

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Departamento:

Desarrollo de Aplicaciones para Dispositivos Móviles.

TRABAJO DE DIPLOMA

Título: Prototipo de Aplicación para Móviles para la "Enciclopedia Cubana 1000 Preguntas 1000 Respuestas"

Autor: César Daniel Marichal Reyes

Tutores: Dra.C Yanet Rodríguez Sarabia

MSc. Deymis Tamayo Rueda

Santa Clara, Mes y Año
Copyright©UCLV

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Academic Departament:
Development Applications for Mobile Devices.

DIPLOMA THESIS

Title: Mobile Application Prototype for the "Cuban Encyclopedia
1000 Questions 1000 Answers".

Author: César Daniel Marichal Reyes

Mentors: Dra.C Yanet Rodríguez Sarabia

MSc. Deymis Tamayo Rueda

Santa Clara, Month, Year
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419

Agradecimientos

En primer lugar a mi familia por estar siempre a mi lado y brindarme su apoyo.

A mis tutores por su ayuda incondicional y su paciencia.

A mis amigos de la Facultad con los cuales compartí momentos inolvidables.

Por su ayuda incondicional y abundante cariño a todos muchas gracias.

Resumen

El trabajo siguiente, aborda el desarrollo de un prototipo de aplicación para dispositivos móviles en el sistema operativo Android para la Enciclopedia Cubana 1000 preguntas 1000 respuestas. Para el desarrollo de este prototipo se realizó un análisis de tecnologías de desarrollo de aplicaciones para móviles, se diseñó una base de datos en SQLite, que permitió almacenar la información que posee esta Enciclopedia. Para validar este prototipo de aplicación, se realizaron diferentes pruebas como: pruebas de integración, pruebas de compatibilidad y pruebas de campo. En las pruebas realizadas se obtuvieron excelentes resultados.

Abstract

The following work deals with the development of a prototype application for mobile devices in the Android operating system for the Cuban Encyclopedia 1000 questions 1000 answers. For the development of this prototype an analysis of mobile application development technologies was made, a SQLite database was designed, which allowed to store the information held by this Encilclopedia. In order to validate this application prototype, different tests were carried out such as: integration tests, compatibility tests and field tests. In the tests carried out, excellent results were obtained.

Tabla de contenidos

Introducción	1
1. Consideraciones generales sobre el desarrollo de aplicaciones para móviles como soporte para libros electrónicos	4
1.1. Conceptos básicos para el desarrollo de aplicaciones para dispositivos móviles .	4
1.2. Diferencias entre Aplicaciones y Web Móviles	6
1.2.1. Aplicaciones Móviles	7
1.3. El Proceso de diseño y desarrollo de una aplicación para móviles	7
1.4. Tipos de aplicaciones según su desarrollo	10
1.4.1. Aplicaciones nativas	10
1.4.2. Aplicaciones Web	11
1.4.3. Aplicaciones Híbridas	11
1.5. Diseño de aplicaciones móviles para el sistema operativo Android	12
1.5.1. Principio de experiencia de usuario	13
1.5.2. Simplicidad	13
1.5.3. Consistencia	14
1.5.4. Navegación intuitiva	14
1.5.5. El estilo de las interfaces	15
1.5.6. Interfaces nativas o personalizadas	15
1.6. Sistema operativo Android para dispositivos móviles	16
1.6.1. Material Design	17
1.6.2. Almacenamiento de Información de aplicaciones móviles en Android .	17
1.7. Metodologías de desarrollo de Software para aplicaciones móviles	18
1.7.1. Metodologías ágiles	18
1.7.2. Extreme Programming (XP)	19
1.7.3. Scrum	20
1.7.4. Mobile-D	21

1.7.5. Comparación entre XP, Scrum y Mobile-D	22
1.8. Desarrollo de aplicaciones móviles para enciclopedias	23
1.8.1. Wikipedia	23
1.8.2. Kiwix, Wikipedia Offline	24
1.8.3. Encyclo	25
2. Análisis, Diseño e Implementación de la Aplicación	27
2.1. Primera Fase: Análisis	27
2.2. Segunda Fase: Diseño	29
2.2.1. Inicialización	29
2.3. Tercera Fase: Codificación	42
2.3.1. Producción y Estabilización	42
3. Descripción de las pruebas realizadas a la aplicación y Manual de usuario	46
3.1. Pruebas del sistema y correcciones a la aplicación	46
3.1.1. Pruebas de integración Manuales	46
3.1.2. Pruebas de compatibilidad	48
3.1.3. Pruebas de campo	50
3.2. Manual de Usuario de la Aplicación: PREGUNTAS 1000 RESPUESTAS . . .	53
3.2.1. Pantalla Principal de la Aplicación	54
3.2.2. Tomos de la Aplicación	54
3.2.3. Menú Lateral de la Aplicación	55
Conclusiones	59
Recomendaciones	60
Bibliografía	61
A.	65

Lista de figuras

1.1. Cantidad de trabajo entre el diseñador y el programador en el desarrollo de software.	8
1.2. Ciclo de vida de Scrum para dispositivos móviles	21
1.3. Ciclo de vida de Mobile-D	22
1.4. Comparación entre las metodologías ágiles: XP, Scrum y Mobile-D	22
1.5. Logo Wikipedia	24
1.6. Menú principal de la app Wikipedia	24
1.7. Logo Kiwix	25
1.8. Actividad principal de la app Kiwix	25
1.9. Logo Kiwix	25
1.10. Actividad principal de la app Kiwix	26
2.1. Diagrama de Entidad Relación: preguntas_mil_respuestas.db	30
2.2. Diagrama de Componentes	30
2.3. Diagrama de Casos de Uso del Sistema	31
2.4. Descripción del Caso de Uso: Consultar Tomos	33
2.5. Descripción del Caso de Uso: Crear Marcador	34
2.6. Descripción del Caso de Uso: Abrir Menú Lateral	35
2.7. Storyboard de la Aplicación	36
2.8. Prototipo de Pantalla: Splash	37
2.9. Prototipo de Pantalla: Pantalla Principal	37
2.10. Storycard de la Pantalla Principal	38
2.11. Prototipo de Pantalla: Splash	39
2.12. Prototipo de Pantalla: Enciclopedia Cubana	39
2.13. Prototipo de Pantalla: Marcadores	40
2.14. Prototipo de Pantalla: Opciones	40
2.15. Prototipo de Pantalla: Acerca de	41

2.16. Storycard del Menú de Navegación Lateral	41
2.17. Prototipos de Pantallas: Tomos	42
2.18. Estructura de directorios más significativos	43
3.1. Prueba de Integración a la Pantalla Principal	47
3.2. Prueba de Integración al Menú Lateral	47
3.3. Prueba de Integración a la Pantalla de Marcadores	48
3.4. Dispositivos de Prueba	49
3.5. Encuesta Realizada como prueba de campo	51
3.6. Gráfico que muestra los resultados obtenidos en la encuesta de campo aplicada	52
3.7. Splash de la aplicación	53
3.8. Pantalla Principal de la Aplicación	54
3.9. Pantalla de los Tomos de la Aplicación	54
3.10. Menú Lateral de la Aplicación	55
3.11. Menú Lateral: Enciclopedia Cubana	56
3.12. Menú Lateral: Marcadores	56
3.13. Menú Lateral: Opciones	57
3.14. Menú Lateral: Acerca de	57

Lista de tablas

2.1. Requerimientos funcionales	28
2.2. Requerimientos no funcionales	28

Introducción

Los dispositivos móviles, principalmente los teléfonos inteligentes y las tabletas, se han convertido en artefactos cotidianos en la vida de las personas. Según cifras mundiales del 2018, se vendieron más de 1969,6 millones de celulares, de los cuales el 95,6 % fueron teléfonos inteligentes o “smartphones” (ROLDAN and THOMPSON, 2018).

El viejo concepto del teléfono utilizado para establecer una comunicación oral o escrita ha evolucionado a un concepto de aparato inteligente, portátil, con acceso a Internet, que permite movilidad y que ha cambiado la forma en que se interactúa en el mundo utilizando aplicaciones móviles, las cuales establecen nuevos modelos para hacer negocios, para estudiar, para comprar y vender, para trabajar, para buscar información; en fin, el contar con un dispositivo móvil inteligente junto con las respectivas aplicaciones, permite a sus dueños disponer de una poderosa herramienta capaz de traer su mundo al alcance del bolsillo de una camisa o de un pantalón (Cuello and Vittone, 2013).

Los dispositivos móviles también han sido utilizados en la consulta del conocimiento, no solo en internet, sino también a los libros, en este caso, los libros digitales o electrónicos. Un libro electrónico, libro digital o ciberlibro, conocido en inglés como e-book o eBook, es la versión o evolución electrónica o digital de un libro. Es importante diferenciar el libro electrónico o digital de uno de los dispositivos más popularizados para su lectura: el lector de libros electrónicos, o e-reader (Agenjo-Bullón and Hernández-Carrascal, 2010).

En Cuba al igual que el resto de los países del mundo se ha extendido el uso de las tecnologías de la información y las telecomunicaciones donde las generaciones más jóvenes, fundamentalmente en el ambiente universitario conviven con los smartphones en todo momento (Peñuela Epalza et al., 2014). A partir de esta problemática el ministerio de informática y comunicaciones tiene una política de informatización de la sociedad cubana reflejada en los Lineamientos resultantes del 7mo Congreso del Partido Comunista de Cuba (de Cuba, 2017) en su artículo 108 plantea:

“Avanzar gradualmente, según lo permitan las posibilidades económicas, en el proceso de in-

formatización de la sociedad, el desarrollo de la infraestructura de telecomunicaciones y la industria de aplicaciones y servicios informáticos. Sustentar este avance en un sistema de ciberseguridad que proteja nuestra soberanía tecnológica y asegure el enfrentamiento al uso ilegal de las tecnologías de la información y la comunicación. Instrumentar mecanismos de colaboración internacional en este campo”.

El proceso de informatización de la sociedad cubana busca utilizar ordenadamente y de forma masiva las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, buscando satisfacer todas las necesidades de la sociedad, logrando cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos (Jover and Arríete, 2018)

La Editorial Félix Varela en el año 2014 publicó una colección de libros compuesta por cuatro tomos titulada: Enciclopedia cubana ”1000 preguntas, 1000 respuestas”, la cual cuenta con un vasto cúmulo de información ordenada por temáticas: Universo, Vida, Invenciones y Sociedad. Esta Enciclopedia posee una gran importancia pues está concebida de manera dinámica en forma de preguntas y respuestas, acorde al lenguaje de los jóvenes y del pueblo de Cuba en general; la misma abarca varios tomos que contextualizan la realidad cubana actual. El acceso a esta colección se ve limitada debido a la poca portabilidad de los libros al ser de un grueso considerable así como la escasa cantidad de ejemplares impresos que existen en la actualidad.

De lo anterior expuesto se deriva el siguiente **problema de investigación**:

La poca accesibilidad, limitada impresión de sus ediciones y poca facilidad de uso por los usuarios potenciales de la colección Enciclopedia Cubana ”1000 preguntas, 1000 respuestas”.

Para contribuir a la solución del problema de investigación antes planteado, se formuló el siguiente **objetivo general** de investigación:

Desarrollar una Aplicación para Móviles para el sistema operativo Android que permita una mayor portabilidad y acceso a los contenidos de la colección: Enciclopedia Cubana ”1000 preguntas, 1000 respuestas”.

Este objetivo general fue desglosado en los **objetivos específicos** siguientes:

1. Caracterizar el marco teórico referencial de la investigación, en especial lo relativo a las metodologías y herramientas para el desarrollo de aplicaciones informáticas sobre el sistema operativo Android, así como las técnicas para la implementación y visualización de libros en formato de aplicación para móviles.
2. Diseñar una Base de Datos que almacene la información de la colección: Enciclopedia

cubana ”1000 preguntas, 1000 respuestas”.

3. Implementar una aplicación para móviles sobre los contenidos de la Enciclopedia cubana “1000 preguntas, 1000 respuestas”.
4. Realizar pruebas de usuario para comprobar el correcto funcionamiento de la aplicación.

Preguntas de Investigación:

- ¿Cuáles herramientas son las más adecuadas para desarrollar una aplicación en el sistema operativo Android para el acceso a la colección: Enciclopedia cubana “1000 preguntas, 1000 respuestas”?
- ¿Qué arquitectura y gestor de Base de Datos es apropiado para una aplicación móvil que almacena el contenido de libros?

Justificación de la Investigación

El producto responde a una demanda de la Editorial «Felix Varela» ante la necesidad de incrementar la disponibilidad de esta Enciclopedia. La aplicación a desarrollar cuenta con una gran importancia, la misma se encuentra contextualizada a la situación cubana, desarrollada de una manera peculiar basada en preguntas y respuestas, abarca muchos temas apoyados en imágenes que ayudan a un mejor aprendizaje, puesto que logra que las nuevas generaciones se interesen por sus contenidos. La aplicación para dispositivos móviles que se propone, facilitará el acceso a esta importante Enciclopedia de una manera amena y rápida.

