,7802E+12	On Film Mulhall, Step Routledge			2002	PN1997.A32253 -- F \$100.00		\$125.00	\$150.00	http://site.e
5003203	9,78193E+12		Windows Proffitt, Brian Paraglyph Press			2001	QA76.76.O63 -- P7(\$29.99		\$44.98		http://site.e
5001435	9,78042E+12	9,7802E+12	Visual Dig Darley, Andre Routledge			2000	GV1469.17.563 -- D \$165.00	\$206.25	\$247.50		http://site.e
5000368	9,78185E+12	9,78185E+12	Asperger: Attwood, Ton Jessica Kingsley			1997	RC553.A88 -- A825 \$19.95	\$24.94	\$29.92		http://site.e
2004755	9,78157E+12	9,78157E+12	CyberCitie Boyer, M. Chr Princeton Archit			1996	QA76.9.C66 -- B69 \$35.00	\$43.75	\$52.50		http://site.e
2004750	9,78157E+12	9,78157E+12	Plumbing Friedman, Da Princeton Archit			1997	NA2542.35 -- .P58 \$19.95	\$24.94	\$29.92		http://site.e
2004687	9,78042E+12	9,7802E+12	Contestec Holliday, Rutl Routledge			2001	HM636 -- .H65 2001 \$175.00	\$218.75	\$262.50		http://site.e
2002666	9,78042E+12	9,7802E+12	Virtual Ge Green, Eileen Routledge			2001	HQ1233 -- .V57 200 \$195.00	\$243.75	\$292.50		http://site.e
2001123	9,78075E+12	9,78185E+12	Alien Ider Whelehan, In Pluto Press			1999	PN1995.9.A47 -- A4 \$90.00	\$112.50	\$135.00		http://site.e
2000642	9,78052E+12	9,78051E+12	Figments Cohen, Jack Cambridge Univ			1997	BF311 -- .S679 1997 \$125.00				http://site.e
11095911	9,78023E+12	9,78023E+12	Tokyo Cyt Brown, Steve Palgrave Macmil			2015	NC1766.J3BPN \$110.00	\$165.00			http://site.e
11095898	9,78083E+12	9,78083E+12	Internet F Romanosky, S RAND, National :			2015	TK5105 \$9.95	\$12.44	\$14.92		http://site.e
11095893	9,78114E+12	9,78114E+12	The Palgr: Hauskeller, N Palgrave Macmil			2015	\$185.00	\$277.50			http://site.e
11095856	9,78114E+12	9,78114E+12	Palgrave S Smith, Russel Palgrave Macmil			2015	\$100.00	\$150.00			http://site.e
11095832		9,78163E+12	Dictionary Watson, Jame Bloomsbury Acai			2015	\$102.00	\$127.50	\$153.00		http://site.e
11095831		9,7815E+12	Thinking H Schmerheim, Bloomsbury Acai			2015	\$140.00	\$175.00	\$210.00		http://site.e
11095829		9,7815E+12	The End o Schaberg, Chr Bloomsbury Acai			2015	\$57.00	\$71.25	\$85.50		http://site.e
11095782	9,78147E+12	9,78147E+12	100 Ideas Bunce, Steve Bloomsbury Edu			2015	\$51.00	\$63.75	\$76.50		http://site.e
11095738	9,78147E+12	9,78147E+12	New Hori; Ara?jo, Susan Bloomsbury Acai			2015	\$140.00	\$175.00	\$210.00		http://site.e
11095371	9,78163E+12	9,78163E+12	Synthesis Kehtarnavaz, Morgan & Claypc			2015	TK5102.9 -- .K484 2 \$150.00	\$187.50	\$225.00		http://site.e

Figura 40: Archivo de resultados de eBrary.

Fuente: eBrary, 2015.

Por medio de los siguientes comandos en Java, el servidor es capaz de ingresar al sitio web de eBrary, ingresar el texto de búsqueda y descargarse el archivo con los resultados:

```

URL url = new URL("http://site.ebrary.com/lib/alltitles");
Document doc = Jsoup.parse(url, 300000);
Elements buscar = doc.select("input#text");
buscar.val("android");
Elements botonBuscar = doc.select("input#search");
Elements linkDescarga = doc.select("#downloadCsv");
File csv=new File("temp.csv");
URL urlDescarga=new URL (linkDescarga.attr("target"));
FileUtils.copyURLToFile(urlDescarga, csv);

```

Figura 41: Código para búsqueda en eBrary.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

De tal manera que los pasos que se realizan para consultas en eBrary quedaron de la siguiente manera:

1. El usuario consulta en la aplicación.
2. El texto es enviado al servidor.
3. El servidor se conecta a la página de consulta de eBrary.
4. El servidor realiza la consulta en el sitio web de eBrary.
5. El servidor obtiene el archivo con resultados de la consulta.
6. El servidor procesa el archivo con los resultados.
7. El servidor devuelve los datos de la consulta a la aplicación en formato JSON.
8. La aplicación procesa y presenta los resultados.

## Integración de la Biblioteca General de la PUCE

La Pontificia Universidad Católica del Ecuador cuenta con una página web exclusiva para buscar textos en su biblioteca<sup>42</sup>, en esta página se puede ingresar el texto a buscar y el número específico de resultados que se desea obtener.



Buscar  Buscar Borrar Nueva búsqueda  
Resultados 10 ▼

Figura 42: Pagina de búsqueda de la PUCE

Fuente: PUCE, 2015.

Una vez que se realiza la consulta, el sitio web muestra una página con los resultados de la búsqueda. En esta página se presentan solamente los siguientes campos: título, autor y año. Sin embargo, cada texto cuenta con un link para consultar el detalle del documento y los siguientes campos adicionales: piso en el que se encuentra el texto dentro de la biblioteca, código de la biblioteca, país, editorial, páginas, dimensiones y disponibilidad.

---

<sup>42</sup> Sitio oficial Biblioteca PUCE. <http://www.puce.edu.ec/sitios/biblioteca/>, 2015.

Buscar

Resultados  ▼

Resultados para android :11

- [ver](#) **El gran libro de programación avanzada con Android**  
Amaro Soriano, José Enrique
- [ver](#) **Análisis y diseño de un Sistema Domótico, su administración y control a través de un dispositivo móvil con Sisteam Operativo Android**  
Cajas Flores, Leslie Elizabeth
- [ver](#) **Análisis y aplicación de software para la recuperación forense de evidencia digital en dispositivos móviles Android**  
Guerra Castro, Christian David
- [ver](#) **Android : curso práctico para todos los niveles : cómo extraer todo el potencial del smartphone**  
Muñiz Troyano, Javier
- [ver](#) **HTML5 para iOS y Android**  
Nixon, Robin
- [ver](#) **Desarrollo de aplicaciones para Android**  
Ribas Lequerica, Joan
- [ver](#) **Desarrollo de una aplicación para equipos Android, basada en geolocalización para obtener información de atractivos turísticos en la ciudad de Tulcán**  
Robles Ruano, Juan Carlos
- [ver](#) **El gran libro de Android**  
Tomás Gironés, Jesús
- [ver](#) **El gran libro de Android**  
Tomás Gironés, Jesús
- [ver](#) **El gran libro de Android avanzado / Jesús Tomás Gironés ... [et al.]**  
Tomás Gironés, Jesús
- [ver](#) **Crea tus propias aplicaciones Android con google app inventor / David Wolber ... [et al.]**  
Wolber, David

Figura 43: Pagina de resultados de la PUCE

Fuente: PUCE, 2015.

Gracias a esto fue necesario obtener cada uno de estos links desde el servidor para poder entrar al detalle de cada documento y de esta manera obtener todos los datos necesarios de los textos consultados.

[Piso : PB](#)

2013

978-607-707-551-6

005.43/Am13g

Amaro Soriano, José Enrique

El gran libro de programación avanzada con Android

[México, México] : Alfaomega, c2013

398 p. ; 23 cm.\*il.

Figura 44: Información por documento de la PUCE

Fuente: PUCE, 2015

Este formato de los resultados presentó una dificultad especial, ya que no se especifica el dato al que pertenece cada campo que devuelve la búsqueda de un texto, por lo que para cada usuario debe inferir esto. En la aplicación se resolvió esto tomando en cuenta el orden y el contenido de los datos. Por ejemplo, se notó que siempre el primer texto que se presenta es el piso, el autor siempre está antes que el título y la editorial cuenta con dos puntos en su contenido y siempre se presenta después del título. También cabe recalcar que no siempre se presenta el mismo detalle de datos para cada texto, al final lo que se utilizó para poder inferir a que se refiere cada campo fue el uso del orden de los datos antes descritos y adicionalmente se utilizaron expresiones regulares para poder determinar el significado de cada uno, como se muestra a continuación.

```

public boolean esPaginasCM(String txt) {
    boolean retorno = false;

    if (txt.indexOf("p.") > -1 || txt.indexOf("cm.") > -1) {
        retorno = true;
    }

    return retorno;
}

public boolean isNumeric(String str) {
    return str.matches("\\d+");
}

public boolean esAutor(String str) {
    return str.matches("[A-zÀ-ÿ]+\\s?([A-zÀ-ÿ])?,(\\s?[A-zÀ-ÿ])");
}

public boolean isISO(String str) {
    return str.matches("\\d+-\\d+-\\d+.*");
}

```

Figura 45: Código para el campo en resultados de la PUCE.

Fuente: Estefano Valdez, Sebastián Dávalos. Eclipse, 2015.

De esta manera, el proceso que se estableció para realizar consultas en la Biblioteca General de la PUCE es el siguiente:

1. El usuario consulta en la aplicación.
2. El texto es enviado al servidor.
3. El servidor se conecta a la página de consulta de la Biblioteca de la PUCE.
4. El servidor realiza la consulta en el sitio web de Biblioteca de la PUCE.
5. El servidor obtiene los links con el detalle de cada documento que coincide con la búsqueda.
6. El servidor ingresa a cada link e identifica los datos del detalle de los documentos.
7. El servidor devuelve los datos de la consulta a la aplicación en formato JSON.
8. La aplicación procesa y presenta los resultados.

## Funcionalidades desarrolladas en la aplicación

De acuerdo a la etapa de diseño realizada en el capítulo 2, la aplicación muestra la siguiente interfaz cuando se abre:

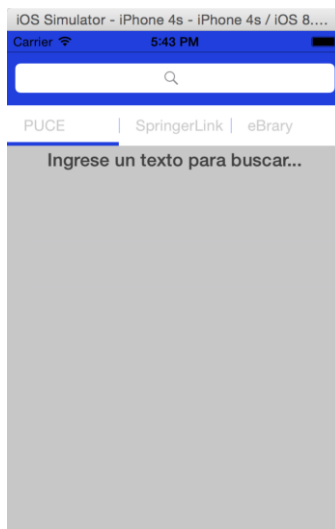


Figura 46: Interfaz inicial.

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015

Todos los componentes están basados en el ancho y alto (en orientación vertical) del dispositivo, utilizando los siguientes comandos:

```
Ti.UI.orientation = Ti.UI.PORTRAIT;  
var alto = Ti.Platform.displayCaps.platformHeight;  
var ancho = Ti.Platform.displayCaps.platformWidth;
```

Figura 47: Código para obtener tamaño

Fuente: Estefano Valdez, Sebastián Dávalos. Titanium Studio, 2015.

