

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“APLICATIVO MÓVIL MULTIPLATAFORMA PARA LA
PROMOCIÓN DE UN DESTINO TURÍSTICO, CASO:
CENTRO HISTÓRICO DE AREQUIPA”**

Tesis presentada por el Bachiller:

Alfonso Rodrigo Chicata Abarca

Para optar el título profesional de:

INGENIERO DE SISTEMAS

AREQUIPA – PERÚ

2014

DEDICATORIA

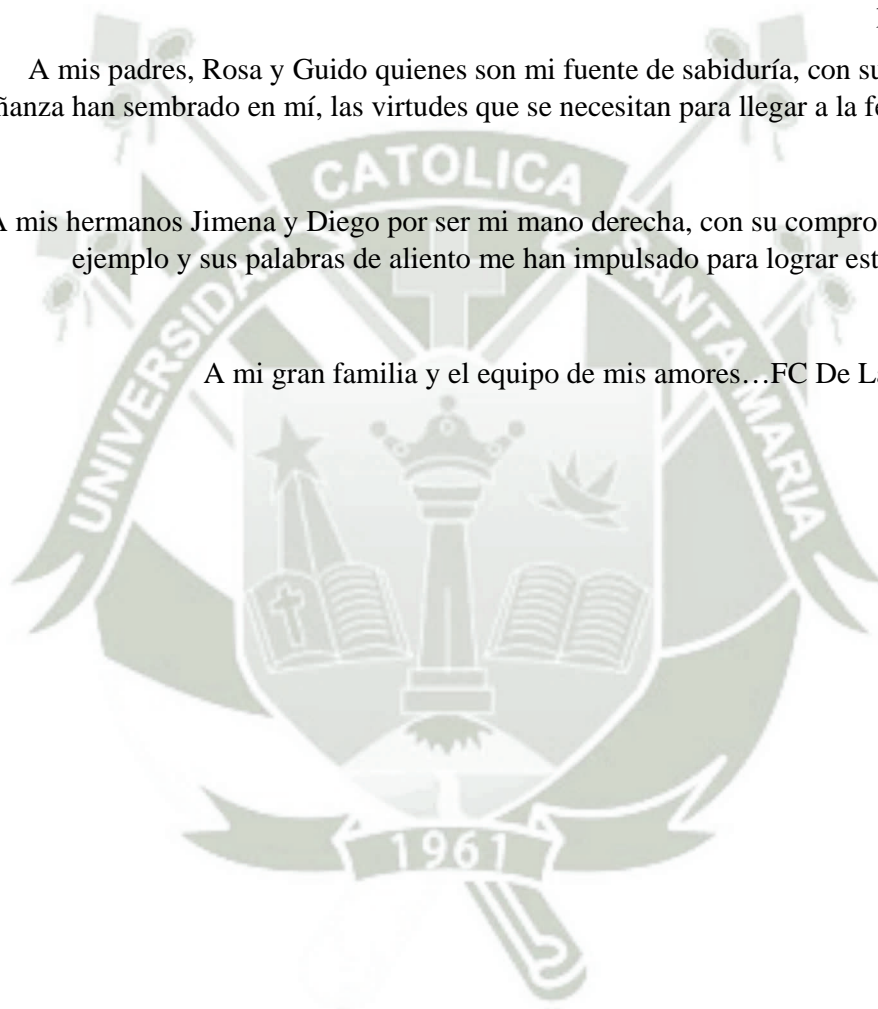
La presente Tesis está dedicada a Dios, por ser la luz que me ilumina por el buen camino, gracias a él puedo dar esta alegría a mis seres queridos.

A mis padres Mayrene y Alfonso porque además de darme todo en la vida, son la fuente de inspiración y motivación para el desarrollo de esta Tesis. Por estar siempre a mi lado para darme su apoyo incondicional y guiarme para ser de mí cada vez una mejor persona.

A mis padres, Rosa y Guido quienes son mi fuente de sabiduría, con su amor y enseñanza han sembrado en mí, las virtudes que se necesitan para llegar a la felicidad.

A mis hermanos Jimena y Diego por ser mi mano derecha, con su compromiso, su ejemplo y sus palabras de aliento me han impulsado para lograr este sueño.

A mi gran familia y el equipo de mis amores...FC De La Salle!!



ÍNDICE

	Pág.
RESUMEN	
ABSTRACT	
INTRODUCCIÓN	
CAPÍTULO I: PLANTEAMIENTO TEÓRICO	
1.1. Problema de Investigación	5
1.1.1. Enunciado del Problema	5
1.1.2. Descripción del Problema	5
1.1.3. Formulación Interrogativa del Problema	6
1.1.4. Justificación de la Investigación	6
1.1.5. Alcances y Limitaciones	6
1.2. Objetivos	7
1.2.1. Objetivo General	7
1.2.2. Objetivos Específicos	7
CAPÍTULO II: MARCO TEÓRICO	
2.1. Estado del Arte	8
2.1.1. Antecedentes de la Investigación	8
2.1.2. Antecedentes de Dispositivos Móviles	12
2.1.3. Antecedentes de Aplicaciones Móviles	15
2.2. Situación Actual	16
2.2.1. Mercado de Dispositivos Móviles	16
2.2.2. Mercado de Aplicaciones Móviles	19

2.2.3. Turismo	21
2.2.4. Turismo en el Centro Histórico de Arequipa	24
2.2.5. Tecnología de Información en el Turismo	28
2.2.6. Aplicaciones Móviles para el Turismo	30
2.3. Sistemas Operativos Móviles	34
2.3.1. Sistema Operativo Android	34
2.3.2. Sistema operativo IOs	38
2.3.3. Sistema Operativo Windows 8.1 y Windows Phone 8	41
2.4. Desarrollo de Aplicaciones Móviles Multiplataforma	43
2.4.1. Principales Características de las Aplicaciones Móviles Multiplataforma	44
a. Experiencia de usuario	44
b. Directrices para la satisfacción del usuario	44
c. Arquetipos para la innovación	46
d. Usabilidad	47
e. Métricas de usabilidad de una aplicación móvil	49
2.4.2. Tipos de aplicaciones móviles multiplataforma	51
2.4.2.1. Aplicaciones basadas en desarrollo web	52
a. Ventajas de las aplicaciones basadas en desarrollo web	55
b. Desventajas de las aplicaciones basadas en desarrollo web	55
2.4.2.2. Aplicaciones nativas	57
a. Características de las aplicaciones móviles nativas	58
b. Ventajas de las aplicaciones móviles nativas	59
c. Desventajas de las aplicaciones móviles nativas	59

2.4.2.2. Aplicaciones móviles híbridas	60
a. Características de las aplicaciones híbridas	61
b. Ventajas de las aplicaciones híbridas	61
c. Desventajas de las aplicaciones híbridas	62
2.5. Frameworks para el desarrollo de aplicaciones móviles multiplataforma	62
2.5.1. JQuery Mobile	62
a. Anatomía de una App Sencha	64
2.5.2. Phonegap	65
a. Ventajas	66
b. desventajas	67
2.5.3. Titanium Appcelerator	67
2.5.4. Strawberry	69
2.5.5. Qt 5.2	72
a. Flexibilidad	74
2.5.6. Xamarin	74
2.5.7. Código Compartido	77
2.6. Computación en la Nube para Aplicaciones Móviles	77
2.6.1. La Fiabilidad y el Tiempo de Actividad	78
2.6.2. Apoyo Técnico	78
2.6.3. Espacio de Almacenamiento	79
2.6.4. Facilidad de Uso	79
2.6.5. Los Enfoques de la Computación en la Nube	79
2.6.6. Infraestructura como Servicio	79

2.6.7. Software como Servicio	80
2.6.8. Plataforma como Servicio	81
2.6.9. Backend como Servicio	81
2.6.10. Propósito	82
2.6.11. Modelo de Negocio	82
2.6.12. Windows Azure	83
a. Características de Windows Azure	84
b. Componentes Azure Platform	85
c. Beneficios de Windows Azure	87
d. Microsoft Azure Storage	88
e. Windows Azure Mobile Services	91
f. Escenarios Reales con los Servicios móviles de Windows Azure	92
CAPÍTULO III: DESARROLLO DEL APLICATIVO MÓVIL	
3.1. Metodología de Desarrollo	95
3.2. Análisis de Requerimientos	96
3.2.1. Requerimientos Funcionales	97
3.2.2. Requerimientos no Funcionales	98
3.3. Diagrama de Casos de Uso	98
3.4. Especificación de Casos de Uso	99
3.4.1. Caso de Uso Registrar Usuario	99
3.4.2. Caso de Uso Iniciar Sesión	100
3.4.3. Caso de Uso Seleccionar Destino	101
3.4.4. Caso de Uso Escribir Comentario	102
3.4.5. Caso de Uso Valorar Destino Turístico	103

3.4.6. Caso de Uso Notificar Nuevo Comentario	104
3.4.7. Caso de Uso Ir a la Vista de Lectura	105
3.5. Diagrama de Clases	106
3.6. Wireframes	107
3.7. Arquitectura	110
3.7.1. Aplicación del Patrón de Diseño MVVM	110
a. Modelo	110
b. Vista	111
c. Modelo de Vista	112
3.7.2. Estrategia de Organización de un Proyecto	112
a. Librerías de clase portables o PCL	112
3.7.3. Ventajas y Desventajas de Utilizar una Librería de Clases Portable	113
a. Ventajas	113
b. Desventajas	114
3.7.4. Proyecto Compartido	114
3.7.5. Enlace de Datos entre Capas	115
3.7.6. Interfaces Importantes	117
3.7.7. Definición de la Arquitectura	119
3.7.8. Capas de la Aplicación	121
CAPÍTULO IV: PRUEBAS Y RESULTADOS	
4.1. Capturas de Pantalla de la Aplicación	123
4.1.1. Registro e Inicio de Sesión	123
4.1.2. Escribir un Comentario	126
4.2. Usabilidad e Interactividad del Aplicativo	127

4.3. El Contenido del Aplicativo	129
4.4. Análisis de los Resultados	130
4.4.1. Rendimiento	135
4.4.2. Puntos Crash	136
4.4.3. Conectividad	136
CONCLUSIONES	137
RECOMENDACIONES	139
REFERENCIAS BIBLIOGRÁFICAS	
ANEXOS	
GLOSARIO DE TÉRMINOS	



ÍNDICE DE TABLAS

	Pág.
Tabla 1: Tipos de destinos turísticos	23
Tabla 2: Tipos de turista	24
Tabla 3: Actividades realizadas por el turista nacional	25
Tabla 4: Actividades realizadas por el turista extranjero	26
Tabla 5: Llegada de visitantes al monasterio de Santa Catalina. Enero 2005 – Diciembre 2012.	27
Tabla 6: De métricas utilizadas para la evaluación de características móviles.	50
Tabla 7: Requerimiento funcional	97
Tabla 8: Requerimiento no funcional	98
Tabla 9: Caso de uso registrar usuario	99
Tabla 10: Caso de uso iniciar sesión	100
Tabla 11: Caso de uso seleccionar destino	101
Tabla 12: Caso de uso escribir comentario	102
Tabla 13: Caso de uso valorar destino turístico	103
Tabla 14: Caso de uso notificar nuevo comentario	104
Tabla 15: Caso de uso ir a la vista de lectura	105

ÍNDICE DE FIGURAS

	Pág.
Figura 1: Participación de páginas visitadas desde diferentes dispositivos móviles	16
Figura 2: Porcentaje de usuarios por sistema operativo	18
Figura 3: Participación por marcas en el mercado global de tabletas, 2012-2016	19
Figura 4: Participación de marcas de dispositivos móviles en el mercado peruano	21
Figura 5: Aplicación turística Be your Guide- Toledo	32
Figura 6: Aplicación turística: Perú Natural	33
Figura 7: Aplicación turística del centro histórico de Lima	33
Figura 8: Arquitectura del sistema operativo Android	37
Figura 9: Logo de la tienda virtual de Android	39
Figura 10: Arquitectura del sistema operativo IOs	39
Figura 11: Arquitectura del sistema operativo Windows Phone	42
Figura 12: Flujo de la definición de usabilidad de una aplicación móvil	48
Figura 13: Soluciones Multiplataforma para dispositivos móviles	53
Figura 14: Arquitectura de una aplicación móvil basada en la web	54
Figura 15: Arquitectura de una aplicación móvil multiplataforma nativa	66
Figura 16: Logo de JQuery	63
Figura 17: Arquitectura de una aplicación Sencha	64
Figura 18: Logo de PhoneGap	66

Figura 19:	Logo AppCelerator	68
Figura 20:	Logo Strawberry	70
Figura 21:	Logo QT	72
Figura 22:	Logo Xamarin	75
Figura 23:	Arquitectura de una aplicación móvil en Xamarin	76
Figura 24:	Tipo de servicios en la nube	80
Figura 25:	Arquitectura de una aplicación móvil multiplataforma utilizando Azure	84
Figura 26:	Servicios de Windows Azure Storage	90
Figura 27:	Arquitectura de Windows Azure Services	92
Figura 28:	Mapa conceptual de la metodología de desarrollo del aplicativo móvil	95
Figura 29:	Diagrama de Casos de uso.	98
Figura 30:	Diagrama de clases	106
Figura 31:	Wireframe de la vista principal	107
Figura 32:	Wireframe de la vista de comentarios	108
Figura 33:	Wireframe de la vista de mapas	108
Figura 34:	Wireframe de la vista principal de comentarios	109
Figura 35:	Wireframe de la vista de lectura	109
Figura 36:	Equivalentes para telefonos	110
Figura 37:	Arquitectura del patrón Model View ViewModel	112
Figura 38:	Proyecto Compartido	115
Figura 39:	Código base para los enlaces de datos	116
Figura 40:	Código de interface namespace	118

Figura 41:	Código de interface namespace	118
Figura 42:	Arquitectura propuesta para la aplicación Móvil multiplataforma	119
Figura 43:	Vista de login y registro	124
Figura 44(a):	Inicio de sesión y registro	124
Figura 44(b):	Inicio de sesión y registro	125
Figura 45:	Login y registro con Facebook en teléfonos inteligentes	125
Figura 46:	Vista multimedia principal	126
Figura 47:	Usabilidad e interactividad del aplicativo	127
Figura 48:	Funcionalidad de la interactividad de la aplicación.	128
Figura 49:	Funcionalidad de la interactividad de la aplicación.	128
Figura 50:	Vista de Contenido informativo brindado por el aplicativo	129
Figura 51:	Vista de Mapa del Centro Histórico	129
Figura 52:	Vista de Contenido informativo brindado por el aplicativo	130
Figura 53:	¿La aplicación móvil multiplataforma cumple con lo propuesto, para promocionar un destino turístico?	131
Figura 54:	¿Cómo evalúa el diseño de interfaz de usuario de la aplicación?	132
Figura 55:	¿Cómo evalúa el desempeño de la aplicación en un dispositivo real?	132
Figura 56:	¿Cómo evalúa la usabilidad de la aplicación en una tableta?	133
Figura 57:	¿Cómo evalúa la usabilidad de la aplicación en un teléfono inteligente?	134
Figura 58:	¿Usarías la aplicación en tus viajes?	135

RESUMEN

Al día de hoy, millones de usuarios utilizan asiduamente sus dispositivos móviles para interactuar con su entorno. En la mayoría de los casos, el proceso de interacción está gestionado por una aplicación software específica. Debido a las características de este tipo de aplicaciones, en la mayoría de casos deben de ser desarrolladas como aplicaciones nativas para cada plataforma móvil destino. Tomando esto como premisa, uno de los retos a los que la informática móvil ha sido llamada, es superar la fragmentación de dispositivos y sistemas operativos móviles.

Cuando se trata de desarrollo nativo de una aplicación para varias de las plataformas existentes requiere una inversión grande en cuanto a recursos destinados para el desarrollo. A través de la introducción del desarrollo multiplataforma, los desarrolladores pueden crear aplicaciones para múltiples plataformas utilizando un mismo código base, reutilizable entre las diferentes plataformas.

Para solucionar los problemas de compatibilidad entre las diferentes plataformas, las herramientas de desarrollo multiplataforma ofrecen las mismas ventajas que un desarrollo aislado para cada plataforma, tales como *performance*, acceso a los accesorios del dispositivo, independencia de una conexión a internet, también se aprecia que el desarrollo multiplataforma era más barato y lleva menos tiempo.

En esta investigación se desarrolla un aplicativo móvil para la promoción del centro histórico de Arequipa, aplicación disponible en español y desarrollado para las tres plataformas más importantes del mercado global (Android, Windows y IOs), utilizando los servicios de la nube para la gestión de usuarios, así como las redes sociales para la promoción del aplicativo móvil.

ABSTRACT

Nowadays millions of users regularly use their mobile devices to interact with their environment. In most cases, the interaction process is managed by a specific software application. Because the features of this type of application, in most cases must be developed as native applications on each mobile platform destination. Taking this as a premise, one of the challenges that mobile computing has been called for, is to overcome the fragmentation of mobile operating systems and devices.

When it comes to native application development for multiple of existing platforms requires a large investment in terms of resources for development. Through the introduction of cross-platform development, developers can build applications for multiple platforms using a single code base, reusable between the different platforms.

To fix the compatibility problems between different platforms, cross-platform development tools offer the same advantages as isolated development for each platform, such as performance, access device accessories, regardless of an internet connection, also shows that the platform development was cheaper and takes less time.

In this research a mobile application for the promotion of the historic center of Arequipa, application available in Spanish and developed for the three major platforms of the global market (Android, Windows and IOs) was developed, using cloud services for managing users and social networks to promote mobile application.

INTRODUCCIÓN

Las aplicaciones para dispositivos móviles, aplicaciones móviles, están en la etapa en que deben existir para múltiples plataformas de manera que cubran la mayor cantidad de usuarios posibles.

El desarrollo nativo para múltiples plataformas requiere la inversión más grande recursos de desarrollo de software, algo que resulta demasiado costoso. A través de desarrollo multiplataforma, los desarrolladores pueden crear aplicaciones para múltiples plataformas utilizando el mismo código base.

Los escenarios de las aplicaciones móviles se han convertido en uno de los más variados campos, el campo de la computación móvil trajo consigo aplicaciones de movilidad y conectividad completa. El turismo es una actividad económica que está muy relacionada con el desarrollo de aplicaciones móviles, este avance tecnológico es importante para la mejora de sus procesos, así como para la promoción efectiva de sus productos y servicios.

Las aplicaciones pueden ser desarrolladas haciendo uso de una gran variedad de herramientas que existen actualmente, metodologías variadas de desarrollo usando la metodología tradicional y también nuevas metodologías que surgen de la evolución de la computación móvil. El número de herramientas que prometen una vista limpia y sin problemas, acelerando el proceso de desarrollo. Las aplicaciones móviles pueden dividirse en dos tipos principales que son aplicaciones nativas y aplicaciones multiplataforma.

En el capítulo I, Planteamiento Teórico, se plantea el problema que motivó a la realización de la arquitectura, así como también los objetivos trazados para obtenerlo.

En el capítulo II, Marco Teórico, se abordan los antecedentes de los dispositivos móviles así como de los aplicativos móviles, también se expone como se encuentra la situación actual del mercado de los mismos. Por otro lado se explica el turismo en general y local, así como la tecnología de la información que se aplica en esta actividad económica y por último las aplicaciones móviles que existen en el turismo.

En este capítulo se explica también los aspectos teóricos de los sistemas operativos, el desarrollo de aplicaciones móviles multiplataforma, los *frameworks* para el desarrollo de aplicaciones móviles y la computación en la nube para aplicaciones móviles.

En el capítulo III, Desarrollo del Aplicativo Móvil, se describe la metodología de desarrollo, el análisis de requerimientos, los diagramas de casos de uso, las especificaciones de los mismos, los diagramas de clases, los *wireframes* y la arquitectura de la solución.

En el capítulo IV, Pruebas y Resultados, se prueban los escenarios creados usando la arquitectura propuesta.

Finalmente, se expone las conclusiones, recomendaciones y hacia donde se enfoca este trabajo en el futuro.

CAPÍTULO I: PLANTEAMIENTO TEÓRICO

1.1. Problema de Investigación

1.1.1. Enunciado del Problema

“Aplicativo móvil multiplataforma para la promoción de un destino turístico caso: Centro Histórico de Arequipa”

1.1.2. Descripción del Problema

La forma más óptima de desarrollar aplicaciones móviles es haciendo uso de herramientas nativas. Sin lugar a dudas el rendimiento de la aplicación tanto en uso de recursos y de la interacción del usuario será de las mejores, pero cabe señalar que el costo profesional para su desarrollo será relativamente alto, cuando se trata de tener la misma aplicación en la mayor cantidad de dispositivos posibles.

Uno de los principales retos del desarrollo nativo es la existencia de una variedad de plataformas y dispositivos en el mercado móvil, haciendo que el desarrollo de aplicativos móviles termine siendo muy fragmentado y costoso es sus estrategias de desarrollo de software.

Una aplicación móvil multiplataforma es ampliamente adaptable, lo que significa que se desarrolla una sola vez y puede ser desplegada en cualquier dispositivo, o en una variedad importante de los mismos. Actualmente el mercado provee una vertiginosa cantidad de herramientas de desarrollo móvil multiplataforma.

1.1.3. Formulación Interrogativa del Problema

El desarrollo de aplicaciones móviles nativas para múltiples sistemas operativos móviles, genera diferentes costos en las estrategias de desarrollo de aplicaciones móviles. Además el tiempo empleado crece proporcionalmente a la cantidad de sistemas operativos para lo que se desea desarrollar la aplicación multiplataforma. ¿Cómo se puede optimizar estos recursos para la elaboración de aplicativos móviles multiplataforma sin afectar las características que se obtienen al desarrollar nativamente?

1.1.4. Justificación de la Investigación

El desarrollo móvil multiplataforma compartido propicia la optimización de la construcción y un mayor aprovechamiento del conocimiento para lo cual se plantea una arquitectura que permite la reutilización de código fuente.

1.1.5. Alcances y Limitaciones

El presente trabajo de investigación, desarrollará un aplicativo móvil para la promoción del centro histórico de Arequipa. Esta aplicación va estar disponible en idioma español y se va a desarrollar para las tres plataformas más importantes del mercado global (Android, Windows y IOs). Además se utilizará los servicios de la nube para la gestión de usuarios así como las redes sociales para la promoción del aplicativo móvil.

Una vez analizados los alcances es necesario considerar las limitaciones que se presentan ante un proyecto de estas características. En primer lugar nos encontramos frente a tecnologías relativamente nuevas, con terminales móviles de última generación.

En segundo lugar, la inestabilidad de los *framework* de desarrollo multiplataformas existentes y por último la constante actualización de las versiones de los sistemas operativos móviles, lo cual implica tener en cuenta nuevas características en el desarrollo

1.2. Objetivos

1.2.1. Objetivo General

Proponer un prototipo de aplicación móvil multiplataforma para el despliegue de una guía turística para la promoción del Centro Histórico de Arequipa.

1.2.2. Objetivos Específicos

- Aplicar el patrón de diseño MVVM (Modelo, Vista, Modelo de Vista) al desarrollo de un aplicativo multiplataforma que permita compartir código fuente entre las diferentes plataformas, de tal forma que el desarrollo sea óptimo.
- Implementación de un *back-end* compartido en la nube, que permita el inicio de sesión de usuarios y compartir opiniones sobre el aplicativo.
- Implementar interfaces de usuario interactivas que tengan impacto en la interacción de usuario y un óptimo uso de las ventajas del hardware de los dispositivos.

CAPÍTULO II: MARCO TEÓRICO

2.1. Estado del Arte

2.1.1. Antecedentes de la Investigación

- Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma Android de Google

Autor: Jaime Aranz Tudela

Institución: Universidad Carlos III de Madrid

Año: 2009

El presente proyecto busca conocer y comprender las características y el funcionamiento de este nuevo sistema operativo, averiguando sus posibilidades y ventajas frente a otras alternativas. Además, abarca el desarrollo completo de una aplicación de nombre ContactMap, que permite localizar a los contactos mediante mapas y con la que se busca ilustrar de forma práctica la construcción y naturaleza de las aplicaciones para Android.