Este trabajo está estructurado en tres capítulos. En el Capítulo 1 se describen los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado. Se llega a conclusiones sobre los métodos, tecnologías y software a desarrollar. El Capítulo 2 muestra la arquitectura de la solución y se documentan los principales artefactos de la metodología de desarrollo seleccionada. Haciendo uso para algunos de UML 2.0. En el Capítulo 3 se muestra la interfaz de usuario y una descripción de cómo se utiliza la aplicación desarrollada por los usuarios. Este documento culmina con las conclusiones, seguidamente las recomendaciones derivadas de la investigación y finalmente la bibliografía consultada.

Capítulo 1

Consideraciones generales sobre el desarrollo de aplicaciones para móviles como soporte para libros electrónicos

En este capítulo se describen los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado, para esto se realiza un estudio detallado de las metodologías de desarrollo de software relacionadas al desarrollo de aplicaciones móviles. Se llega a conclusiones sobre los métodos y tecnologías a utilizar, los cuales permitirán una aplicación acorde al avance de los dispositivos móviles en cuanto a la compatibilidad con las distintas versiones del sistema operativo Android. Se realiza un análisis sobre el uso de las aplicaciones móviles como soporte de libros electrónicos.

1.1. Conceptos básicos para el desarrollo de aplicaciones para dispositivos móviles

Los dispositivos móviles en la actualidad están presentes en todas las esferas de la vida, tanto en lo profesional como lo personal, y a pesar de que no hace mucho resultaba complejo construir aplicaciones para ellos, hoy en día es relativamente sencillo desarrollar programas, basta tomar algo de la abundante literatura existente, elegir un lenguaje de programación y descargar el software necesario que permita la instalación de un entorno de desarrollo apropiado para dicha actividad (Pérez Guerrero, 2018).

Una aplicación móvil, o app es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes (Smartphones), tabletas u otros dispositivos móviles, que permiten al usuario efectuar una tarea concreta de cualquier tipo: profesional, ocio, educativa, de acceso a servicios, etc., facilitando las gestiones o actividades a desarrollar (Santiago and Trbaldo, 2015).

Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Mobile, entre otros. Existen aplicaciones móviles gratuitas y otras de pago (Wolfram and Schilling, 2015).

Al ser aplicaciones residentes en los dispositivos están escritas en algún lenguaje de programación compilado, y su funcionamiento y recursos se encaminan a aportar una serie de ventajas tales como: un acceso más rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso, un almacenamiento de datos personales que, a priori, es de una manera segura, una gran versatilidad en cuanto a su utilización o aplicación práctica; la atribución de funcionalidades específicas así como mejoras en la capacidad de conectividad y disponibilidad de servicios y productos (usuario-usuario, usuario-proveedor de servicios, etc.).

Es importante destacar que una app no es una aplicación web, tampoco es un sistema operativo, ni un servicio de alojamiento informático o web (Tracy, 2012).

Actualmente encontramos aplicaciones de todo tipo, forma y color, pero en los primeros teléfonos, estaban enfocadas en mejorar la productividad personal: se trataba de alarmas, calendarios, calculadoras y clientes de correo. Al mismo tiempo, también mejoraron las herramientas de las que disponían diseñadores y programadores para desarrollar apps, facilitando la tarea de producir una aplicación y lanzarla al mercado, incluso por cuenta propia.

Cuando el uso de dispositivos móviles se hizo algo común, también la programación de aplicaciones para estos se tomó más en serio, así diversas compañías crearon entornos de programación o complementos a *Integrated Development Enviroment* (IDE).

A continuación se muestran algunos de los IDEs y emuladores existentes en la actualidad:

- Pda Toolbox,
- Satellite Forms,
- Scotbuilder,
- Palm Os Developer Studio,
- HS Pascal,

- Mobile Visual Basic,
- Embedded Visual Tools.

Con el auge alcanzado, las aplicaciones para dispositivos móviles y los distintos sistemas operativos existentes para dichos dispositivos se han desarrollado plataformas de desarrollo que permiten la creación multiplataforma de aplicaciones con el mínimo de configuración para su implementación en ambientes específicos (Torres Tapia and Solano Mosquera, 2013) como son:

- Xamarin,
- Appcelerator Titanium,
- Phonegap,

1.2. Diferencias entre Aplicaciones y Web Móviles

Las aplicaciones comparten la pantalla del teléfono con las webs móviles (Arroyo-Vázquez, 2009), pero mientras las primeras tienen que ser descargadas e instaladas antes de usar, a una web puede accederse simplemente usando Internet y un navegador; sin embargo, no todas pueden verse correctamente desde una pantalla más pequeña que la de un ordenador de escritorio (Arroyo-Vázquez, 2009). Las aplicaciones que se adaptan especialmente a un dispositivo móvil se llaman responsive web y son ejemplo del diseño adaptativo (Martín, 2010), ya que se puede pensar en ellas como un contenido que toma la forma del contenedor, mostrando la información según sea necesario.

Quienes cuentan ya con una web responsive pueden plantearse la necesidad de diseñar una aplicación, pero la respuesta a si esto es o no necesario, depende de entender tanto los objetivos de negocio, como las características que diferencian las aplicaciones de las webs. Por ejemplo, las aplicaciones pueden acceder a ciertas características del teléfono como los sensores, capacidades que actualmente están fuera del alcance de las webs.

Por lo anterior, puede decirse que una aplicación ofrece una mejor experiencia de uso, evitando tiempos de espera excesivos y logrando una navegación más fluida entre los contenidos. No siempre hay que elegir entre una u otra. Webs y aplicaciones no son competidoras, más bien, pueden complementarse entre ellas; por ejemplo, una web puede ser útil como canal de información para motivar la descarga de la aplicación (Aguado-Terrón and Martínez-Martínez, 2009).

1.2.1. Aplicaciones Móviles

En ocasiones al implementar una aplicación, es posible que ya exista una web como antecedente, en casos como este la aplicación debe utilizar este contenido de la web y darle un valor añadido mediante funcionalidades que permitan una mejor interacción con el mismo debido a que una aplicación está pensada para ser ejecutada en una pantalla más pequeña y puede que necesite adaptarse o utilizar características extras para ser visualizado correctamente.

En otros casos, el diseño comienza desde cero, cuando todavía no hay ni web ni aplicación, y hay que decidirse por cuál de ellas empezar. Aquí es donde adquiere más trascendencia el concepto de mobile first, que implica plantear el proceso de diseño teniendo en cuenta el móvil en primer lugar. La ventaja de esta forma de trabajar es que el pensar en el móvil como punto de partida, obliga a concentrarse en lo esencial de un producto y a hacer foco solo en lo que tiene sentido para este dispositivo.

Una vez que la aplicación está diseñada, puede preguntarse cuál es la mejor forma de llevar lo hecho para el teléfono a una pantalla de ordenador o a otros dispositivos, extendiendo y escalando el contenido y repensando la diagramación. Todos los dispositivos tienen usos diferentes, y en el momento de adaptar el diseño, hay que tener en cuenta las características particulares de cada uno de ellos. Mobile first es una propuesta de trabajo que ha surgido recientemente; una tendencia emergente que aún está por consolidarse. Actualmente, es solo una manera de afrontar el proceso de diseño y como tal, puede evaluarse la comodidad que se tiene trabajando de esta forma antes de empezar (Cuello and Vittone, 2013).

1.3. El Proceso de diseño y desarrollo de una aplicación para móviles

El proceso de diseño y desarrollo de una aplicación, abarca desde la concepción de la idea hasta el análisis posterior a su publicación en las tiendas. Durante las diferentes etapas, diseñadores y desarrolladores trabajan la mayor parte del tiempo de manera simultánea y coordinada. Se ha resumido las fases de este proceso solo desde la perspectiva del diseño y desarrollo, es decir, sin tener en cuenta los roles de coordinación, la participación del cliente, ni los accionistas de la empresa. A continuación se especifican las fases de este proceso según (Cuello and Vittone, 2013).

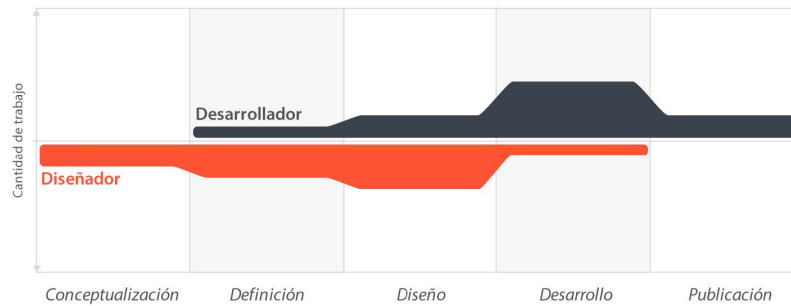


Figura 1.1: Cantidad de trabajo entre el diseñador y el programador en el desarrollo de software.

Conceptualización

El resultado de esta etapa es consolidar una concepto que, teniendo en cuenta las necesidades y problemas de los usuarios, responda a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto (Piattini Velthuis et al., 2007).

- Idealización,
- Investigación,
- Formalización de la idea.

Definición

En este paso del proceso se describe con detalle a los usuarios para quienes se diseñará la aplicación, usando metodologías como “PERSONAS” y “Viaje del usuario” (Pichler, 2016). También se sientan las bases de la funcionalidad, lo cual determinará el alcance del proyecto y la complejidad de diseño y programación de la app.

- Definición de usuarios
- Definición de funcionalidades

Diseño

En la etapa de diseño se llevan a un plano tangible los conceptos y definiciones anteriores, primero en forma de wireframes, que permiten crear los primeros prototipos para ser probados con usuarios, y posteriormente, en un diseño visual acabado que será provisto al desarrollador, en forma de archivos separados y pantallas modelo, para la programación del código.

Los wireframes varían desde un dibujo en una hoja de papel hasta un prototipo completamente comprensible utilizado para documentar el diseño a los programadores. Un wireframe realizado

en corto plazo puede transmitir la idea de que trata el software que se realizará. Independientemente de los medios utilizados para llevar a cabo un wireframe, éste solo tendrá un centro periodo de utilidad por lo que suelen tener un costo muy bajo en recursos y tiempo. Debido a su rápida creación es vital en las primeras fases de desarrollo ya que su diseño puede sufrir cambios constantemente.

Los prototipos se utilizan durante la fase de diseño conceptual presentando propuestas de las pantallas que componen el software. Estos dibujos pueden ser dibujados en papel o hechos en algún software y posteriormente impresos. Para simular el funcionamiento del sistema, se crean en papel múltiples pantallas e interfaces de elementos y una persona hace el rol de dispositivo, mostrando las pantallas que sean necesarias a lo largo de la presentación. Mediante esta interacción, usuarios y desarrolladores evalúan las diferentes características del software.

Las pruebas con usuarios son siempre un reto, si se tiene acceso a los usuarios finales del producto, éstos deberían ser los principales protagonistas del proceso, aunque los usuarios que son afectados directamente por el producto siempre proveen información que puede ser utilizada para mejorar características o funcionalidades o alternativas de usabilidad.

El diseño visual es el lenguaje visual y la composición de los elementos utilizados para expresar la estructura del software a los usuarios. Encontrar una forma de alcanzar una calidad visual y lograr un completo entendimiento con los usuarios es algo muy difícil de lograr y por lo general toma mucho tiempo. En muchas ocasiones los usuarios diferirán en lo que ven dependiendo de su punto de vista (Snyder, 2003).

Desarrollo

El programador de software podría considerarse como un trabajador simbólico-analítico. Serían aquellos trabajadores que identifican problemas y ofrecen soluciones a los mismos, comparten información a través de comunidades públicas y semipúblicas y privadas conformando redes de conocimiento como una de sus actividades prioritarias para acceder a información que se convertirá en un activo para resolver problemas de programación (Rodríguez-Gutiérrez, 2015).

Publicación

El proceso de publicación es cuando el software se pone a disposición de los usuarios ya sea en las respectivas tiendas de aplicaciones u otras formas de distribución. Una vez lanzado el software se realiza un seguimiento a través de analíticas, estadísticas y comentarios de usuarios, para evaluar el comportamiento y desempeño de la app, corregir errores, realizar mejoras y actualizarla en futuras versiones (Agustín, 2002).

1.4. Tipos de aplicaciones según su desarrollo

En los últimos años el mercado de los dispositivos móviles, en especial los Smartphone, ha mostrado un crecimiento notable en todo el mundo. Siendo las plataformas más sobresalientes iOS y Android. Cada una de estas plataformas cuenta con una infraestructura de desarrollo particular, siendo un reto para los desarrolladores de aplicaciones proporcionar soluciones para todas las plataformas, lo cual tiene un alto costo. La solución ideal a este problema es crear y mantener una única aplicación para todas las plataformas. El desarrollo multiplataforma tiene como objetivo mantener la misma base de código para diversas plataformas. De esta forma el esfuerzo y costo de desarrollo se reduce notablemente.

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico. Aunque a primera vista esto no parezca incumbencia del diseñador, la realidad es que el tipo de aplicación que se elija, condicionará el diseño visual y la interacción (Delía et al., 2013).

1.4.1. Aplicaciones nativas

Las aplicaciones nativas son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamado genéricamente Software Development Kit o SDK. Así, Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK.

Este tipo de apps se descarga e instala desde las tiendas de aplicaciones, con ciertas excepciones en el caso de Android, que sacan buen partido de las diferentes herramientas de promoción y marketing de cada una de ellas.