De esta manera los componentes se adaptan dependiendo del tamaño de cada dispositivo. Adicionalmente, en esta interfaz el usuario puede ingresar un texto para buscar documentos en las tres librerías, por las cuales se puede navegar por medio de las tres pestañas en la aplicación.

A continuación se muestran las principales funcionalidades desarrolladas en la aplicación, entre las cuales destacan la búsqueda de documentos en las bibliotecas: PUCE, Springer y eBrary:

F01 Buscar documentos en la biblioteca de la PUCE.

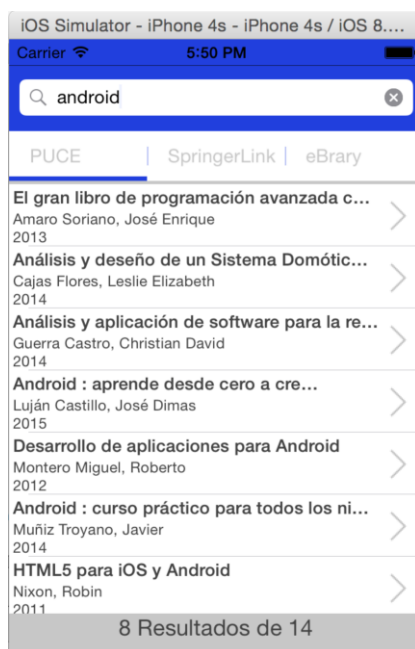


Figura 48: Interfaz documentos PUCE.

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015.

Los datos para la búsqueda de documentos en la biblioteca de la PUCE se reciben en el siguiente formato JSON:

```
{
  "Codigo": "005.43/Am13g ",
  "Editorial": "[México, México] : Alfaomega, c2013",
  "Piso": "Piso : PB ",
  "ISO": "978-607-707-551-6",
  "Tema": "LENGUAJES DE PROGRAMACION - ANDROID",
  "Anio": "2013",
  "Titulo": "El gran libro de programación avanzada con Android",
  "Autor": "Amaro Soriano, José Enrique",
```

```

"Aux8": "ANDROID (SISTEMA OPERATIVO) ",
"Ejemplares": "1 ",
"Aux5": "El gran libro de programación avanzada con Android",
"Disponible": "1",
"Paginas": "398 p. ; 23 cm.*il."
}

```

F02 Buscar documentos en la biblioteca virtual SpringerLink.

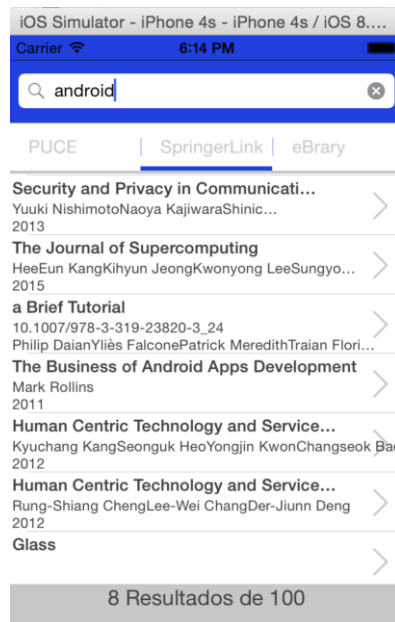


Figura 49: Interfaz documentos SpringerLink.

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015.

Los datos para la búsqueda de documentos en la biblioteca virtual de Springer se reciben en el siguiente formato JSON:

```

{
  "JournalIssue": "",
  "BookSeriesTitle": "",
  "JournalVolume": "",
  "PublicationYear": "2013",
  "Authors": "Yuuki NishimotoNaoya KajiwaraShinichi MatsumotoYoshiaki
HoriKouichi Sakurai",

```

```

"URL": "http://link.springer.com/chapter/10.1007/978-3-319-04283-1_25",
"ItemTitle": "Detection of Android API Call Using Logging Mechanism within
Android Framework",
"PublicationTitle": "Security and Privacy in Communication Networks",
"ContentType": "Chapter",
"ItemDOI": "10.1007/978-3-319-04283-1_25"
}

```

F03 Buscar documentos en la biblioteca virtual eBrary.

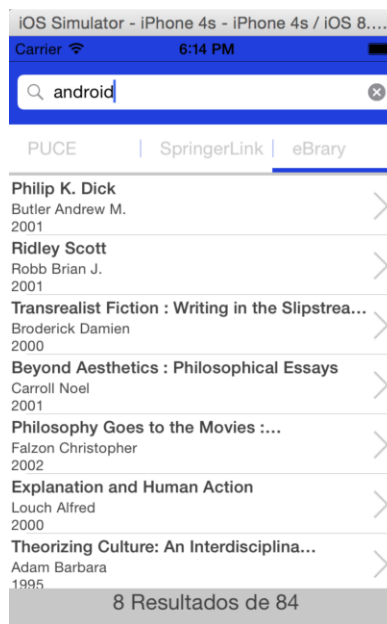


Figura 50: Interfaz documentos eBrary

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015.

Los datos para la búsqueda de documentos en la biblioteca de eBrary se reciben en el siguiente formato JSON:

```

{
  "Multi-User-Price": "",
  "Publisher": "Pocket Essentials",
  "Single-User-Price": "",
  "LCCallNumber": "PS3554.I3 -- Z8 2000eb",
  "pISBN": "9781903047293",

```

```

"Three-User-Price": "",
"?DocID": "5007507",
"Author": "Butler Andrew M.",
"URL": "http://site.ebrary.com/lib/alltitles/docDetail.action?docID=5007507",
"Publishedyear": "2001",
"eISBN": "",
"Title": "Philip K. Dick"
}

```

F04 Visualizar descripción del documento y disponibilidad.

Por medio de esta funcionalidad el usuario puede ingresar a cualquiera de los documentos que se muestren en la consulta y obtener datos adicionales, tales como: la editorial, el tema, el número de páginas, entre otros. Para el caso de la biblioteca de la PUCE se muestra adicionalmente la disponibilidad, el número de ejemplares y el piso donde se encuentra físicamente el documento.

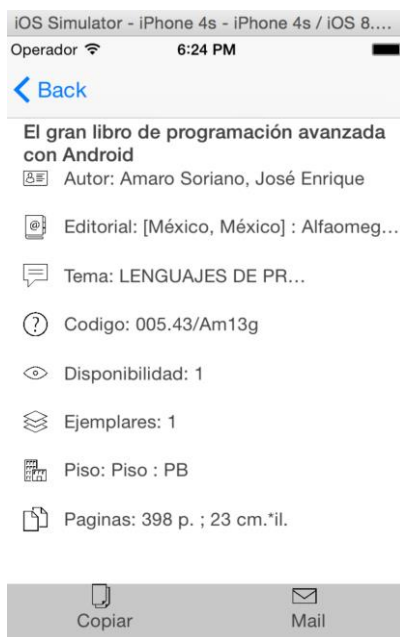


Figura 51: Interfaz detalle de documentos.

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015.

## F05 Generar citas bibliográficas.

La funcionalidad de generar citas bibliográficas se logró concatenando los campos: autor, título, editorial, país y año, de tal manera que el texto generado quede en el formato mencionado en el capítulo 2:

- Apellidos y nombre del autor. Título. Lugar: Editorial, año de edición. ISBN (opcional).

De esta manera el usuario puede generar una cita bibliográfica de cualquier texto seleccionado y posteriormente usarla en cualquier otra aplicación como mensajes, calendario o notas del dispositivo.

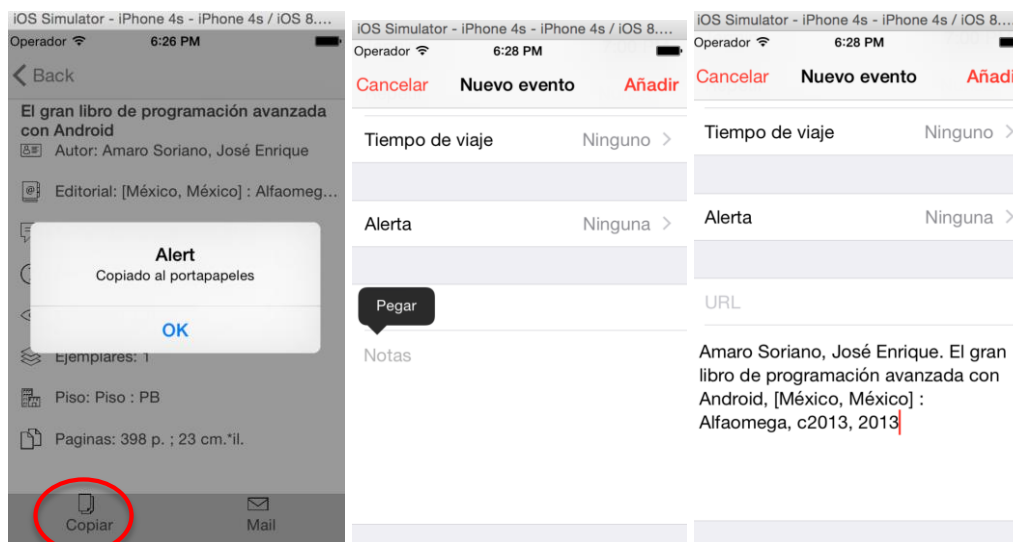


Figura 52: Generación de cita al portapapeles.

Fuente: Estefano Valdez, Sebastián Dávalos. iOS Simulator – iPhone 4s, 2015.

Los siguientes comandos permiten copiar texto al portapapeles del dispositivo:

```
Ti.UI.Clipboard.clearText();
alert("Copiado al portapapeles");
Ti.UI.Clipboard.setText(citaPuce);
```

Figura 53: Comandos de cita al portapapeles.

Fuente: Estefano Valdez, Sebastián Dávalos. Titanium Studio, 2015.

## F06 Enviar citas por correo electrónico.

Esta funcionalidad es similar a la funcionalidad de copiar la cita al portapapeles, sin embargo, tiene la ventaja de que la aplicación abre directamente la ventana de mail con la cita copiada en el cuerpo del mail, lista para ser enviada. Para que esto funcione de manera correcta el usuario debe tener una cuenta registrada en el dispositivo.

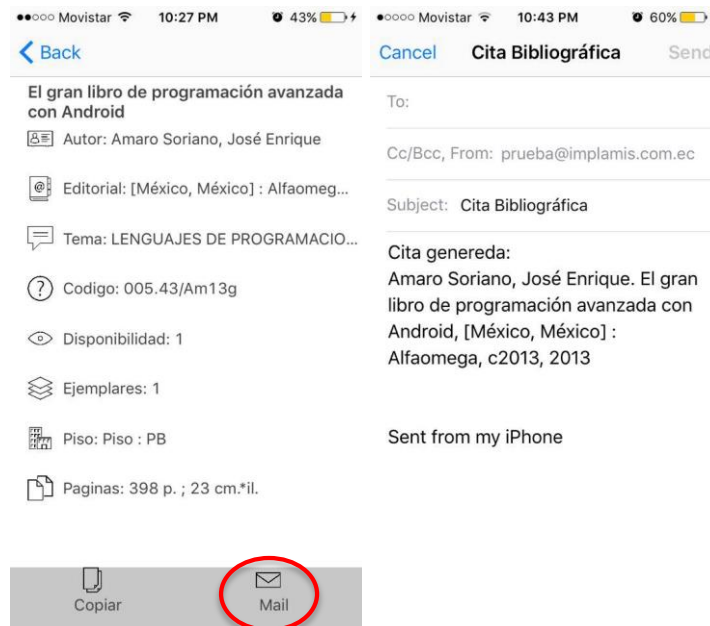


Figura 54: Generación de mail.