- “Propuesta metodológica para implementar aplicaciones móviles utilizando Java me partir del Threat modeling”

Autor: Edwin Junior Martínez Pacheco

Institución: Universidad Católica de Santa María

Año: 2011

En esta tesis, el autor propone un planteamiento metodológico que cubra todas las cuestiones posibles inherentes al desarrollo de aplicativos móviles. De una manera sencilla y clara, se propone un enfoque proactivo, donde se cubran y mitiguen posibles riesgos en el diseño de aplicaciones, mediante la utilización del concepto de modelamiento de amenazas más conocido como “threat modeling”. Se desarrollará y evaluará un aplicativo de prueba utilizando el enfoque propuesto con el objetivo de ejemplificar el uso de la metodología. Se identificarán partes esenciales del aplicativo para su correcto análisis y prueba. De esta manera el autor prueba que la metodología propuesta es viable en cada una de las etapas que propone.

- “Propuesta de un sistema de monitoreo en dispositivos móviles sobre la plataforma Android para un sistema de ventas en empresas distribuidoras”

Autor: Antony Escobar Mamani, Luis García Santillana

Institución: Universidad Católica de Santa María

Año: 2011

La tesis propone un sistema de monitoreo desarrollado sobre la plataforma Android en dispositivos móviles para ayudar a la gestión de una empresa distribuidora y de esta manera lograr un mejor acceso a la información para mejorar la toma de decisiones. Esta herramienta brinda la posibilidad de visualizar en cualquier momento reportes y gráficos de manera online con datos actualizados de los movimientos que hay en la empresa, también cuenta el trazo y mapeo de los movimientos de nuestros trabajadores que realizan todos los días y de esta manera facilitar el análisis de cada proceso para poder optimizarlo.

- Native cross-platform mobile application development using void

Autor: Mathieu Bruning

Institución: Delft University of Technology, Netherlands

Año: 2011

Esta tesis es el resultado de la investigación que se centra en cómo hacer frente a la heterogeneidad de los dispositivos móviles, mientras que el desarrollo de aplicaciones móviles. Dado que las soluciones actuales de desarrollo multiplataforma de aplicaciones móviles de la industria no ofrecen la flexibilidad, ya que tienden a tener un carácter estático en términos de plataformas y funciones compatibles, un nuevo prototipo marco de desarrollo de plataforma cruzada se presenta en esta tesis. Utilizando el framework de desarrollo llamado Voind, las aplicaciones móviles se pueden definir en un modelo abstracto con el fin de utilizar este modelo para la generación de múltiples versiones para diferentes plataformas móviles.

- Cross-platform mobile application development: A pattern-based approach

Autores: Christian G. Acord, Corey C. Murphy

Institución: Naval Postgraduate School - Monterey, California

Año: 2012

En esta tesis se compara y contrasta dos plataformas, iOS y Android smartphones, y analiza cómo se podría aplicar el patrón Modelo, Vista, Controlador de patrón con el fin de minimizar las diferencias inherentes entre las plataformas. Por otra parte, esta tesis

describe el proceso de diseño unificado que se puede utilizar para implementar iOS y Android aplicaciones nativas de un solo proceso de diseño. Este proceso de diseño reduce la cantidad de tiempo requerido para el desarrollo de aplicaciones y mantiene estilos de interfaz de usuario específicos de la plataforma para las diferentes plataformas.

Los autores utilizan este proceso para diseñar y construir un prototipo funcional de la aplicación NPS Muster en ambas plataformas. Esta aplicación es capaz de mostrar anuncios y permite a los estudiantes de NPS para realizar los pases de revista diarias.

- “Método para optimizar el consumo de energía utilizando conocimiento del contexto asociado a cada aplicación en dispositivos móviles”

Autor: José Carlos Valdivia Bedregal

Institución: Universidad Católica de Santa María

Año: 2013

La siguiente tesis busca dar una solución a cómo ahorrar el consumo de baterías de smartphones, la propuesta que presenta es el aprendizaje del contexto del usuario: las aplicaciones que utiliza, el consumo de cada una de estas aplicaciones por los componentes del dispositivo móvil y el tiempo que estas aplicaciones están siendo utilizadas en primer plano. Con esta información se realizan acciones automatizadas con el fin de optimizar el consumo de energía en el dispositivo móvil.

- “Propuesta de una estrategia de elección de redes basada en un contexto para mejorar el ahorro de energía en dispositivos móviles con sistema operativo Android en la ciudad de Arequipa”

Autor: Brenda Tamayo Oporto

Institución: Universidad Católica de Santa María

Año: 2013

La tesis mencionada, presenta una propuesta de estrategia de elección de la interfaz de red con menor consumo energético para el envío de datos desde un dispositivo móvil Android. La solución muestra la elección adecuada de la red, de acuerdo a las características del contexto de conectividad, pueden llevar a un ahorro de energía en la batería.

2.1.2. Antecedentes de Dispositivos Móviles

Los dispositivos móviles son aquellos aparatos suficientemente pequeños para ser transportados y empleados durante su transporte. Normalmente se sincronizan con un sistema de sobremesa para actualizar aplicaciones y datos. Fernández (2006).

Estos dispositivos contribuyen a la sociedad, para que esta se mantenga comunicada en todo momento y en cualquier lugar. Los dispositivos móviles están diseñados básicamente para ser portátiles y a menudo los podemos movilizar ajustado en la palma de la mano o en el bolsillo. Algunos dispositivos móviles son más potentes y permiten hacer muchas de las mismas cosas que puedes hacer con un ordenador de sobremesa o portátil. Algunos de los ejemplos de estos dispositivos pueden ser: (a) Paginadores, (b) comunicadores de bolsillo, (c) *Smartphone*, (d) *PDA's*, (e) *Tablets*, (f) *E-Book*, etc.

Otra característica importante es la conexión a una red inalámbrica. Esta particularidad es denominada con el término en inglés "*wireless*" (en español, inalámbrico). Un dispositivo inalámbrico es aquel que es capaz de comunicarse o

acceder a una red sin cables. Por ejemplo, un teléfono móvil, una *Tablet*, un *PDA*, etc. Este tipo de dispositivos se comportan como si estuvieran directamente conectados a una red mediante un cable, dando la impresión al usuario que los datos están almacenados en el propio dispositivo.

El lanzamiento de los dispositivos móviles fue un icono revolucionario que rápidamente se convirtió en una tendencia en la sociedad. La evolución de los mismos crece exponencialmente a nivel nacional y mundial. Con el pasar de los años la tecnología ha superado dificultades que los dispositivos presentaban como por ejemplo: (a) El tamaño y peso del dispositivo, (b) El inalcanzable precio, (c) Desarrollo de baterías enormes y de poca duración, (c) Pantallas poco nítidas y con menor variedad de colores y (d) Sistemas operativos complejos, etc. Sin embargo en la actualidad estos retos fueron superados a nivel de hardware y, teniendo como resultado dispositivos móviles con tecnología más desarrollada que facilita la utilidad de estos aparatos electrónicos.

De acuerdo con Fidler (1997), existen varias razones importantes y determinantes para justificar el éxito en la adopción de una nueva tecnología. Además de las motivaciones tradicionales como mantener la conexión y comunicación entre familiares/amigos y reducir el aislamiento en zonas alejadas, podemos decir que también son importantes las motivaciones de movilidad y entretenimiento Williams, Dordick y Jesuale (1985), la sensación de más seguridad personal Dimmick, Sikand y Patterson (1994) y la moda/estatus Souza (2006).

En el año 2005, las empresas DuPont Global Mobility Innovation Team y T38 propusieron los siguientes estándares para la clasificación de dispositivos móviles en función al tamaño de sus pantallas y las funcionalidades de sus equipos: (a)

Computadores PC portátiles y *Netbooks*, (b) Tabletas PC (*Tablet*), (c) *PDA*s, Palm PC y *Pokec PC*, (d) Reproductores multimedia y *pocket-book*, (e) Videoconsola de juegos, y (f) Teléfonos móviles. Estos equipos avanzados incorporan sistemas operativos tales como *Symbian*, *Android*, *iOS* (p.e. *iPhone*), *Windows Phone* y *BlackBerry*. Fombona & Pascual & Madeira (2012).

Tanto los dispositivos como los sistemas operativos que hacen posible su funcionamiento han ido evolucionando según las necesidades de los usuarios finales. En la actualidad existe otra manera de clasificarlos, estos son principalmente dos tipos de dispositivos móviles: los que cuentan con un teclado y aquellos que están basados en una pantalla táctil, para la introducción de datos.

Al sustituir todo el teclado por un lápiz que se utiliza sobre la propia pantalla, el peso y tamaño del dispositivo se pueden reducir, así como el consumo de energía, lo cual contribuye a la mayor duración de las baterías. Se plantea, sin embargo, el problema de la introducción de datos. Existen básicamente dos opciones: utilizar un teclado en pantalla, sobre el cual se puede ir pulsando con el propio lápiz, o bien reconocer la escritura directa del usuario.

Uno de los mayores problemas que plantean los dispositivos móviles es el mecanismo a utilizar para que el usuario pueda introducir información. Si las dimensiones no son suficientes para incluir un teclado útil, las soluciones posibles son varias: reconocimiento de escritura, un teclado en pantalla o incluso un teclado externo.

El teclado en pantalla, en el que se pulsa con la punta de un pequeño lápiz, es una eficaz alternativa al teclado físico representando, un considerable ahorro de espacio, peso y en menor medida, energía.

Así mismo, también se tiene el método que podría considerarse más natural: la escritura directa sobre la pantalla del dispositivo. Para que esto sea posible, no obstante, el dispositivo debe contar con un software de reconocimiento de caracteres suficientemente eficiente ya que, de lo contrario, se perderá más tiempo en efectuar correcciones que en la propia introducción de datos.

Sin embargo, en la actualidad podemos decir que la función principal de cualquier dispositivo móvil es servir como plataforma para aplicaciones móviles que puedan resolver inquietudes o problemas a los usuarios finales.

2.1.3. Antecedentes de Aplicaciones Móviles

Las aplicaciones móviles se han convertido en una herramienta indispensable en nuestra vida cotidiana, tanto es así que la primera pregunta que se hace un desarrollador de aplicaciones móviles es: ¿Qué problema puedo solucionar con esta *App*?, es decir que la función principal de una aplicación móvil es resolver una inquietud o problema que se le presenta a una persona.

A finales de los años 90s se desarrollaron las primeras aplicaciones móviles, eran aplicaciones que cumplían funciones muy elementales, por tal motivo su diseño era básico, por ejemplo teníamos aplicaciones como la agenda, *arcade games*, editores de *ringtone*, etc. Sin embargo su evolución creció rápidamente gracias a la tecnología *WAP* y la transmisión de data (*EDGE*). Cornella (2014).

Uno de los principales problemas en el crecimiento de las aplicaciones móviles fueron las restricciones de los fabricantes de los dispositivos móviles, no permitían desarrolladores externos de sus sistemas operativos. Sin embargo, en el año 2007 se

lanzó el *Iphone* de *Apple*, con la característica revolucionaria de aceptar aplicaciones móviles externas por medio de su tienda *App Store*. (Cornella, 2014).

Hace algunos años el acceso a internet estaba dominado por los ordenadores personales, en la actualidad las visitas a internet están siendo realizadas en mayor cantidad desde un dispositivo móvil, por tal motivo se proyecta con mayor expectativa el crecimiento del mercado de aplicaciones móviles.

2.2. Situación Actual

2.2.1. Mercado de Dispositivos Móviles

Según el informe realizado en junio del 2014 por *Comscore* “En el Perú y América latina existe una gran oportunidad de crecimiento en acceso a internet desde dispositivos móviles” en nuestro país, el mayor porcentaje de visitas a internet realizadas desde diferentes dispositivos móviles son realizadas desde los *Smartphone* y las tabletas, Ver Figura 1.

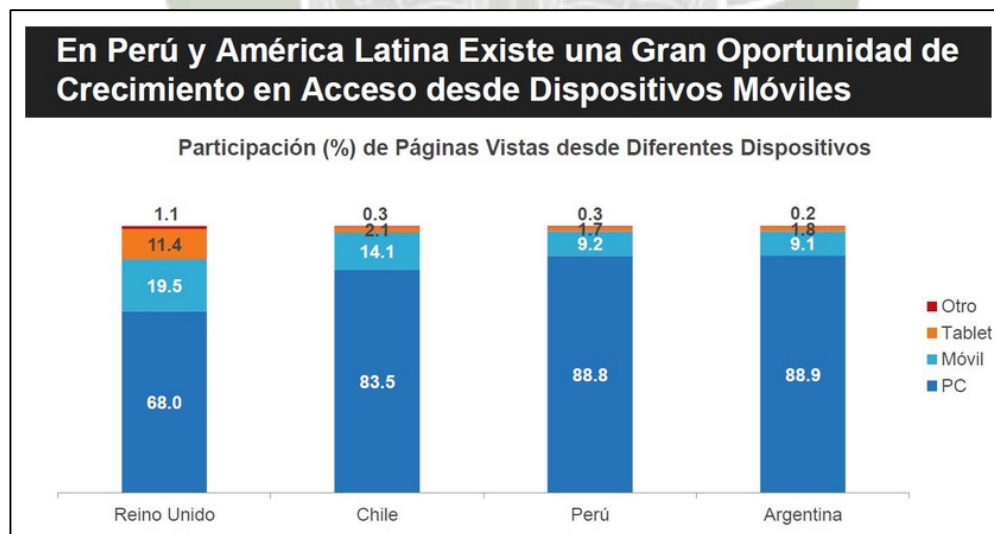


Figura 1: Participación de páginas visitadas desde diferentes dispositivos móviles.
Fuente: Comscore, Device Essentials, Abril 2014.

Un teléfono inteligente (*smartphone*) es un término comercial para denominar a un teléfono móvil que ofrece más funciones que un teléfono móvil común. Casi todos los teléfonos móviles inteligentes se comportan como un organizador personal de un cliente, por ejemplo tienen incorporado una agenda o su correo electrónico.

Una de las características más importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero.

El mercado actual de *smartphones* está dominado por *Android* y por *iOS*. Los pronósticos de la consultora *IHS iSupply* afirman que durante el año 2013 ambas plataformas concentraron el 76% del mercado. El resto es compartido por sistemas operativos como *Symbian*, *Windows Phone*, *BlackBerry*, *Bada*, *Firefox* y otras plataformas menores. SEGITUR (2014). A principios del 2012, los países con más penetración de *smartphones* fueron China, USA, UK, España, Japón, Francia y Alemania, alcanzando un total de 470 millones de dispositivos, independientemente del sistema operativo. Ver Figura 2.

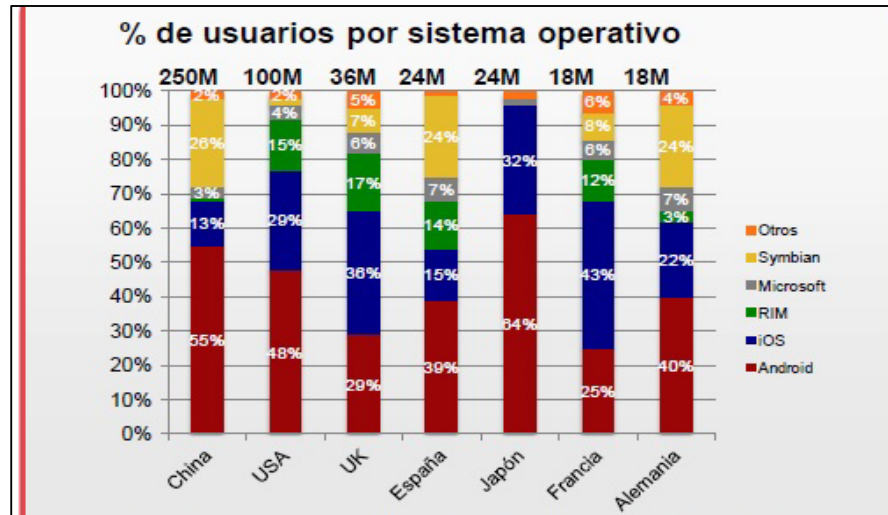


Figura 2: Porcentaje de usuarios por sistema operativo.
Fuente: Comscore, Go-Móvil.es, Newmediatrendwatch.com, Análisis Interno TADG.

Al igual que las computadoras portátiles, las tabletas se han diseñado para ser portátil. Sin embargo, proporcionan una experiencia informática muy diferente. La diferencia más obvia que tienen con las computadoras personales es que las tabletas no tienen teclados, en cambio sí tienen pantallas táctiles, toda la pantalla es sensible al tacto, que le permite escribir en un teclado virtual y utilizar su dedo como puntero del ratón.

Los dispositivos *Tablet* están diseñados principalmente para los medios de comunicación que consumen, y están optimizados para tareas como la navegación *web*, multimedia, leer libros electrónicos y los juegos. Para muchas personas, un equipo "regular" como un escritorio o portátil sigue siendo necesario para poder utilizar algunos programas. Sin embargo, la comodidad de una computadora tableta significa que puede ser ideal como un segundo ordenador.

Según el estudio de mercado de *apps* turísticas realizado por SEGITUR, en el año 2012, se alcanzaron los 118 millones de *tablets* vendidos a nivel mundial. *Apple* cede cada año su cuota de mercado, respecto al 2011 ha perdido un 8%. A finales de 2012, su cuota se

sitúa en un 43%. Samsung es su competidor más directo con el 15% del mercado, seguido de los *tablets* de Amazon 12%, Asus 6% y Barnes & Noble 2%. Ver Figura 3

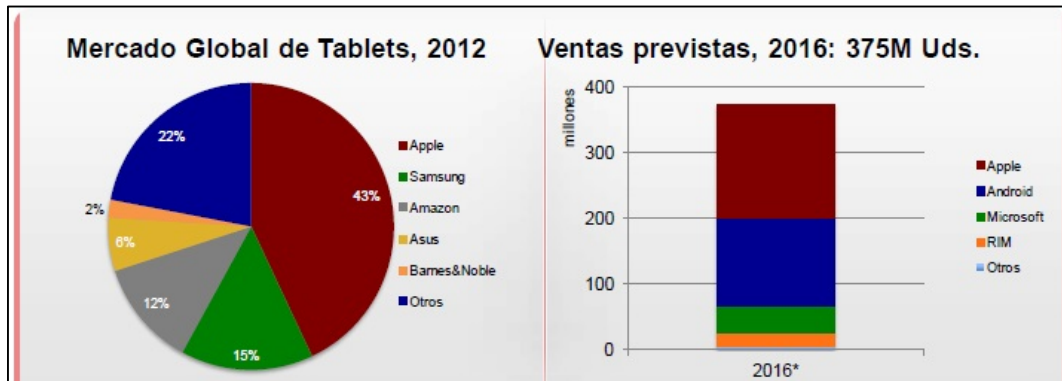


Figura 3: Participación por marcas en el mercado global de tabletas, 2012-2016.

Fuente: Forbes.com, Telegraph.co.uk, Análisis interno TADG, Pmewswire.com, Flurry.

2.2.2. Mercado de Aplicaciones Móviles

Las aplicaciones móviles permiten crear experiencias sensoriales y emocionales que van más allá de la simple información. Además son capaces de capturar la atención del usuario, invitándolo a interactuar constantemente y a mantenerse en contacto con la marca por medio de su *App Store*. El éxito de una aplicación depende del análisis y creatividad que surge a partir de la necesidad del cliente.

La tendencia en el uso de los “móviles inteligentes” o *Smartphone*, ha provocado la apertura de un nuevo nicho de negocio, hablamos del mercado de aplicaciones móviles. En el Perú, la industria registra un constante crecimiento y las cifras apuntan a que seguirá ese camino: Según la Oficina Comercial de Washington, el mercado de aplicaciones móviles (*apps*) en el Perú crece 80% al año, con lo cual se estima que se alcanzarían a desarrollar un total de 1,500 *apps* durante el 2015. (Canal TI)

Estos números alentadores incentivan la apertura de nuevas empresas desarrolladoras de aplicativos móviles en el Perú, según (Canal TI), En la actualidad

muchas corporaciones en el Perú han identificado una gran oportunidad para el desarrollo de su modelo de negocio en las *apps*, especialmente el sector textil, financiero, de salud y, por supuesto, las *startups*. No sólo para poder tener un mayor acercamiento con sus empleados y potenciales clientes, quienes tienen fácil acceso o ya disponen de un *smartphone*, sino también, para optimizar sus procesos empresariales y tener un mayor control de todo lo que sucede dentro de la organización.

En el ecosistema móvil, las aplicaciones representan actualmente la segunda mayor fuente de ingresos por detrás del consumo de datos, tanto por pago de contenidos como por publicidad. En el 2011 se descargaron más de 23.000 millones de aplicaciones en todo el mundo. (SEGITUR, EM).

El *App Store* de *Apple* ha superado los 40.000 millones de descargas en enero del 2013. Sólo en el 2012 se han producido casi 20.000 millones de descargas (tasa de crecimiento de 2.000 millones de descargas al mes), el 50% del total de descargas en toda su historia. (SEGITUR, EM).

Mientras tanto el mercado *Android* ha superado los 10.000 millones de descargas en febrero del 2012, con una tasa de crecimiento de 1.000 millones de descargas al mes desde julio del 2011. *Android* espera alcanzar la cifra alcanzada por el *App Store* de *Apple* en 2016. (SEGITUR, EM).

Según Laudon (2012), las principales áreas de crecimiento en el comercio electrónico móvil son los servicios basados en la ubicación, con cerca de \$ 215 millones de ingresos en 2010, las ventas de aplicaciones de software en tiendas como *iTunes* (cerca de \$1.8 mil millones); las descargas de entretenimiento compuestas por tonos de llamadas, música, video y programas TV (cerca de \$1 mil millones); los anuncios

móviles (\$ 784 millones); los servicios de compra directa como Slifter (\$200 millones), y las ventas de libros electrónicos (\$ 338 millones).

El sistema operativo *Android* es el más utilizado entre los usuarios peruanos seguidamente los sistemas operativos *iOS* y *Windows Phone*. comScore (2014). Ver Figura 4.

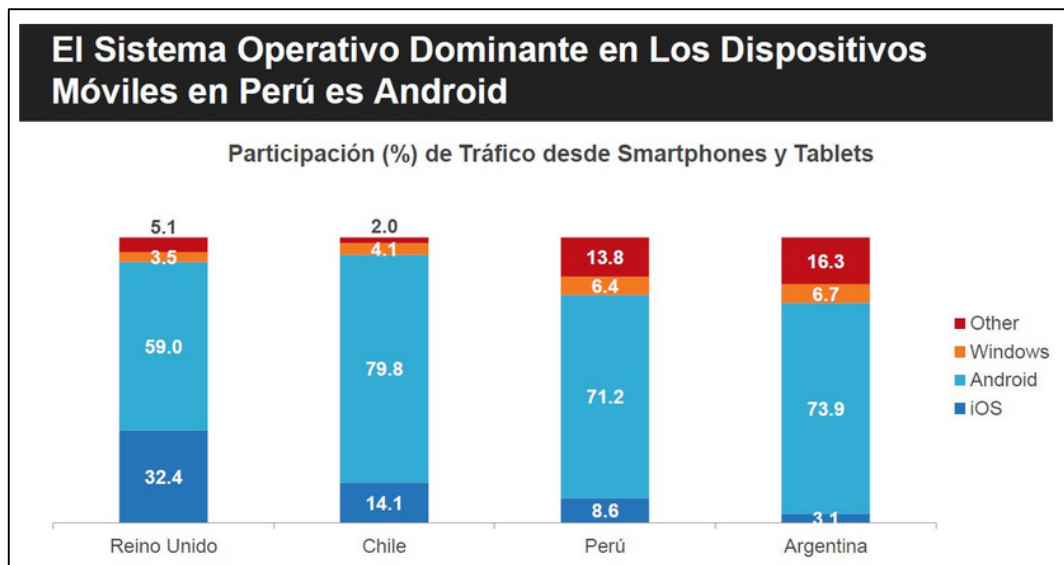


Figura 4: Participación de marcas de dispositivos móviles en el mercado peruano.
Fuente: Comscore, Device Essentials, Abril 2014

La aparición de los *smartphones* ha generado un cambio en la industria de la telefonía móvil. Actualmente la decisión de comprar un celular pasa por el sistema operativo o plataforma móvil que tenga este dispositivo, las aplicaciones potencian y dan un valor agregado a estos terminales móviles.