Las aplicaciones nativas se actualizan frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras. Una característica generalmente menospreciada de las apps nativas, pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación, ejemplo de esto son los mensajes generados por la aplicación.

Las aplicaciones nativas no requieren Internet para funcionar, por lo que ofrecen una experiencia de uso más fluida y están realmente integradas al teléfono, lo cual les permite utilizar todas las características de hardware del terminal, como la cámara y los sensores (GPS, acelerómetro, giroscopio, entre otros).

A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio sistema operativo. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

1.4.2. Aplicaciones Web

La base de programación de las aplicaciones web, también llamadas web apps, es el HTML, conjuntamente con JavaScript y CSS (Cascade Style Sheet) (Lie et al., 2005), estas son herramientas muy utilizadas por los programadores web.

En este caso no se emplea un SDK, lo cual permite programar de forma independiente al sistema operativo en el cual se usará la aplicación. Por eso, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular.

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón, no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente.

Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero, a diferencia de las apps nativas, requieren de una conexión a Internet para funcionar correctamente (Luján-Mora, 2002).

Adicionalmente, tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono. Las aplicaciones web suelen tener una interfaz más genérica e independiente de la apariencia del sistema operativo, por lo que la experiencia de identificación del usuario con los elementos de navegación e interacción, suele ser menor que en el caso de las nativas (Delía et al., 2013).

1.4.3. Aplicaciones Híbridas

Las aplicaciones híbridas son una especie de combinación entre las aplicaciones web y nativas. La forma de desarrollarlas es parecida a la de una aplicación web, usando HTML, CSS y JavaScript, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal,

que el resultado final es como si se tratara de una aplicación nativa. Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuirlas en cada una de sus tiendas. A diferencia de las aplicaciones web, estas permiten acceder, usando bibliotecas, a las capacidades del teléfono, tal como lo haría una app nativa (Solanky et al., 2016).

Las aplicaciones híbridas, también tienen un diseño visual que no se identifica en gran medida con el del sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para adherirse más a la estética propia de cada una. Existen algunas herramientas para desarrollar este tipo de aplicaciones. Apache Cordova es una de las más populares, pero existen otras alternativas como Icenium o Xamarin .

¿Qué tipo de aplicación implementar?

Dadas las características de cada una de las aplicaciones, decidirse por una u otra estará determinado por un conjunto de factores fundamentales y por la forma en que afectan finalmente la experiencia de uso. Cuando la disponibilidad de la app sin Internet, la posibilidad de usar notificaciones y el acceso a los recursos de hardware del teléfono sean importantes, una aplicación nativa será la opción más indicada.

Si ninguna de estas características es realmente importante para la aplicación, diseñar una aplicación web sería una mejor opción, si es que ya se dispone del conocimiento para ello, heredado del desarrollo de sitios web. En este caso, el costo de desarrollo es más bajo y la forma de trabajar un poco más ágil. Independientemente de esto, las aplicaciones nativas son las que ofrecen una mejor experiencia de uso y, sobre todo, rendimiento. Algunas apps como Facebook o LinkedIn, que antes eran híbridas, han pasado a ser nativas por este motivo. Adicionalmente, ellas responden más a las guías de diseño de cada sistema operativo.

1.5. Diseño de aplicaciones móviles para el sistema operativo Android

El diseño de aplicaciones para dispositivos móviles es fundamental, pues de ello depende el éxito una vez que sea desplegada y los usuarios comiencen a utilizarla. Se debe hacer énfasis en tres objetivos fundamentales a tener en cuenta en el desarrollo y diseño de estas aplicaciones. Las apps deben ser estéticamente placenteras en cada nivel o sección. Las transiciones deben ser rápidas y limpias, la organización y las tipografías deben ser atractivas y llenas de significado.

Los iconos son como obras de arte si están bien utilizados ya que una simple imagen, bien conformada, dice más que muchas palabras. Como una herramienta bien hecha, las aplicaciones deben combinar belleza, simpleza y propósito para crear una verdadera experiencia con mínimo esfuerzo, pero poderoso a su vez.

Deben simplificarnos las tareas que realizamos comúnmente y a su vez deben ser fáciles de entender, intuitivas. Cuando una persona utiliza una app por primera vez, ésta debe ser capaz de intuir fácilmente las características o funcionalidades más importantes. Hacer las operaciones más complejas de una forma transparente, dejando al usuario la menor cantidad de interacción posible. Las personas de cualquier edad o cultura deben sentirse totalmente en control y no abrumadas por muchas decisiones o señalamientos innecesarios. No es suficiente con desarrollar una aplicación que sea fácil de utilizar. Las aplicaciones tienen que inspirar a las personas a probar cosas nuevas y utilizar las aplicaciones de formas innovadoras. Permitir a las personas combinar aplicaciones a diferentes flujos de trabajo a través de multitareas, notificaciones e incluso compartir contenido entre aplicaciones (Treude and Aniche, 2018).

Todos los sistemas operativos proponen diferentes formas de interactuar con los elementos en pantalla. Conocer la diferencia entre ellos y utilizar elementos familiares para el usuario, asegura que se sienta cómodo y seguro usando la aplicación.

1.5.1. Principio de experiencia de usuario

Cada sistema operativo tiene su propia identidad que es reflejada en la apariencia y comportamiento de cada uno de los elementos que componen su interfaz. En ellos imprime su personalidad, lo que hace que la experiencia sea diferente a las demás. Sin embargo, todos comparten algunos puntos de vista fundamentales que se manifiestan en el diseño de sus interfaces (Hasan Montero and Martín Fernández, 2005).

1.5.2. Simplicidad

La simplicidad visual está directamente relacionada con la usabilidad. Ser simple implica en cierta medida ser mínimo, contar con pocos elementos, sobre todo, que aquellos presentes en la interfaz tengan una función bien definida que contribuya a cumplir el objetivo de la app y ayude al usuario.

Los móviles no son dispositivos para mostrar mucha información en pantalla. Por esta misma

razón, la simplicidad consiste también en manejar la economía visual y tener un buen criterio para determinar qué incluir y qué no en el diseño. Una gran cantidad de elementos puede abrumar al usuario, por eso, lo que está en pantalla tiene que ser necesario en ese momento y en esa situación de uso. Hacer el diseño simple es bastante complicado, pero reporta grandes beneficios en la experiencia de uso de la aplicación.

1.5.3. Consistencia

Una app tiene diferentes pantallas que la componen y al mismo tiempo, está dentro de un sistema operativo que propone un determinado aspecto visual e interacción. El usuario de Android, iOS o Windows Phone ya está habituado a ellos y espera que las aplicaciones se comporten de la misma manera. La consistencia, entonces, se trata de respetar estos conocimientos y costumbres del usuario, no solo en el interior de la aplicación, sino también en relación con el resto del Sistema Operativo. Esto favorece el uso intuitivo de la app, ya que el usuario puede prever su comportamiento sin demasiado esfuerzo.

La relación existente entre apariencia y comportamiento también tiene que ser consistente. El aspecto visual de un elemento interactivo determinado (como un botón con un icono) puede llevar a esperar un comportamiento específico de acuerdo a la forma en que se ve. Por ejemplo, si se usa un botón que representa la acción “eliminar” en el sistema operativo, el usuario esperará que también dentro de la app haga lo mismo. Cumplir con esa expectativa habla de consistencia.

1.5.4. Navegación intuitiva

Un aspecto que merece mucha atención en una aplicación es la forma de navegar entre contenidos, de manera que resulte fácil de comprender para el usuario, evitando la sensación de desorientación que puede ocasionar una navegación confusa. La navegación intuitiva está también relacionada con la consistencia. Cada sistema operativo propone diferentes elementos para navegar por la app como botones, pestañas y paneles. Hacer uso de ellos hará que el usuario los reconozca a primera vista y, solo con estos componentes, ya sepa cómo ir de una sección a otra.

Por otro lado, para el usuario es importante saber y prever qué pasará después de pulsar un botón o cómo se mostrarán las pantallas. Intuir dónde se está dentro de los contenidos de la aplicación y conocer cómo volver hacia atrás son factores muy importantes que alivian al usuario y le ahorran esfuerzos inútiles por intentar comprender cómo ir de un sitio a otro. Una navegación intuitiva permite, justamente, lograr un uso fluido y sin esfuerzo de la aplicación.

1.5.5. El estilo de las interfaces

La interfaz de una aplicación es como la ropa que viste. Es también la capa que hay entre el usuario y el corazón funcional de la app, el lugar donde nacen las interacciones. En mayor medida está compuesta por botones, gráficos, íconos y fondos, que tienen una apariencia visual diferente en cada uno de los sistemas operativos, porque Android, iOS y Windows Phone tienen su propia forma de entender el diseño. El trabajo del diseñador consiste en interpretar la personalidad de cada sistema operativo, aportando su propia visión y estilo de diseño, para conseguir aplicaciones que, además de ser fáciles de usar, sean distintas a las demás y tengan coherencia visual con la plataforma que las acoge (Farman, 2013).

1.5.6. Interfaces nativas o personalizadas

Las interfaces nativas se basan en elementos (botones, listas y encabezados) que vienen preestablecidos en cada plataforma. Tienen un aspecto ya definido en cuanto a las características básicas de su apariencia como color, tamaño o tipo de fuente, que pueden ajustarse en mayor o menor medida para que se correspondan con la estética buscada. En el momento de comenzar a diseñar es recomendable definir la interfaz con elementos nativos. De esta forma, se consigue una buena base sobre la cual trabajar y no es necesario crear todos los elementos desde cero.

El inconveniente de las interfaces nativas es que limitan la personalidad del diseño y, en algunos casos, es necesario ir un paso más allá. En situaciones como esta, todos o algunos elementos de la interfaz pueden ser personalizados, lo cual se logra creándolos de nuevo como imágenes. Por ejemplo, un elemento visual personalizado podría ser un campo de entrada de texto de un formulario que se genera como imagen, para aprovechar la posibilidad de incluir texturas, relieves o sombras específicas, que un elemento nativo no ofrece.

Diseñar una interfaz personalizada tiene que planearse de antemano porque representa una mayor complejidad y tiempo de desarrollo. De la misma manera, no siempre los diseños de este tipo de interfaces se trasladan a la aplicación funcional de forma fidedigna, pues su correcta implementación queda en manos de la pericia del desarrollador.

¿Qué tipo de interfaz utilizar?

En la mayoría de los casos no se trata de elegir entre una u otra, sino de alcanzar el balance adecuado combinando ambas. La situación más habitual suele ser partir de una interfaz nativa y personalizar solo aquellos elementos que se consideren necesarios.

El tipo de aplicación también tiene incidencia directa en este asunto. Por ejemplo, las aplicaciones que dan valor a los detalles visuales y a la experiencia en general, suelen contar con más de un elemento personalizado. Tal es el caso de Path y su forma de añadir “momentos”, algo diseñado y desarrollado a medida. Por el contrario, hay otras aplicaciones que dan más valor al cumplimiento de tareas y que se benefician de tener una apariencia más limpia que no distraiga del proceso. Un ejemplo de ellas es WhatsApp, cuyo aspecto visual está completamente basado en elementos nativos del sistema operativo.

Además del objetivo o del tipo de aplicación, hay otras variables para considerar. Las interfaces nativas tienen un punto a favor y es que están constituidas por elementos que el usuario ya conoce y a los que está habituado, por lo tanto, no representan un nuevo aprendizaje. Esto puede incidir favorablemente en la usabilidad de la app. Por otro lado, las interfaces personalizadas pueden ofrecer una apariencia más acabada, pero al trabajar con ellas hay que considerar la compatibilidad con múltiples dispositivos, es decir cómo se visualiza el mismo diseño en teléfonos diferentes y el rendimiento general.

1.6. Sistema operativo Android para dispositivos móviles

Android es un paquete de software para dispositivos móviles que incluyen un sistema operativo, middleware y aplicaciones. Los componentes de Android están diseñados en forma de paquete o stack, las aplicaciones conforman la capa superior del paquete, mientras que un kernel de Linux conforma las demás capas. Las aplicaciones que conforman el sistema operativo son para realizar las operaciones básicas como revisar el correo electrónico, enviar SMS, mapas, navegador web, contactos, entre otras. Todas las aplicaciones están escritas en el lenguaje de programación Java por tanto es en el que se desarrollan todas las aplicaciones para este sistema.

El framework de desarrollo de aplicaciones tiene el mismo acceso al mismo framework con que se desarrollan las aplicaciones por defecto del sistema, su arquitectura está diseñada para simplificar y reutilizar componentes; las capacidades de cualquier aplicación pueden ser publicadas para ser utilizadas por cualquier otra aplicación (sujeto a requerimientos de seguridad establecidos por el framework). Este mismo mecanismo permite a los componentes ser reemplazados por el usuario. Por ejemplo, si se tiene una pequeña aplicación para almacenar notas y deseas buscar una ubicación específica cuya localización acabas de escribir, puedes utilizar las funcionalidades de la aplicación de mapas dentro de la misma aplicación en vez de considerar cambiar a la aplicación de mapas y buscar la ubicación manualmente. Android incluye un set de bibliotecas C/C++ que son utilizadas por varios componentes del sistema operativo. Estas funcionalidades

están a disposición a través del framework de desarrollo de aplicaciones (Developers, 2011).