Fuente: Estefano Valdez, Sebastián Dávalos. iPhone 6, 2015.

Los siguientes comandos permiten generar un mail con texto:

```
var emailDialog = Ti.UI.createEmailDialog();
emailDialog.subject = "Cita Bibliográfica";
emailDialog.toRecipients = [''];
emailDialog.messageBody = 'Cita generada: '+citaPuce;
emailDialog.open();
```

Figura 55: Comandos de cita al portapapeles.

Fuente: Estefano Valdez, Sebastián Dávalos. Titanium Studio, 2015.

En caso de que algún sitio web no esté disponible se agregó un control en la aplicación notificando al usuario que debe realizar la consulta en otro momento, también se desarrolló un control para informar si se presentó algún error al procesar la información en el servidor, por ejemplo cuando un portal haya cambiado su estructura HTML. El siguiente código permite validar las verificaciones mencionadas:

```
function cargarDatos(url, enviar, callback) {
    var response = null;
    var client = Ti.Network.createHTTPClient({
        onload: function(e) {
            response = JSON.parse(this.responseText);
            if (response.success)
                callback(response);
            else
                alert("Error en el servidor: "+response.error);
        },
        onerror: function(e) {
            alert('El servicio no está disponible, intente más tarde');
        },
        timeout: 1000000
    });
    client.open("POST", servidor + url);
    client.send(enviar);
}
```

Figura 56: Controles de conexión.

Fuente: Estefano Valdez, Sebastián Dávalos. Titanium Studio, 2015.

En el siguiente capítulo se van a realizar las pruebas funcionales de aplicación, las cuales van a estar basadas en los casos de prueba elaborados en el capítulo anterior.

## CAPÍTULO 4

### PRUEBAS FUNCIONALES DE LA APLICACIÓN

En el presente capítulo se van a documentar las pruebas de la aplicación para revisar las funcionalidades desarrolladas, para esto se va a tomar en cuenta el plan de pruebas realizado en el capítulo 2. Los dispositivos que se van a utilizar para realizar este proceso son: iPhone 6, iPad 2, simulador iPhone 4s, simulador iPhone 5 y simulador iPad Air.

#### Caso de prueba (C1)

**Descripción:** Ingreso a la aplicación

**Entradas:** El usuario abre la aplicación.

**Resultado esperado:** La aplicación se abre mostrando la opción de búsqueda de documentos.

**Interfaz:** UI1

**Estatus:** Aprobado

**Capturas de pantalla:**

Simulador iPhone 4s



Figura 57: C1 - Simulador iPhone 4s.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPhone 5

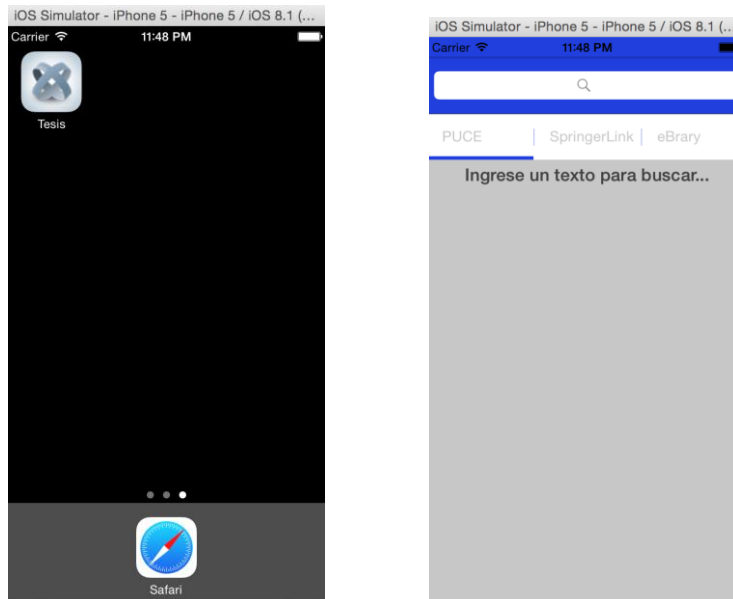


Figura 58: C1 - Simulador iPhone 5.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPad Air

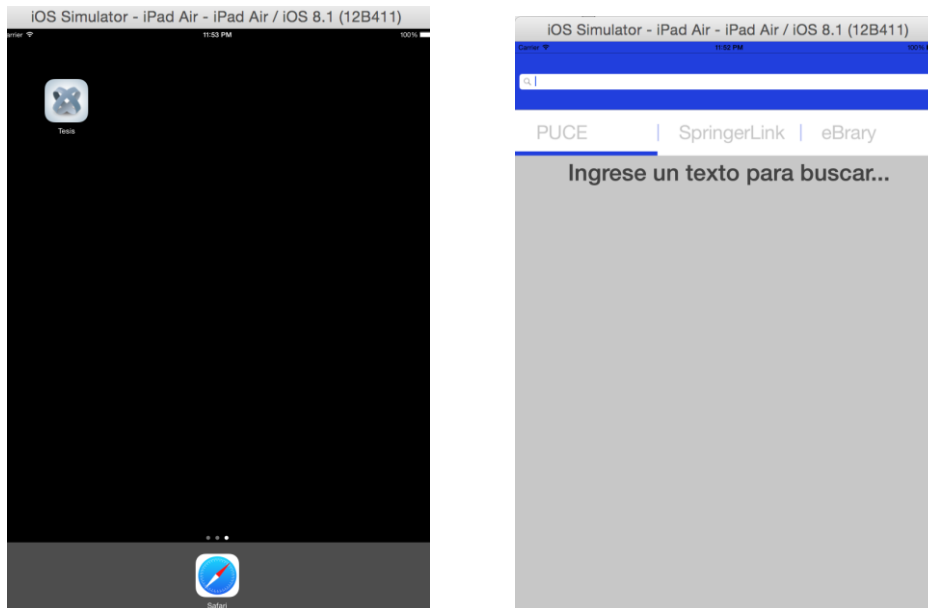


Figura 59: C1 - Simulador iPad Air.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

iPhone 6

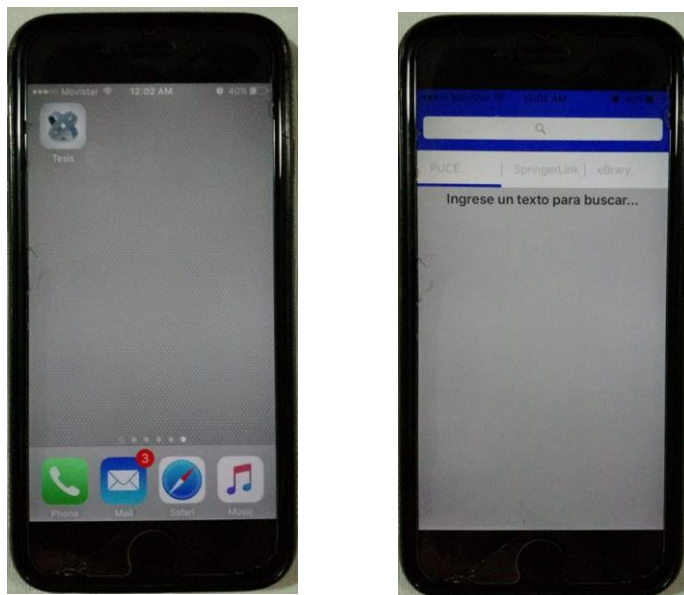


Figura 60: C1 - iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

iPad 2

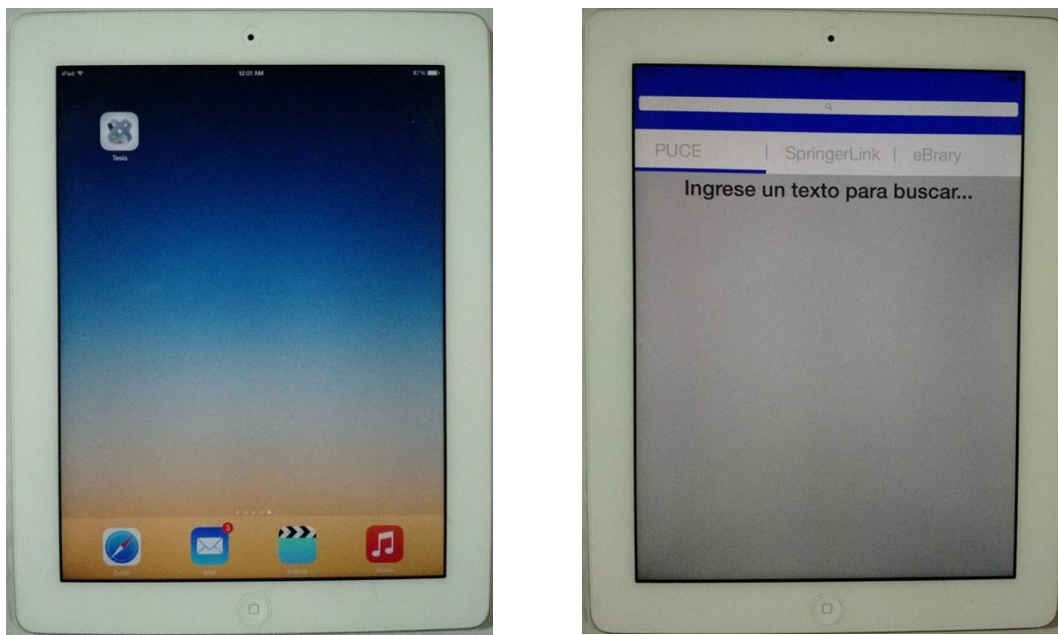


Figura 61: C1 - iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

**Comentario:** La aplicación se instaló sin ningún inconveniente en los distintos dispositivos de prueba. Una vez abierta, mostró el campo para realizar la búsqueda como se había esperado.

## Caso de prueba (C2)

**Descripción:** Búsqueda de documentos

**Entradas:** El usuario ingresa el texto de referencia para buscar en las librerías.

**Resultado esperado:** La aplicación muestra todos los documentos que tengan coincidencia con la referencia ingresada, en las librerías PUCE, Springer y eBrary. Los campos que se presentan son: título, autor y año.