2.2.3. Turismo

El turismo es considerado como una de las principales actividades de desarrollo para diferentes países que cuentan con potencialidades turísticas y que constituyen la motivación para que muchos visitantes elijan dentro de la diversidad de destinos, aquellos que

representan una especial motivación e interés ante la necesidad de conocer y aprovechar el tiempo disponible, en aquellos que se denomina actualmente la cultura del ocio.

Climent (2003) señaló que el turismo es la actividad o hecho de viajar por placer, el conjunto de los medios conducentes y finalmente el conjunto de personas que realiza este tipo de viajes. Para la OMT (2005), el turismo es un fenómeno social, cultural y económico relacionado con el movimiento de las personas a lugares que se encuentran fuera de su lugar de residencia habitual por motivos personales y/o profesionales.

Valls (2007) definió el destino turístico como el espacio geográfico determinado, con rasgos propios de clima, raíces culturales-étnicas, infraestructuras y servicios y con cierta capacidad administrativa para desarrollar instrumentos comunes de planificación. Este espacio atrae a turistas con productos perfectamente.

Según Mochón (2004) las formas básicas de clasificar la práctica del turismo, según la procedencia del viajero son: (1) turismo interior, el de los residentes del país que viajan dentro del interior del país, (2) turismo receptor, el de los no residentes que viajan dentro del país en cuestión, (3) turismo emisor, el de los residentes del país cuando viajan a otro país, (4) turismo interior, que incluye el turismo interno y el turismo receptor, (5) turismo nacional, que incluye el turismo interno y el turismo emisor, (6) turismo internacional, que se compone de turismo receptor y turismo emisor.

Según los autores Bigné, E., Font, X. & Andreu, L. (2000). Clasifican los destinos turísticos de la siguiente manera:

Tabla 1

Tipos de destinos turísticos

Tipo de destino	Clientes	Ejemplos de atractivos y actividades
Urbano	Negocios-RICF	Reuniones, incentivos, congresos, ferias Educación, religión, salud
	Ocio	Visitar lugares de interés, ir de compras, eventos culturales, pasar el fin de semana
De costa	Negocios-RICF	Reuniones, incentivos, congresos, ferias
	Ocio	Sol, playa, sexo, deportes
De montaña	Negocios-RICF	Reuniones, incentivos, congresos, ferias
	Ocio	Esquí, deportes de montaña, salud
Rural	Negocios-RICF	Reuniones, incentivos, congresos, ferias
	Ocio	Relajación, agricultura, deporte, aprender actividades de campo
Sin explorar	Negocios-RICF	Explorar nuevas oportunidades de negocio, incentivos
	Ocio	Aventura, autenticidad, misiones y otros especializados
Único, exótica, exclusivo	Negocios-RICF	Reuniones, incentivos
	Ocio	Ocasión especial, luna de miel, aniversario

Nota: Bigné, E., Font, X. & Andreu, L. (2000). *Marketing de destinos turísticos: Análisis y estrategias de desarrollo*. ISBN 84-7356 -262 -3. ESIC Editorial. Madrid, España.

La organización mundial de turismo (OMT) divide al tipo de turista en dos tipos: (a) turista internacional y (b) turista interno. Dentro de estos dos tipos presenta cuatro clases de turistas de acuerdo con las características y el tiempo de permanencia.

Tabla 2

Tipos de turista

	Turismo internacional	Turismo interno
Residente	Permanece al menos 12 meses consecutivos	Permanece al menos 6 meses consecutivos
Visitante	Persona que permanece menos de un año	Persona que permanece menos de un año
Turista	Persona que pernocta por lo menos una noche	Persona que pernocta por lo menos una noche
Excursionista	Persona que permanece menos de 24 horas sin pernoctar en el destino turístico	Persona que permanece menos de 24 horas sin pernoctar en el destino turístico

Nota: Tipo de turistas. Jiménez, L & Jiménez, W. (2013). TURISMO: Tendencias globales y planificación estratégica. 1Ed.- Bogotá, Colombia: Ecoe Ediciones.

De acuerdo a la información brindada en los párrafos anteriores podemos deducir que el centro histórico de Arequipa es un destino Urbano debido a sus atractivos turísticos como la catedral, templos o lugares culturales. De igual manera podemos decir que los destinos urbanos atraen visitas educativas, así como al turista vacacional que aprovecha los fines de semana o las vacaciones para poder visitar nuevas ciudades.

2.2.4. Turismo en el Centro Histórico de Arequipa

El centro de la ciudad de Arequipa presenta diversos atractivos turísticos y es considerada como una zona histórica y monumental de la nación, su delimitación comprende espacios y lugares arquitectónicos como la plaza de armas, la catedral, las iglesias, los monasterios, el barrio de San Lázaro, las casonas o solares, el trazo de las calles, los puentes, etc.

En la Tabla 3 podemos apreciar que las actividades que más realiza el turista nacional son el turismo de cultura, donde visita Iglesias, catedrales, conventos y museos. De igual manera en la Tabla 4 se puede observar que el turista extranjero realiza el mismo tipo de visitas en la ciudad de Arequipa.

Tabla 3

Actividades realizadas por el turista nacional

ACTIVIDADES REALIZADAS	TOTAL %
TURISMO DE CULTURA	49
Visitar iglesias, catedrales, conventos	35
Visitar museos	22
Visitar sitios arqueológicos	18
Visitar inmuebles históricos y monumentos	9
Participar en festividades locales	8
City tour guiado	8
Participar en festividades religiosas	4
Visitar comunidades nativas / andinas / campesinas	3
Visitar chamanes / curanderos	1
TURISMO DE NATURALEZA	38
Visitar áreas / reservas naturales	25
Observación de flora o fauna en su ambiente natural	22
Pasear por lagos, lagunas y ríos	14
Observación de aves	4
TURISMO DE SOL Y PLAYA	49
Ir a la playa a pasar el día	49
Acampar en la playa	8
COMPRAS	35
Dulces típicos	23
Productos comestibles no dulces	12
Artículos de vestir	7
Artesanía	7
DIVERSIÓN Y ENTRETENIMIENTO	30
Ir a discotecas / pubs	20
Ir a peñas y shows folklóricos	13
Asistir a conciertos y festivales no folklóricos	3
Ir a casinos / bingos y tragamonedas	2
Ir a espectáculos deportivos	1
Ir a cines y teatros	1
TURISMO DE AVENTURA	5
Andinismo	3
Trekking	2
Ciclismo de montaña	1
OTROS	10
Visitar bodegas de vinos / piscos / otros	4
Visitar aguas termales	3
Visitar criaderos	2
Ir a zoológicos	1
Ir a parques de diversión	1

Nota: Actividades realizadas, Perfil del vacacionista nacional 2013, Vacacionistas que visitaron el departamento de Arequipa. PROMPERÚ 2013.

Tabla 4

Actividades realizadas por el turista extranjero

Actividades realizadas en Arequipa	%
TURISMO CULTURAL	88%
Visitar iglesias / catedrales / conventos	61%
Pasear, caminar por la ciudad	56%
Visitar parques, plazuela de la ciudad	53%
Visitar museos	33%
Visitar inmuebles históricos	32%
City tour guiado	24%
Visitar sitios arqueológicos	18%
Visitar comunidades nativas / andinas / campesinas	3%
Visitar rutas gastronómicas	3%
Participar en festividades locales	3%
TURISMO DE NATURALEZA	48%
Visitar áreas / reservas naturales	43%
Observación de aves	22%
Observación de flora	13%
Observación de mamíferos	5%
Observación de insectos	1%
TURISMO DE AVENTURA	13%
Trekking / Senderismo / Hiking	10%
Canotaje	1%
SOL Y PLAYA	4%

Nota: Actividades realizadas, Perfil del turista extranjero 2012, Turistas extranjeros que visitaron el departamento de Arequipa. PROMPERÚ 2012.

También es importante mencionar el incremento en los últimos años de visitas a estos lugares culturales en la ciudad de Arequipa, por ejemplo en la tabla 5 se puede ver que en el año 2005 la cantidad total de turistas que visitan el monasterio de santa catalina fue de 125047 turistas, mientras que en el año 2012 la cantidad de turistas fue de 173781 entonces se puede decir que hubo un incremento del 38.9 % de visitantes al convento arequipeño.

Tabla 5

Llegada de visitantes al monasterio de Santa Catalina. Enero 2005 – Diciembre 2012.

AREQUIPA: LLEGADA DE VISITANTES AL MONASTERIO DE SANTA CATALINA, ENERO 2005 - DICIEMBRE 2012

Mes	2005			2006			2007			2008			2009			2010			2011			2012		
	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT	Total	NAC	EXT
Enero	5,909	1,389	4,520	7,431	2,061	5,370	8,286	2,287	6,009	9,103	2,612	6,491	9,884	3,103	6,781	10,614	3,287	7,327	9,824	3,466	6,358	10,818	3,236	7,582
Febrero	5,937	1,363	4,574	6,198	1,647	4,551	8,152	2,526	5,626	8,452	2,082	6,370	8,291	2,410	5,881	8,574	2,717	5,857	7,834	2,789	5,045	8,737	2,694	6,043
Marzo	7,422	1,413	6,009	6,365	1,213	5,152	8,667	1,726	6,941	9,756	1,984	7,772	8,791	1,647	7,144	7,749	1,623	6,126	8,923	2,003	6,920	8,657	1,858	6,799
Abril	7,403	791	6,612	8,335	1,371	6,964	10,928	2,207	8,721	10,951	1,486	9,465	12,058	2,542	9,516	10,610	2,845	7,765	12,208	2,610	9,598	13,681	2,940	10,741
Mayo	9,786	1,667	8,099	8,962	1,766	7,196	11,802	2,340	9,462	12,898	2,431	10,467	12,299	2,256	10,043	12,442	2,756	9,686	12,749	2,853	9,896	15,099	3,098	12,001
Junio	8,317	1,162	7,155	7,706	1,652	6,054	10,814	2,152	8,662	11,529	2,647	8,882	11,164	2,363	8,801	10,492	2,300	8,192	11,574	2,355	9,219	11,667	3,241	8,426
Julio	13,662	2,513	11,349	14,239	3,912	10,327	17,385	2,600	14,785	22,290	5,435	16,855	19,175	4,296	14,879	19,734	5,140	14,594	21,532	5,618	15,914	20,395	4,684	15,711
Agosto	23,077	8,542	14,535	19,214	5,119	14,095	21,189	4,232	16,957	22,721	4,698	18,023	18,750	3,255	15,495	21,286	5,683	15,403	23,647	5,621	17,826	23,066	7,797	15,269
Setiembre	11,260	2,052	9,208	12,310	2,506	9,804	14,283	2,618	11,665	17,177	3,386	13,791	14,073	2,767	11,306	15,732	3,903	11,829	16,851	3,535	13,316	17,718	4,610	13,108
Octubre	15,629	5,432	10,197	15,630	5,160	10,470	18,436	5,053	13,383	19,988	5,488	14,500	16,220	3,789	12,431	17,975	5,690	12,005	19,708	5,504	14,204	20,410	6,443	13,967
Noviembre	10,354	2,844	7,510	11,032	3,094	7,938	15,140	4,271	10,869	15,431	4,687	10,744	13,413	3,661	9,752	13,627	4,595	9,032	14,746	4,283	10,463	14,575	4,591	9,984
Diciembre	6,391	1,870	4,521	6,299	2,372	3,927	7,819	2,651	5,168	7,702	2,173	5,529	7,688	2,438	5,260	7,804	2,695	5,109	7,718	2,294	5,424	8,758	3,064	5,674
Total	125,047	31,058	93,989	123,721	32,073	91,648	152,911	34,663	118,248	168,008	39,131	128,877	151,816	34,727	117,089	156,659	43,534	113,105	167,114	43,131	123,983	173,791	48,476	125,305

Nota: Monasterio de Santa Catalina – Promociones Turísticas del Sur. Elaboración: MINCETUR/SG/OGEE- Oficina de Estudios Turísticos y Artesanales. Con información disponible a enero de 2013.

2.2.5. Tecnología de Información en el Turismo

Las tecnologías de información han revolucionado el panorama de los negocios en el mundo. Las tecnologías han modificado las industrias hoteleras, de restaurantes, de servicios de viajes, del sector de intermediarios como touroperadores y agencias de viajes y ahora juegan un papel fundamental en las reglas que rigen el mundo de negocios y en la forma de acercarse a los clientes. Las ventajas de las nuevas tecnologías en cuanto a incremento de la competitividad, reducción de errores y creación de nuevas funcionalidades son incuestionables en cualquier sector, incluyendo el turístico.

Según (Cornella, 2014), dos hechos permiten pensar en una estrecha interrelación entre la industria turística y la informática, por un lado el hecho del que el turismo al ser la primera industria mundial se convierte, potencialmente en el principal cliente de la industria informática, y por otro lado el hecho de que la información está en los mismos cimientos de la industria turística. Además la industria turística era y es atractiva para el desarrollo de tecnologías de información: el turismo es una actividad interterritorial que promociona y comercializa actividades ofrecidas lejos del lugar donde se encuentra el cliente y por otro lado, al formar parte de una industria que involucra ocio y entretenimiento, necesita medios de promoción basados en medios audiovisuales que resulten atractivos.

Según Cornella (2014) la relación entre la tecnología de información y el turismo empieza en los años 50, con el desarrollo de sistemas computarizados para administrar la información que este nuevo medio de transporte genera, son los “*Computer Reservation Systems*” (CRSs), que permiten la gestión automática de las reservas.

A partir de los años setenta y debido al hecho de que los agentes de viajes necesitan acceder a diferentes *CRS* pertenecientes a muchas compañías, se plantea la necesidad de articular los diferentes sistemas de reserva, desarrollándose los “*Global Distribution Systems*” (*GDSs*) o Sistemas de Distribución Global. Estos sistemas engloban diferentes compañías como por ejemplo: AMADEUS con Lufthansa, Air France, Iberia y Continental Airlines; GALILEO opera con United Airlines, British Airways, Swiss Air, KLM Royal Dutch Airlines, US Air, Alitalia, Olympic, Air Canada, TAP Air Portugal, Austrian Airlines, Aer Lingun; WORLDSPAN al que están afiliadas Delta Airlines, Northwest Airlines, TransWorld Airlines y Abacos.

A partir de los años 90 cabe destacar a Internet como elemento revolucionario en el mundo del turismo. Este desarrollo tecnológico, así como las comunicaciones móviles, banda ancha, satélites, etc. – están produciendo cambios significativos en la estructura económica y social, y en el conjunto de las relaciones sociales.

El sector del turismo está aprendiendo rápidamente que Internet puede satisfacer mucho mejor que cualquier otra tecnología actual la necesidad de los usuarios de recibir información de calidad que sea confiable. Las empresas turísticas tienen las oportunidades de crear propios sitios *web* así como aplicativos móviles que son de gran importancia para la comunicación de marketing en Internet ya que disponen del potencial de proveer de altos niveles de información, crear imagen de marca, conseguir una respuesta directa y además de crear experiencias virtuales del producto. Las empresas que se encuentran más favorecidas son los *PYMEs* porque ellos pueden ocupar nichos del mercado para alcanzar un mayor porcentaje de su grupo destinatario y de expandir su radio de alcance. Cornella (2014).

El sector turístico está viviendo un crecimiento espectacular en los últimos años y ya supone más del 10% del Producto Interior Bruto Mundial. Su desarrollo ha supuesto también que se produzcan variaciones notables en el mercado laboral. En la actualidad, el Turismo genera más de 200 millones de empleos. Esa evolución va acompañando con el creciente desarrollo y significación de las tecnologías informáticas en este sector.

2.2.6. Aplicaciones Móviles para el Turismo

El uso de dispositivos móviles se extiende y aumenta a gran velocidad. La sociedad utiliza estos dispositivos como herramientas de trabajo, ocio y comunicación, por tal motivo se ha incrementado la utilización y descarga de aplicativos móviles que facilitan la vida del usuario. En el caso del turismo es importante la incorporación de aplicativos móviles porque suma a la mejora de sus proceso y además es de gran ayuda para la promoción de sus productos y servicios. Además, las aplicaciones para dispositivos móviles, tanto *smartphones* como *tablets*, tienen un papel destacado, ya que cada vez dan más servicio al turista en la organización del viaje durante las distintas etapas.

Según *SEGITUR EM* el TOP 20 DE LAS APPS MÁS DESCARGADAS DE TURISMO está conformado con mayor numero por las aplicaciones de recomendaciones, buscadores de hoteles y vuelos. *Google Maps* registra el mayor número de descargas con 227 millones. *Tripadvisor* y *Kayak* son las *apps* dedicadas al viajero más descargadas.

La Sociedad Estatal para la Gestión de la Innovación y las Tecnologías Turísticas, S. A. (*SEGITTUR*) en colaboración con la feria internacional del turismo (*FITUR*) organizaron en España el primer concurso de *apps* turísticas, denominado “The

Apptourism Awards 2014”. El objetivo de este proyecto es mejorar los nuevos mecanismos para impulsar la innovación en los destinos, con el despliegue y desarrollo de las *TICS*, de forma que se puedan crear servicios diferenciales y altamente competitivos. SEGITUR (2014)

El concurso se dividió en dos categorías: nacional (España) e internacional. La categoría nacional se subdividía a su vez en: “Sol y playa”, “Turismo activo”, “Turismo cultural” y “Servicios turísticos”; mientras que la categoría internacional se subdividía en “Guía por destino” y “Mejora de la experiencia del turista en el viaje”.

Postularon 125 *apps*, de las cuales 84 se registraron en la categoría nacional y 41 en la internacional. En la categoría nacional resultaron ganadores las siguientes aplicaciones: (a) Be Your Guide, una completa guía de Toledo, fue la ganadora en la categoría de Turismo cultural (ver Figura 5); (b) NaturApps dedicada a rutas se alzó con el premio de Turismo Activo; (c) AsturPlaya ha logrado el premio en Sol y Playa y (d) eltenedor.es recibió el galardón en la categoría de Servicios Turísticos. Mientras que en la categoría internacional, (e) Visitnorway fue la ganadora como guía de destino; y (f) skyscanner fue reconocida como Mejora de la experiencia del turista. EUROPAPRESS (2014).



Figura 5: Aplicación turística Be your Guide- Toledo

Fuente: Aplicación turística Be your Guide- Toledo recuperado de <http://www.europapress.es/turismo/nacional/noticia-fitur-seis-aplicaciones-turisticas-premiadas-concurso-the-apptourism-awards-2014-20140124094501.html>

A nivel nacional nos encontramos con una gran escasez de aplicativos móviles dirigidos al turismo o a la promoción de destinos turísticos, sin embargo es importante mencionar algunos de los aplicativos móviles que se encuentran en el Perú como por ejemplo: (a) Perú Natural, esta herramienta de información contiene fotos, videos, mapas y datos claves en español e inglés, para que todos los usuarios puedan acceder y planificar un viaje singular a sitios únicos como las Áreas Naturales Protegidas. (Ver Figura 6); (b) Audioguías, esta aplicación contiene información visual y auditiva del centro histórico de lima. Ver Figura 7; (c) Servicios turísticos, es una herramienta que ayuda a encontrar servicios turísticos como hoteles, restaurantes, etc., alrededor de donde te encuentres, todos ellos geo-referenciados; y (d) Pisco Perú, te proporciona información para preparar el famoso Pisco Sour por medio de recetas y videos.



Figura 6: Aplicación turística: Perú Natural.

Fuente: Aplicación turística: Perú Natural recuperado de <http://www.sernanp.gob.pe/sernanp/noticia.jsp?ID=1929>

AUDIUGUÍAS | CENTRO HISTÓRICO DE LIMA

CÓMO UTILIZAR MICROSOFT TAG | ENGLISH VERSION | compartir

RUTA 01 Plaza Mayor, palacios e iglesias

Desde la Plaza Mayor, centro de la ciudad, recorramos los alrededores para admirar sus principales atractivos religiosos y culturales, además de las principales edificaciones gubernamentales.

RUTA 01 | RUTA 02 | RUTA 03 | RUTA 04 | RUTA 05 | RUTA 06 | RUTA 07 | MAPA GENERAL | DESCARGA LAS RUTAS

Mira aquí algunos puntos de la ruta

Ver en Google Maps

perú Perú

Figura 7: Aplicación turística del centro histórico de Lima

Fuente: Aplicación turística del centro histórico de Lima, recuperado de <http://audioguias.peru.travel/esp/lima-01.html>.

Además hay empresas privadas como Sony y Movistar, que se encuentran desarrollando campañas para el desarrollo de aplicaciones móviles para el turismo, la campaña XPERIA PERÚ es una aplicativo que permite comentar y compartir experiencias de viajes de los turistas. También existen aplicaciones móviles relacionadas con la realidad aumentada.

Es importante mencionar que a comparación de España nos encontramos muy por debajo de la cantidad de aplicaciones móviles desarrolladas en el sector Turismo. Asimismo, la mayoría de aplicaciones móviles para el turismo a nivel mundial se enfocan en un solo sistema operativo.

2.3. Sistemas Operativos Móviles

El sistema operativo (S.O.) es el puente de conexión que existe entre el hardware (dispositivo móvil) y el usuario. Actualmente existen diferentes sistemas operativos en los dispositivos móviles, entre los cuales podemos encontrar principalmente a los siguientes S.O.: (a) *Android*, (b) *IOs* y (c) *Windows 8.1* y *Windows Phone 8*.

2.3.1. Sistema Operativo *Android*

Android es un sistema operativo móvil basado en Linux y Java. Se enfoca en dar facilidades al programador proporcionando un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones móviles. Asimismo, busca que los programadores desarrollen sus aplicativos una sola vez para que así sean compatibles con varios terminales. Otra característica de este S.O. es que proporciona una plataforma de desarrollo gratuita y flexible en el desarrollo de aplicaciones móviles.

Según Gonzales (2012) el diseño de la arquitectura del sistema operativo *Android* cuenta con las siguientes características:

- Busca el desarrollo rápido de aplicaciones, que sean reutilizables y verdaderamente portables entre diferentes dispositivos.
- Los componentes básicos de las aplicaciones se pueden sustituir fácilmente por otros.
- Cuenta con su propia máquina virtual, Dalvik, que interpreta y ejecuta código escrito en Java.
- Permite la representación de gráficos 2D y 3D.
- Posibilita el uso de bases de datos.
- Soporta un elevado número de formatos multimedia.
- Servicio de localización *GSM*.
- Controla los diferentes elementos hardware: *Bluetooth*, *Wi-Fi*, cámara fotográfica o de vídeo, *GPS*, acelerómetro, infrarrojos, etc., siempre y cuando el dispositivo móvil lo contemple.
- Cuenta con un entorno de desarrollo muy cuidado mediante un *SDK* disponible de forma gratuita.
- Ofrece un plug-in para uno de los entornos de desarrollo más populares, Eclipse, y un emulador integrado para ejecutar las aplicaciones.

En la primera capa se encuentra el núcleo de *Android* (Linux 2.6). *Android* utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los

dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes.