1.6.1. Material Design

Google, a partir de la versión de Android 5.0 – Lollipop, ha estandarizado ciertas normas en cuanto al diseño de aplicaciones.

Material Design, es una normativa de diseño enfocada en la visualización del sistema operativo Android, en la Web o en cualquier otra plataforma. La filosofía se ha aplicado a la mayoría de las aplicaciones desarrolladas por la empresa, y a muchos de los productos que se pueden encontrar hoy en el mercado; proporcionando una experiencia consistente en todas las plataformas. Se trata de un diseño más limpio, en el que predominan las animaciones y transiciones en cada respuesta, el relleno y los efectos de profundidad, así como la iluminación y las sombras. El diseño de Material tiene superficies físicas y bordes. Las escenas y sombras proporcionan significado sobre lo que se puede tocar y cómo se va a mover, así como resaltar diversos contenidos (Benbourahla, 2015).

1.6.2. Almacenamiento de Información de aplicaciones móviles en Android

Las aplicaciones Android profesionales se diseñan cada vez más como los sistemas de escritorio, con bases de datos para gestionar el almacenamiento y recuperación de información, que ayudan a mejorar sus capacidades y prestaciones (Vanegas, 2012).

Android Studio se perfila desde el año 2014 como el único e inmejorable IDE para desarrollar aplicaciones Android, no sólo el hecho de que este es multiplataforma sino que, no se debe instalar el SDK por separado tal como se hacía anteriormente con Eclipse, ya que este por ser promocionado por Google ya viene incorporado en sus diferentes instaladores, el único requisito que se solicita tener instalado es la versión más reciente del JDK (Java Developer Kit).

Entre las características positivas que posee están que utiliza Gradle lo que facilita en gran medida reusar código y recursos, facilita configurar, extender y personalizar el proceso, facilita la distribución del código y por tanto trabajar en equipos. Gestiona las dependencias de una forma cómoda y potente (está basado en Maven). Hace increíblemente fácil crear distintas versiones de la aplicación, por ejemplo, para distintos dispositivos, o una versión de prueba que carga

distintos recursos, apunta a webservices distintos. Siendo el IDE oficial para Android (Cinar, 2015).

1.7. Metodologías de desarrollo de Software para aplicaciones móviles

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub - fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo”(Avison et al., 2006).

Las metodologías tradicionales inician el desarrollo de un proyecto con un riguroso proceso de establecimiento de requerimientos, previo a etapas de análisis y diseño. En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto. En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más adelante nombradas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala (Cadavid et al., 2013).

1.7.1. Metodologías ágiles

En (Poole, 2009) se define al desarrollo ágil como aquel que, en comparación con el desarrollo tradicional, provee beneficios de mayor flexibilidad, retorno de inversión más alto, realización más rápida del retorno de inversión, alta calidad y mayor visibilidad.

En febrero del 2001, tras una reunión celebrada en Utah, nace el término ”ágil.” aplicado al desarrollo de software. El objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y responder a los cambios que pueden surgir a lo largo

del proyecto. Esto pretende ser una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas (Blanco et al., 2009).

Tras la reunión se creó The Agile Alliance dedicada a promover el desarrollo ágil de software y ayudar a las empresas que lo adoptaran. El punto de partida fue el Manifiesto Ágil, documento que resume esa filosofía y que expone cuatro valores a tener en cuenta:

- El individuo y las interacciones del equipo de desarrollo están por encima del proceso y las herramientas. Construir un buen equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir buena documentación. No producir documentos a menos que sean necesarios de una forma inmediata. Si el software no funciona, los documentos no valen de nada.
- La colaboración con el cliente es más importante que la negociación de contratos. Tiene que haber una interacción constante entre el cliente y el equipo de desarrollo.
- La respuesta ante el cambio es más importante que el seguimiento de un plan. La planificación no debe ser estricta sino flexible y abierta, la habilidad de responder a los cambios que surjan determina el éxito o el fracaso del proyecto.

1.7.2. Extreme Programing (XP)

Se centra en las mejores prácticas para el desarrollo de software. Consta de doce prácticas: el juego de planificación, pequeñas emisiones, la metáfora, el diseño sencillo, las pruebas, la refactorización, la programación en parejas, la propiedad colectiva, integración continua, semana 40-h, los clientes en el lugar, y los estándares de codificación (Balaguera, 2013).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Las historias del usuario son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos

funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Canós and Letelier, 2012).

1.7.3. Scrum

Scrum centra la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. Entonces, el dueño del producto decide qué elementos del backlog se deben desarrollar en el sprint siguiente.

La 1.2 muestra el ciclo de vida de Scrum para dispositivos móviles destacándose la planificación de los Sprint. Un Sprint corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint es considerado como un proyecto independiente. Su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos: reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint. Utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas (Canós and Letelier, 2012).

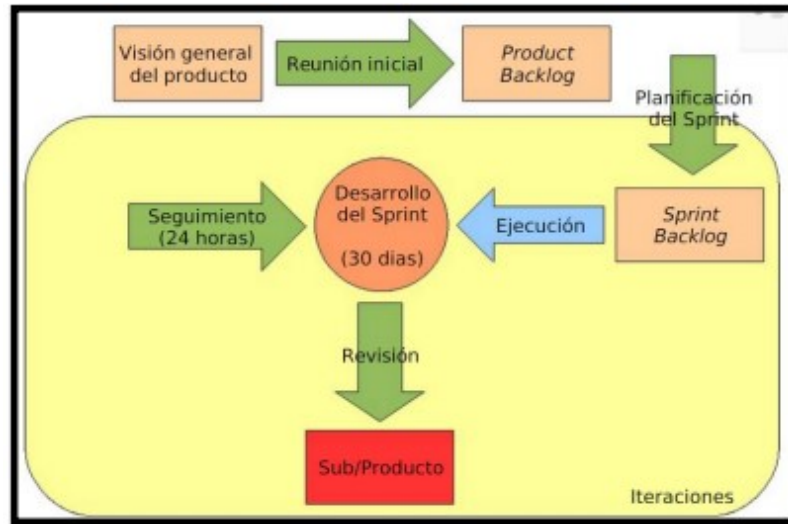


Figura 1.2: Ciclo de vida de Scrum para dispositivos móviles

1.7.4. Mobile-D

Una metodología de desarrollo, especialmente diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés Technical Research Centre of Finland) en Finlandia que lideran una corriente muy importante de desarrollo ágil muy centrada en las plataformas móviles. Las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software; según Abrahamsson Mobile-D debe ser utilizado por un equipo de no más de diez desarrolladores, trabajando en conjunto para suministrar un producto listo en un plazo máximo de diez semanas (Balaguera, 2013).

La aproximación de Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: Extreme Programming (XP), Crystal methodologies y Rational Unified Process (RUP). Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron un input muy valioso en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida.

En la 1.3 se muestran las cinco fases de Mobile-D: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas.



Figura 1.3: Ciclo de vida de Mobile-D

1.7.5. Comparación entre XP, Scrum y Mobile-D

	Metodología Extreme Programing	Metodología Scrum para móviles	Metodología Mobile D
Fases	Fases: planificación, diseño, desarrollo y pruebas.	Se define por Sprints de dos semanas aproximadamente y de un producto entregable.	Fases: exploración, inicialización, producción, estabilización y prueba del sistema.
Comunicación con el cliente	Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo	El Product Owner maneja la comunicación con el cliente	La comunicación con el cliente es menor que en Scrum.
Programación	Programación en parejas, en jornadas largas revisión de código mutuo.	El tiempo de programación se determina de acuerdo a la puntuación dada a cada tarea.	Es la parte más importante, y que mayor tiempo posee junto con las pruebas.
Documentación	Lo principal son las historias de usuario y tarjetas CRC.	Lo principal lo constituyen las historias de usuario.	Lo principal es el modelo del dominio, storycards y storyboards.
Pruebas	Se realizan unidades de pruebas, pruebas de aceptación y pruebas unitarias.	Se realizan pruebas unitarias, pruebas de integración, pruebas de aceptación.	Se realizan pruebas unitarias, pruebas de interfaz, pruebas de datos ingresados y pruebas funcionales.

Figura 1.4: Comparación entre las metodologías ágiles: XP, Scrum y Mobile-D

Luego de analizar la Figura 1.4, se determinó que las fases de las metodologías XP y Mobile-D se parecen a las fases de análisis, diseño, codificación y pruebas; en cambio Scrum se caracteriza por ser iterativo, se podría decir que realiza las fases de análisis, diseño, codificación y pruebas por cada iteración. La documentación en la metodología XP y en Scrum tiene un nivel bajo en comparación con la metodología Mobile-D. Sin embargo, Mobile-D no exige una gran cantidad de documentación, solo lo necesario para guiarse en el desarrollo de la aplicación.

Después de comparadas las metodologías anteriores, teniendo en cuenta que el grupo de desarrollo está conformado por una persona y se dispone de poco tiempo, se decidió elegir la metodología Mobile-D ya que esta fue creada especialmente para el desarrollo de aplicaciones móviles y la entrega de resultados es rápida.

1.8. Desarrollo de aplicaciones móviles para enciclopedias

En la actualidad no existe conocimiento inalcanzable, sobretodo si nos encontramos delante de un ordenador o de un dispositivo con acceso a internet, podemos encontrar casi cualquier información que necesitemos viajando por la red. Esto sucede ya que disponemos de un conjunto de herramientas y páginas para solucionar cualquier duda que podamos tener. A pesar de las ventajas que nos ofrece internet en cuanto a la búsqueda de información, es importante valorar de que lugares obtenemos estos datos, ya que no siempre ofrecen fiabilidad. Sin duda una buena opción utilizada a nivel mundial es Wikipiedia. Se trata de una enciclopedia libre y accesible para todos a través del esfuerzo colaborativo. Compartir el conocimiento ha llagado también en forma de aplicación para nuestros teléfonos móviles y no solo de la mano de Wikipedia. En esta sección se mencionan 3 aplicaciones “de enciclopedias” que podemos encontrar en forma de aplicación para dispositivos móviles, estas no son las únicas que existen, pero es válido aclarar que son las más representativas, como antecedente a la aplicación que deseamos implementar para la: ^Enciclopedia cubana ”1000 preguntas, 1000 respuestas”.

1.8.1. Wikipedia

Wikipedia es una enciclopedia libre, políglota y editada de manera colaborativa. Es administrada por la Fundación Wikimedia, una organización sin ánimo de lucro cuya financiación está basada en donaciones. Sus más de 48 millones de artículos en 300 idiomas han sido redactados en conjunto por voluntarios de todo el mundo, lo que suma más de 2000 millones de ediciones, y permite que cualquier persona pueda sumarse al proyecto para editarlos, salvo que la página se

encuentre protegida contra vandalismos para evitar problemas. Wikipedia posee una aplicación para móviles para el Sistema Operativo Android, la aplicación permite navegar con más velocidad por los contenidos, que, utilizando la web, brinda la opción de editar un artículo, la app dispone de páginas recientes, páginas guardadas, la posibilidad de compartir en las redes sociales y un buen soporte de idiomas. En las Figuras 1.5 1.6 se muestra el logo y el menú principal de la aplicación.



Figura 1.5: Logo Wikipedia

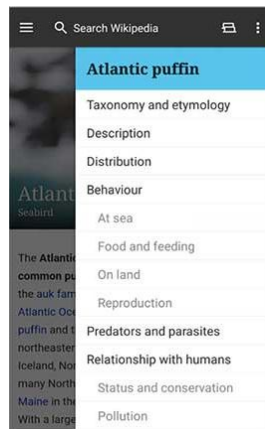


Figura 1.6: Menú principal de la app Wikipedia

1.8.2. Kiwix, Wikipedia Offline

Kiwix es un proyecto de software libre que ofrece la posibilidad de acceder a Wikipedia sin conexión a Internet y otros portales web basados en MediaWiki. Kiwix es el lector offline de Wikipedia soportado oficialmente por la Fundación Wikimedia. Además de ofrecer el acceso a Wikipedia sin conexión a Internet, Kiwix también se emplea para acceder a diferentes tipos de contenido offline. El programa está disponible para varios sistemas operativos como Windows, Mac OS X, Linux, iOS y Android. La app dispone de un software de lectura de archivos ZIM almacenado en nuestro dispositivo o en algún tipo de almacenamiento externo como puede ser una tarjeta SD. En las Figuras 1.7 1.8 se muestra el logo y la actividad principal de la aplicación.



Figura 1.7: Logo Kiwix



Figura 1.8: Actividad principal de la app Kiwix

1.8.3. Encyclo

Encyclo es una enciclopedia creada con la intención de ofrecer una colección completa de significados y definiciones. Su término de búsqueda utiliza más de 2.000 listas de palabras en inglés como Wikipedia y otros recursos en línea. Encyclo ofrece las primeras 250 cartas de cada definición. Si se requiere más detalle o explicación adicional se puede seleccionar a través de la fuente original de la lista de palabras. En las Figuras 1.9 1.10 se muestra el logo y la actividad principal de la aplicación.