**Interfaz:** UI3, UI4, UI5

**Estatus:** Aprobado

**Capturas de pantalla:**

Simulador iPhone 4s

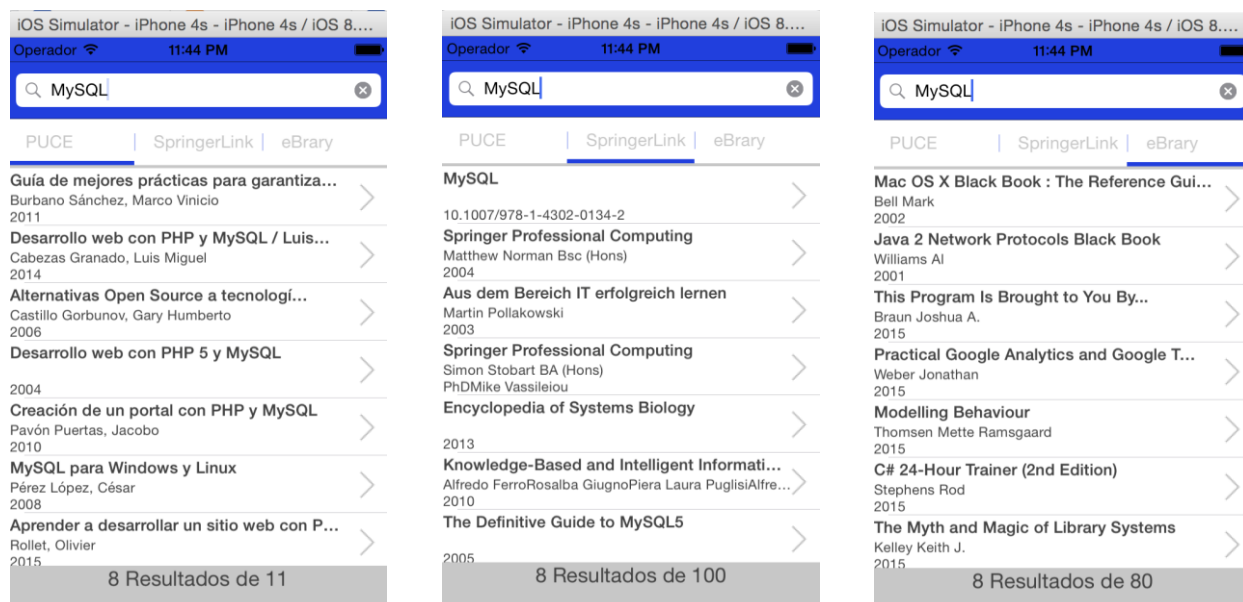


Figura 62: C2 - Simulador iPhone 4s.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPhone 5

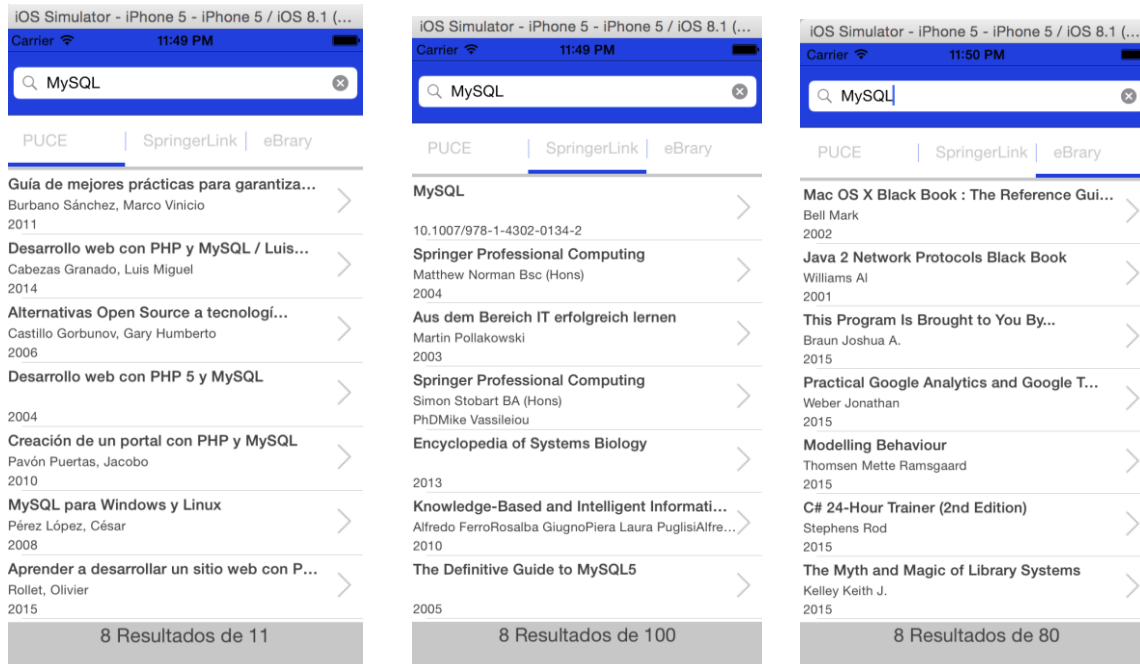


Figura 63: C2 - Simulador iPhone 5.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPad Air

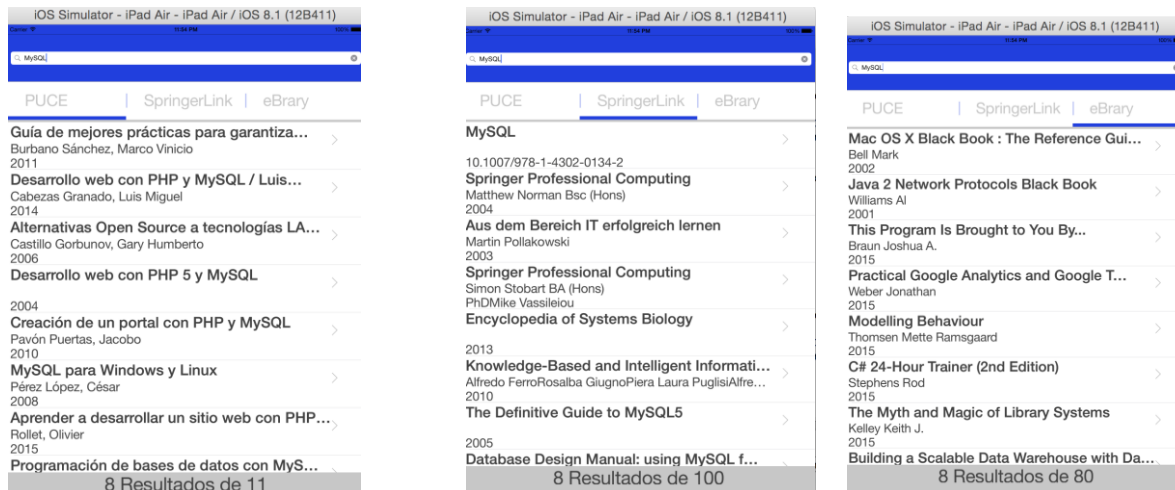


Figura 64: C2 - Simulador iPad Air.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## iPhone 6

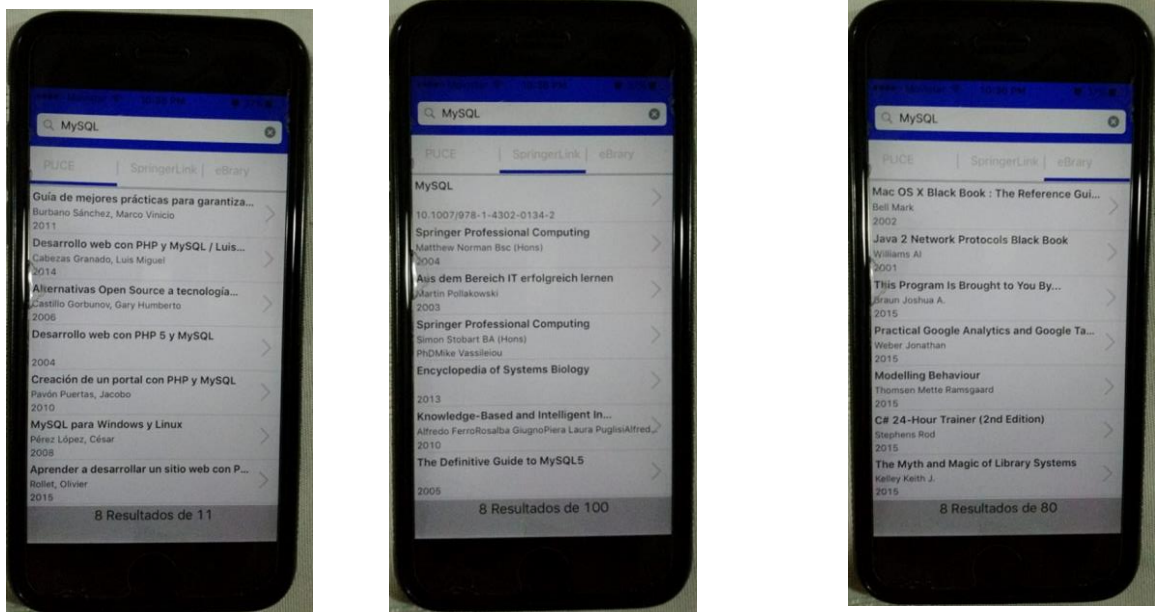


Figura 65: C2 - Dispositivo iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## iPad 2

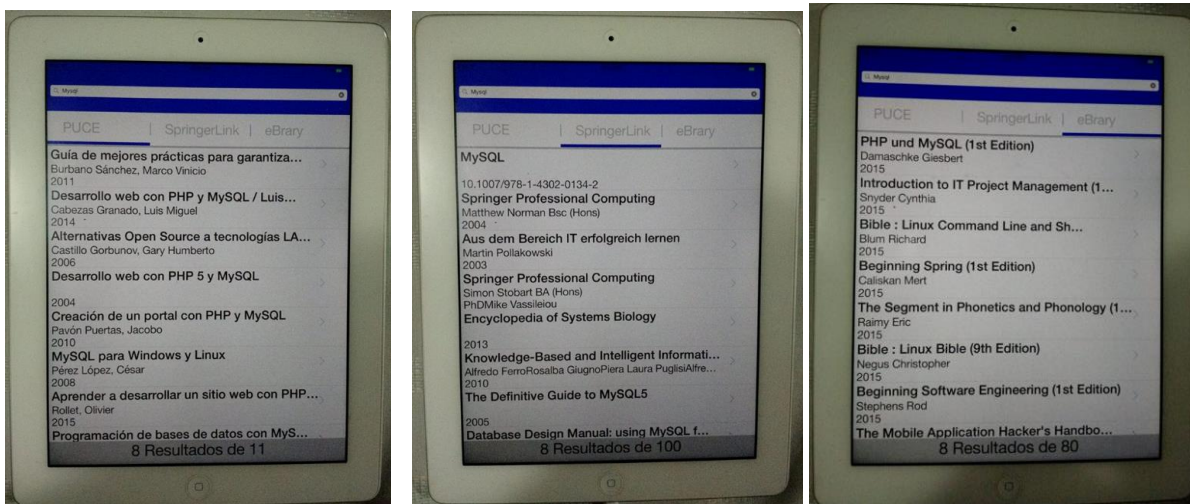


Figura 66: C2 - Dispositivo iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

**Comentario:** El texto utilizado para realizar esta búsqueda fue “MySQL”. El resultado fue positivo ya que se mostraron resultados para las tres librerías. En el caso de la PUCE se cargaron 11 resultados, en Springer 100 y en eBrary 79. Se compararon los resultados con las páginas oficiales y los documentos coinciden con los que se muestran en la aplicación.