La elección de Linux 2.6 se debe principalmente a su naturaleza de código abierto y libre. Esto se ajusta al tipo de distribución que se buscaba para *Android*. Otra razón es que este *kernel* de Linux incluye de por sí numerosos drivers, además de contemplar la gestión de memoria, gestión de procesos, módulos de seguridad, comunicación en red y otras muchas responsabilidades propias de un sistemas operativo.

La segunda capa se corresponde con las librerías utilizadas por *Android*. Éstas han sido escritas utilizando C/C++ y proporcionan a *Android* la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de *Android*.

Al mismo nivel que las librerías de *Android* se sitúa el entorno de ejecución. Éste lo constituyen las CoreLibraries, que son librerías con multitud de clases de Java, y la máquina virtual Dalvik.

Los dos últimos niveles de la arquitectura de *Android* están escritos enteramente en Java. El *framework* de aplicaciones representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para *Android*, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de *API* y el mismo *framework*, representado por este nivel.

El último nivel del diseño arquitectónico de *Android* son las aplicaciones. Éste nivel incluye tanto las incluidas por defecto de *Android* como aquellas que el usuario vaya

añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las *API* y librerías de los niveles anteriores.

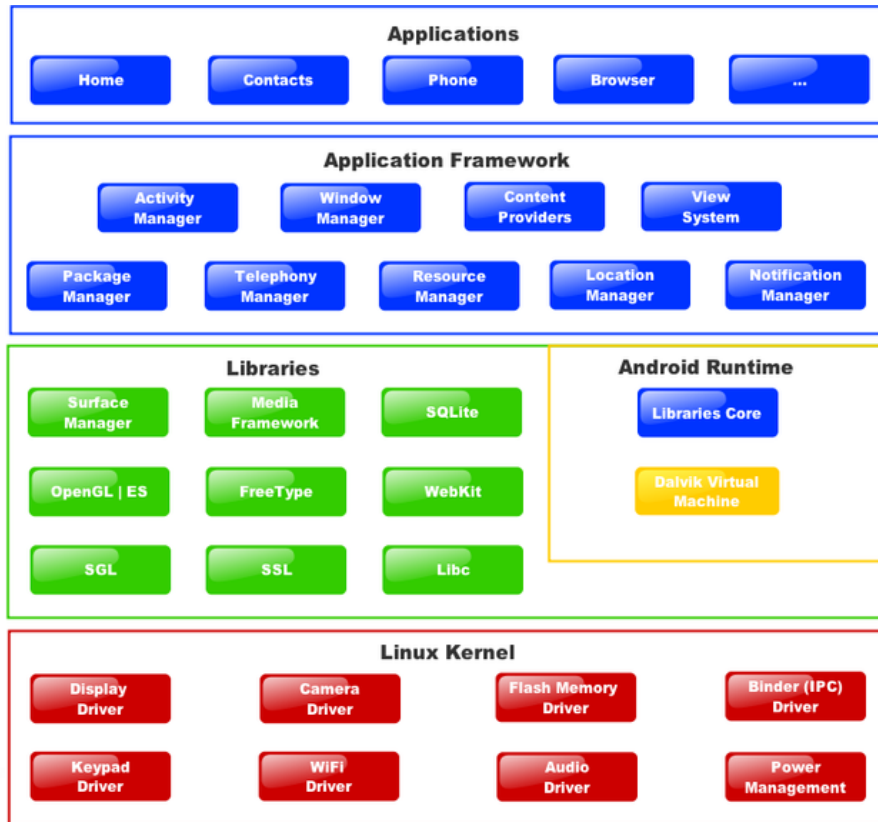


Figura8: Arquitectura del sistema operativo Android.
Fuente: <http://www.xatakandroid.com/sistema-operativo/que-es-android>

La tienda de aplicaciones *Android* conocida como «*Android market*» retribuye a los desarrolladores el 70% del precio de su aplicación. Asimismo, el desarrollo de aplicaciones para *Android* no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «*SDK*» provisto por Google el cual, se puede descargar gratuitamente.



Figura 9: Logo de la tienda virtual de Android.

Fuente: <http://atomcave.com/hd/44665-market-android-wallpaper-images.html>

2.3.2. Sistema Operativo IOS

El IOS es el sistema operativo que utiliza el *iPhone* y el *iPod touch* e *iPad*. Es un derivado de *Mac OS X*, que a su vez está basado en *Darwin BSD*. La arquitectura de IOS tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de Servicios Principales, la capa de Medios de Comunicación y la capa de *Cocoa Touch*.

El sistema operativo móvil *iOS* brinda como medio de desarrollo el kit *iPhone SDK* con el objetivo de incentivar el desarrollo de aplicaciones nativas para *iOS*, no solo de particulares sino también por terceros. El lenguaje de programación principal es el *Objective-C*, un lenguaje orientado a objetos basado en *C*.

Según Martínez (2011) el *SDK* se puede descargar gratis, pero para publicar el software es necesario registrarse en el Programa de Desarrollo del *iPhone*, un paso que requiere el pago y la aprobación por parte de *Apple*. Durante el proceso, se entregan al

desarrollador unas claves firmadas que permiten subir una aplicación a la tienda de aplicaciones de *Apple*.

Las aplicaciones pueden ser distribuidas a través de la *App Store* de *Apple*, por parte de una empresa a sus empleados, o sobre una red “*Ad-hoc*” de hasta 100 *iPhones*. Los desarrolladores de aplicaciones para *iOS* pueden determinar el precio quedándose el 70% del dinero que produzca la aplicación, el 30% restante se lo quedaría la compañía. Los desarrolladores también pueden optar por ofrecer sus aplicaciones gratis y no pagar así nada por la distribución del programa más allá de la cuota de socio. Martínez (2011).



Figura 10: Arquitectura del sistema operativo iOS.

Fuente: <http://www.maestrosdelweb.com/editorial/guia-desarrollo-iphone-ipad/>

Las principales características que tiene el iOS son:

- Notificaciones:
 - Desde la parte superior de la pantalla, deslice el dedo hacia abajo para ver el centro de notificaciones.

- Las notificaciones nuevas aparecen brevemente en la parte superior de la pantalla.
- Puede ver las notificaciones desde la pantalla bloqueada.
- Quiosco:

Las suscripciones a periódicos y revistas se ordenan automáticamente en la pantalla de inicio.
- Recordatorios para gestionar listas de tareas:

Sincronización con iCloud, iCal y Outlook.
- Previsión del tiempo hora a hora.
- Valores bursátiles en tiempo real.
- Sincronización inalámbrica con *iTunes*.
- Mejoras de accesibilidad:
 - En el *iPhone*, cree secuencias de vibración personalizadas para las llamadas entrantes.
 - Nueva interfaz para usar IOS junto con dispositivos pensados para ayudar a personas con movilidad reducida.
- Mejoras en Exchange ActiveSync:
 - Sincronice las tareas remotamente.
 - Marque los mensajes como leídos/no leídos o con un indicador.

- Mejor rendimiento cuando no está conectado.
- Activación y configuración desde el mismo dispositivo con el asistente de configuración.
- Actualizaciones de software disponibles de forma remota sin necesidad de conectar el dispositivo al ordenador.

2.3.3. Sistema Operativo Windows 8.1 y *Windows Phone 8*

Es la versión actual del sistema operativo de Microsoft Windows, producido por Microsoft para su uso en ordenadores personales. Su interfaz de usuario ha sido modificada para hacerla más adecuada para su uso con pantallas táctiles, además de los tradicionales ratón y teclado. Este cambio puede provocar que algunos usuarios se sientan perdidos a la hora de utilizar y moverse por el nuevo sistema operativo de Windows.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las *API* de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Singh (2012).

Además, existe una gran oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de *Windows Marketplace for Mobile*.

Originalmente apareció bajo el nombre de Pocket PC, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas. En la actualidad, la mayoría de los teléfonos con Windows Mobile vienen con un estilete digital, que se utiliza para introducir comandos pulsando en la pantalla. Martínez (2011)

Como todo sistema operativo la arquitectura está diseñada por 2 áreas fundamentales, el *kernel* y nonkernel (traducción a gusto de lector); El *kernel* posee obviamente el núcleo, 2 sistemas de archivos (IMGFS y TexFAT), render gráfico y actualización de sistema, mientras la parte nonkernel contiene las aplicaciones, la shell y espacio de usuario.

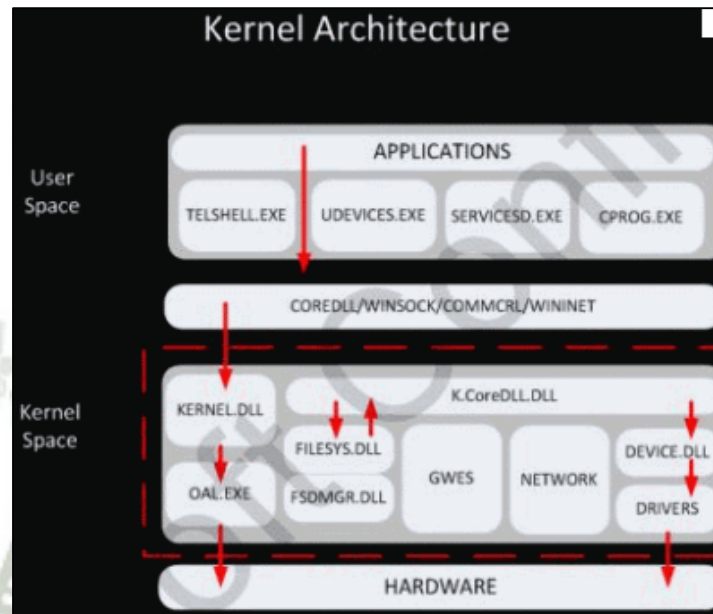


Figura 11: Arquitectura del sistema operativo Windows Phone

Fuente: <http://www.chw.net/2010/04/filtran-arquitectura-de-windows-phone-7/>

Las características principales de este sistema operativo son:

- Su interfaz es bastante fácil e intuitiva; su sistema de ventanas modulares ofrece animaciones actualizaciones y avisos mucho más atractivos que la competencia.
- Sincronización posibilidad de interacción con los archivos office.
- Sistema de seguridad desde nuestro PC, podremos llamar a nuestro teléfono en caso de haberlo perdido, bloquearlo, localizarlo e incluso borrar parte de su contenido de forma remota.

- Cada Hub (*Marketplace*, Office, gente, retratos, Xbox Live y Zune) tiene una estrecha integración con las aplicaciones nativas y de terceros.

2.4. Desarrollo de Aplicaciones Móviles Multiplataforma

El desarrollo de aplicaciones móviles multiplataforma se refiere a la práctica de escribir una sola base de código fuente para aplicaciones que finalmente serán utilizadas en diferentes sistemas operativos móviles. Aklilu (2012). Los sistemas operativos de los dispositivos móviles más importantes, como *Android*, *iOS* y *Windows 8.1* son estructuralmente diferentes en su arquitectura. No es de extrañar, por lo tanto, que el desarrollo causa problemas masivos para los desarrolladores que tratan de hacer que las aplicaciones que puedan ser utilizadas desde las distintas plataformas, por lo que es muy difícil para los programadores escribir una sola base de código que sea eficiente en las plataformas mencionadas. Aranaz (2009). También significa que los desarrolladores tendrán que reescribir el código y seguir ajustando a trabajar para todas plataformas.

En el proceso de desarrollo de aplicaciones móviles, la preocupación de los desarrolladores es elegir la mejor manera de desarrollar una aplicación en múltiples plataformas sin ajustar código para cada plataforma, la mejor forma de controlar las versiones de las aplicaciones es no perder de vista la semántica implícita y el propósito de la aplicación. Por estas razones y muchas más, la demanda de desarrollo de aplicaciones móviles multiplataforma está en constante crecimiento en el mercado.

La necesidad básica para los programadores en cualquier plataforma es la compatibilidad. Un marco de aplicación móvil puede servir de base para múltiples aspectos de una aplicación. Los siguientes aspectos técnicos a menudo se pueden encontrar:

2.4.1. Principales Características de las Aplicaciones Móviles Multiplataforma

a. Experiencia de usuario

La creación de una experiencia eficaz y agradable para esos cientos de millones de usuarios es una gran oportunidad de tener éxito o no en el mundo del desarrollo del software de hoy.

Con el gran número de aplicaciones descargables que existe en la actualidad, se podría pensar que los desarrolladores ya deberían haber perfeccionado el diseño ideal para la experiencia móvil del usuario. Aklilu (2012). Pero los expertos en *interfaces* y usabilidad en general están de acuerdo con que todavía queda mucho por hacer, que no existen estándares, que se promocionan las mejores prácticas pero a menudo se las ignora y que demasiados desarrolladores prefieren concentrarse en la cuestión técnica en vez de pensar en qué es lo mejor para el usuario.

Los desarrolladores han progresado mucho y varios de ellos desarrollaron aplicaciones excepcionalmente útiles para los usuarios. Pero todavía nos encontramos en los comienzos del software para dispositivos móviles y todos los desarrolladores de aplicaciones tienen la oportunidad de dar con la fórmula ideal para crear la mejor experiencia de usuario. Singh (2012).

b. Directrices para la satisfacción del usuario

Los fabricantes de *Android*, *iPhone*, *Windows phone*, y otras plataformas móviles, utilizan directrices de interfaz de usuario. Les proporcionan algunas reglas a los desarrolladores para trabajar con un sistema operativo o un en particular (como por ejemplo los tamaños de los botones y los widgets); y a veces les proporcionan buenas

prácticas para el diseño de una experiencia de usuario ideal. Reunir estas directrices es una buena forma de familiarizarse con algunos de los principios de diseño centrado en el usuario. Wargo (2012).

- Permitir al usuario comenzar una tarea o un proceso de inmediato.
- No permitir que los elementos de diseño opaquen el contenido.
- El texto debe ser breve, conciso, que pueda leerse de un vistazo.
- Ubicar las operaciones utilizadas con mayor frecuencia al principio de un menú o de una lista.
- Mantener al mínimo la entrada de datos por parte del usuario.
- La navegación debe ser previsible.
- Hacer posible el uso con una sola mano o sin manos.
- Utilizar gráficos impresionantes para “el deleite la gente” tal como sugiere *Apple*.
- Hacer que las búsquedas sean fáciles y rápidas.
- Evitar el manejo de archivos.
- Asegurarse de que los elementos de la interfaz sean coherentes.
- Asegurarse de que sea sencillo tocar el botón correcto.
- Hacer posible que los usuarios puedan manejar el contenido de una forma que se sienta natural.

- Probar los elementos de interfaz con usuarios reales.
- Comenzar a realizar pruebas desde la fase de prototipo.
- Hacer un bosquejo de varios flujos de trabajo para cada tarea, probar cada uno con los usuarios.
- Recordar que la gente siempre está ocupada cuando utiliza las aplicaciones.

c. Arquetipos para la innovación

Los arquetipos pueden ser útiles como un modo de concentrarse en la función y el diseño de una aplicación. Es posible concebir una interfaz o un modelo de interacción totalmente nuevos. Entre estos arquetipos se encuentran:

- El centrado en la aplicación: Las herramientas son las que hacen útil al dispositivo. La experiencia de usuario comienza con una gran variedad de aplicaciones en el dispositivo.
- El centrado en la actividad: Lo que le da forma al diseño es aquello que el usuario desea lograr, las actividades constituyen la experiencia de interacción.
- El centrado en la secuencia: El usuario maneja datos con el tiempo como foco principal.
- El centrado en el proceso o en la tarea: Permite al usuario cumplir con ciertas tareas describiendo el proceso de modo secuencial.

- El centrado en las personas: Basado en la identidad. La función principal es permitir actividades relacionadas con redes sociales, como por ejemplo actualizaciones de estados.

d. Usabilidad

Las aplicaciones o productos de software cuando son lanzados al mercado se espera que tengan cierto grado de aceptación entre los usuarios, ese grado va a depender de las características particulares que cada usuario considere importantes. Resulta relevante para los desarrolladores de software poder medir esa calidad o realizar pruebas de calidad a las aplicaciones construidas, pero para poder medir se necesita saber qué es lo que hay que medir y cómo. Huang (2012).

Debido a que en los últimos años el uso de dispositivos móviles (teléfonos móviles, reproductores de audio portátil, asistentes personales digitales, navegadores *gps*, *tablets*, cámaras digitales, etc.) se ha incrementado de manera considerable, es importante disponer de metodologías y herramientas que permitan realizar estudios de usabilidad Aklilu (2012).

La usabilidad en general tiene que ver con la forma en que se usa algún elemento (herramienta, dispositivo electrónico, etc.), es la facilidad con que se usa y si permite hacer lo que se necesita. Particularmente la usabilidad de una aplicación de software se refiere a la facilidad con que los usuarios pueden utilizar la misma para alcanzar un objetivo concreto. Glera (2013). Este nivel de usabilidad no puede medirse o ser evaluado directamente, debido a que depende de diferentes factores.

De la definición anterior se puede observar que la usabilidad está relacionada con los atributos de una aplicación o sistema, así como también de su contexto; se entiende

por atributo la característica o propiedad de una aplicación de software. En la norma mencionada anteriormente los atributos considerados son los siguientes:

- Efectividad: Está relacionada con la precisión y completitud con la que los usuarios utilizan la aplicación para alcanzar objetivos específicos. La calidad de la solución y la tasa de errores son indicadores de efectividad.
- Eficiencia: Es la relación entre efectividad y el esfuerzo o los recursos empleados para lograr el objetivo. Indicadores de eficiencia incluyen el tiempo de finalización de tareas y tiempo de aprendizaje. A menor cantidad de esfuerzo o recursos, mayor eficiencia.
- Satisfacción: Es el grado con que el usuario se siente satisfecho, con actitudes positivas, al utilizar la aplicación para alcanzar objetivos específicos. La satisfacción es un atributo subjetivo, puede ser medido utilizando escalas de calificación de actitud.

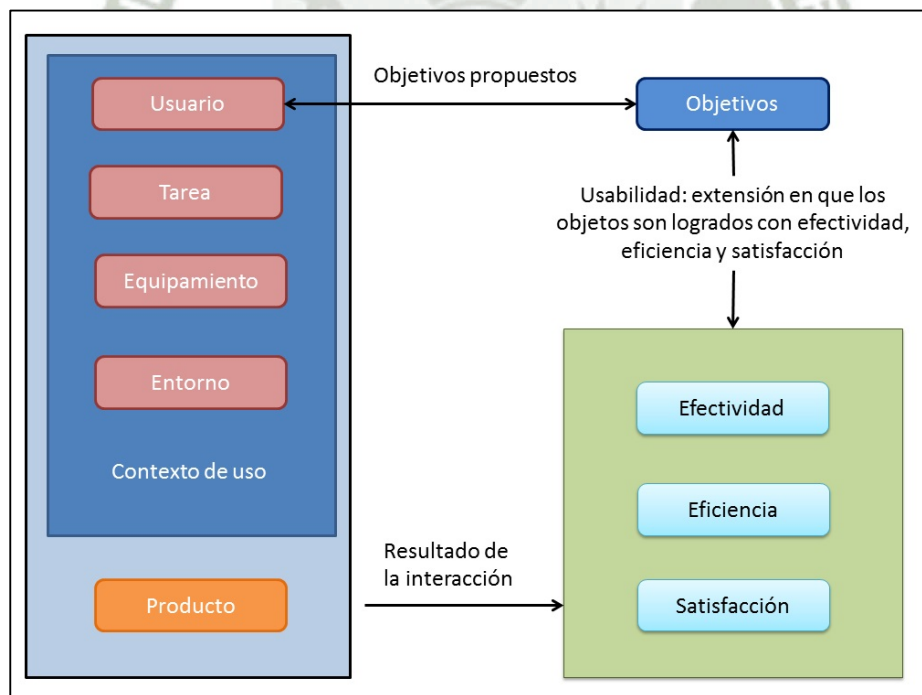


Figura 12: Flujo de la definición de usabilidad de una aplicación móvil.
Fuente: Wargo (2012)

e. Métricas de usabilidad de una aplicación móvil

Una métrica (medida) es un valor numérico o nominal asignado a características o atributos de un objeto, calculado a partir de un conjunto de datos observables y consistentes. Una métrica debe cumplir con ciertas características, las cuales enumeramos a continuación. Wu (2012):

- Debe tener características matemáticas deseables para su medición.
- Una métrica representa una característica que aumenta cuando se presentan rasgos positivos o que disminuye al encontrar rasgos negativos, el valor de la métrica debe aumentar o disminuir en el mismo sentido.
- Cada métrica debe validarse empíricamente en una amplia variedad de contextos antes de publicarse o aplicarse en la toma de decisiones.

En la siguiente tabla se muestra los atributos de usabilidad y las métricas comúnmente asociadas a los aplicativos móviles, para poder cuantificarlos y evaluar sus características más importantes.

Tabla 6

De métricas utilizadas para la evaluación de características móviles.

ATRIBUTOS	MÉTRICAS UTILIZADAS
Efectividad	<ul style="list-style-type: none"> - Tareas resueltas en un tiempo limitado. - Porcentaje de tareas completadas con éxito al primer intento. - Número de funciones aprendidas.
Eficiencia	<ul style="list-style-type: none"> - Tiempo empleado en completar una tarea. - Número de acciones realizadas para completar una tarea. - Tiempo transcurrido en cada pantalla. - Eficiencia relativa en comparación con un usuario experto.
Satisfacción	<ul style="list-style-type: none"> - Nivel de dificultad. - Agrada o no agrada. - Preferencias.
Facilidad de Aprendizaje	<ul style="list-style-type: none"> - Tiempo usado para terminar una tarea la primera vez. - Cantidad de entrenamiento. - Curva de aprendizaje.
Memorabilidad	<ul style="list-style-type: none"> - Número de pasos, toques o vistas usadas para terminar una tarea después de no usar la aplicación por un periodo de tiempo
Errores	<ul style="list-style-type: none"> - Número de errores
Contenido	<ul style="list-style-type: none"> - Cantidad de palabras por vista. - Cantidad total de imágenes. - Número de páginas.
Accesibilidad	<ul style="list-style-type: none"> - Tamaño de letra ajustable. - Cantidad de imágenes con texto alternativo.

Seguridad	<ul style="list-style-type: none"> - Control de usuario. - Número de incidentes detectados. - Cantidad de reglas de seguridad.
Portabilidad	<ul style="list-style-type: none"> - Grado con que se desacopla el software del hardware. - Nivel de configuración.
Contexto	<ul style="list-style-type: none"> - Grado de conectividad. - Ubicación. - Características del dispositivo.

Baker (2011).

Dependiendo de la naturaleza de la aplicación a estimar, en las pruebas se consideran relevantes y diferentes atributos. La usabilidad del sistema no es una simple adición del valor de estos atributos, sino que se define para cada sistema como un nivel a alcanzar por cada uno de los atributos especificados para ese sistema. Huang (2012).

2.4.2. Tipos de Aplicaciones Móviles Multiplataforma

El desarrollo de aplicaciones móviles continúa en expansión, más personas cuentan con un dispositivo móvil y descubren las bondades de tener tecnología al alcance de sus dedos, más personas vuelcan sus miradas a desarrollar nuevas aplicaciones que puedan utilizarse dentro de estos dispositivos, para así llegar a más consumidores. Glera (2013).

Los desarrolladores de aplicaciones móviles deben orientar a los usuarios a escoger de manera adecuada que rumbo coger en su siguiente aplicación, dentro de las tres opciones disponibles:

- Aplicaciones *web* optimizadas para móviles.

- Aplicaciones móviles nativas.
- Aplicaciones híbridas.