Figura 1.9: Logo Kiwix



Figura 1.10: Actividad principal de la app Kiwix

Conclusiones Parciales

En este capítulo se fundamentan las herramientas y la metodología utilizada para dar solución, al desarrollo de la aplicación para plataforma Android. Después de lo expuesto en este capítulo se concluye que:

1. La aplicación a desarrollar se clasifica como nativa, ya que estará diseñada específicamente para el sistema operativo Android y no podrá ser accedida desde el navegador.
2. Para el desarrollo de la aplicación se decidió optar por una metodología ágil ya que estas tienen un tiempo corto para la entrega del producto final y son las más usadas en el desarrollo de aplicaciones para dispositivos móviles.
3. Basándonos en el análisis realizado a las aplicaciones ya existentes para enciclopedias, se decidió trabajar con una base de datos en SQLite que facilite el trabajo offline con la información.

Capítulo 2

Análisis, Diseño e Implementación de la Aplicación

En este capítulo se documentan los artefactos de la aplicación “Preguntas 1000 Respuestas”, de acuerdo con las fases de la metodología ágil Mobile-D, metodología seleccionada para la concepción y desarrollo de la aplicación. Se describen los storyboard y storycard, los cuales representan la principal documentación de la metodología Mobile-D, estos muestran visualmente como se observara cada pantalla de la aplicación. Se confecciona el mapa de navegación, el cual permite conocer cómo acceder a cada pantalla. Se describen los métodos y clases de más interés de la implementación.

2.1. Primera Fase: Análisis

En esta fase de implementación se realiza un análisis de los requisitos funcionales para la aplicación a implementar mediante el estudio de la problemática.

Exploración

En esta fase se definen los requerimientos, el alcance, así como las bases fundamentales para el desarrollo de un producto consecuente con lo que se desea.

Establecimiento de los Grupos de Interés o Stakeholders:

- **Desarrollador:** Es la persona encargada del análisis, desarrollo y pruebas de la aplicación.
- **Consejo Editorial de la Enciclopedia 1000 preguntas 1000 respuestas:** Personas que conocen la estructura de la enciclopedia y la estructura de la información de la misma.

También definen junto al desarrollador la información que deciden llevar a la aplicación.

- **Usuarios que usarán la aplicación:** Toda persona ya sea estudiante de la enseñanza media o superior donde se despliegue la herramienta desarrollada.

Requerimientos Iniciales :

Se pretende desarrollar una aplicación móvil para el sistema operativo Android que permita visualizar los contenidos de la Enciclopedia 1000 preguntas 1000 respuestas para que pueda ser utilizado por todo tipo de usuario no solo por estudiantes de la enseñanza media.

Requerimientos Funcionales :

Código	Descripción
RF1	Listar los tomos disponibles de la enciclopedia.
RF2	Listar contenidos por tomo.
RF3	Establecer un marcador a un contenido de un tomo.
RF4	Visualizar información genral sobre la Enciclopedia 1000 Preguntas 1000 Respuestas
RF5	Consultar marcadores guardados por tomo.
RF6	Personalizar la aplicación.
RF7	Visualizar información de la aplicación desarrollada

Tabla 2.1: Requerimientos funcionales

Requerimientos No Funcionales :

Código	Descripción
RNF1	El Sistema será desarrollado para la plataforma Android.
RNF2	El Sistema será desarrollado bajo la plataforma de programación Java.
RNF3	El sistema utilizará una interfáz amigable y responsive (sensible).
RNF4	El sistema permitirá el acceso a los usuarios sin necesidad de una autenticación.
RNF5	El sistema es instalable en equipos móviles con versión Android Lollipop 5.0- o superior.

Tabla 2.2: Requerimientos no funcionales

Definición del Alcance El alcance del proyecto se determina a través de las limitaciones y de los supuestos y dependencias.

Limitaciones

Las limitaciones de la aplicación son:

- La aplicación sólo puede ser ejecutada en dispositivos con el Sistema Operativo Android.
- Para la actualización de la información brindada por la aplicación es necesario contar con la información proporcionada por la editorial.

Supuestos y dependencias

Los supuestos y dependencias de la aplicación son:

- Los datos en la aplicación son actualizados solo cuando la editorial proporciona la información.
- Los usuarios manejan como idioma principal el español, siendo éste el lenguaje que se muestra en la interfaz de la aplicación.

2.2. Segunda Fase: Diseño

A través del diseño se desean obtener los diagramas que representen el funcionamiento de la aplicación, los procesos que va a ejecutar con el fin de cumplir los requerimientos definidos en la primera fase; además la descripción de cada uno de ellos y las configuraciones necesarias para desarrollar la aplicación móvil.

2.2.1. Inicialización

En esta etapa se realizan las actividades relacionadas a la configuración del ambiente de desarrollo y al diseño de la aplicación.

Configuración del Ambiente de Desarrollo:

En esta actividad se debe realizar la configuración de los elementos técnicos, en este caso se configuró el entorno de desarrollo para aplicaciones Android.

Configuración para aplicaciones móviles Android:

- **Tipo de Proyecto:** Android Application Project.
- **Configuraciones:** Se utilizó la biblioteca externa Duo navigation view y se consulta de un fichero de base de datos SQLite.

Diagrama de Entidad Relación:

El modelo Entidad-Relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Es una técnica de análisis para definir las entidades y las relaciones entre ellas. A continuación, el esquema diseñado para la base de datos consultada por aplicación.



Figura 2.1: Diagrama de Entidad Relación: preguntas_mil_respuestas.db

Diagrama de Componentes:

El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes de software, sus interfaces y las dependencias entre ellos (Ver Figura 2.2).

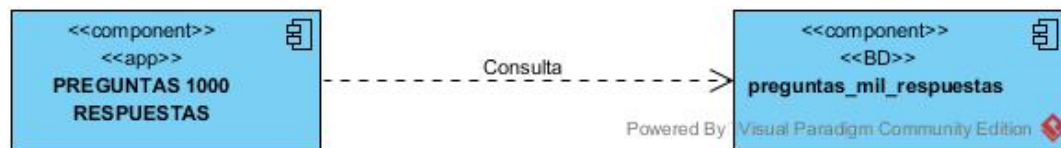


Figura 2.2: Diagrama de Componentes

Diagrama de Casos de Uso:

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. Los casos de uso del actor de este sistema identificados son mostrados a continuación:

- CU1: Consultar Tomos.
- CU2: Crear Marcador.
- CU3: Abrir Menú Lateral.
- CU4: Abrir Información General de la Enciclopedia Cubana.
- CU5: Abrir Marcadores.
- CU6: Abrir Opciones.
- CU7: Consultar Acerca de.

En la figura 2.3 se muestra el Diagrama de Casos de Uso del Sistema

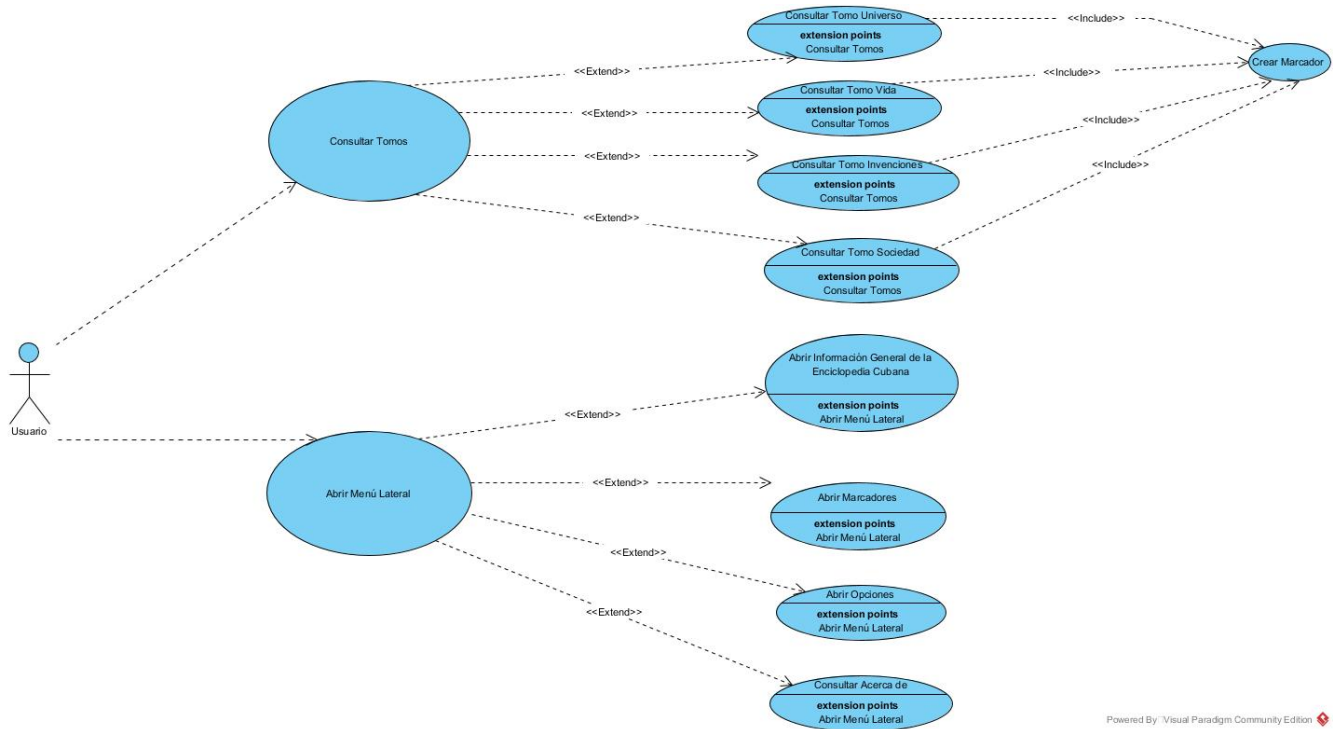


Figura 2.3: Diagrama de Casos de Uso del Sistema

Descripción de los Casos de Uso:

- **Consultar Tomos** La Aplicación después de ejecutar el splash inicia la Pantalla Principal, luego el usuario puede acceder a cada uno de los Tomos de la Enciclopedia, presionando el botón del tomo correspondiente: Universo, Vida, Invenciones y Sociedad Figura 2.4.
- **Crear Marcador** Cuando el usuario se encuentran dentro de un Tomo determinado, puede establecer marcadores a los contenidos que desee, facilitando guardar las preguntas que sean de interés para el usuario.
- **Abrir Menú Lateral** Este Menú brinda un grupo de funcionalidades al usuario como: Abrir Información General de la Enciclopedia Cubana, Abrir Marcadores, Abrir Opciones y Consultar el Acerca de. Cada una de estas funcionalidades se activan al presionar esta opción en el menú lateral.
- **Abrir Información General de la Enciclopedia Cubana** Al presionar esta opción el usuario puede consultar toda la Información Genaral que posee la Encliclopedia.

- **Abrir Marcadores** El usuario al presionar esta opción puede revisar cada uno de los marcadores establecidos, estos se encuentran ordenados por el tomo donde se fijó el marcador.
- **Abrir Opciones** El usuario puede seleccionar un conjunto de opciones que permiten personalizar la aplicación en cuanto a tamaño de texto, tema claro u oscuro y estilo de letra.
- **Consultar Acerca de** Facilita al usuario la información referente a la información general de la app PREGUNTAS 1000 RESPUESTAS y se accede a esta funcionalidad a través del menú lateral.


Caso de Uso del Sistema:	Consultar Tomos
Actor:	Usuario
Propósito:	Consultar el contenido de los tomos de la Enciclopedia 1000 Preguntas 1000 Respuestas
Resumen:	La Aplicación después de ejecutar el splash inicia la Pantalla Principal, luego el usuario puede acceder a cada uno de los tomos de la Enciclopedia, presionando el botón del tomo correspondiente: Universo, Vida, Invenciones y Sociedad. En la Pantalla Principal se puede consultar el Menú Lateral.
Responsabilidad:	Conocer la información del Tomo deseado.
Casos de uso asociado:	Consultar Tomo Universo, Consultar Tomo Vida, Consultar Tomo Invenciones, Consultar Tomo Sociedad.
Requisitos especiales:	-
Precondiciones:	-
Descripción	
IU_CONSULTAR TOMOS	
	
Flujo Normal de los eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el tomo que desea, presionando el botón que posee el icono del tomo correspondiente. 3. El usuario consulta el contenido deseado.	2. El sistema muestra el tomo seleccionado
Flujos Alternativos	
Post condiciones:	-

Figura 2.4: Descripción del Caso de Uso: Consultar Tomos


Caso de Uso del Sistema:	Crear Marcador
Actor:	Usuario
Propósito:	Marcar un contenido que resultó interesante para el usuario
Resumen:	El usuario puede marcar un contenido en específico, presionando el contenido y establecerlo como marcado.
Responsabilidad:	Conocer la información del Tomo deseado.
Casos de uso asociado:	Consultar Tomo Universo, Consultar Tomo Vida, Consultar Tomo Invenciones, Consultar Tomo Sociedad.
Requisitos especiales:	-
Precondiciones:	-
Descripción	
IU_TOMOS 	
Flujo Normal de los eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el contenido que desea marcar. 3. El usuario puede continuar consultando el contenido del Tomo.	2. El sistema marca y guarda el contenido seleccionado
Flujos Alternativos	
Post condiciones:	-

Figura 2.5: Descripción del Caso de Uso: Crear Marcador


Caso de Uso del Sistema:	Abrir Menú Lateral	
Actor:	Usuario	
Propósito:	Acceder a las opciones o funcionalidades que brinda la aplicación	
Resumen:	El usuario utiliza este menú para acceder a: Abrir Información General de la Enciclopedia Cubana, Abrir Marcadores, Abrir Opciones y Consultar el Acerca de.	
Responsabilidad:	Conocer la opción que se desea consultar	
Casos de uso asociado:	Abrir Información General de la Enciclopedia Cubana, Abrir Marcadores, Abrir Opciones, Consultar Acerca de	
Requisitos especiales:	-	
Precondiciones:	-	
Descripción		
IU_MENU LATERAL		
		
Flujo Normal de los eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona la opción que desea en el menú opciones. 3. El usuario interactúa con la opción seleccionada.	2. El sistema muestra la IU de la opción seleccionada.	
Flujos Alternativos		
Post condiciones:	-	

Figura 2.6: Descripción del Caso de Uso: Abrir Menú Lateral

Descripción de la Interfaz de Usuario:

A continuación se describe el storyboard o mapa de navegación dentro de la aplicación, es decir los vínculos entre las distintas pantallas que conforman la aplicación (Ver Figura 2.7).