### **Caso de prueba (C3)**

**Descripción:** No se encuentran resultados.

**Entradas:** El usuario ingresa un texto que no devuelve resultados en la búsqueda.

**Resultado esperado:** La aplicación genera un mensaje anunciando que no se encontraron resultados.

**Interfaz:** UI1

**Estatus:** Aprobado

**Capturas de pantalla:**

Simulador iPhone 4s

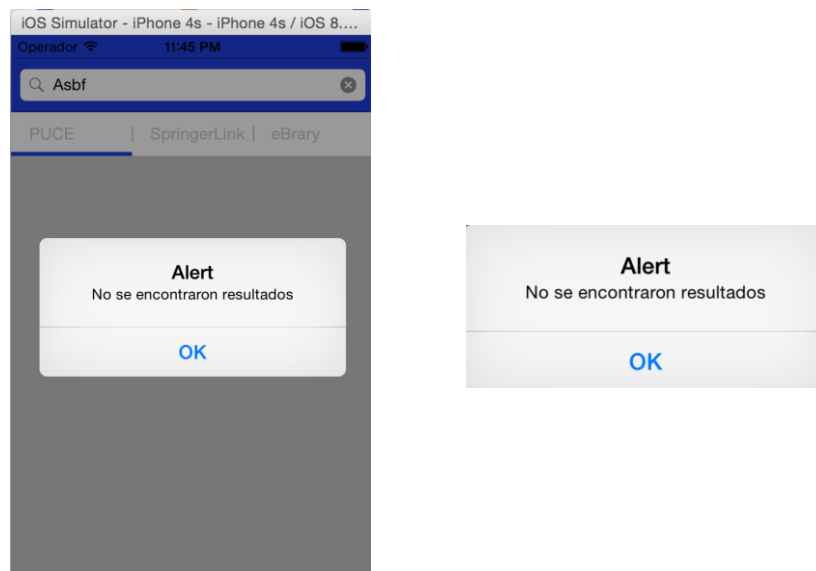


Figura 67: C3 - Simulador iPhone 4s.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPhone 5

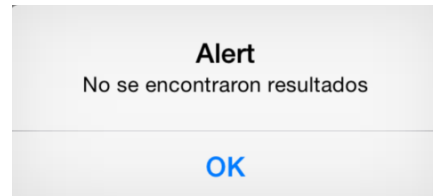
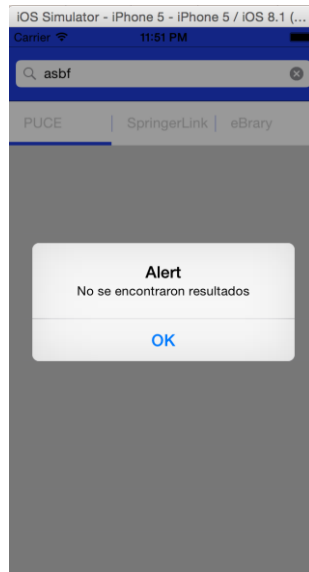


Figura 68: C3 - Simulador iPhone 5.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPad Air

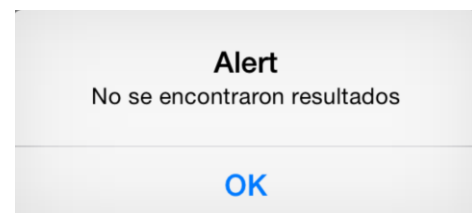
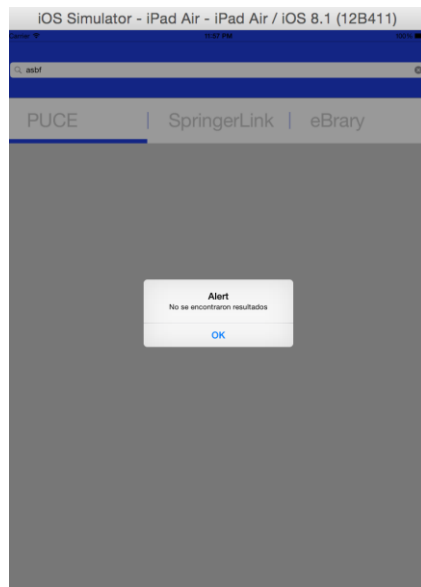


Figura 69: C3 - Simulador iPad Air.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

iPhone 6

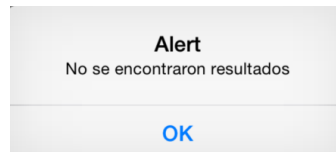
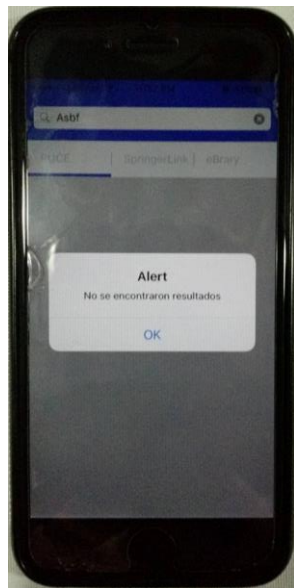


Figura 70: C3 - Dispositivo iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

iPad 2

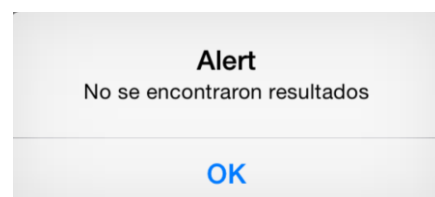
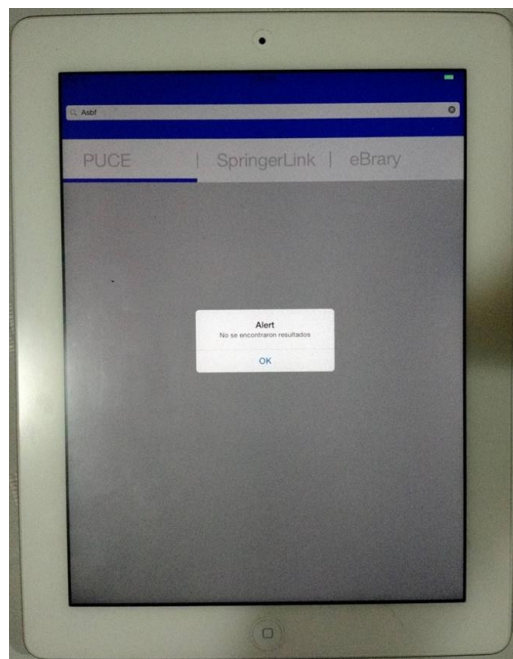


Figura 71: C3 - Dispositivo iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

**Comentario:** El texto utilizado para realizar esta búsqueda fue “Asbf” de tal manera que se reduzca la posibilidad de encontrar documentos que cumplan con esta condición. El resultado fue positivo ya que no se mostraron resultados y la aplicación mostró el mensaje respectivo notificando al usuario, al cerrar este mensaje el usuario puede volver a realizar una búsqueda. Adicionalmente, se probó que la aplicación no se pueda conectar al servicio de los portales de las bibliotecas desactivando la red del dispositivo, en este caso se presentó un mensaje que alerta al usuario para que realice la consulta en otro momento.

## Caso de prueba (C4)

**Descripción:** Mostrar detalle del documento

**Entradas:** El usuario selecciona un documento para visualizar el detalle.

**Resultado esperado:** La aplicación muestra el detalle del documento: título, autor, editorial, edición, año, ciudad, disponibilidad y campos adicionales.

**Interfaz:** UI6

**Estatus:** Aprobado

**Capturas de pantalla:**

Simulador iPhone 4s

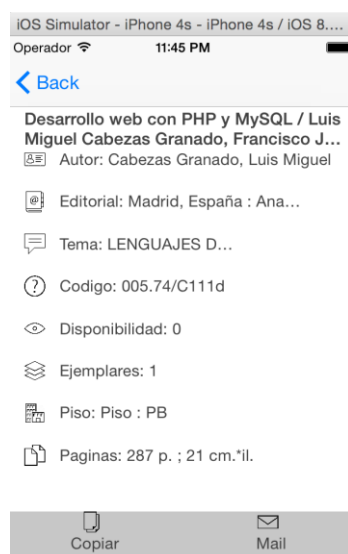


Figura 72: C4 - Simulador iPhone 4s.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPhone 5

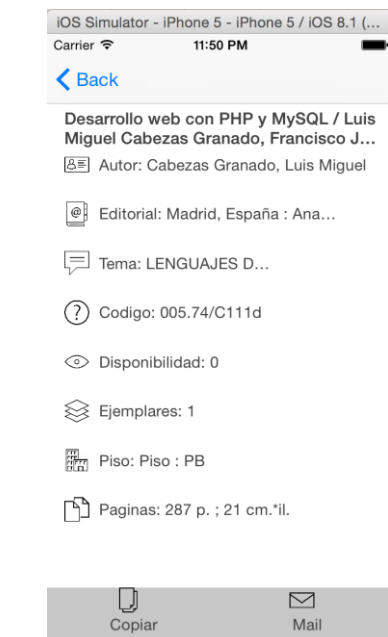


Figura 73: C4 - Simulador iPhone 5.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPad Air

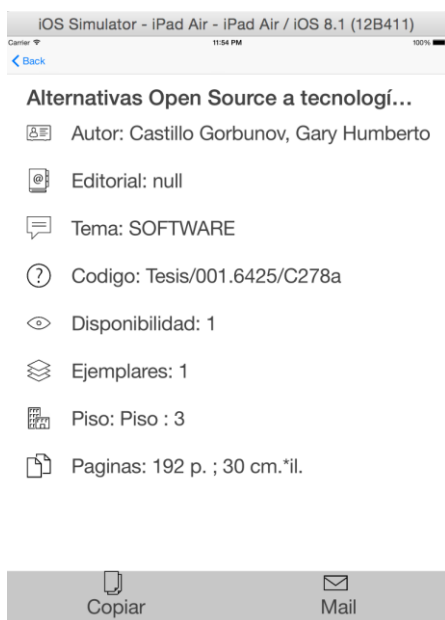


Figura 74: C4 - Simulador iPad Air.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

iPhone 6

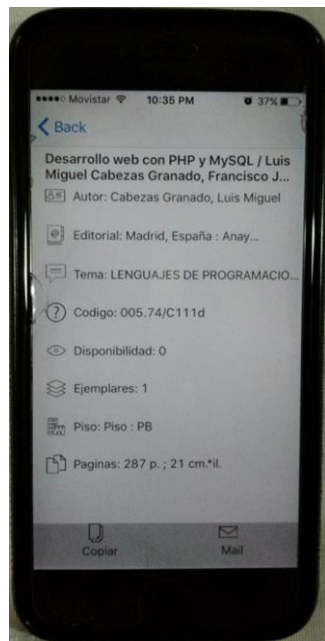


Figura 75: C4 - Dispositivo iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

iPad 2

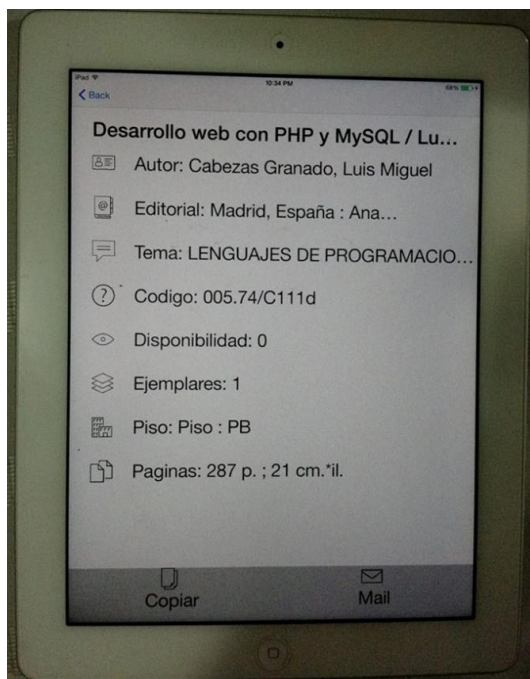


Figura 76: C4 - Dispositivo iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

**Comentario:** En esta prueba de funcionalidad se seleccionó un documento para ver sus campos en detalle. El resultado fue el esperado y adicionalmente se agregó una animación para que el cambio de ventana sea más fluido en la aplicación. Además se tomó en cuenta que no todos los documentos tienen los mismos campos, por lo que se agregó varias condiciones para mostrar diferentes elementos en esta pantalla, con un número limitado de 9 líneas de texto incluyendo el título.