2.4.2.1. Aplicaciones basadas en desarrollo *web*

El desarrollo móvil *web* multiplataforma es prácticamente el desarrollo de aplicaciones *web* regulares, pero optimizadas para ser visualizadas desde un dispositivo móvil. Por definición, estas aplicaciones son accedidas utilizando el navegador que viene por defecto dentro de los dispositivos: ya sea Safari (o algún otro) en *iPhone* o *iPad*, digamos Dolphin o Google Chrome (entre otros) para *Android*, y Bing o Internet Explorer en *Windows 8* y *Windows Phone 8*. Acord (2012).

Una aplicación móvil *web* o *webapp* es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, Javascript y CSS. La principal ventaja con respecto al desarrollo convencional es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones, esta es la manera como se hace multiplataforma. Glera (2013).

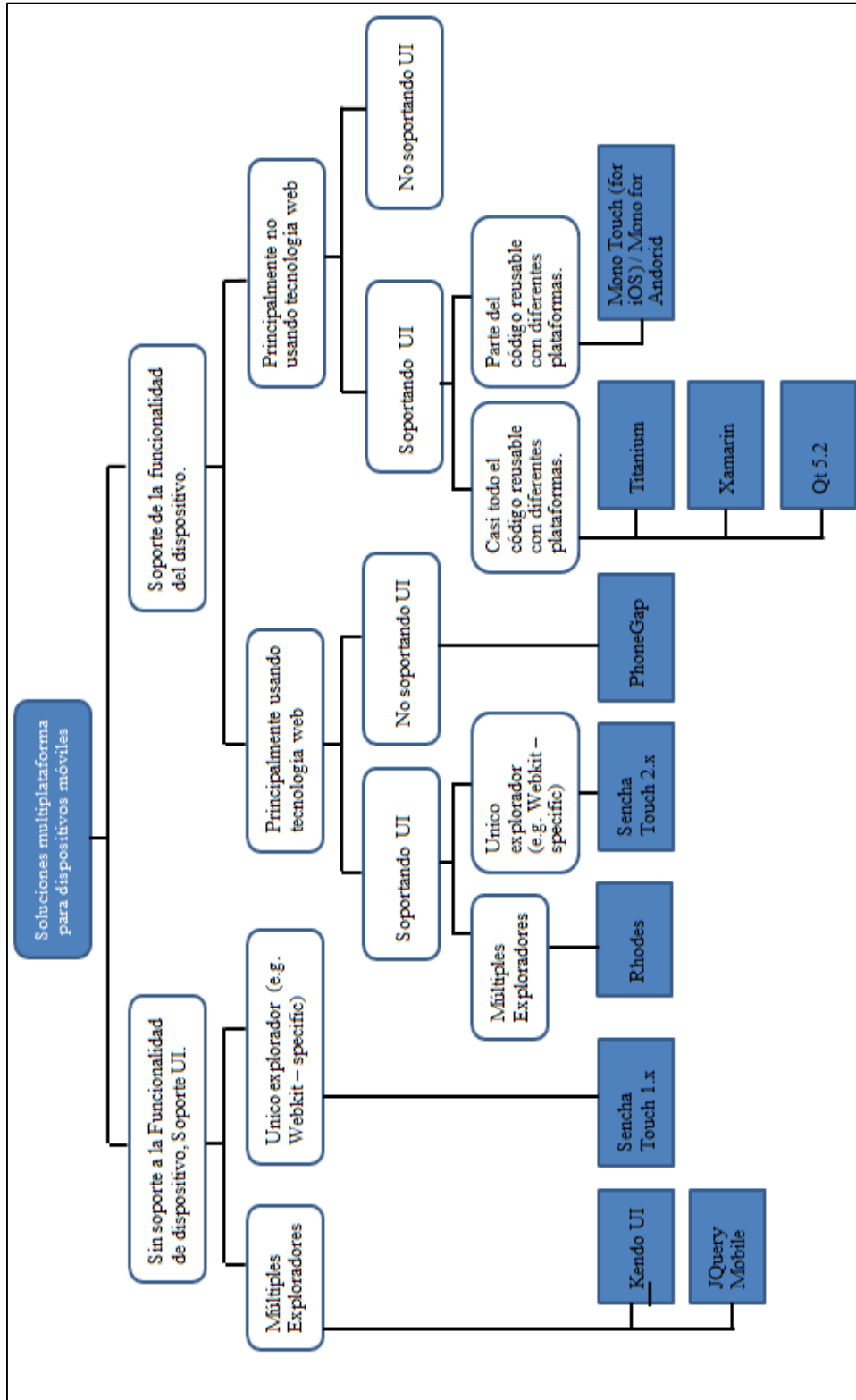


Figura 13: Soluciones multiplataforma para dispositivos móviles.
Fuente: Wargo (2012).

Existen algunas características por las que se considera una aplicación móvil a este tipo de aplicaciones; en realidad la gran diferencia con una aplicación desarrollada de forma convencional es que no necesita instalación por lo que no pueden estar visibles en la diferentes tiendas de aplicaciones móviles y la promoción y comercialización debe realizarse de forma independiente. Acord (2012). De todas formas se puede crear un acceso directo que sería emular el proceso de instalación de la aplicación móvil en el dispositivo.

Las *apps web* móviles son siempre una buena opción si nuestro objetivo es adaptar la *web* a formato móvil, en la siguiente imagen mostramos la arquitectura de una aplicación móvil multiplataforma basada en desarrollo *web*. Saks (2012).

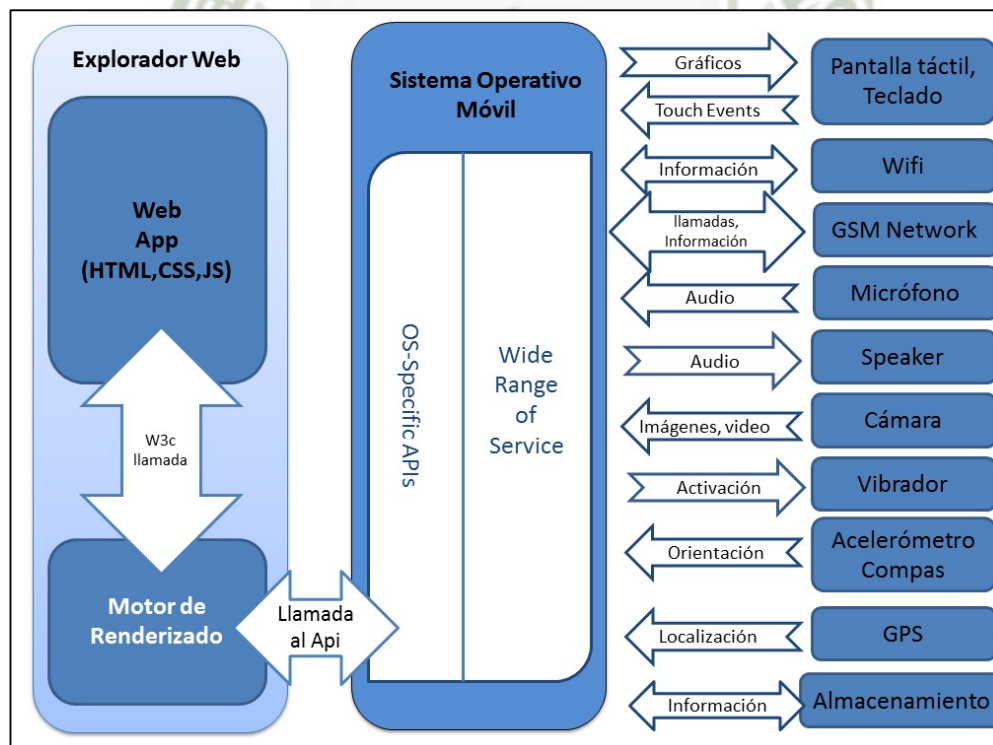


Figura 14: Arquitectura de una aplicación móvil basada en la web
Fuente: (Diep, 2013)

Una de las principales ventajas de los dispositivos móviles es que todos tienen un navegador HTML5, por lo que podemos, simplemente, crear una aplicación *web* y usarla desde el navegador de uno de estos móviles. Glera (2013). Esto tiene sus ventajas y desventajas:

a. Ventajas de las aplicaciones basadas en desarrollo *web*

Fácil diseño: no hay que pensar en desarrollar una aplicación para móviles si no queremos. Basta con hacer un diseño adaptado a una pantalla y resolución más pequeñas, que puede ser simplemente adaptando un *CSS* por cada dispositivo o podemos rediseñar la navegación al completo. Además, las aplicaciones *web* se pueden personalizar para que parezcan aplicaciones nativas: icono de aplicación, pantalla completa, *splash screen*, barra de estado. Akkiraju (2011).

Fácil implementación: las aplicaciones *web* pueden ser desarrolladas en cualquier tecnología de servidor, así que podemos usar nuestro lenguaje favorito (Java, Grails, Php, Ruby, Python, etc.) con la seguridad de que la aplicación se verá igual en todos los terminales. Aklilu (2012).

Seguridad: el desarrollador controla el acceso a la aplicación, se puede actualizar sin tener que pedir permiso. Nadie va a acceder ilegalmente a la versión *premium* de tu aplicativo, ni va a vetar la entrada a las tiendas virtuales de aplicaciones porque no cumple los requisitos.

b. Desventajas de las aplicaciones basadas en desarrollo *web*

APIs nativas: No hay acceso completo a todas las *Apis* nativas del móvil. Aunque la cámara y el micro son accesibles con *Flash*, todos sabemos que esa tecnología está

vetada en *IOS*. Desde *HTML5* y *Javascript*, es posible acceder a las coordenadas del *GPS*, pero no en tiempo real ni de la misma manera que si pudiéramos acceder a la *Api* del móvil directamente. Y olvídate de usar el acelerómetro, la agenda, la brújula, etc.

Difícil de usar, fácil de olvidar: para usar una aplicación *web* en un móvil, es necesario que el usuario abra el navegador y teclee la dirección, ya sea porque la sepa, la haya encontrado en *Google* o la haya recibido por correo o *chat*. Una vez abierta la aplicación, debe añadirla a favoritos o, mejor todavía, crear un icono de acceso directo en el móvil para acceder a ella. Saks (2012). Confiar en que nuestros posibles usuarios acaben haciendo todo esto para poder acceder cómodamente a nuestra aplicación en futuras ocasiones es pecar de ingenuo. Es mucho más fácil descargar una aplicación y que aparezca directamente como un icono en nuestro móvil. Saks (2012).

Lentitud de renderizado: sin contar que renderizar *HTML* e interpretar *Javascript* es sin duda más costoso en uso de hardware que ejecutar una aplicación nativa, cada petición que hagamos en nuestra aplicación implicará una recarga de la página o un acceso, en mayor o menor medida, contra nuestro servidor. Cualquier espera por pequeña que sea impacta en la experiencia de usuario. Una aplicación nativa tiene todos los recursos y procesos guardados en local, y solo accede al servidor para obtener o enviar datos si es que los necesita. Por tanto, una aplicación *web* no tiene la fluidez y velocidad de manejo que una aplicación nativa, ni nunca la tendrá. Aranaz (2009).

Peor monetización: es más fácil que un usuario pague por nuestros servicios si simplemente cobramos una cierta cantidad por nuestra aplicación al descargarla del *App Store*, que no hacer que el usuario se tenga que registrar y efectuar el pago en nuestra *web*, introduciendo manualmente todos sus datos como el número de tarjeta, dirección,

etc. Si el usuario ya tiene sus datos guardados en el *App Store*, para comprarla solo tiene que poner su contraseña y confirmar la compra. Saks (2012).

Tenemos que ser conscientes de que los mercados de aplicaciones son una realidad: *Apple* tiene su *App Store* para *IOS* y *MacOS*, *Google* su *Chrome Web Store*, *Google Apps Marketplace* y el *Android Market*, *Amazon* su *Amazon Appstore*, incluso hay *markets* alternativos como *OpenAppMkt*. Cada vez más empresas invierten en crear un entorno fácil y cómodo para que el usuario pueda descargar, probar y comprar aplicaciones, repartiéndose los beneficios. Pensar que podemos competir con todo esto con una aplicación *web* para móviles, que simplemente se puede acceder desde una dirección en un navegador, es una ilusión. Bruning (2012).

2.4.2.2. Aplicaciones nativas

Una aplicación nativa es una aplicación que ha sido desarrollada específicamente para ejecutarse en el sistema operativo de un dispositivo y que se instala en dicho dispositivo. Este tipo de aplicaciones son diseñadas de forma específica para cada terminal, es decir, se desarrollan en diferentes lenguajes de programación en función del sistema operativo en el que van a ser utilizadas.

Por ejemplo, aquellas aplicaciones que sean desarrolladas para *iOS*, es decir para el sistema operativo de *iphone* e *ipad*, son diseñadas con los lenguajes: *Objective C*. Por otro lado las que son desarrolladas para el sistema operativo de *Android* utilizan el lenguaje *Java*. Aklilu (2012).

a. Características de las aplicaciones móviles nativas

- Serán descargadas desde el mercado de aplicaciones para el sistema operativo, el *AppStore*, *Google Play* o *Windows Store*.
- Es una aplicación distinta por sistema operativo: una para *Android*, otra para iOS y otra más para Windows.
- Se desarrollan con lenguajes como Java (*Android*) u *Objective-C* (iOS), pero hay alternativas como *Appcelerator* que permiten desarrollarlas utilizando JavaScript puro, para ambas plataformas.

Como ya se ha mencionado antes, las aplicaciones nativas tienen acceso total a las utilidades del sistema operativo del dispositivo: Dispositivos de almacenamiento, cámara, *gps*, acelerómetro, etc. Esto hace que la experiencia de usuario sea la más completa. Saks (2012). Todo esto deberá ser considerado en la arquitectura de la aplicación móvil multiplataforma.

Además del total acceso a los elementos del teléfono las aplicaciones nativas no requieren de conexión *web* para ser ejecutadas (aunque esto tiende a cambiar) Aklilu (2012). Por último es importante destacar que las aplicaciones nativas tendrán mucha más visibilidad ya que se distribuyen a través de las tiendas de aplicaciones de los fabricantes. Glera (2013).

Por otra parte, el desarrollo móvil nativo es el desarrollo de aplicaciones que serán instaladas en el sistema de archivos de cada dispositivo y serán distribuidas por los mercados de aplicaciones móviles, como el *AppStore* (iOS), el *Play Market* (*Android*) o el *Windows Store*.

b. Ventajas de las aplicaciones móviles nativas

- Posibilidad de utilizar funcionalidades nativas: siempre puedes utilizar funcionalidades nativas, como la cámara, el acelerómetro, etc.
- Nuevas alternativas para desarrollar multiplataforma: Aun cuando lo regular es desarrollar estas aplicaciones utilizando las herramientas oficiales (como el *Android SDK* y Java en *Android*), han surgido varias nuevas tecnologías para llegar a varios dispositivos con casi el mismo código fuente, como *Titanium Appcelerator*.
- *Performance*: Típicamente llegan a desempeñarse mejor que las aplicaciones *web*.
- Hay un mercado para estas aplicaciones: Las personas pueden encontrar tu aplicación utilizando el *AppStore* o el *Play Market*. en cualquiera de los casos, estas aplicaciones pueden ser puestas a la venta.
- Modo offline: Tu aplicación puede utilizarse sin necesidad de una conexión a internet.

c. Desventajas de las aplicaciones móviles nativas

- Típicamente son más costosas de llevar a cabo: Por lo general, son más costosas de llevar a cabo, ya que se necesita invertir mucho más tiempo.
- Mayores costos de mantenimiento: De igual forma, mantener varias versiones de la aplicación es costoso.
- Actualizaciones y aprobaciones: Poner una aplicación en el *AppStore* es tedioso, puede llevar hasta más de una semana. De igual forma, si tienes una nueva

funcionalidad o descubriste un bug que quieres arreglar, debes esperar otra semana para que los chicos de *Apple* aprueben tus cambios.

- Aun cuando hay opciones como *Appcelerator*, no llegas a todos: *Appcelerator*, por ejemplo, permite desarrollar para *Android*, *iPhone*, iPad y móvil-web (nuevo).

2.4.2.3. Aplicaciones móviles híbridas

Estas aplicaciones son una combinación de los dos tipos de aplicaciones anteriores: aplicaciones nativas y aplicaciones *web*. El concepto de aplicación híbrida consiste en una página basada en HTML mediante las herramientas estándar HTML5, CSS3 y JavaScript que posteriormente es envuelta en código nativo, de modo que el paquete resultante puede ser distribuido, al igual que las aplicaciones nativas, a través de los canales de compra de aplicaciones. Saks (2012).

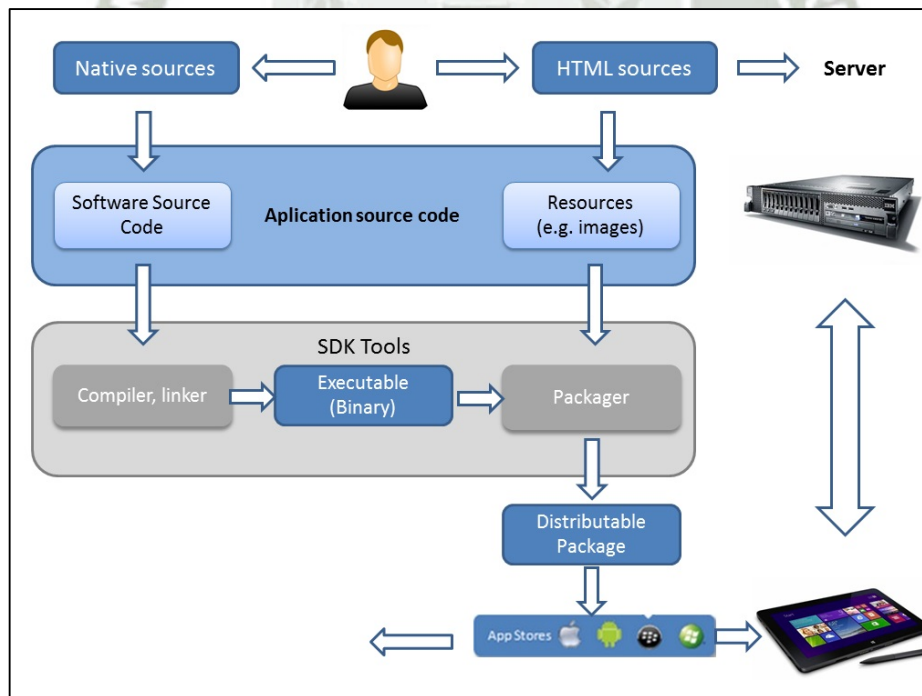


Figura 15: Arquitectura de una aplicación móvil multiplataforma nativa
Fuente: (Diep, 2013)

La idea es fácil: utilizar el desarrollo nativo cuando es mejor o es necesario (para utilizar la cámara, por ejemplo), pero utilizar tecnologías *web* y el desarrollo *web* cuando es más práctico (por ejemplo en *interfaces* más complejas).

a. Características de las aplicaciones híbridas

- Todas las listas infinitas en la aplicación son nativas.
- Todas las ventanas son nativas, estas funcionan como contenedores de vistas *web*.
- Todas las páginas que son detalles con información son JavaScript y HTML, por su fácil desarrollo.

b. Ventajas de las aplicaciones híbridas

- La principal ventaja es que son multiplataforma, permiten que el código fuente se pueda ejecutar en diferentes plataformas.
- Estas aplicaciones permiten acceder a todos los recursos del móvil, como pueden ser la cámara o el sistema *GPS*.
- No es necesaria conexión a internet para ejecutar la aplicación a excepción de partes concretas de la aplicación que requieran dicha conexión.
- Aúnan lo mejor de los dos tipos de aplicaciones, de modo que se tratan de una buena alternativa cuando existe duda en qué tipo de aplicación realizar.
- No es necesario utilizar un lenguaje específico para crear la aplicación íntegra, solo es necesario para la plataforma contenedor. La mayor parte utiliza HTML5, un

leguaje más sencillo y por lo tanto el desarrollo de estas aplicaciones no supone tanto tiempo.

c. Desventajas de las aplicaciones híbridas

- Las actualizaciones son manuales, el usuario debe acceder al canal de compra de aplicaciones de las distintas plataformas para conseguir las actualizaciones de la aplicación.
- Al igual que en el caso de las aplicaciones nativas para poder distribuir las híbridas en los *market place* es necesario pasar unos procesos de validación que en ocasiones pueden resultar duros.
- La parte nativa que envuelve el código fuente basado en HTML, si debe ser programado con un lenguaje específico para las distintas plataformas.

2.5. Frameworks para el Desarrollo de Aplicaciones Móviles Multiplataforma

2.5.1. JQuery Mobile

Es un *framework* de desarrollo móvil multiplataforma basado en HTML5, y esta optimizado para dispositivos móviles táctiles. Este *framework* es de fácil uso y cuenta con una documentación amplia sobre su *interface* de desarrollo. Como su nombre indica, está desarrollado bajo jQuery y jQuery UI, cuenta con un código ligero de poco peso basado en mejora progresiva, esto al momento del desarrollo. Greenberg(2012). Este framework permite utilizar la herramienta de diseño *web* ThemeRoller para el diseño de *interfaces*, las cuales luego se adaptaran perfectamente a las pantallas táctiles.



Figura 16: Logo de JQuery.
Fuente: Wargo (2012)

Este *framework* permite también hacer aplicaciones con funciones en el lado del servidor, esta es una técnica muy común en sitios *web*, instalar jQuery Mobile enlazando directamente a los archivos almacenados en la nube, se recomienda ampliamente no aplicar esa técnica para aplicaciones que utilicen la combinación con *PhoneGap*. Muy frecuentemente la aplicación será utilizada sin ninguna conexión a internet o con conexión intermitente. Si los archivos de jQuery están enlazados a documentos en la nube, la aplicación quedaría posiblemente inutilizable. Acord (2012).

Sencha Touch 2 es un *framework* para desarrollo de aplicaciones móviles basado en HTML5. Este *framework* permite construir aplicaciones móviles para los sistemas operativos iOS, *Android*, *BlackBerry*, *Kindle Fire* y otros. El *framework* de desarrollo se puede descargar gratuitamente. Se ha de instalar de manera local y es necesario tener un servidor local ejecutándose. Wargo (2012).

Las aplicaciones en *Sencha Touch* funcionan mejor cuando siguen estructuras simples que el mismo *framework* proporciona. Son un conjunto de estándares, convenciones y clases que hacen que el código sea mantenerle, especialmente cuando el trabajo es parte de un equipo. Bruning (2012).

Sencha Touch está optimizado para la creación de aplicaciones que funcionan a través de múltiples plataformas. Para escribir aplicaciones lo más simple posible, proporcionamos una sencilla pero potente arquitectura de aplicación que aprovecha el

patrón *Model View Controller (MVC)*. Este enfoque mantiene su código limpio, comprobable, y fácil de mantener, y proporciona una serie de beneficios a la hora de escribir sus aplicaciones:

Historia de Apoyo: Permite navegar dentro de sus aplicaciones utilizando de nuevo el apoyo botón y acceder a cualquier parte de su aplicación que puede estar relacionado.

Deep Linking: Permite compartir enlaces profundos que abren cualquier pantalla en su aplicación, al igual que la vinculación a una página *web*.

Perfiles de dispositivo: Permite personalizar fácilmente la interfaz de usuario de su aplicación para móviles, *tablets* y otros dispositivos, mientras que compartir código común.

a. Anatomía de una *app* sencha

Una aplicación es una colección de modelos, vistas, controladores, Tiendas y perfiles, además de metadatos adicionales para las entidades relacionadas de aplicaciones, tales como iconos de aplicaciones e imágenes de pantalla de lanzamiento. Wargo (2012).