Al iniciar la aplicación se puede visualizar el Splash, pantalla en con el logo de los tomos de la enciclopedia. Normalmente pasaría a visualizarse la pantalla principal la cual posee un conjunto de botones que permite seleccionar el tomo de la enciclopedia que se desea.

La pantalla principal también cuenta con un menú lateral, el cual permite cuatro funcionalidades, las cuales son: Enciclopedia Cubana, Marcadores, Opciones y Acerca de.

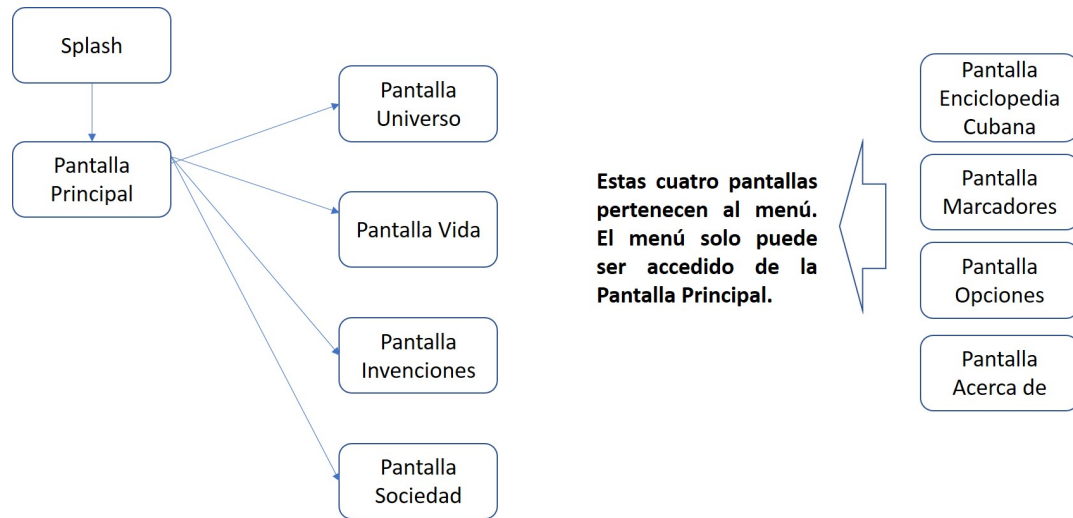


Figura 2.7: Storyboard de la Aplicación

- **Pantalla Principal** La Pantalla Principal está formada por cuatro botones, donde cada uno posee el contenido correspondiente a los 4 tomos de la Enciclopedia Cubana 1000 Preguntas 1000 Respuestas. Además será precedida de un splash con el logo de los cuatro tomos de le Enciclopedia. Se observa el Splash (Figura 2.8) y la Pantalla Principal (Figura 2.9) en un móvil y un tablet.



Figura 2.8: Prototipo de Pantalla: Splash



Figura 2.9: Prototipo de Pantalla: Pantalla Principal

En la Figura 2.10, se describe el StoryCard de prototipo de la pantalla principal. El story-card muestra una descripción del funcionamiento de la Pantalla Principal y de las fechas de su desarrollo.

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
01	Nuevo Fijo Mejora	Fácil	Fácil			Baja	
		Moderado	Moderado			Media	
		Duro	Duro			Alta	
Descripción							
Cuando el usuario abre la aplicación se visualiza el Splash, posteriormente se visualiza la Pantalla Principal la cual está formada por cuatro botones, donde cada uno posee el contenido correspondiente a los 4 tomos de la "Enciclopedia Cubana 1000 Preguntas Mil Respuestas". Cuenta con un menú lateral donde se puede acceder a: Enciclopedia Cubana, Marcadores Opciones y Acerca de.							
Fecha	Estado	Comentario					
10/03/19	Definido	Sin comentario					
15/03/19	Implementado	Sin comentario					
20/03/19	Hecho	Sin comentario					
10/04/19	Verificado	Sin comentario					

Figura 2.10: Storycard de la Pantalla Principal

- Menú de Navegación Lateral:** El menú de navegación lateral (Figura 2.11), muestra una lista con las opciones: Enciclopedia Cubana (Figura 2.12), Marcadores (Figura 2.13), Opciones (Figura 2.14) y Acerca de (Figura 2.15).
 - ▶ Enciclopedia Cubana, muestra la información general del conjunto de los cuatro tomos de la Enciclopedia 1000 Preguntas Mil Respuestas.
 - ▶ Marcadores, permite ver los marcadores establecidos dentro de cada tomo de la enciclopedia en la aplicación. Estos marcadores están organizados por tomo.
 - ▶ Opciones, permite personalizar la aplicación en cuanto a tipo y tamaño de letra, estilo del texto, tema de la aplicación y establecer un acceso directo rápido en la pantalla principal del dispositivo móvil.
 - ▶ Acerca de, esta opción permite ver la información de la aplicación desarrollada.

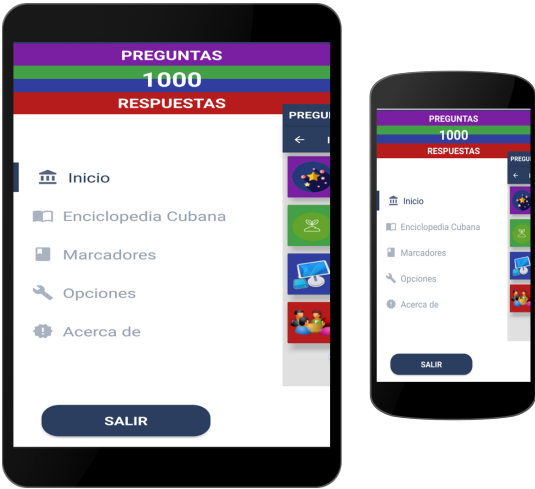


Figura 2.11: Prototipo de Pantalla: Splash

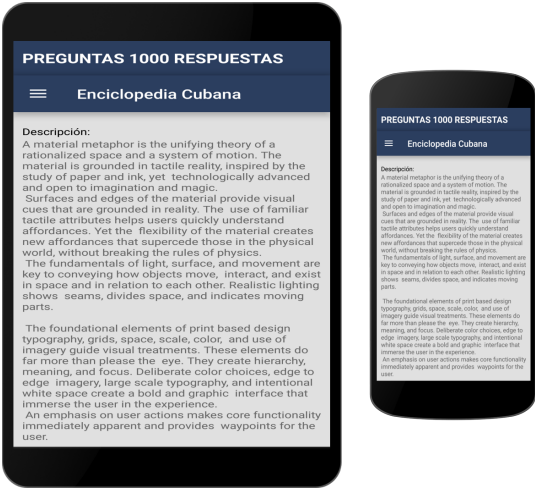


Figura 2.12: Prototipo de Pantalla: Enciclopedia Cubana



Figura 2.13: Prototipo de Pantalla: Marcadores



Figura 2.14: Prototipo de Pantalla: Opciones



Figura 2.15: Prototipo de Pantalla: Acerca de

En la Figura 2.16, se describe el StoryCard del Prototipado de Pantalla: Menú de Navegación Lateral.

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
02	Nuevo	Fácil	Fácil			Baja	
	Fijo	Moderado	Moderado			Media	
	Mejora	Duro	Duro			Alta	
Descripción							
El usuario presiona el botón de navegación, y aparece un menú lateral con las opciones de: Enciclopedia Cubana, Marcadores Opciones y Acerca de.							
Fecha	Estado		Comentario				
12/03/19	Definido		Sin comentario				
16/03/19	Implementado		Sin comentario				
21/03/19	Hecho		Sin comentario				
12/04/19	Verificado		Sin comentario				

Figura 2.16: Storycard del Menú de Navegación Lateral

● Pantallas de los Tomos de la Enciclopedia:

Las cuatro pantallas correspondientes a cada uno de los tomos de la enciclopedia poseen las mismas características, donde se pueden establecer marcadores en el contenido. La única diferencia existente entre estas pantallas consiste en el color que poseen, siendo este color el que posee el respectivo tomo a la enciclopedia en formato impreso (Figura 2.17).

- Layouts: Eos layouts tienen todo su nombre en minúsculas, si poseen más de dos palabras tendrán un guión bajo para separar cada una de ellas.

Estructura de Directorios:

En el Figura 2.18, se describe la estructura de la aplicación mostrando los directorios mas importantes.

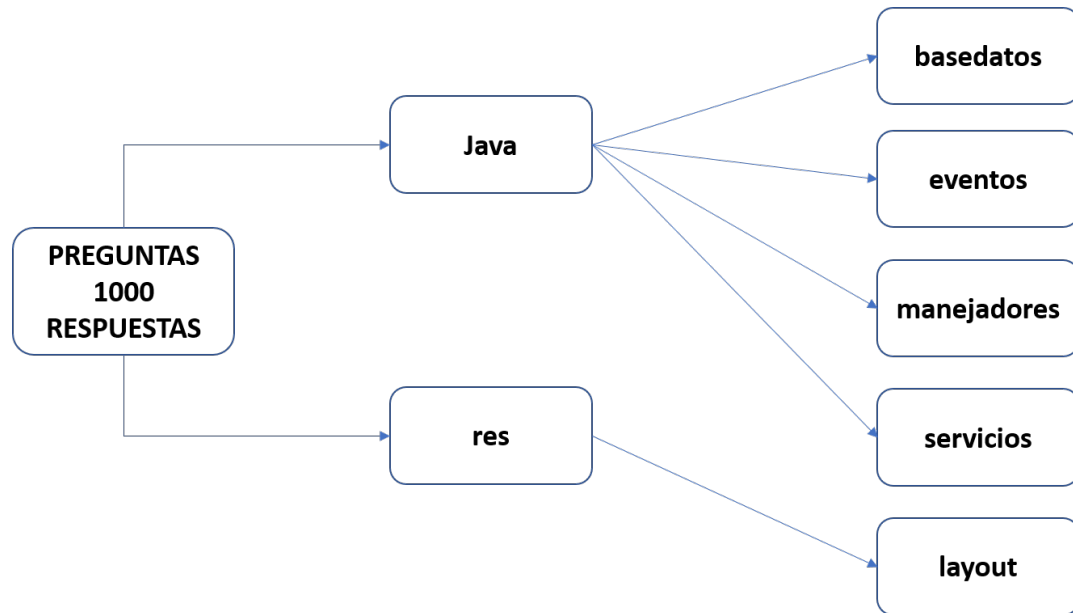


Figura 2.18: Estructura de directorios más significativos

El directorio eventos posee todos los eventos utilizados en la aplicación para controlar los procesos que se realizan dentro de la misma. Son de gran importancia para el directorio manejadores, el cual se encarga de desarrollar la lógica de la aplicación en segundo plano y mediante eventos se controla la terminación de métodos o invocación de objetos. El directorio servicios posee la clase que se encarga de brindar la posibilidad de salir de la aplicación mientras se obtiene la información de la base de datos.

Codificación:

La codificación se realizó basándose en cada uno de los storycards realizados en la fase anterior. A continuación se muestran las clases y métodos más importantes.

Pantalla Principal y Menú de Navegación Lateral:

Para la pantalla principal se realizó la construcción del Splash, se implementó la conexión a la base de datos, así como las respectivas consultas para obtener información de la misma.

- **Construcción del Splash** Para la creación del Splash se creó una actividad llamada *activity_splash* y se utilizaron varios métodos como los siguientes:

setRequestedOrientation(): Este método sirve para fijar una orientación al dispositivo, en este caso *ActivityInfo.SCREEN_ORIENTATION_PORTRAIT* se utiliza para el modo retrato; permite que el dispositivo no cambie de orientación mientras se carga el splash y así éste no se reinicie con algún movimiento del dispositivo.

Thread: Se utilizó esta clase para que el splash se ejecutara durante dos segundos.