## Caso de prueba (C5)

**Descripción:** Generar cita bibliográfica.

**Entradas:** El usuario selecciona la opción de generar cita.

**Resultado esperado:** La aplicación genera la cita en formato estándar y la copia al portapapeles.

**Interfaz:** UI6

**Estatus:** Aprobado

**Capturas de pantalla:**

Simulador iPhone 4s

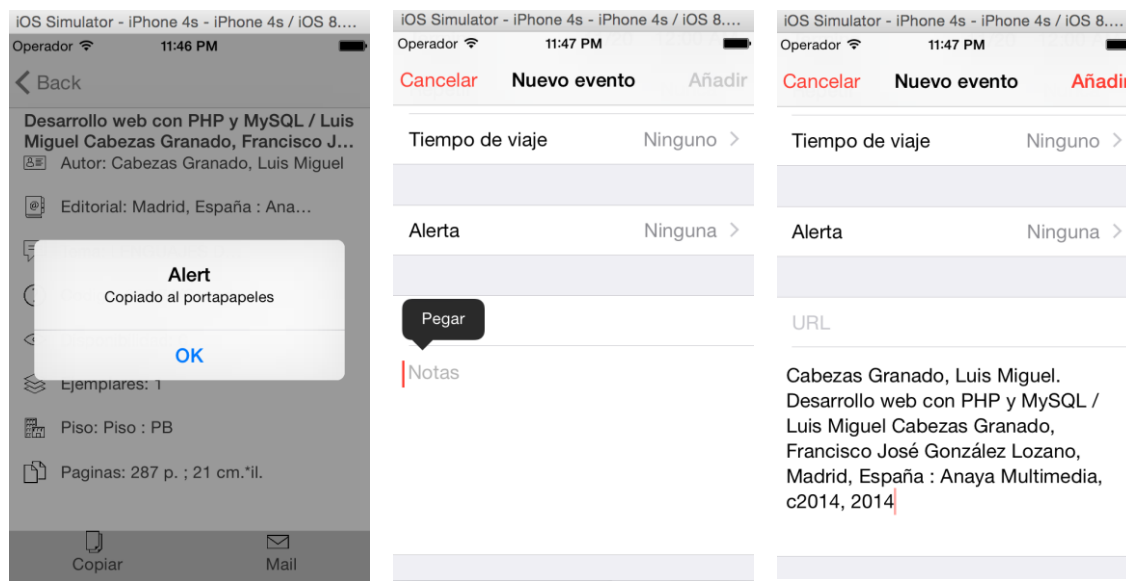


Figura 77: C5 - Simulador iPhone 4s.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPhone 5

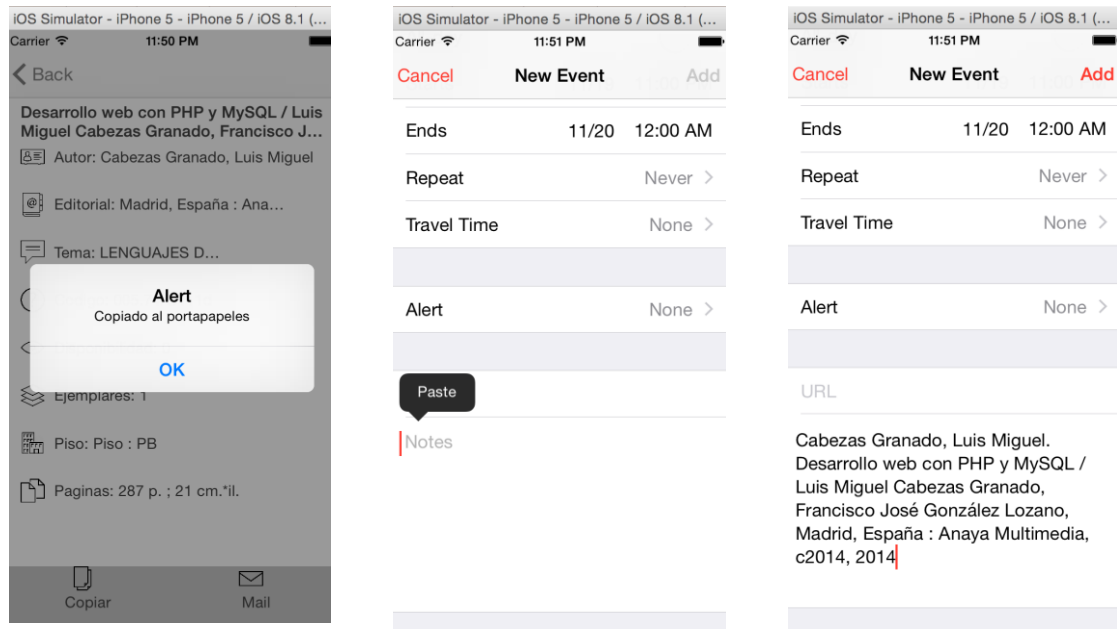


Figura 78: C5 - Simulador iPhone 5.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## Simulador iPad Air

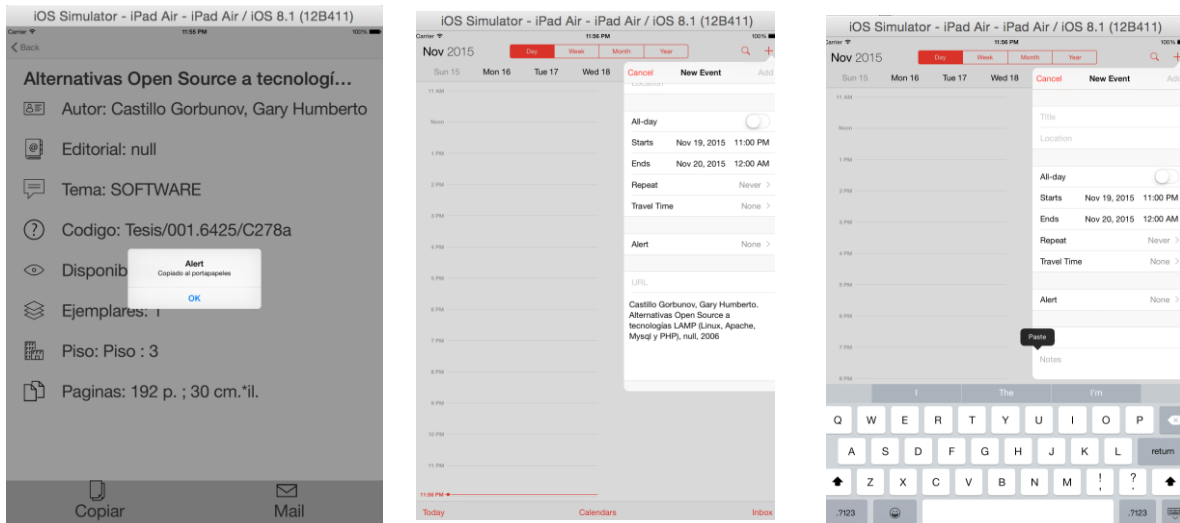


Figura 79: C5 - Simulador iPad Air.

Fuente: Estefano Valdez, Sebastián Dávalos. Simulador iOS, 2015.

## iPhone 6

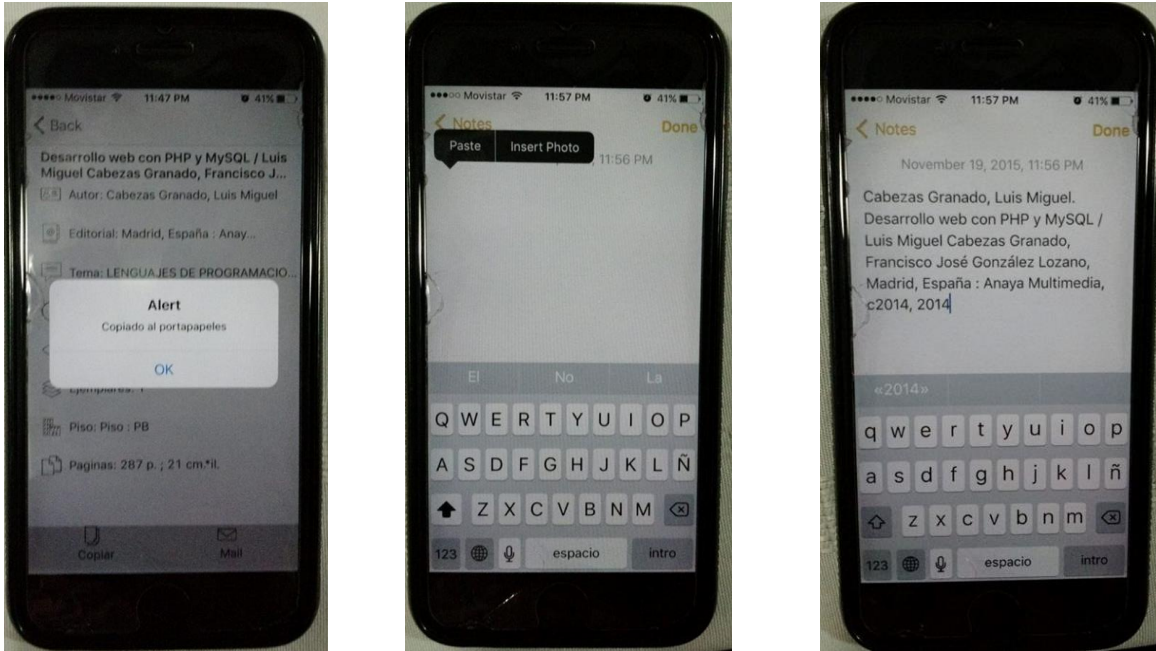


Figura 80: C5 - Dispositivo iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