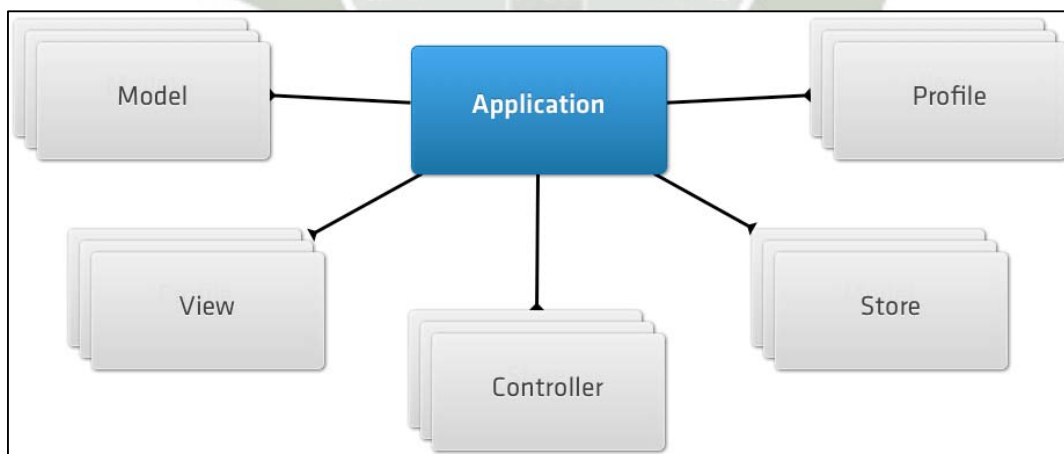


Figura 17: Arquitectura de una aplicación Sencha
Fuente: McWherter (2012)

- **Modelos:** representan un tipo de objetos de datos en su aplicación – puede ser una aplicación de comercio electrónico que podría tener modelos de usuario, del producto, y Orden.
- **Vistas:** se encargan de mostrar los datos de sus usuarios y para el aprovechamiento de los componentes integrados en *Sencha Touch*.
- **Controladores:** manejan la interacción con la aplicación, por la escucha de las interacciones del usuario, tales como grifos y golpes, y tomar medidas en consecuencia.
- **Tiendas:** son responsables de la carga de datos en su aplicación y de los componentes que alimentan tales como Listas y *DataViews*.
- **Perfiles:** le permiten personalizar fácilmente la interfaz de usuario de su aplicación para las tabletas y los teléfonos, mientras que compartir tanto código como sea posible.

2.5.2. Phonegap

El desarrollo de una aplicación en *Phonegap* se realiza en *HTML5*, *CSS3* y *Javascript*. Son aplicaciones *web* empaquetadas para ser tratadas como aplicaciones móviles. Es importante diseñarlas pensando en la usabilidad y experiencia de usuario de un móvil y no una *web* convencional. *Phonegap* lo que hace es embeber la aplicación en el navegador *web* del dispositivo. Esto facilita el soporte de muchas plataformas pero a la vez las aplicaciones no se suelen ver igual en todas las plataformas ya que se depende del navegador de cada una. Springer (2013).

Además *Phonegap* dispone de un servicio en la nube para compilar y crear la aplicación para las diferentes plataformas. Este servicio facilita la compilación de las

aplicaciones de *PhoneGap* y libera de esta tarea al desarrollador que debería configurar los entornos de cada plataforma y compilar la aplicación una por una. Aklilu (2012).



Figura 18: Logo de PhoneGap
Fuente: McWherter, (2012)

a. Ventajas

- Independientemente de la plataforma del lado del servidor y la experiencia del lenguaje de programación, un número significativo de los desarrolladores tienen experiencia con HTML, JavaScript y CSS.
- *PhoneGap* sigue una arquitectura de *plugin*, lo que significa que el acceso a las API nativas de los dispositivos se puede ampliar de forma modular. Hay muchos *plugins PhoneGap* para elegir - permitiendo a los desarrolladores centrarse en las habilidades basadas en la *web* que ya tienen. (Esta es una debilidad, así, como veremos posteriormente).
- Soluciones *PhoneGap* existían en este espacio desde el principio, y han madurado hasta el punto en el que las ofertas de valor añadido en la parte superior de la *CPT* básica son la norma. Por ejemplo, *PhoneGap* tanto de Adobe Build o Icenium de Telerik, permite a los desarrolladores construir plataformas de destino soportados en la nube, sin *SDKs* locales (es decir, los usuarios de Mac no pueden construir aplicaciones

de iOS). Además de la nube de Icenium construir servicios, Telerik también ofrece Kendo UI Mobile (un marco MVVM objetivo para el rendimiento en el móvil), análisis de aplicaciones a través de *EQATEC* y un *backend-as-a-Service (BAAS)* ofreciendo llamado *everlive*. Adobe ha integrado *PhoneGap Build* capacidades en soportes (un *IDE* basado en la *web*) y *Dreamweaver*.

b. Desventajas

- Por supuesto, ser libre no es garantía de éxito. La fuerza de ser de código abierto - y el aprovechamiento de las características de una amplia gama de contribuyentes - es a la vez una bendición y una maldición. Si necesita extender su aplicación con una costumbre *PhoneGap plugin*, más probable es que se encuentra uno. Sin embargo, puede estar fuera de fecha y es compatible con las plataformas de destino que usted necesita. Springer (2013).
- El rendimiento de aplicaciones *PhoneGap* ha sido a menudo criticado. La interfaz de usuario nativa siempre superará a una solución híbrida, pero las mejoras en el hardware del dispositivo y las implementaciones *web* han reducido la brecha. Sus desarrolladores *web* tendrán que prestar mucha atención al rendimiento, lo que significa que su conocimiento de las herramientas de perfiles, así como que los marcos de interfaz de usuario *web* son móviles-amigable es esencial Springer (2013).

2.5.3. Titanium Appcelerator

El desarrollo con el *framework* Titanium *Appcelerator* se realiza en Javascript. Dispone de un entorno de desarrollo propio (Titanium Studio) basado en Eclipse que facilita el desarrollo, la configuración y compilación de las aplicaciones.

Appcelerator, a diferencia de *Phonegap*, al compilar traduce a código nativo. Esto hace que normalmente las aplicaciones con *Appcelerator* tengan mejor experiencia de usuario y mejor rendimiento. A cambio, la curva de aprendizaje es más costosa ya que no es programación *web* como *Phonegap*. Acord (2012).



Figura 19: Logo AppCelerator

Fuente: <http://www.appcelerator.com/titanium/>

Un entorno de desarrollo abierto y extensible para la creación de hermosas aplicaciones nativas a través de diferentes dispositivos móviles y sistemas operativos, incluyendo *iOS*, *Android* y *BlackBerry*, así como híbridos y *HTML5*. Incluye una fuente abierta *SDK* con más de 5.000 *APIs* de dispositivos y del sistema operativo para móviles, *Studio*, un potente *IDE* basado en *Eclipse*, *aleación*, un *framework MVC* y servicios en la nube para un *backend* móvil listo para su uso. Glera (2013).

Titanium es el entorno de desarrollo móvil líder de la opción para cientos de miles de desarrolladores. Con más de 74.000 aplicaciones móviles desplegadas en 250 millones de dispositivos, el galardonado entorno de titanio ayuda a las organizaciones a obtener en el mercado un 60% más rápido y lograr una ventaja competitiva significativa. Sommer (2012).

Titanium simplifica el desarrollo con una sola base de código. Rápidamente construir, probar, empaquetar y publicar aplicaciones móviles utilizando sólo JavaScript y una sola base de código. Ahora, los desarrolladores con conocimientos de desarrollo *web* pueden construir de inmediato ricas aplicaciones móviles, nativas. Usted ya no tendrá que gestionar múltiples conjuntos de herramientas para desarrolladores, lenguajes y metodologías. Greenberg (2012).

Cerca del 70-80% de nuestro código se puede reutilizar en aplicaciones. Eso nos ahorra un montón de tiempo y mucha habilidad. Es muy difícil encontrar diferentes características nativas y combinarlos en un solo equipo, pero es relativamente fácil superar la curva de aprendizaje de JavaScript. (Marijn Deurloo, CEO, imgZine)

2.5.4. Strawberry

El desarrollo de aplicaciones en *Strawberry* se programa: la lógica en C++ y el diseño en HTML/CSS. De esta manera se separa completamente el diseño de la programación y facilita la separación de roles: maquettador y desarrollador.

Strawberry permite obtener el máximo rendimiento: utiliza la aceleración por hardware *OpenGL* y está programado íntegramente en C++. *Strawberry* dispone de su propio renderizador lo que ayuda a que las aplicaciones se visualicen igual en todas las plataformas y dispositivos.



Figura 20: Logo Strawberry

Fuente: <http://www.strawberrysdk.com/whatis.php>

Strawberry *SDK* es una herramienta de desarrollo que permite a los desarrolladores a reducir el costo de desarrollo de aplicaciones multiplataforma.

No existe la necesidad de tener que desarrollar para cada plataforma por separado. El *SDK* de Strawberry es una herramienta multiplataforma que puede compilar sus aplicaciones y juegos para *Android*, *iOS* y *Mac OS X* con el mismo código utilizando un solo proyecto. Sommer (2012). El desarrollador no necesita aprender nada nuevo, Strawberry *SDK* está escrito en *C++* y *OpenGL ES 1.0*, así como un desarrollador puede escribir todo el código en *C++*, exprimir el potencial de código nativo en cada plataforma. Para la creación de los diseños ha creado un *HTML* personalizado y *CSS*. Haciendo que la vida del diseñador sea mucho más fácil. Greenberg (2012).

Con Strawberry *SDK* se pueden crear diseños personalizados que se ven exactamente igual en todas las plataformas. Se puede crear diferentes interfaces de usuario si desea crear la mejor *UX* para cada plataforma. La obtención de los resultados deseados siempre que lo desee es ahora sin esfuerzo y la iluminación rápida. Es más, con strawberry *SDK* puede crear aplicaciones nativas que aprovechan al máximo las capacidades del hardware del dispositivo, así como por medio de sensores como el *GPS* y el acelerómetro de una manera fácil. Usted obtendrá el máximo rendimiento: perfecto para juegos y aplicaciones pesadas.

Strawberry *SDK* no es sólo una herramienta para los desarrolladores y diseñadores, también es una nueva metodología de trabajo se centró en **el dinamismo y el trabajo en paralelo**. Diseño de la disposición está completamente separado del código (Glera, 2013). De esta manera ajustar la interfaz de usuario no va a interferir con la programación, o viceversa. Gracias a este marco, es mucho más simple de optimizar los tiempos de desarrollo, mediante los desarrolladores y diseñadores de la interfaz de usuario funcionan en paralelo. Sommer (2012).

Strawberry SDK facilita y mejora la comunicación entre el desarrollador y el diseñador de la *interfaz* de usuario ya que no es una herramienta única para los *techies*. Ha hecho que sea más fácil de iterar sobre los diseños y la facilidad de uso sin necesidad de programación lógica, no se utilizara mayor tiempo ajustar la interfaz de usuario y la compilación de una y otra vez.

Gracias al *player Strawberry* (incluido en Strawberry *SDK*) puede crear diseños para juegos y aplicaciones a velocidad de la luz a través de nuestro HTML y CSS personalizado especialmente creado para el diseño de diseño. Para la creación de diseños

sólo se necesita un editor de texto y luego ver lo que está realizando con la Strawberry Player. Piense en ello como un navegador para sus *interfaces* de usuario que le permite crear el diseño perfecto con precisión de *píxel* perfecto.

Como el código es completamente independiente del esquema de trazado, Strawberry *SDK* le ofrece diseño súper flexible que le permite implementar fácilmente las cosas interesantes como las pruebas, diferentes diseños para teléfonos, tabletas y ordenadores por cada plataforma, o incluso modificar completamente el diseño de interfaz de usuario de un aplicación o el juego sin tener que preocuparse sobre el código fuente.

Strawberry *SDK* le ayudará a reducir las aplicaciones y juegos tiempo de desarrollo en un 42%. Como marco de plataforma cruzada, se utiliza el mismo código en *Android*, *iOS* y *Mac OS*, con lo que sus productos más rápido a sus usuarios y lo que es más fácil de mantener el código.

2.5.5. Qt 5.2



Figura 21: Logo QT
Fuente: <http://qt-project.org/>

El mercado nos está diciendo experiencia de usuario multiplataforma es una necesidad absoluta ahora. Estamos escuchando. Creemos que el desarrollo de software moderno tiene que tener esto en cuenta -. Pensar fuera de la caja de un sistema operativo o una plataforma

de su estrategia de tecnología debe incluir la capacidad de fácilmente crear dispositivos conectados, el IEU y aplicaciones que se ejecutan en cualquier lugar en cualquier dispositivo, en cualquier sistema operativo en cualquier momento. Greenberg (2012).

Con Qt puede crear *interfaces* de usuario innovadoras y únicas para los adaptados para su público objetivo, ya sea para un panel de operador industrial o una interfaz de usuario del auricular llamativo. Lo que hace este rápido desarrollo es el conjunto de altamente intuitivos clases C++ biblioteca, de hecho, proporciona una gama completa de funcionalidad. Sommer (2012). No sólo son fáciles de aprender y utilizar, sino también el marco es significado modular puede seleccionar sólo las clases que necesita, haciendo la depuración y pruebas más rápido y su huella de dispositivo más pequeño, de desarrollo rápido y rendimiento rápido. La aceleración de la productividad aún más también se puede lograr con los controles Qt Empresa prediseñadas y personalizables. Anglin (2013).

Estas clases ofrecen más que apenas diseño *GUI*. Usted también consigue clases de plataforma cruzada para cosas como, sólo para nombrar unos pocos: el acceso de base de datos, roscado, multimedia, y la creación de redes que permite centrarse en que las competencias clave y diferenciadores de productos en lugar de los fundamentos.

Reutilización de código para el desarrollo de aplicaciones y creación de dispositivos. Qt es un *framework* multiplataforma utilizado para crear dispositivos con *interfaces* impresionantes y potentes aplicaciones nativas. Es una elección inteligente para cualquier tipo de desarrollo de software. Imagínense poder renderizar o mostrar que también funciona básicamente en todas partes sin tener que reescribir el código. Si desea la diversidad entre las plataformas, como un diseño de interfaz de usuario sensible para diferentes tamaños de pantalla, esto es fácil de implementar con Qt, también. Se puede pensar que esto es

demasiado bueno para ser verdad, pero no hay soluciones de compromiso con Qt. Aranaz (2009). Se pueden crear *interfaces* de usuario impresionante y obtener un rendimiento increíble mientras que apunta múltiples pantallas, dispositivos, sistemas operativos, plataformas, cualquier palabra que desea utilizar para describir que Qt corre por todas partes.

a. Flexibilidad

La flexibilidad en el desarrollo de software es muy importante, con Qt usted puede elegir su método de desarrollo. Se puede optar por el uso de las *API* altamente intuitivas Qt para el lenguaje de programación C++ o utilizar el lenguaje declarativo Qt, QML, lo que hace que está comenzando súper rápido. No es ninguna coincidencia que la tecnología se llama Qt Quick. De igual manera no se tiene que elegir uno u otro, se puede tener ambas cosas. Usar Qt / C++ para su lógica de la aplicación *backend* y *QML* para el diseño de su interfaz de usuario. Aranaz (2009).

2.5.6. Xamarin

Al considerar cómo construir aplicaciones *iOS*, *Android* o *Windows 8.1*, muchas personas piensan que los lenguajes nativos de desarrollo, *Objective-C* y *Java*, respectivamente, son la única opción. Sin embargo, en los últimos años, ha surgido un nuevo ecosistema entero de plataformas para crear aplicaciones móviles. Peppers (2014).

Xamarin es único en este espacio, ofreciendo un solo lenguaje de programación el C#, la biblioteca de clases, y el tiempo de ejecución que funciona a través de las tres plataformas móviles de *iOS*, *Android*, *Windows 8* y *Windows Phone* sin dejar de

compilación nativa (no -interpretado) aplicaciones que tienen el *performance* suficiente incluso para los juegos más exigentes. Peppers (2014).



Figura 22: Logo Xamarin –
Fuente: <http://xamarin.com/>

Xamarin es único, ya que combina toda la potencia de las plataformas nativas y añade una serie de características de gran alcance de su cuenta, incluyendo:

- Completo enlace para los *SDK* Nativos: Xamarin contiene enlaces durante casi todo el *SDK* de la plataforma subyacente tanto en *iOS* y *Android*. Además, estos enlaces están fuertemente-mecanografiadas, lo que significa que son fáciles de navegar y usar, y proporcionar la comprobación de tipos en tiempo de compilación robusto y durante el desarrollo. Esto conduce a un menor número de errores en tiempo de ejecución y las aplicaciones de mayor calidad.
- Interoperabilidad con *Objective-C*, Java, C y C ++: Xamarin ofrece instalaciones para invocar directamente las bibliotecas de *Objective-C*, Java, C y C ++, que le da el poder de utilizar una amplia gama de tercera código partido que ya se ha creado. Esto le permite tomar ventaja de *iOS* existentes y bibliotecas *Android* escritos en *Objective-C*, Java o C / C ++. Peppers (2014).
- Lenguaje de programación moderno: Xamarin aplicaciones están escritas en C #, un lenguaje moderno que incluye mejoras significativas en *Objective-C* y Java, tales

- como funciones de idioma dinámico, las construcciones funcionales como *Lambdas*, *LINQ*, características de programación en paralelo.
- Entorno de desarrollo integrado y moderno (IDE) - Xamarin utiliza Xamarin Studio en Mac OS X, y también Xamarin Studio o Visual Studio 2013 en Windows. Se trata de dos de *IDE* modernas que incluyen características tales como la finalización de código automático, un proyecto sofisticado y sistema de gestión de soluciones, una biblioteca completa plantilla de proyecto, control de código fuente integrada, y muchos otros.
 - Soporte Móvil Multiplataforma: Xamarin ofrece sofisticado soporte multiplataforma para las tres principales plataformas móviles de *iOS*, *Android* y *Windows Phone*. Las aplicaciones pueden ser escritas para compartir hasta el 90% de su código, y nuestra biblioteca Xamarin. Mobile ofrece una *API* unificada para acceder a recursos comunes en las tres plataformas. Esto puede reducir significativamente los costes de desarrollo y el tiempo de comercialización para los desarrolladores móviles que se dirigen a las tres plataformas móviles más populares.

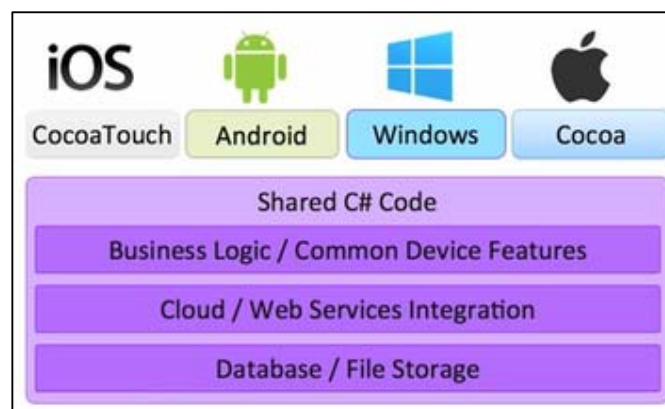


Figura 23: Arquitectura de una aplicación móvil en Xamarin
Fuente: Marin (2012)

2.5.2. Código Compartido

Aplicaciones Xamarin están escritas enteramente en *C #* y tener pleno acceso a la *NET Framework* en cada plataforma. Esto significa que puede escribir código que utiliza cualquiera de las funciones de NET, incluyendo las operaciones de archivos, acceso a la *Web*, el almacenamiento de base de datos, manipulación de *XML* y más, a continuación, compartir ese código a través de *iOS*, *Android* y *Windows*. Peppers (2014).

2.6. Computación en la Nube para Aplicaciones Móviles

Con el crecimiento del almacenamiento en la nube y servidores de la nube, se ha hecho más fácil que nunca para copia de seguridad de todos sus archivos importantes de la computación en línea. Por este motivo, y para satisfacer las necesidades de los sistemas de computación más exigentes, se ha producido una interesante evolución de las arquitecturas de cálculo, basada fundamentalmente en la ejecución simultánea de procesos en múltiples equipos informáticos. Aranaz (2009).

Ahora se le da la flexibilidad de acceder a todos sus archivos desde cualquier parte del mundo, con la ventaja de saber que todas sus fotografías importantes, vídeos, música, archivos, documentos, así como otros programas y los datos se almacenan de forma segura y disponible la 24 horas al día 7 días a la semana.

A lo largo del presente estudio se ha optado por emplear el término en idioma inglés *cloud computing* y no el de computación en nube, en idioma español tal y como lo traduce la *Unión Internacional de Telecomunicaciones (UIT)* y la Unión Europea debido a que es el término en idioma inglés el que aparece citado con mayor frecuencia en la mayor parte de la literatura especializada y en la mayor

parte de las numerosas publicaciones consultadas durante la elaboración del mismo. Peppers (2014).

Se destaca principalmente lo que es más importante para el consumidor, con categorías tales como la cantidad de espacio de almacenamiento que ofrece la empresa, así como el grado de fiabilidad de la empresa de almacenamiento de archivos de la nube. Se muestra a continuación las características más importantes de los servicios de almacenamiento en la nube:

2.6.1. La Fiabilidad y el Tiempo de Actividad

Esta característica se basa en datos estadísticos relacionados con el tiempo de actividad del servicio. Se espera que las compañías de almacenamiento en la nube tengan un tiempo de actividad del 99,9%; sin embargo algunos pueden caer por debajo de que, debido a circunstancias imprevistas. Peppers (2014). El consumidor debe ser consciente de que, para que puedan tomar la decisión más informada en la selección de su proveedor de servicios de almacenamiento en línea.

2.6.2. Apoyo Técnico

El soporte técnico y servicio al cliente están estrechamente relacionados; sin embargo, esta característica es una calificación de la capacidad técnica del personal de soporte técnico. Ciertamente, una cosa es tener un agente de soporte técnico amable, servicial y otra muy distinta es tener uno que es todo eso y realmente resuelve el problema rápidamente.

2.6.3. Espacio de Almacenamiento

Esta característica es una comparación de la cantidad de espacio de almacenamiento ofrecido por la empresa. El espacio de almacenamiento también es un factor en el costo de mantenimiento que significa tener un servidor propio. Peppers (2014).

2.6.4. Facilidad de Uso

Al probar los servicios de almacenamiento en la nube, el usuario debe comprobar lo fácil que es utilizar el servicio. Esto se compara con un punto de referencia de los usuarios en general y usuarios técnicamente competentes. Una buena calificación representa una mediana entre la facilidad de uso para los usuarios generales y técnicos.

2.6.5. Los Enfoques de la Computación en la Nube

La idea detrás de cualquier propuesta de la computación en nube es que usted sólo paga por lo que usa, la ampliación o hacia abajo de acuerdo con las necesidades del negocio. Los proveedores de apoyo *cloud computing* pueden interpretar esta declaración de manera diferente, proporcionando diferentes niveles de servicios para lograr este resultado. Los tres enfoques de la computación en la nube son Infraestructura como Servicio (IaaS), Software como Servicio (SaaS) y Plataforma como servicio (PaaS). Peppers (2014).

2.6.6. Infraestructura como Servicio

Modelo de distribución de infraestructura de computación como un servicio, normalmente mediante una plataforma de virtualización. En vez de adquirir servidores,

espacio en un centro de datos o equipamiento de redes, los clientes compran todos estos recursos a un proveedor de servicios externo. Una diferencia fundamental con el *hosting* virtual es que el aprovisionamiento de estos servicios se hace de manera integral a través de la *web*. Ejemplos: Amazon Web Services EC2 y GoGrid.