- **Manejo de la Base de Datos**

En la clase *DataPreguntasRespuestas.java* (Algoritmo 2.1) se realiza todo el manejo de la base de datos, que permite obtener la información de los tomos .

Algoritmo 2.1: Clase *DataPreguntasRespuesta.java*

```

1 begin
2   public class DataPreguntasRespuestas
3   private SQLiteDatabase db;
4   private Context context;
5   private BDpregunatas_respuestas database;
6   public DataPreguntasRespuestas(Context pcontext)
7   public ArrayList<PreguntaRespuesta>listarUniverso()
8   public ArrayList<PreguntaRespuesta>listarVida()
9   public ArrayList<PreguntaRespuesta>listarInvenciones()
10  public ArrayList<PreguntaRespuesta>listarSociedad()
11  public ArrayList<PreguntaRespuesta>getListUniversoByKeyword(String search)
12  public ArrayList<PreguntaRespuesta>getListVidaByKeyword(String search)
13  public ArrayList<PreguntaRespuesta>getListInvencionesByKeyword(String
    search)
14  public ArrayList<PreguntaRespuesta>getListSociedadByKeyword(String search)
15  public ArrayList<PreguntaRespuesta>getMarcadoresUniverso()
16  public ArrayList<PreguntaRespuesta>getMarcadoresVida()
17  public ArrayList<PreguntaRespuesta>getMarcadoresInvenciones()
18  public ArrayList<PreguntaRespuesta>getMarcadoresSociedad()
19  public void setColumnMarcador(int id,int value)
20  public boolean isMarcado(int id)
21  public void copyMarcadores(int idTable,int id,String pregunta,String
    respuesta,String img ,int tomo)
22  public void deleteMarcador(int id)
23 end

```

Conclusiones Parciales

En este capítulo se diseñaron los principales artefactos de la metodología Mobile-D. Esto permitió:

1. Implementar cada uno de los storyboard confeccionados junto al usuario.
2. Desarrollar cada pantalla sin errores, ya que Mobile-D propone durante la fase de producción realizar varias pruebas durante cada interacción.
3. Hacer un correcto diseño de la base de datos así como del diagrama de componentes y casos de uso evitando errores durante la fase de producción de Mobile-D.

Capítulo 3

Descripción de las pruebas realizadas a la aplicación y Manual de usuario

Una vez culminado el desarrollo del software, es de vital importancia verificar y evaluar su funcionamiento y calidad. Debe verificarse si el software cumple con los requisitos definidos con anterioridad y genere los resultados esperados por los usuarios. Para lograr este objetivo es necesario ejecutar la aplicación en diversos dispositivos y con diferentes usuarios. Para realizar la entrega del producto terminado antes deben llevarse a cabo una serie de pruebas que ayudarán al proceso de corrección de errores y así mejorar la interacción con el usuario final.

3.1. Pruebas del sistema y correcciones a la aplicación

Luego de haber terminado con la codificación se deben realizar las pruebas que comprueben que es una aplicación de calidad y que cumple con los requerimientos por los que fue creada. Las pruebas que la metodología Mobile-D establece para una aplicación son la comprobación de requerimientos, los resultados en tiempos de respuestas y tiempos de accesos, además de una lista de dispositivos en dónde se ha comprobado el adecuado funcionamiento de la aplicación.

3.1.1. Pruebas de integración Manuales

En este tipo de prueba no solo se verifica el funcionamiento individual de cada componente del software móvil, sino también su integración con los demás componentes de la aplicación, especialmente con el servidor de datos con el cual se realiza el proceso de sincronización de la

información utilizada en el dispositivo (Yagüe and Garbajosa, 2009).

Se realizó una prueba exhaustiva en cada una de las posibles pantallas, verificando el correcto funcionamiento de cada uno de los componentes interactivos o no, y en el caso de los que presentan interacción se comprobó que cada uno de estos realizaba la función esperada, como se describe en las Figuras 3.1, 3.2, 3.3.



Figura 3.1: Prueba de Integración a la Pantalla Principal



Figura 3.2: Prueba de Integración al Menú Lateral

Acción: Pulsar
Mostrar Pantalla de cada
marcador por tomo.



Figura 3.3: Prueba de Integración a la Pantalla de Marcadores

3.1.2. Pruebas de compatibilidad

En las pruebas de compatibilidad se asegura que la aplicación móvil funcione correctamente, para esta prueba se seleccionaron varios dispositivos con diferentes tamaños de pantalla, resolución y versión de sistema operativo Android. Pueden ser de inspección manual o automatizada mediante la captura de pantallas que son posteriormente revisadas por los testers (Díaz, 2013).

La aplicación está diseñada para funcionar en cualquier Smartphone o Tablet con sistema operativo Android 5.0 – Lollipop o superior, por tanto, se realizaron pruebas en distintos dispositivos, estos dispositivos se muestran en la Figura 3.4.

	<ul style="list-style-type: none"> - Motorola Moto G XT-1072 (2015) - CPU Qualcomm Snapdragon Quad-Core 1.2GHz - Memoria RAM de 1GB - Almacenamiento interno 8GB. - Pantalla táctil 5.0" 720x1280 HD 16:9 - Android 4.0.3 - Jellybean
	<ul style="list-style-type: none"> - Samsung Galaxy S6 SM-G920F - CPU Cortex-A53 Quad-Core 1.5GHz & Exynos 7420 Quad-Core 2.1GHz & Qualcomm Snapdragon Quad-Core 1.2GHz - Memoria RAM de 3GB - Almacenamiento interno 32GB. - Pantalla táctil 5.1" 1440x2560 QuadHD 16:9 - Android 7.1.1 - Nougat
	<ul style="list-style-type: none"> - Blu Dash J - CPU MediaTek Dual-Core 1.3GHz + ARM Mail 400 GPU - Memoria RAM de 256MB - Almacenamiento interno 512MB. - Pantalla táctil 4.0" 480x800, 4:3 - Android 4.4 - KitKat
	<ul style="list-style-type: none"> - Samsung P4 - CPU ARM Cortex-A9 Dual-Core 1.6ghz + Nvidia Tegra GPU - Memoria RAM De 768MB - Almacenamiento Interno 16GB. - Pantalla Táctil 10.0" 800x1280 HD 16:9 - Android 5.1.1 - Lollipop
	<ul style="list-style-type: none"> - Huawei T1-7.0 - CPU Spreadtrum Quad-Core 1.2GHz + GPU Mail - Memoria RAM de 1GB - Almacenamiento interno 8GB. - Pantalla táctil 7.0" 600x1024, 16:9 - Android 6.0 - Marshmallow

Figura 3.4: Dispositivos de Prueba

En los Smartphones probados se pudo apreciar como la app se ajusta de manera automática a las variaciones de tamaño de pantalla y densidad de pixeles, además al instalarse en distintos sistemas operativos la aplicación respondió como se esperaba ya que al instalarla en dispositivos con sistema operativo Android inferior a la versión 5.0 – Lollipop, la aplicación no funcionó, ya que se implementó para versiones superiores o la 5.0 -Lollipop.

En los Tablets la aplicación se comporta correctamente, variando la forma en que se visualiza el contenido, para ajustarse a pantallas con un mayor tamaño se utilizaron layouts diferentes con el objetivo de aprovechar las ventajas de tener más espacio para mostrar información en dispositivos más grandes. También se probó en versiones distintas de Android para evaluar su comportamiento, el cual fue exitoso.

3.1.3. Pruebas de campo

Esta aplicación surge como resultado de una encuesta sobre la preferencia y el conocimiento de existencia que tenían los usuarios de la enseñanza media superior sobre los tomos impresos de la Enciclopedia 1000 Preguntas 1000 Respuestas, donde la mayoría de los usuarios no conocían esta enciclopedia y luego de conocer sus contenidos preferían tenerla en el móvil que portarla en formato impreso. Esta conclusión se extrajo de la siguiente encuesta ??

¡ ¡¡HOLA! ! !

Estamos haciendo una tesis para digitalizar una enciclopedia sobre temáticas desde vida, sociedad, inversiones y universo. Queremos consultarte tu opinión.

Esta encuesta es anónima. Sé sincero para ayudarnos por favor. Gracias por tu colaboración.

1) ¿Has consultado la Enciclopedia cubana "1000 Preguntas 1000 Respuestas"?



2) ¿Te gustaría conocerla?



3) ¿Por qué vía accediste?

☐ Te la presto alguien.

☐ La compraste.

☐ La consultaste en una institución.

4) ¿Cuál de los 4 tomos te gusto más?

a) Vida: ☐  ☐  ☐ 

b) Universo: ☐  ☐  ☐ 

c) Sociedad: ☐  ☐  ☐ 

d) Inversiones: ☐  ☐  ☐ 

5) ¿Te agrada que las imágenes acompañen a la respuesta?



6) ¿Consideras que varias preguntas estaban relacionadas?



7) ¿Te gustaría tener esa enciclopedia en tu celular?



8) Para descargar el contenido de la enciclopedia, marca la opción preferida.

☐ Toda la enciclopedia que tiene más de 20Gb.

☐ Cada tomo cuando lo necesites, tamaño aproximado entre 3 y 5 Gb.

☐ Cada dimensión que tiene entre 5 y 10 preguntas relacionadas.

Figura 3.5: Encuesta Realizada como prueba de campo

De esta encuesta se pudo obtener los siguientes datos:

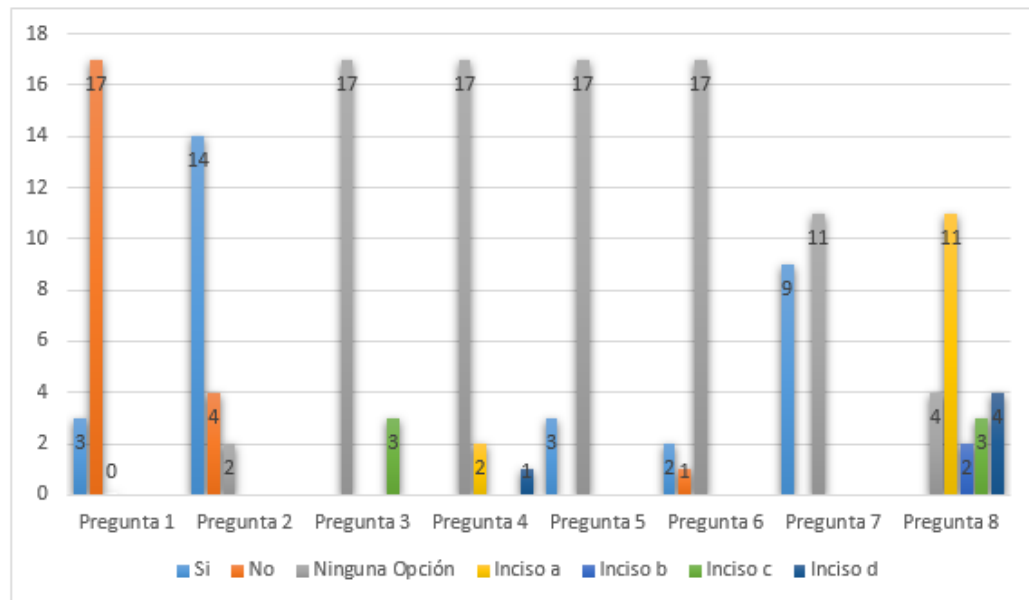


Figura 3.6: Gráfico que muestra los resultados obtenidos en la encuesta de campo aplicada

La encuesta se realizó a 20 usuarios de la enseñanza media superior, donde se pretendía averiguar el conocimiento de existencia de la Encilopedia 1000 Preguntas 1000 Respuestas. Se pudo constatar allí que la mayoría de los usuarios no conocían de la existencia de esta Encilopedia y algunos preferían portar esta en el móvil. De esta forma se pudo evidenciar que existe poca disponibilidad de los tomos impresos de la misma y esto conlleva a que no se conozca de su existencia. De igual modo, se pudo constatar que es preferida esta Enciclopedia en forma de aplicación para móviles que en formato impreso.

3.2. Manual de Usuario de la Aplicación: PREGUNTAS 1000 RESPUESTAS



Figura 3.7: Splash de la aplicación

Requisitos del Sistema:

1. Soporte: La versión del sistema operativo Android debe ser 5.0 (Lollipop) o superior.
2. Seguridad: A partir de la versión 6.0 (Marshmallow) del sistema operativo Android el usuario debe otorgar manualmente los permisos de escritura y lectura de la memoria interna.

3.2.1. Pantalla Principal de la Aplicación



Figura 3.8: Pantalla Principal de la Aplicación

Al precionar cualquier botón se muestra el contenido del tomo correspondiente, permitiendo navegar por la información del mismo. Es posible además utilizar un conjunto de opciones que son brindadas en el menú lateral que ofrece esta Pantalla Principal.