## iPad 2

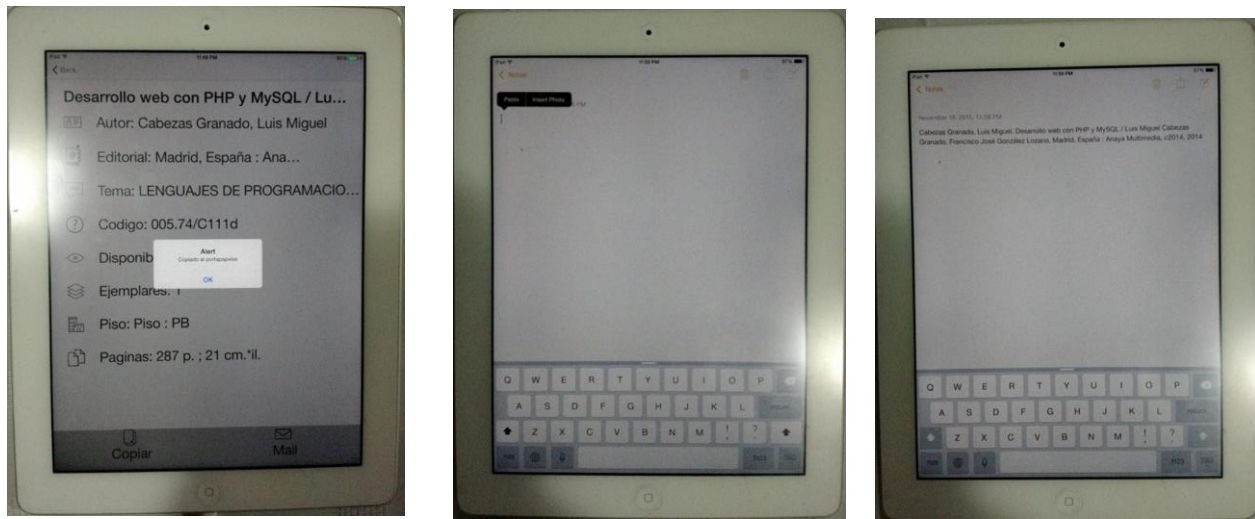


Figura 81: C5 - Dispositivo iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

**Comentario:** En esta prueba se revisó la funcionalidad de generar la cita bibliográfica y copiarla al portapapeles, el resultado también fue positivo. Una vez que la aplicación se encargó de copiar el texto, se utilizó otras aplicaciones como notas o calendario para pegar la cita y revisar que todos los campos concatenados estaban copiados.

## Caso de prueba (C6)

**Descripción:** Enviar cita por mail.

**Entradas:** El usuario selecciona la opción de enviar cita por correo electrónico.

**Resultado esperado:** La aplicación llama al cliente de correo del dispositivo con la cita generada en el cuerpo del correo.

**Interfaz:** UI6

**Estatus:** Aprobado

### Capturas de pantalla:

iPhone 6

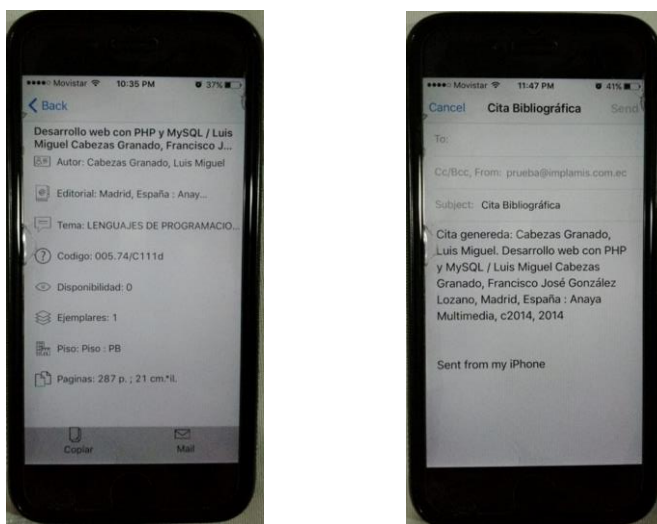


Figura 82: C6 - Dispositivo iPhone 6.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

## iPad 2

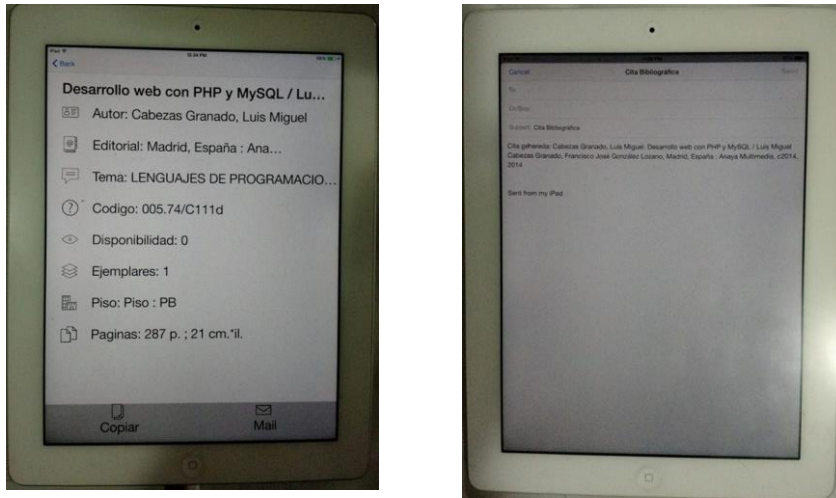


Figura 83: C6 - Dispositivo iPad 2.

Fuente: Estefano Valdez, Sebastián Dávalos. Dispositivo iOS, 2015.

**Comentario:** Esta funcionalidad se probó solamente en dispositivos físicos ya que los simuladores no tienen la aplicación de mail y por lo tanto tienen una cuenta de correo registrada. El resultado fue satisfactorio ya que la aplicación se encargó de generar un correo nuevo con la cita concatenada y copiada en el cuerpo del correo. El correo se manda desde la cuenta registrada en el dispositivo y el campo de destinatarios está vacío por defecto para que sea completado por el usuario.

## **CAPÍTULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

Para concluir, se recuerda que el objetivo de este trabajo de investigación fue proponer la implementación de una herramienta que pueda responder a algunas necesidades que tienen actualmente los estudiantes universitarios respecto a búsqueda y creación de citas bibliográficas. Esta problemática se intentó resolver a través del desarrollo de un prototipo de aplicación móvil por medio de la herramienta Titanium y un servidor en la nube de OpenShift, utilizando lenguajes de programación comunes como JavaScript del lado del cliente y Java del lado del servidor; las pruebas demostraron el buen funcionamiento del código desarrollado y la factibilidad de desarrollar aplicaciones multiplataforma. Por lo tanto, después de la realización de este trabajo se exponen a continuación una serie de conclusiones y recomendaciones.

#### **Conclusiones**

- Gracias al desarrollo de esta aplicación se logró concentrar más de 1 millón 300 mil documentos entre los cuales destacan: libros, revistas, publicaciones científicas, documentos electrónicos, hojas topográficas, diapositivas, depositarios del Fondo de Publicaciones de las Naciones Unidas, entre otros. De esta manera, los usuarios pueden encontrar una gran cantidad de documentos relacionados con su interés de investigación y generar citas bibliográficas para cualquier trabajo o consulta.
- El desarrollo de aplicaciones híbridas multiplataforma en herramientas como Titanium reduce significativamente el tiempo de implementación, ya que se puede reutilizar el código fuente y exportarlo a distintos dispositivos sin la necesidad de realizar muchas modificaciones en el mismo; sin embargo, siempre va a ser necesario tener acceso al SDK de las plataformas de interés para que el código pueda ser traducido a cada sistema operativo.
- Las metodologías ágiles en desarrollo de software facilitan el proceso gracias a su flexibilidad, orientación a la comunicación, y simplicidad en la organización de las actividades del equipo; a pesar de esto, siempre va a ser necesario utilizar algunos de los

elementos de las metodologías tradicionales (modelos en cascada, TSP) los cuales son esenciales para entender el problema que se quiere resolver, principalmente en la etapa de diseño.

- La técnica web scraping resultó ser positiva a la hora de realizar esta aplicación, ya que se logró concentrar varios sitios de consulta sin la necesidad de armar una base de datos propia. Grandes empresas como Facebook y Google tienen implementada esta técnica en sus portales como buena práctica al compartir links o realizar búsquedas en la web.
- La principal desventaja de web scraping es que la codificación se genera a partir de una estructura definida, en caso de que este diseño cambie es necesario revisar y actualizar el software. Otro inconveniente es que la extracción de información de un sitio externo va a depender del tiempo de conexión al portal y de que el servicio esté disponible por parte del proveedor de la información.
- La configuración de servidores en la nube provee una gran ventaja a los desarrolladores en términos monetarios y de gestión de tiempo, ya que existen varias empresas que ofrecen una solución lista para ser usada, no sólo en infraestructura, sino incluso en plataforma o software. Por lo tanto, una solución de computación en la nube puede apoyar a los equipos de desarrollo para que pongan mayor atención en el producto, confiando en que la solución va a ser escalable de acuerdo al uso que tenga la aplicación.
- El uso de lenguajes de programación populares como Java y JavaScript permite tener acceso a varios recursos como: tutoriales, librerías, documentos, foros y comunidades. Esto es de gran ayuda al momento de desarrollar software ya que se pueden encontrar respuestas a dificultades que surjan en el transcurso del proyecto. Por ejemplo, en este caso resultó muy útil la librería jsoup y todos los recursos en internet acerca del lenguaje Java, lo cual facilitó la implementación de web scraping de una manera mucho más rápida.

## Recomendaciones

- Al utilizar la metodología Kanban es muy importante que los miembros del equipo realicen reuniones periódicas para revisar el avance y dividir las nuevas tareas que surjan en el transcurso. En caso de que el proyecto involucre un cliente o usuario final es recomendable juntar Kanban con otra metodología ágil como Scrum, de esta manera se puede asignar un rol específico a cada integrante del proyecto. No obstante, siempre se debe hacer un estudio para determinar qué metodología es la que mejor se acopla según las características del proyecto, de tal manera que se alcance la mejor productividad en el equipo.
- Para el desarrollo de software en equipo es recomendable usar un repositorio de código, con el cual cada integrante pueda obtener y publicar cada versión del código fuente desarrollado en la aplicación. De esta manera se guarda un historial de las modificaciones y cada desarrollador puede llevar un registro de los cambios realizados, sin interferir con el avance del resto del equipo.
- Para el uso de la técnica web scraping es necesario revisar que las páginas tengan una buena estructura HTML, de esta manera es más fácil acceder a cada uno de los links de donde se requiere obtener información. Así mismo, se debe comprobar que el tiempo de respuesta del sitio web se encuentre en un rango aceptable, para que esto no perjudique el rendimiento del software con el cual se va a extraer información.
- Cuando se obtiene información de un sitio externo es necesario tener en cuenta que si el servicio no está disponible se debe informar al usuario para que pueda realizar la consulta en otro momento. Por otro lado, si el portal cambia la estructura de su página es importante programar controles que notifiquen al desarrollador que se debe revisar el código. Por ejemplo, en la obtención de links se puede verificar que la URL cargada sea válida, este control se puede aplicar sobre otros campos de la página como identificadores, tipos de datos, entre otros.

- Los sitios web que no deseen exponer su contenido a aplicaciones como la desarrollada en esta tesis pueden hacer uso de herramientas como captcha<sup>43</sup>, para que un servidor externo no pueda ingresar a su contenido. Por otro lado, los sitios web que deseen compartir su información de manera abierta pueden hacer uso de API's<sup>44</sup> que estén disponibles a los desarrolladores.
- Al realizar la configuración de un servidor en la nube se debe tomar en cuenta las características del proyecto y las necesidades que van a tener los usuarios, adicionalmente se debe verificar que tenga soporte sobre el lenguaje de programación requerido, de esta manera se pueden ahorrar costos, manteniendo un buen funcionamiento de la aplicación.
- Finalmente, a la hora de publicar la aplicación es muy importante tomar en cuenta algunos componentes importantes que no forman parte de la funcionalidad, como son: nombre de la aplicación, logo, una imagen de inicio, un identificador, entre otras características necesarias para que el software pueda estar disponible en una tienda o para un usuario final. Adicionalmente, es recomendable revisar las guías de desarrollo que pone a disposición cada tienda, ya que de esta manera se reduce la posibilidad de que la aplicación sea rechazada por la empresa.