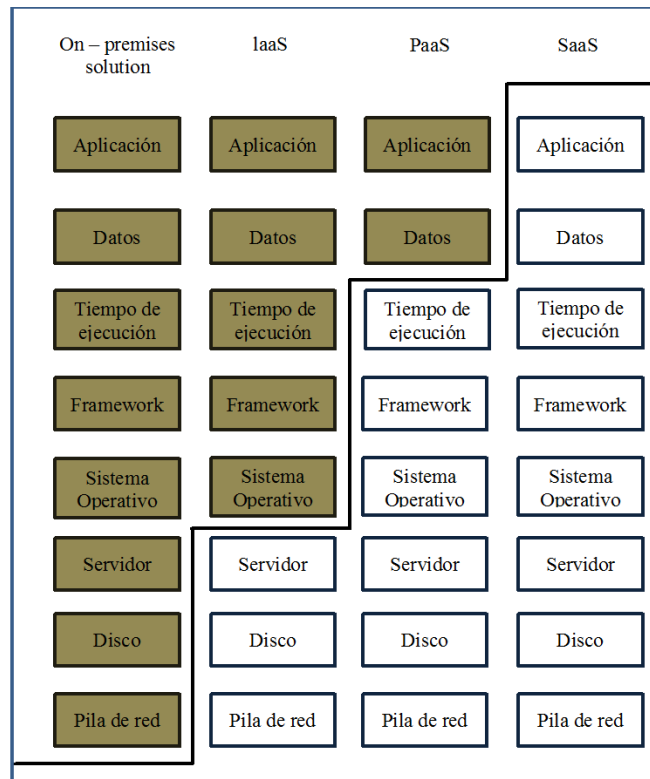


Figura 24: Tipo de servicios en la nube
Fuente: Marin (2012)

2.6.7. Software como Servicio

En español Software como Servicio. Modelo de distribución de software donde una empresa sirve el mantenimiento, soporte y operación que usará el cliente durante el tiempo que haya contratado el servicio. El cliente usará el sistema alojado por esa empresa, la cual mantendrá la información del cliente en sus sistemas y proveerá los recursos necesarios para explotar esa información. Ejemplos: Salesforce, Basecamp.

2.6.8. Plataforma como Servicio

Aunque suele identificarse como una evolución de *SaaS*, es más bien un modelo en el que se ofrece todo lo necesario para soportar el ciclo de vida completo de construcción y puesta en marcha de aplicaciones y servicios *web* completamente disponibles en la Internet. Peppers (2014). Otra característica importante es que no hay descarga de software que instalar en los equipos de los desarrolladores. *PaaS* ofrece múltiples servicios, pero todos provisionados como una solución integral en la *web*. Aunque algunos servicios de Amazon Web Services como SimpleDB y SQS yo los considero *PaaS*, esta afirmación puede ser discutida. Otro ejemplo es *Google App Engine*.

2.6.9. Backend como Servicio

También conocido como *Backend as a service (BaaS)* por su nombre en inglés, es un modelo para proporcionar a los desarrolladores *web* y de aplicaciones móviles una forma de vincular estas aplicaciones al almacenamiento en nube (cloud storage), servicios analíticos y/o otras características tales como la gestión de usuarios, la posibilidad de enviar *notificaciones push* y la integración con servicios de redes sociales Sommer (2012). Estos servicios se prestan a través de la utilización de *kits* personalizados de desarrollo de software (*SDK*) y las *interfaces* de programación de aplicaciones (*API*). *BaaS* es un modelo relativamente reciente en la computación en la nube, donde la mayoría de empresas datan del 2011 o posterior. Peppers (2014).

2.6.10. Propósito

Las aplicaciones *web* y móviles requieren un conjunto similar de características en su *backend*, incluyendo *notificaciones push*, analíticas, integración con redes sociales y almacenamiento en la nube. Cada uno de estos servicios tiene su propia *API* que debe de ser incorporada individualmente a una aplicación, proceso que puede ser largo y complicado para los desarrolladores de aplicaciones. Los proveedores *BaaS* forman un puente entre el *frontend* de una aplicación y varios servicios *backend* en la nube a través de una *API* y *SDK* unificada.

Proporcionar una forma coherente para gestionar los datos del *backend* significa para los desarrolladores no tener que desarrollar su propio *backend* para cada uno de los servicios que sus aplicaciones necesitan acceder, ahorrando potencialmente tiempo y dinero. Peppers (2014).

Aunque es similar a otras herramientas de desarrollo *web* de computación en la nube, como el software como servicio (*SaaS*), infraestructura como servicio (*IaaS*) y plataforma como servicio (*PaaS*), *BaaS* se distingue de estos otros servicios ya que está específicamente dirigido a las necesidades de la computación en la nube de la *web* y de los desarrolladores de aplicaciones móviles, proporcionando un medio unificado de conectar sus aplicaciones a servicios en la nube.

2.6.11. Modelo de Negocio

Los proveedores de *BaaS* generan ingresos de sus servicios de diversas maneras, a menudo utilizando un modelo *freemium*. Bajo este modelo, un cliente recibe un determinado número de usuarios o llamadas *API* al mes y paga una cuota por cada usuario o

llamada que excede este límite. Como alternativa, los clientes pueden pagar una cuota fija por un paquete que permita un mayor número de llamadas o usuarios activos por mes, Peppers (2014).

2.6.12. Windows Azure

Microsoft Azure (anteriormente Windows Azure y Azure Services Platform) es una plataforma ofrecida como servicio y alojada en los *Data Centers de Microsoft*. Anunciada en el *Professional Developers Conference de Microsoft (PDC)* del 2008 en su versión beta, pasó a ser un producto comercial el 1 de enero del 2010. Windows Azure es una plataforma general que tiene diferentes servicios para aplicaciones, desde servicios que alojan aplicaciones en alguno de los centros de procesamiento de datos de Microsoft para que se ejecute sobre su infraestructura (Cloud Computing) hasta servicios de comunicación segura.

Windows Azure utiliza un sistema operativo especializado, llamado de la misma forma, para correr sus Capas— un *cluster* localizado en los servidores de datos de Microsoft que se encargan de manejar los recursos almacenados y procesamiento para proveer los recursos(o una parte de ellos) para las aplicaciones que se ejecutan sobre Windows Azure. Peppers (2014).

Windows Azure se describe como una “capa en la nube” funcionando sobre un número de sistemas que utilizan Windows Server, estos funcionan bajo la versión 2008 de Windows Server y una versión *customizada* de *Hyper-V*, conocido como el Hipervisor de Windows Azure que provee la virtualización de los servicios. La capa controladora de Windows Azure se encarga de escalar y de manejar la confiabilidad del sistema evitando así que los servicios se detengan si alguno de los servidores de datos

de Microsoft tiene problemas y a su vez maneja la información de la aplicación *web* del usuario dando como ejemplo los recursos de la memoria o el balanceo del uso de esta.

Dentro de la plataforma, el servicio de Windows Azure es el encargado de proporcionar el alojamiento de las aplicaciones y el almacenamiento no relacional. Dichas aplicaciones deben funcionar sobre Windows Server 2008 R2. Pueden estar desarrolladas en .NET, PHP, C++, Ruby, Java. Además del servicio de ejecución, dispone de diferentes mecanismos de almacenamiento de datos: *tablas NoSQL*, *blobs*, *blobs para streaming*, colas de mensajes o *'drives'* NTFS para operaciones de lectura / escritura a disco.

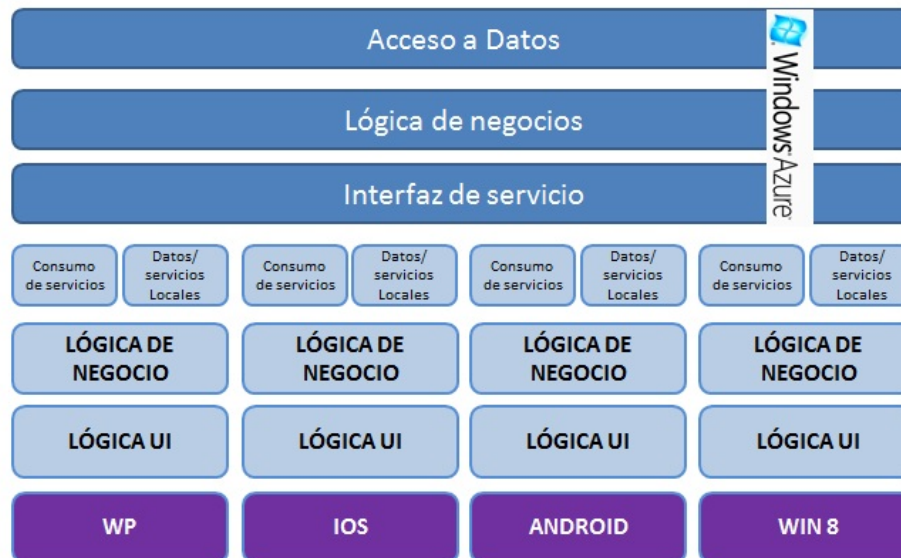


Figura 25: Arquitectura de una aplicación móvil multiplataforma utilizando Azure.
Fuente: Marin (2012)

a. Características de Windows Azure

- Proceso: el servicio de proceso de Windows Azure ejecuta aplicaciones basadas en Windows Server. Estas aplicaciones se pueden crear mediante *.NET Framework* en lenguajes como C# y Visual Basic, o implementar sin *.NET* en C++, Java y otros lenguajes. Peppers (2014).

- Almacenamiento: objetos binarios grandes (blobs) proporcionan colas para la comunicación entre los componentes de las aplicaciones de Windows Azure y ofrece un tipo de tablas con un lenguaje de consulta simple. Martínez (2011).
- Controlador de tejido: Windows Azure se ejecuta en un gran número de máquinas. El trabajo del controlador de tejido es combinar las máquinas en un solo centro de datos de Windows Azure formando un conjunto armónico. Los servicios de proceso y almacenamiento de Windows Azure se implementan encima de toda esta eficacia de procesamiento. Martínez (2011).
- *Red de entrega de contenido (CDN)*: el almacenamiento en caché de los datos a los que se accede frecuentemente cerca de sus usuarios agiliza el acceso a esos datos.
- Connect: organizaciones interactúan con aplicaciones en la nube como si estuvieran dentro del propio *firewall* de la organización.

b. Componentes Azure Platform

- Windows Azure Compute es una plataforma para hospedar y administrar aplicaciones en los centros de datos de Microsoft. Una aplicación de Windows Azure consta de uno o varios componentes denominados ‘roles.’ Los roles pueden ser de tres tipos: rol web, rol de trabajo y rol de máquina virtual (VM).
- *Windows Azure Storage* tiene servicios de básicos como parte de la cuenta de almacenamiento de Windows Azure. Los *blobs*, tablas y colas están accesibles a aplicaciones o instancias de aplicaciones simultáneamente. Marin (2012).

- Microsoft SQL Azure es un servicio de base de datos en la nube basado en las tecnologías de SQL Server. Los servicios de SQL Azure incluyen: Base de datos SQL Azure, SQL Azure Reporting y SQL Azure Data Sync Aspectos destacados de la base de datos de SQL Azure.
- *Content Delivery Network (CDN)* de Windows Azure coloca copias de los datos cerca de donde estos se encuentran. La *CDN* de Windows Azure entrega actualmente muchos productos de Microsoft, como Windows Update, vídeos de Zune y Bing Maps, que los clientes conocen y usan todos los días. Gracias a la incorporación de la *CDN* a los servicios de Windows Azure, ahora esta red a gran escala está disponible a todos los usuarios de Windows Azure. Marin (2012).
- Azure AppFabric El servicio de AppFabric (en fase beta se llamaba .NET Services) ofrece diferentes servicios para aplicaciones. Los servicios de autenticación, autorización y mensajería permiten la comunicación segura entre aplicaciones y servicios desplegados tanto en la nube y en local. Los diferentes servicios que ofrece el servicio de AppFabric se pueden dividir en dos grandes bloques: AppFabric Service Bus y AppFabric Access Control.
- *Azure Market Place* es un mercado en línea global compartir, comprar y vender aplicaciones *SaaS* completas y conjuntos de datos. La sección de datos de Windows *Azure Marketplace* incluye datos, imágenes y servicios *Web* en tiempo real de proveedores de datos comerciales, líderes en el sector y orígenes de datos públicos acreditados. Singh (2012).
- Azure Virtual Network es una serie de funciones de red. Windows Azure Connect es la primera característica de *Azure Virtual Network* que configura la conectividad

de red basada en *IP* entre recursos locales y de Windows Azure. Windows Azure Traffic Manager equilibra la carga del tráfico en servicios hospedados.

c. Beneficios de Windows Azure

- Ejecutar procesos genéricos en la nube.
- Crear, modificar y distribuir aplicaciones escalables con un mínimo de recursos internos.
- Realizar almacenamiento de alto volumen, procesamiento de lotes y cómputos intensos o de alto volumen.
- Crear, evaluar, depurar y distribuir servicios *web* con rapidez y de forma accesible.
- Llevar sus ideas al mercado con mayor rapidez y paga cuando lo obtiene.
- Reduce costes de generación y extensión de recursos internos.
- Reduce el esfuerzo y los costes de administración de *TI* Responde con rapidez a los cambios de las necesidades de su empresa y sus clientes.
- Amplía y reduce sus recursos de *TI* en función de sus necesidades.
- Consume recursos de informática SOLO cuando surgen la necesidad.
- Se enfoca menos en administrar restricciones y recursos operativos.
- Elimina la necesidad de administrar hardware.
- Utiliza sus actuales habilidades de desarrollo para crear aplicaciones en la nube.

d. Microsoft Azure Storage

La computación en la nube permite nuevos escenarios para las aplicaciones que requieren de almacenamiento escalable, duradero y de alta disponibilidad para sus datos, que es exactamente el por qué Microsoft desarrolló Azure Storage. Scott (2013). Además de hacer posible que los desarrolladores puedan crear aplicaciones de gran escala para apoyar a estos nuevos escenarios, *Azure Storage* también proporciona la base de almacenamiento para equipos virtuales de Azure, un testimonio más de su solidez. Singh (2012).

Azure Storage es masivamente escalable, por lo que puede almacenar y procesar cientos de terabytes de datos para apoyar los escenarios de grandes datos requeridos por el análisis científico, financiero, y aplicaciones multimedia. Scott (2013). O usted puede almacenar pequeñas cantidades de datos necesarios para un sitio *web* de pequeñas empresas. Donde quiera que caigan sus necesidades, usted sólo paga por la información que almacena. *Azure Storage* almacena en la actualidad decenas de miles de millones de objetos únicos de los clientes, y maneja millones de peticiones por segundo en promedio.

Azure Storage es elástica, por lo que puede diseñar aplicaciones para una gran audiencia global, y escalarlas según sea necesario - tanto en términos de la cantidad de datos almacenados y el número de solicitudes formuladas en su contra. Usted sólo paga por lo que usa, y sólo cuando lo utiliza.

Azure Storage utiliza un sistema de auto-partición que da forma automática y equilibra la carga de los datos en función del tráfico. Esto significa que a medida que las demandas de la aplicación crezcan, *Azure Storage* asigna automáticamente los recursos adecuados para cumplir con ellos.

Azure Storage se puede acceder desde cualquier parte del mundo, desde cualquier tipo de aplicación, ya sea que se ejecuta en la nube, en el escritorio, en un servidor en las instalaciones, o en un dispositivo móvil o *tablet*. Usted puede utilizar *Azure Storage* en escenarios móviles donde la aplicación almacena un subconjunto de los datos del dispositivo y los sincroniza con un conjunto completo de los datos almacenados en la nube. Singh (2012).

Azure Storage apoya a los clientes que utilizan un conjunto diverso de sistemas operativos (incluyendo Windows y Linux) y una variedad de lenguajes de programación (incluidas. NET, Java y C++) para el desarrollo conveniente. *Azure Storage* también expone sus recursos de datos a través de las API REST, que están disponibles para cualquier cliente capaz de enviar y recibir datos a través de *HTTP / HTTPS*.

Introducción a los servicios Azure Storage.- Los servicios de Azure de almacenamiento son el almacenamiento Blob, almacenamiento en Tablas (Table Storage), almacenamiento de colas (Queue Storage), y el almacenamiento de archivos (File Storage).

- **Blob Storage** Un blob puede ser cualquier tipo de texto o datos binarios, como un documento, archivo multimedia, o el instalador de la aplicación.
- **Table Storage:** Almacenamiento de la tabla es un almacén de datos clave-atributo NoSQL, lo que permite el rápido desarrollo y el acceso rápido a grandes cantidades de datos.
- **Queue Storage:** proporciona mensajería confiable para el procesamiento de flujo de trabajo y para la comunicación entre los componentes de servicios en la nube.

- **File Storage:** ofrece almacenamiento compartido para las aplicaciones heredadas que utilizan el protocolo estándar *SMB 2.1*. Máquinas virtuales y Azure servicios en la nube pueden compartir datos de archivos a través de componentes de la aplicación a través de acciones montadas, y las aplicaciones *on-premise* pueden acceder a los datos de archivo en un recurso compartido a través de la *API* del servicio de archivos.

Blob, *Table*, y *Queue* se incluyen en cada cuenta de almacenamiento, mientras que el almacenamiento de archivos está disponible por solicitud a través de la página Azure Vista previa.

La cuenta de almacenamiento es un espacio de nombres único que te da acceso a Azure Storage. Cada cuenta de almacenamiento puede contener hasta 500 TB de blob combinado, la cola, tablas y los datos de archivo. Scott (2013).

La imagen de abajo muestra las relaciones entre los recursos de almacenamiento de Azure:

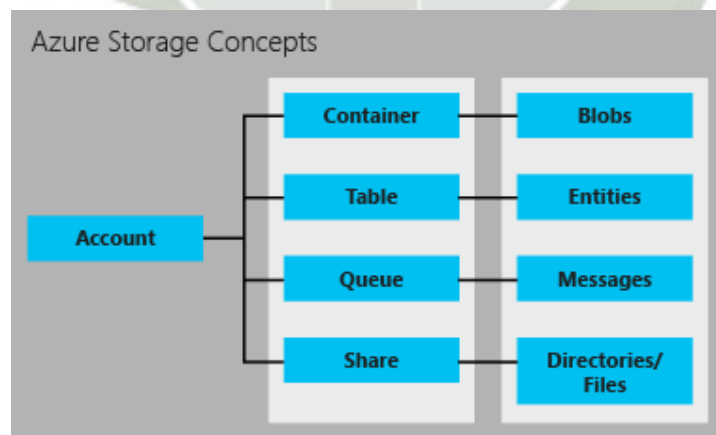


Figura 26: Servicios de Windows Azure Storage
Fuente: Marin (2012)

e. Windows Azure Mobile Services

Mobile Services es un servicio de Windows Azure diseñado para facilitar la creación de aplicaciones móviles altamente funcionales. *Mobile Services* reúne un conjunto de servicios Azure que habilitan diferentes posibilidades para las aplicaciones como: almacenamiento de datos, implementación de servicios y lógica del negocio (*backend*). Singh (2012).

Mobile Services provee las siguientes características para apoyar el *backend* de una aplicación móvil:

- Manejo simple de tablas para almacenamiento.
- Integración con servicios de notificación para entrega de notificaciones *push* a la aplicación móvil.
- Integración con proveedores de autenticación bien renombrados.
- Control granular para el acceso a tablas.
- Personalización de la lógica de negocios en el lado del servidor.
- Integración con otros servicios en la nube.
- Soporta la habilidad de escalar la infraestructura.
- Monitoreo de servicios y *logs*.

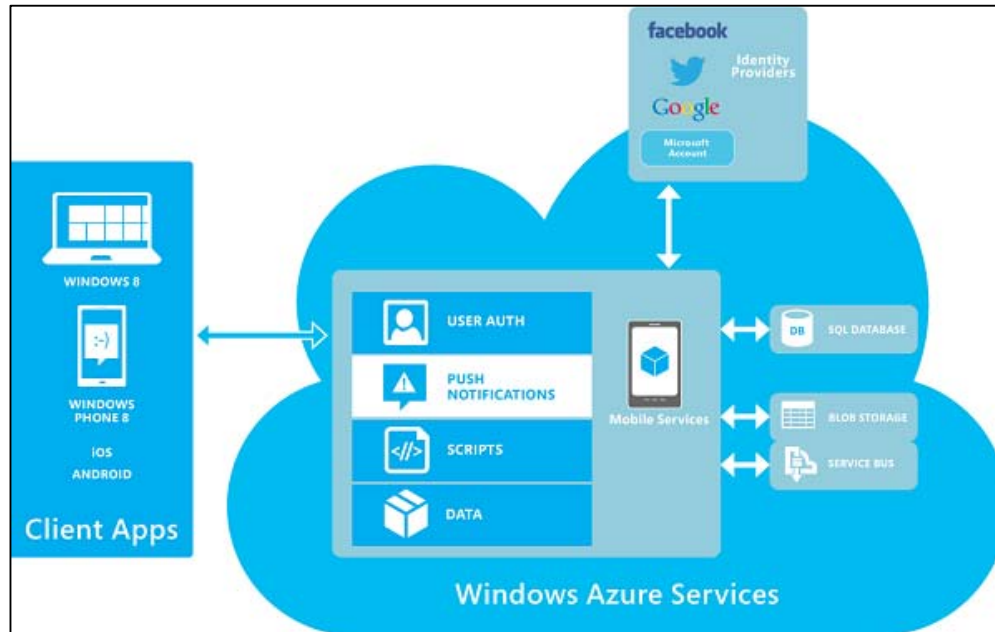


Figura 27: Arquitectura de Windows Azure Services

Fuente: <https://azure.microsoft.com/es-es/>

f. Escenarios reales con los servicios móviles de Windows Azure

Notificaciones Push

Notificaciones Push son un componente vital de aplicaciones conectadas. Permiten a los servicios de *back-end* para enviar actualizaciones frecuentes y oportunas a los usuarios y los dispositivos específicos que luego pueden aparecer como cambios de baldosas, notificaciones tostadas, actualizaciones insignia, y más. Como tal, notificaciones push son quizás el más poderoso mecanismo de participación del cliente a disposición de los desarrolladores de aplicaciones a través de teléfonos, tabletas y ordenadores.

La razón principal de esto es que las notificaciones push se entregan a los dispositivos por los servicios específicos de la plataforma. Por ejemplo, usted tiene que utilizar el *servicio de notificación de Windows (WNS)* para empujar las notificaciones a

Windows Store aplicaciones, *Push Notification Service de Microsoft (MPNS)* para empujar a Windows Phone, el servicio *Apple Push Notification (APN)* para iOS y *Google Cloud Mensajería (GCM)* para Android. Scott (2013).