3.2.2. Tomos de la Aplicación

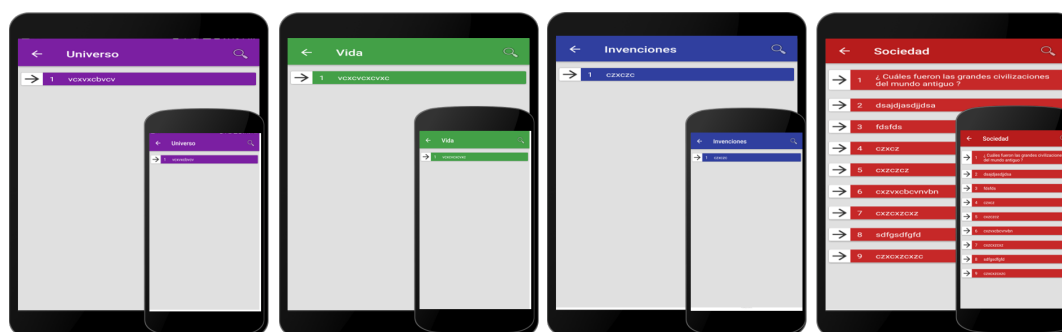


Figura 3.9: Pantalla de los Tomos de la Aplicación

Ya dentro de la información de cada tomo podemos consultar la información y establecer marcadores al contenido, esto es posible realizando un longpress sobre la pregunta preferida.

3.2.3. Menú Lateral de la Aplicación

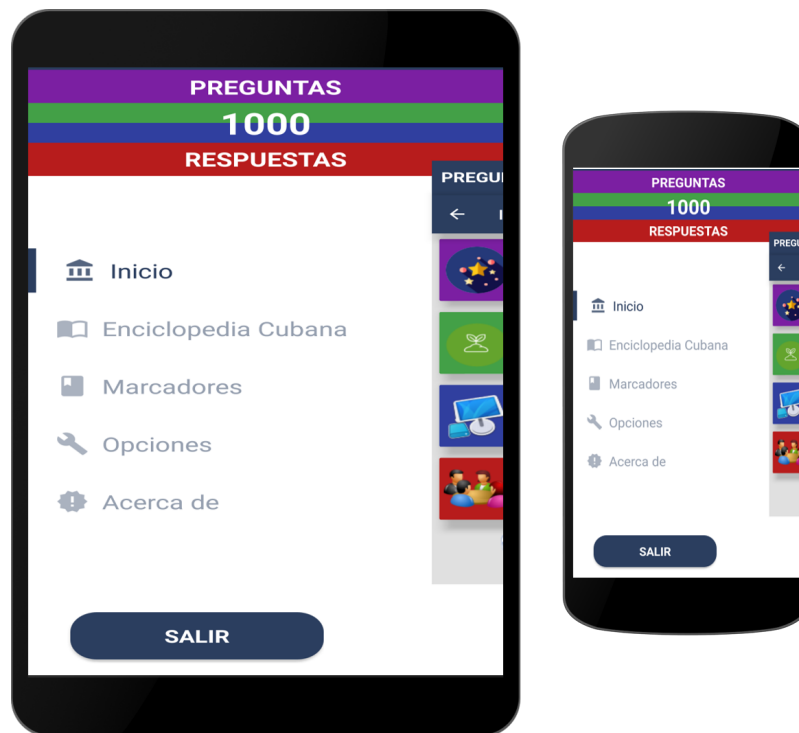


Figura 3.10: Menú Lateral de la Aplicación

El menú lateral brinda un conjunto de opciones como: Consultar la información general de la Enciclopedia 1000 Preguntas 1000 Respuestas, consultar los marcadores establecidos en los tomos de la Enciclopedia, un conjunto de opciones que permiten personalizar la aplicación y consultar el Acerca de, el cual contiene la información de los desarrolladores de la aplicación.

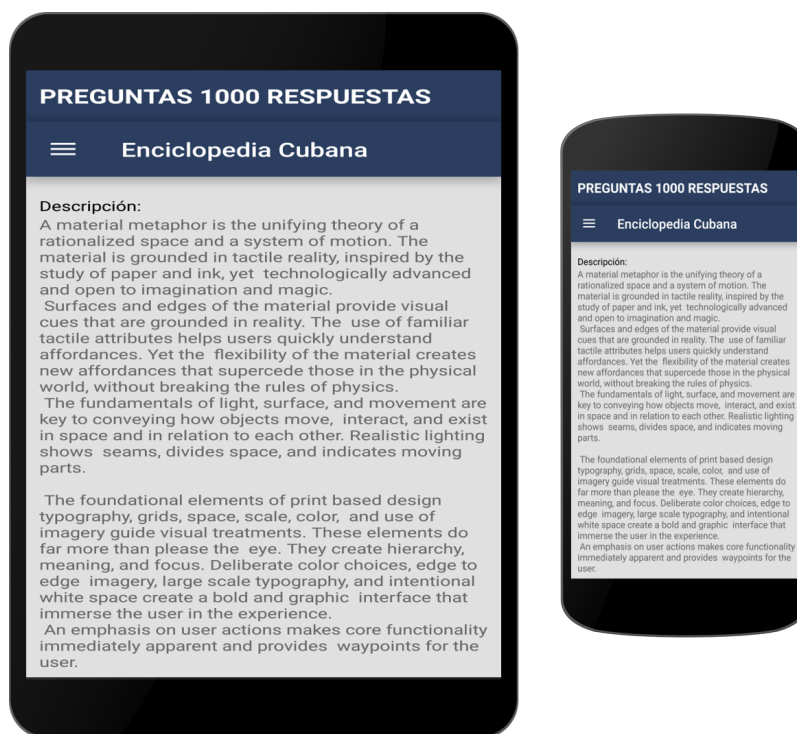


Figura 3.11: Menú Lateral: Enciclopedia Cubana



Figura 3.12: Menú Lateral: Marcadores



Figura 3.13: Menú Lateral: Opciones



Figura 3.14: Menú Lateral: Acerca de

Conclusiones Parciales

En este capítulo se describen las pruebas realizadas a la aplicación desarrollada, las mismas arrojaron los siguientes resultados:

1. Las pruebas realizadas por la herramienta online Monkop sobre los dispositivos seleccionados, mostró que la aplicación utiliza pocos recursos físicos de los dispositivos móviles, y obtuvo resultados satisfactorios.
2. Se pudo comprobar que el inicio de la aplicación es rápido y sin errores de ejecución o bloqueo.
3. La interfaz es responsive y se adapta a distintas dimensiones de pantalla sin perder información ni sobrecargar de elementos la misma.
4. Se confeccionó un pequeño manual de usuario para facilitar el uso de la aplicación por los usuarios.

Conclusiones

A partir del siguiente trabajo, donde se implementó una aplicación para móviles que complementa la existencia en formato impreso de la Enciclopedia Cubana 1000 Preguntas 1000 Respuestas. Se pudo concluir que:

1. Para el desarrollo de la aplicación se decidió optar por una metodología ágil ya que estas tienen un tiempo corto para la entrega del producto final y son las más usadas en el desarrollo de aplicaciones para dispositivos móviles. Entre los Framework de desarrollo de aplicaciones móviles para la plataforma Android se utilizó Android Studio, ya que consiste en la herramienta que brindan los desarrolladores del Sistema Operativo Android para el desarrollo de aplicaciones para el mismo.
2. Para el almacenamiento de la información se diseñó una base de datos SQLite que permite la portabilidad de la información en forma de un fichero y garantiza una excelente compatibilidad con el Sistema Operativo Android.
3. Se realizaron pruebas a la aplicación desarrollada comprobando que la misma funciona tanto en dispositivos con diferentes características de hardware y versiones de Sistema Operativo Android superiores a la 5.0 así como en diferentes dimensiones de pantalla.
4. Se confeccionó un manual de usuario que explica las principales funcionalidades de la aplicación.

Recomendaciones

1. Implementar la funcionalidad autores que permite visualizar los autores con sus preguntas y respuestas.
2. Agregar contenido multimedia que permita ejemplificar el contenido consultado.
3. Crear una red semántica para relacionar las preguntas entre los tomos.

Bibliografía

- Agenjo-Bullón, X., Hernández-Carrascal, F., 2010. La biblioteca virtual: función y planteamiento.
- Aguado-Terrón, J.-M., Martínez-Martínez, I. J., 2009. De la web social al móvil 2.0: el paradigma 2.0 en el proceso de convergencia mediática de la comunicación móvil. *El profesional de la información* 18 (2).
- Agustín, G. C., 2002. Gestión del proceso software. EDITORIAL RAMÓN ARECES.
- Arroyo-Vázquez, N., 2009. Web móvil y bibliotecas. *El profesional de la información* 18 (2), 129–136.
- Avison, D., Cole, M., Fitzgerald, G., 2006. Reflections on teaching information systems analysis and design: from then to now! *Journal of Information Systems Education* 17 (3).
- Balaguera, Y. D. A., 2013. Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. estado actual. *Revista de Tecnología* 12 (2), 111–123.
- Benbourahla, N., 2015. Android 5: principios del desarrollo de aplicaciones Java. Ediciones Eni.
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P., 2009. Metodología de desarrollo ágil para sistemas móviles. introducción al desarrollo con android y el iphone. *Dr. en Ing. Sist. Telemáticos*, 1–30.
- Cadavid, A. N., Martínez, J. D. F., Vélez, J. M., 2013. Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva* 11 (2), 30–39.
- Canós, J. H., Letelier, M. C. P. P., 2012. Metodologías ágiles en el desarrollo de software.
- Cinar, O., 2015. Android quick APIs reference. Apress.
- Cuello, J., Vittone, J., 2013. Diseñando apps para móviles. José Vittone—Javier Cuello.

- de Cuba, P. C., 2017. Lineamientos de la política económica y social del partido y la revolución para el período 2016-2021. Tech. rep.
- Delía, L. N., Galdamez, N., Thomas, P., Pesado, P. M., 2013. Un análisis experimental de tipo de aplicaciones para dispositivos móviles. In: Congreso Argentino de Ciencias de la Computación (CACIC). Vol. 18.
- Developers, A., 2011. What is android? android developers.
- Díaz, S., 2013. Mejores prácticas en las pruebas de aplicaciones móviles. España: ATSistemas.
- Farman, J., 2013. Mobile interface theory: Embodied space and locative media. Routledge.
- Hassan Montero, Y., Martín Fernández, F. J., 2005. La experiencia del usuario. No sólo usabilidad (4).
- Jover, J. N., Arriete, L. F. M., 2018. La política de ciencia, tecnología e innovación en la actualización del modelo económico cubano: evaluación y propuestas/science, technology and innovation policy in the updating of the cuban economic model: evaluations and proposals. Revista Economía y Desarrollo (Impresa) 150 (2).
- Lie, H. W., Bos, B., Lilley, C., Jacobs, I., 2005. Cascading style sheets. Pearson India.
- Luján-Mora, S., 2002. Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.
- Martín, D., 2010. Diseño de aplicaciones adaptativas. No Solo Usabilidad (9).
- Pérez Guerrero, J. X., 2018. Desarrollo de una aplicación basado en plataforma android y aplicativo web utilizando html para el registro de ciudadanos basado en perfiles socio delictivos del distrito esteros. Ph.D. thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas
- Peñuela Epalza, M., Paternina Del Rio, J., Moreno Santiago, D., Camacho Pérez, L., Acosta Barrios, L., De León De León, L., 2014. El uso de los smartphones y las relaciones interpersonales de los jóvenes universitarios en la ciudad de barranquilla (colombia). Salud uninorte 30 (3).
- Piattini Velthuis, M. G., Garcia Rubio, F., Caballero Muñoz-Reja, I., 2007. Calidad de sistemas informáticos. Tech. rep., Alfaomega Ra-Ma,.
- Pichler, R., 2016. Strategize: product strategy and product roadmap practices for the digital age. Pichler Consulting.

- Poole, D. B., Nov. 3 2009. Systems and methods for determining the software components included in a view of a software development project at a particular time. US Patent 7,614,038.
- Rodríguez-Gutiérrez, M., 2015. Hacia una problematización del proceso de trabajo en los programadores de software. una propuesta desde la perspectiva de la sociología del trabajo.
- ROLDAN, M. M., THOMPSON, D. N., 2018. Aplicaciones móviles nativas orientadas a servicios y recursos de bibliotecas universitarias. In: XVI Congreso Edutec Costa Rica.
- Santiago, R., Trbaldo, S., 2015. Mobile learning. Nuevas realidades en el aula. Editorial Océano.
- Snyder, C., 2003. Paper prototyping: The fast and easy way to design and refine user interfaces. Morgan Kaufmann.
- Solanky, J., Patil, K., Patel, G., 2016. Resemblance of phonegap and titanium for mobile application development. International Journal of Computer Applications 144 (10).
- Torres Tapia, I. A., Solano Mosquera, H. R., 2013. Análisis de frameworks para el desarrollo de aplicaciones móviles en la plataforma android. B.S. thesis, Universidad del Azuay.
- Tracy, K. W., 2012. Mobile application development experiences on apple, s ios and android os. Ieee Potentials 31 (4), 30–34.
- Treude, C., Aniche, M., 2018. Where does google find api documentation? In: Proceedings of the 2nd International Workshop on API Usage and Evolution. ACM, pp. 19–22.
- Vanegas, C. A., 2012. Desarrollo de aplicaciones sobre android. Revista vínculos 9 (2), 129–145.
- Wolfram, W., Schilling, N., 2015. American English: dialects and variation. Vol. 25. John Wiley & Sons.
- Yagüe, A., Garbajosa, J., 2009. Comparativa práctica de las pruebas en entornos tradicionales y ágiles. REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software 5 (4).

ANEXOS

Anexo A