---

<sup>43</sup> Un programa permite distinguir entre una entrada humana y la de una máquina, para evitar una extracción automatizada de datos de sitios web.

<sup>44</sup> Interfaz de programas de aplicaciones.

## BIBLIOGRAFÍA

- "Alloy." Titanium 3.X. N.p., n.d. Web. 28 octubre 2014.
- "American Psychological Association (APA)." *Http://www.apa.org*. N.p., n.d. Web. 08 noviembre 2014.
- "Apple Developer." IOS Dev Center -. N.p., n.d. Web. 28 octubre 2014.
- "IEEE." - *The World's Largest Professional Association for the Advancement of Technology*. N.p., n.d. Web. 09 noviembre. 2014.
- "Modern Language Association (MLA): Format, Bibliography, Style, Convention." *Modern Language Association (MLA): Format, Bibliography, Style, Convention*. N.p., n.d. Web. 06 noviembre 2014.
- American Psychological Association. Internet. <http://www.apastyle.org/index.aspx> (30/07/2015)
- Anderson, David. Kanban - Successful Evolutionary Change for your Technology Organizations. Blue Hole Press. 2010. Print
- Apple. Apple Reinvents the Phone with iPhone. Internet. <https://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html> (30/07/2015)
- Apple. Apple's App Store Downloads Top Two Billion. <http://www.apple.com/pr/library/2009/09/28Apples-App-Store-Downloads-Top-Two-Billion.html> (30/07/2015)
- Apple. Welcome to iTunes U Support. Internet. <https://www.apple.com/support/itunes-u/>. (30/07/2015)
- Apple. Welcome to iTunes U Support. Internet. <https://www.apple.com/support/itunes-u/>. (30/07/2015)
- Dannen, Chris, and Christopher White. Desarrollo De Aplicaciones iOS Para Redes Sociales. Madrid: Anaya Multimedia, 2012. Print.
- David J. Anderson, Giulio Concas. A comparative study of Scrum and Kanban approaches on a real case study using simulation. Seattle.

- Dirección General Académica PUCE. Internet. “Las disertaciones en la PUCE” Guía e Instructivo. <http://www.puce.edu.ec/documentos/disertacionesPUCE.pdf> (30/07/2015)
- Ebrary. ebrary Solutions. Internet. <http://www.ebrary.com/corp/solutions.jsp> (30/07/2015)
- EBSCO Publishing. EBSCOhost. <https://itunes.apple.com/ec/app/ebscohost/id433269587?mt=8> (30/07/2015)
- Gartner, Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013. Internet. <http://www.gartner.com/newsroom/id/2592315>. (30/07/2015).
- Gartner, Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013. Internet. <http://www.gartner.com/newsroom/id/2324917>. (30/07/2015)
- Hall, Marty. Core Servlets and JavaServer Pages. Upper Saddle River, NJ: Prentice Hall PTR, 2000. Print
- Harvard Library. HOLLIS+. <http://hollis.harvard.edu/> (30/07/2015)
- Harvard University. Harvard Library. Internet. <http://library.harvard.edu/> (30/07/2015)
- Harvard University. Report to the Digital Library Federation. Internet. [http://old.diglib.org/pubs/news04\\_01/harvard.htm](http://old.diglib.org/pubs/news04_01/harvard.htm) (30/07/2015)
- IDC. Smartphone OS Market Share, Q1 2015. Internet. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> (30/07/2015)
- Imagine Easy Solutions. EasyBib: Automatic Bibliography Generator and Citation Machine. <https://itunes.apple.com/us/app/easybib/id436768184?mt=8> (30/07/2015)
- iOS Human Interface Guidelines. Apple. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/> (21/08/2015)
- Kolbeck, Rainer, julio Arroyo, and Heinz Koschmieder. El Gran Libro De JavaScript. España: Marcombo, 1997. Print
- Krug, Steve. Don't Make Me Think!: A Common Sense Approach to Web Usability. Berkeley, Calif: New Riders Pub., 2006.
- Online Writing Lab, Purdue University. Internet. <https://owl.english.purdue.edu/owl/resource/560/01/> (30/07/2015)

- Propiedad Intelectual del Ecuador, “¿Qué es la Propiedad intelectual?”. Internet. <http://www.propiedadintelectual.gob.ec/propiedad-intelectual/> (30/07/2015)
- PUCE. Bibliotecas Virtuales. Internet. <http://www.puce.edu.ec/portal/content/Bibliotecas%20Virtuales/174?link=oln30.redirect> (30/07/2015)
- PUCE. Información General de la Biblioteca. Internet. <http://www.puce.edu.ec/portal/content/Informaci%C3%B3n%20General%20de%20la%20Biblioteca/92?link=oln30.redirect> (30/07/2015)
- Rapid Value Solutions. Internet. How to Choose the Right Architecture For Your Mobile Application. United States. <http://www.rapidvaluesolutions.com/wp-content/uploads/2013/04/How-to-Choose-the-Right-Technology-Architecture-for-Your-Mobile-Application.pdf> (Nov, 2012).
- Springer. Our business is publishing. Internet. <http://www.springer.com/la/about-springer> (30/07/2015)
- Springer. SpringerLink. <https://itunes.apple.com/us/app/springerlink/id473166018?mt=8> (30/07/2015)
- TIOBE. TIOBE Index for August 2015. Internet. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. (26/08/2015).
- Watts S. Humphrey. The Team Software Process. Pittsburgh. Addison Wesley, 1999.
- Wesley Dyson. iSource APA. <https://itunes.apple.com/us/app/isource-apa/id349729106?mt=8> (30/07/2015)
- Yuan, Michael. A Java Developer’s Guide to PaaS. Internet. [http://www.infoq.com/articles/paas\\_comparison](http://www.infoq.com/articles/paas_comparison). (21/09/2015)

## ANEXOS

### Índice de figuras

Figura 1: Número de descargas de aplicaciones móviles por año. ....	2
Figura 2: Características funcionales de las principales aplicaciones. ....	10
Figura 3: Cuadro comparativo metodologías de desarrollo.....	13
Figura 4: Tablero Kanban. ....	14
Figura 5: Cuadro alternativas de desarrollo de aplicaciones móviles ..... 16	16
Figura 6: Cuadrante mágico de plataformas de desarrollo de aplicaciones móviles ..... 18	18
Figura 7: Cuadrante mágico de aplicaciones empresariales de PaaS. ....	20
Figura 8: Cuadro comparativo de PaaS que ofrecen Java ..... 21	21
Figura 9: Diseño arquitectónico. ....	24
Figura 10: Casos de uso a nivel general. ....	24
Figura 11: Barra de status (iOS). ....	25
Figura 12: Barra de navegación (iOS),.....	26
Figura 13: Barra de búsqueda (iOS).....	26
Figura 14: Íconos estándar.....	27
Figura 15: Cuadro de interfaces.....	27
Figura 16: UI1 ..... 28	28
Figura 17: UI2 ..... 28	28
Figura 18: UI3 ..... 29	29
Figura 19: UI4 ..... 29	29
Figura 20: UI5 ..... 30	30
Figura 21: UI6 ..... 30	30
Figura 22: Casos de prueba. ....	31
Figura 23: Comentario de una línea en JavaScript. ....	33
Figura 24: Comentario de varias líneas en JavaScript.....	33
Figura 25: Función en JavaScript. ....	34
Figura 26: Espacio después de palabras clave en JavaScript. ....	34
Figura 27: Sentencias en JavaScript. ....	34
Figura 28: Uso de llaves en JavaScript.....	35

Figura 29: Comentario de una línea en Java.....	36
Figura 30: Comentario de varias líneas en Java. ....	36
Figura 31: Función en Java.....	36
Figura 32: Uso de llaves en Java. ....	36
Figura 33: Nombre de clases en Java. ....	37
Figura 34: Página de resultados de SpringerLink.....	38
Figura 35: Opción de descarga de resultados de SpringerLink.....	39
Figura 36: Archivo de resultados de SpringerLink. ....	39
Figura 37: Código para búsqueda en SpringerLink.....	40
Figura 38: Pagina de resultados de búsqueda de eBrary. ....	41
Figura 39: Opción de descarga de resultados de eBrary. ....	42
Figura 40: Archivo de resultados de eBrary. ....	42
Figura 41: Código para búsqueda en eBrary .....	43
Figura 42: Pagina de búsqueda de la PUCE.....	44
Figura 43: Pagina de resultados de la PUCE.....	45
Figura 44: Información por documento de la PUCE.....	46
Figura 45: Código para el campo en resultados de la PUCE.....	47
Figura 46: Interfaz inicial. ....	48
Figura 47: Código para obtener tamaño .....	48
Figura 48: Interfaz documentos PUCE.....	49
Figura 49: Interfaz documentos SpringerLink.....	50
Figura 50: Interfaz documentos eBrary .....	51
Figura 51: Interfaz detalle de documentos. ....	52
Figura 52: Generación de cita al portapapeles.....	53
Figura 53: Comandos de cita al portapapeles. ....	54
Figura 54: Generación de mail. ....	54
Figura 55: Comandos de cita al portapapeles.....	55
Figura 56: Controles de conexión.....	55
Figura 57: C1 - Simulador iPhone 4s. ....	56
Figura 58: C1 - Simulador iPhone 5.....	57
Figura 59: C1 - Simulador iPad Air.....	57

Figura 60: C1 - iPhone 6.....	58
Figura 61: C1 - iPad 2.....	58
Figura 62: C2 - Simulador iPhone 4s. ....	59
Figura 63: C2 - Simulador iPhone 5.....	60
Figura 64: C2 - Simulador iPad Air.....	60
Figura 65: C2 - Dispositivo iPhone 6. ....	61
Figura 66: C2 - Dispositivo iPad 2. ....	61
Figura 67: C3 - Simulador iPhone 4s. ....	62
Figura 68: C3 - Simulador iPhone 5.....	63
Figura 69: C3 - Simulador iPad Air.....	63
Figura 70: C3 - Dispositivo iPhone 6. ....	64
Figura 71: C3 - Dispositivo iPad 2. ....	64
Figura 72: C4 - Simulador iPhone 4s. ....	65
Figura 73: C4 - Simulador iPhone 5.....	66
Figura 74: C4 - Simulador iPad Air.....	66
Figura 75: C4 - Dispositivo iPhone 6. ....	67
Figura 76: C4 - Dispositivo iPad 2. ....	67
Figura 77: C5 - Simulador iPhone 4s. ....	68
Figura 78: C5 - Simulador iPhone 5.....	69
Figura 79: C5 - Simulador iPad Air.....	69
Figura 80: C5 - Dispositivo iPhone 6. ....	70
Figura 81: C5 - Dispositivo iPad 2. ....	70
Figura 82: C6 - Dispositivo iPhone 6. ....	71
Figura 83: C6 - Dispositivo iPad 2. ....	72