Hub de Notificaciones

En este momento existe una vista previa del servicio *Hubs* de notificaciones con las siguientes capacidades:

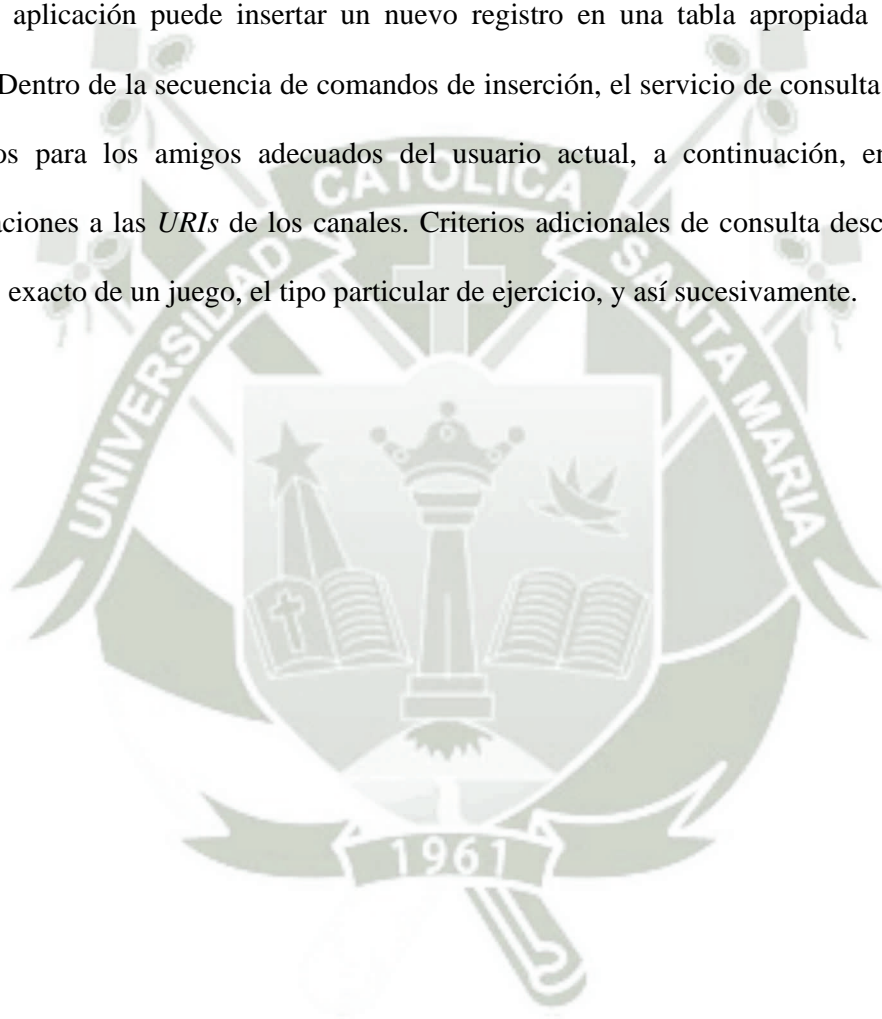
Soporte de notificaciones *Push* multiplataforma: Los *Hubs* de notificación proporcionan una *API* común para enviar notificaciones *push* a múltiples plataformas de dispositivos. Su aplicación puede enviar notificaciones en formatos específicos de la plataforma o de forma independiente de la plataforma. A partir de enero de 2013, de notificación *Hubs* son capaces de notificaciones *push* a Windows 8 aplicaciones y aplicaciones de *iOS*. Soporte para *Android* y *Windows Phone* se añadirán en breve. Singh (2012).

Escalabilidad: El *Hub* de notificación permite llegar a millones de dispositivos sin tener que rediseñar o fragmentar su aplicación. El mecanismo *pub/sub routing* le permite transmitir las notificaciones de una manera muy eficiente. Esto hace que sea increíblemente fácil para enrutar y entregar mensajes de notificación a millones de usuarios sin tener que construir su propia infraestructura de enrutamiento.

Utilizable desde el *Backend* de cualquier Aplicación: Los *Hubs* de notificación se pueden integrar fácilmente en cualquier aplicación de servidor back-end. Se trabajará sin problemas con aplicaciones integradas con *Windows Azure Mobile Services*. También puede ser utilizado por aplicaciones de servidores alojados en *IaaS* máquinas

virtuales (ya sea Windows o Linux), servicios en la nube o sitios *web*. Esto hace que sea fácil para que usted tome ventaja de ello de inmediato, sin tener que cambiar el resto de la arquitectura de su aplicación *backend*. Scott (2013).

El uso de las redes sociales: Las aplicaciones pueden utilizar la red social del usuario para implementar características como la emisión de los retos a los amigos. Para ello, la aplicación puede insertar un nuevo registro en una tabla apropiada servicio móvil. Dentro de la secuencia de comandos de inserción, el servicio de consulta su base de datos para los amigos adecuados del usuario actual, a continuación, envía las notificaciones a las *URIs* de los canales. Criterios adicionales de consulta describen el aspecto exacto de un juego, el tipo particular de ejercicio, y así sucesivamente.



CAPÍTULO III: DESARROLLO DEL APLICATIVO MÓVIL

3.1. Metodología de Desarrollo

En este proyecto se ha utilizado una metodología en cascada centrada en la planificación del proyecto, ya que se cuenta con una visión clara del proyecto. Para este desarrollo se consideran cuatro etapas, cada una depende de la conclusión de sus antecesoras. Las fases consideradas en el desarrollo, como se muestra en el siguiente gráfico, son las siguientes:

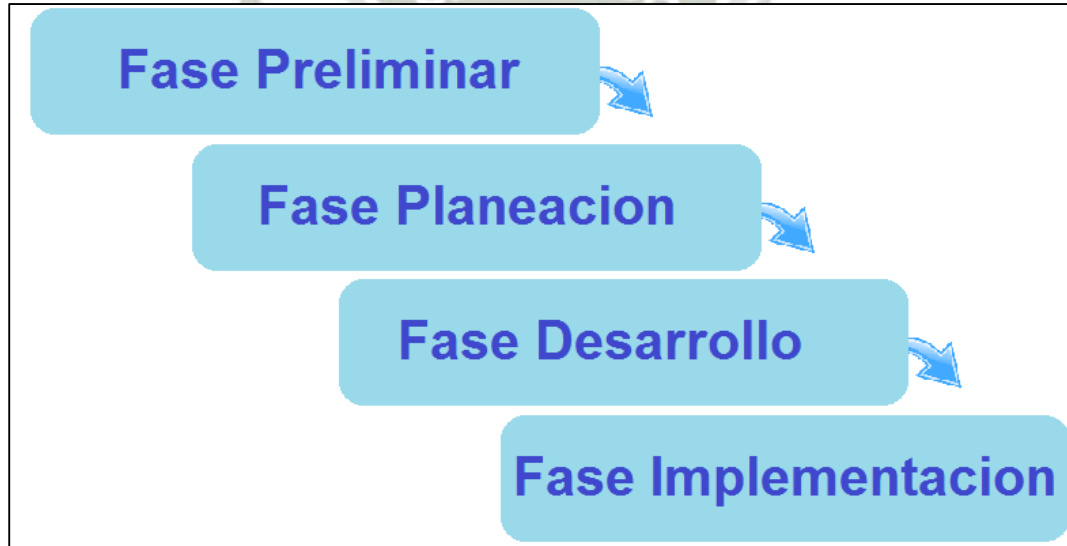


Figura 28: Mapa conceptual de la metodología de desarrollo del aplicativo móvil
Fuente: Elaboración propia

- Fase preliminar, se identifica la necesidad de crear una aplicación para dispositivos móviles, quiénes serán los usuarios, la plataforma para la que se desarrollará y el alcance de la aplicación. En esta etapa es importante considerar el dispositivo al que está destinada la aplicación, es decir, capacidad de memoria, generación del equipo, peso de la aplicación, conectividad y tamaño del dispositivo.

- Fase de planeación, se realizan la indagaciones sobre la aplicación, de tal manera que sea posible recopilar información relevante que permita delimitar el alcance de la aplicación y diseñar un bosquejo de la misma, este bosquejo puede elaborarse en papel o cualquier aplicación que nos permita tener la idea del diseño de las pantallas. El objetivo es conocer las rutas de navegación que tendrá la aplicación, la cual debe ser simple y con el menor número de gestos táctiles (tales como Tap, Pinch, Swipe) posible para acceder a la información.
- Fase de desarrollo, una vez elaborado y aprobado el bosquejo, es necesario centrarse en el desarrollo de la aplicación, en cada una de las plataformas, se debe tomar en cuenta las especificaciones de diseño con respecto a la interfaz de usuario. Se sugirió agregar una etapa de pruebas locales de la aplicación antes de continuar con la implementación.
- Fase de implementación, en esta etapa se pone a disposición de los usuarios la aplicación realizada, publicándola en los medios que proporcionan los diferentes proveedores, en este caso las tiendas virtuales de capa plataforma móvil.

3.2. Análisis de Requerimientos

Debido a la importancia de contar con una aplicación móvil de promoción turística del destino Arequipa que este a la vanguardia en tecnología y considerando el aumento de usuarios de dispositivos móviles; se decidió desarrollar una aplicación que sustituya al tradicional libro de mano, que carga siempre un turista; a continuación describimos los requerimientos obtenidos a partir de un análisis de un ejemplar físico de este libro de mano del turista y de un estudio sobre el estado del arte de aplicaciones similares

aplicadas al rubro de promoción turística, obteniendo los siguientes requerimientos funcionales y no funcionales:

3.2.1. Requerimientos Funcionales

Tabla 7

Requerimiento funcional

RI-001	Un usuario puede ser de un solo tipo.
RI-002	Un usuario debe poder crear una cuenta e iniciar sesión con su nombre usuario y contraseña.
RI-003	Un usuario con sesión iniciada podrá acceder a la información completa de la guía.
RI-004	Un usuario con sesión iniciada podrá escribir un comentario sobre el contenido de la guía.
RI-005	Un usuario con sesión iniciada podrá valorar mediante un puntaje el contenido de la guía.
RI-006	Un usuario con sesión iniciada podrá publicar un comentario sobre la guía en las redes sociales.
RI-007	Un usuario podrá navegar mediante un mapa la ubicación de los destinos turísticos.
RI-008	La información deberá estar ordenada jerárquicamente para la facilidad de exploración.

3.2.2. Requerimientos no Funcionales

Tabla 8

Requerimiento no funcional

En esta parte se define los atributos de calidad que debe tener la aplicación.

RNF-001	Accesibilidad al sistema: Este o no el dispositivo móvil conectado a una red de internet.
RNF-002	Portabilidad del sistema: Soportar plataformas móviles Android, iOS y Windows.
RNF-003	Interfaz de usuario amigable: El uso del aplicativo debe ser intuitivo y la curva de aprendizaje de uso mínima.
RNF-004	Información confiable: La información que se brinde a través del aplicativo debe ser correcta.
RNF-005	Escalabilidad del sistema: Permitir que el número de usuarios y cantidad de información sea escalable.
RNF-006	Flexibilidad del sistema y recuperación ante fallos, ya sean de conectividad o de energía.

3.3. Diagrama de Casos de Uso

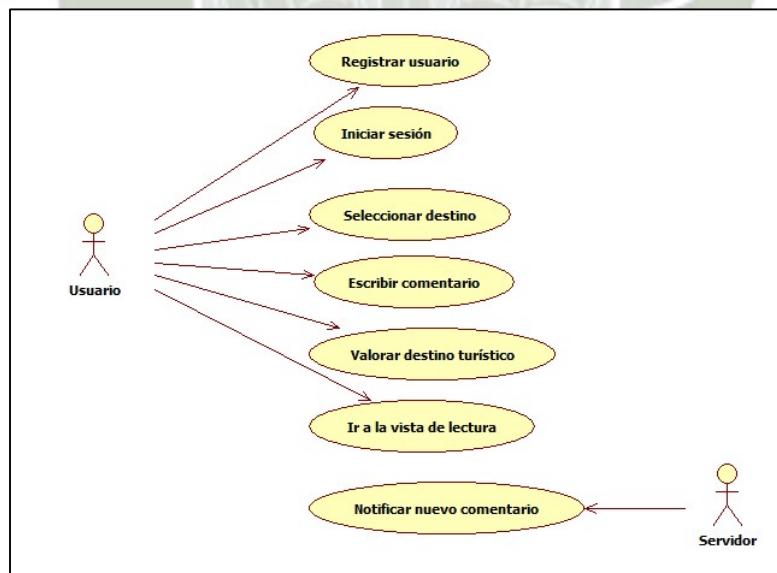


Figura 29: Diagrama de casos de uso.

Fuente: Elaboración propia

3.4. Especificación de Casos de Uso

3.4.1. Caso de Uso Registrar Usuario.

Tabla 9

Caso de uso registrar usuario

Nombre	Registrar usuario
Descripción	El usuario ingresa sus datos para poder registrarse en la aplicación, esto permitirá que pueda acceder a las características completas de la aplicación.
Requisitos asociados	RN-001
Pre-condición	El registro de usuario se realiza mediante el sistema de autenticación de Facebook.
Participantes	El actor usuario y el actor servidor
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario ingresa la información necesaria para el registro. 2. El sistema verifica que los datos ingresados no corresponde a un usuario ya registrado. 3. Se almacena los datos del usuario en la base dataos en la nube. 4. El sistema navega hacia la vista principal.
Secuencia alternativa	No Existe
Post- condición	El usuario está registrado en la aplicación
Excepciones	<ol style="list-style-type: none"> 1. Si el usuario no ingresa toda la información necesaria para el registro, el sistema notificara al usuario el error en esta acción. 2. Si el usuario intenta registrarse con información no válida para el registro.
Importancia	Vital
Comentarios	La navegación a la vista de registro se realiza desde la vista inicio de sesión mediante los servicios de autenticación de Facebook.

3.4.2. Caso de Uso Iniciar Sesión

Tabla 10

Caso de uso iniciar sesión

Nombre	Iniciar sesión
Descripción	El usuario ingresa los datos necesarios para poder iniciar sesión en el sistema.
Requisitos asociados	RN-001, RN-002
Pre-condición	El inicio de sesión se realiza mediante el sistema de autenticación de Facebook.
Participantes	El actor usuario.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario ingresa su ID y contraseña 2. El usuario presiona el botón de acceso. 3. El sistema de autenticación verifica y valida los datos ingresados. 4. El sistema navega hacia la vista principal
Secuencia alternativa	No existe
Post- condición	El usuario esta con sesión iniciada y se encuentra en la vista principal
Excepciones	Si los datos son incorrectos no se iniciara sesión.
Importancia	Vital
Comentarios	Si el usuario se encuentra registrado en el sistema, no se guardara en la base datos los nuevos datos obtenidos.

3.4.3. Caso de Uso Seleccionar Destino

Tabla 11

Caso de uso seleccionar destino

Nombre	Seleccionar destino
Descripción	El usuario selecciona el destino el cual desea explorar
Requisitos asociados	RN-001, RN-002, RN-004
Pre-condición	El usuario debe encontrarse en la vista principal
Participantes	El actor usuario
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario realiza la búsqueda del destino que quiere explorar. 2. El usuario presiona sobre el destino que quiere explorar. 3. El sistema navega hacia la vista general del destino seleccionado.
Secuencia alternativa	<ol style="list-style-type: none"> 1. El usuario navega hacia la vista general de destino escogiendo cualquier opción. 2. En la vista general de destino el usuario interactúa con la vista principal hasta llegar a la vista del destino que desea ver.
Post- condición	El usuario se encuentra en la vista general de destino que seleccionó.
Excepciones	Una vista del sistema no cargo completamente por temas de capacidad de hardware.
Importancia	Media
Comentarios	Luego de tener un registro usuario valido este deberá iniciar sesión en el aplicativo, para acceder a toda la funcionalidad del aplicativo móvil.

3.4.4. Caso de Uso Escribir Comentario.

Tabla 12

Caso de uso escribir comentario

Nombre	Escribir comentario
Descripción	El usuario tendrá la posibilidad de interactuar con el sistema haciendo comentarios en la vista que se encuentre.
Requisitos asociados	RN-004, RN-005
Pre-condición	El usuario debe encontrarse en la vista de lectura
Participantes	El actor usuario
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario presiona el botón de comentario. 2. El sistema desglosa la barra de comentarios. 3. El usuario escribe su comentario. 4. El usuario presiona el botón publicar comentario. 5. El sistema publica el comentario acompañado de la imagen en la vista que se encuentra.
Secuencia alternativa	No existe.
Post- condición	El comentario se encuentra publicado en las redes sociales, y almacenado en la base de datos.
Excepciones	Un error en la conexión de internet puede ocasionar que el comentario no sea publicado
Importancia	Media
Comentarios	Además de publicar un comentario, usuario recibirá en tiempo real notificaciones de los comentarios realizados por otros usuarios, siempre y cuando el dispositivo móvil se encuentre conectado a una red de internet.

3.4.5. Caso de Uso Valorar Destino Turístico.

Tabla 13

Caso de uso valorar destino turístico

Nombre	Valorar destino turístico
Descripción	El usuario dará una valoración al destino turístico en la aplicación.
Requisitos asociados	RN-006
Pre-condición	El usuario debe encontrarse en la vista principal de destino.
Participantes	El actor usuario
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario selecciona la valoración que le dará al destino en la aplicación. 2. El aplicativo registra la valoración que selecciona el usuario. 3. El aplicativo calcula el promedio de la aplicación en su base datos.
Secuencia alternativa	No existe.
Post- condición	La valoración del usuario estará registrada en la base de datos.
Excepciones	Un error en la conexión de internet puede ocasionar que no se registre la valoración dada por el usuario.
Importancia	Media
Comentarios	Esta funcionalidad estará disponible solo en cada una de las vistas correspondientes a un destino determinado.

3.4.6. Caso de Uso Notificar Nuevo Comentario

Tabla 14

Caso de uso notificar nuevo comentario

Nombre:	Notificar nuevo comentario
Descripción	El usuario recibe las notificaciones de los comentarios realizados por otros usuarios.
Requisitos asociados	RN-006, RN-008
Pre-condición	El usuario debe encontrarse con sesión iniciada en el sistema.
Participantes	El actor servidor
Secuencia normal	<ol style="list-style-type: none"> 1. El servidor recibe un nuevo comentario en la base de datos. 2. El servidor envía una notificación de tipo Push a los usuarios que tengan la aplicación activa.
Secuencia alternativa	No existe.
Post- condición	El usuario recibirá las notificaciones de actualización de comentarios.
Excepciones	Si el usuario no se encuentra conectado a internet, no recibirá la notificación en tiempo real.
Importancia	Media
Comentarios	Las notificaciones son enviadas a todos los despóticos, la forma de visualización o no de los mismos dependerá de la configuración que realice cada usuario en su dispositivo.

3.5. Diagrama de Clases

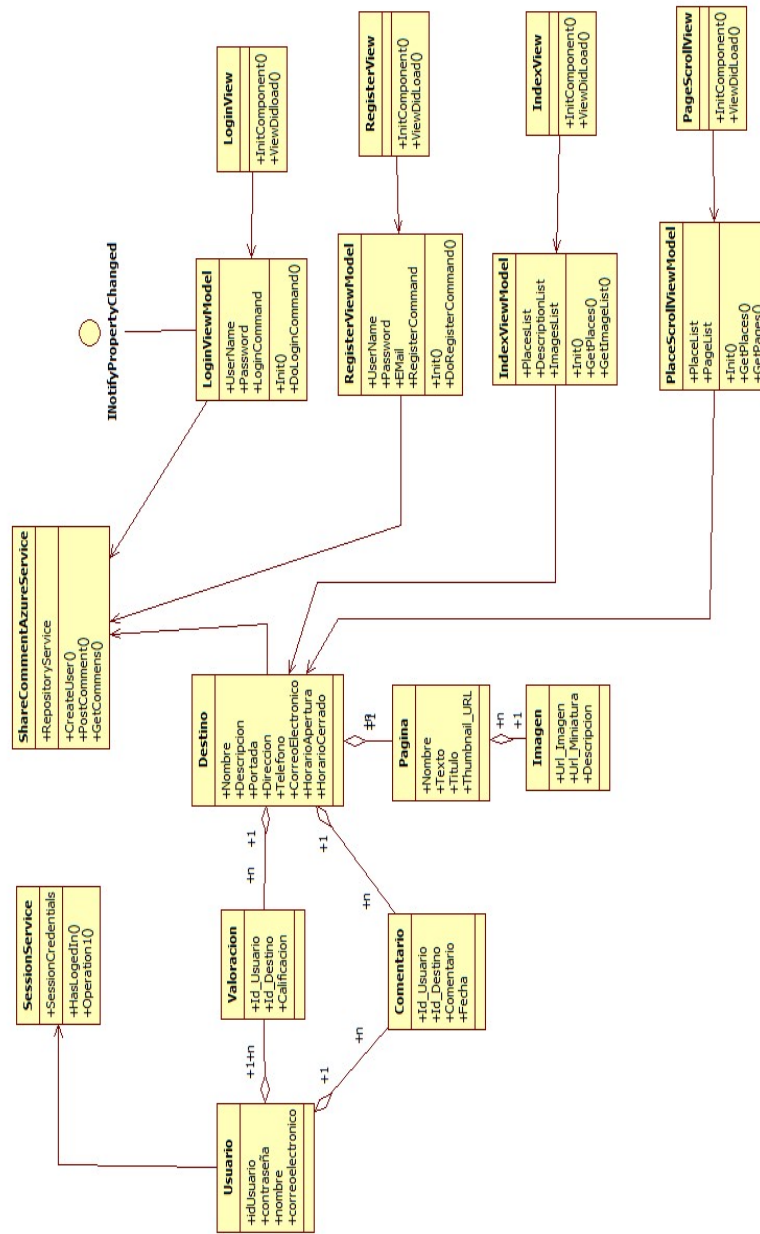


Figura 30: Diagrama de clases
Fuente: Elaboración propia.

3.4.7. Caso de Uso Ir a la Vista de Lectura

Tabla 15

Caso de uso ir a la vista de lectura

Nombre	Ir a la vista de lectura
Descripción	El usuario ingresa a la vista de lectura.
Requisitos asociados	RN-007, RN-008
Pre-condición	El usuario debe encontrarse en la vista general de destino
Participantes	El actor usuario
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario selecciona la miniatura de la página que desea leer e interactúa mediante un gesto, el cual puede ser un gesto Tap o un gesto Pinch. 2. El sistema mostrara la vista de lectura.
Secuencia alternativa	No existe.
Post- condición	El usuario se encontrara en la vista de lectura
Excepciones	Problemas con el hardware grafico del dispositivo móvil.
Importancia	Media
Comentarios	El idioma disponible para la lectura es el español.

3.6. Wireframes

A continuación se muestra los *wireframes* que son la base del aplicativo móvil, esto es tomado como referencia al momento de implementar las vistas, las cuales tienen un acabado final mucho más complejo y vistoso, ya que esta representación es muy limitada en cuanto a temas de diseño. Como se observa a continuación el diseño para tabletas es funcionalmente diferente al que está orientado a teléfonos inteligentes.

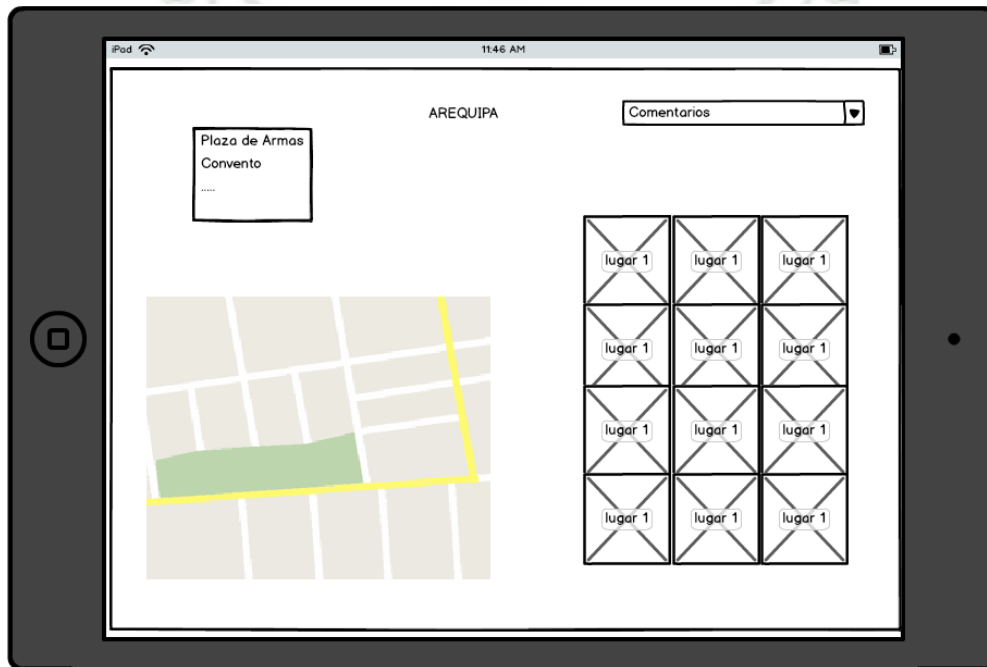


Figura 31: Wireframe de la vista principal
Fuente: Elaboración propia

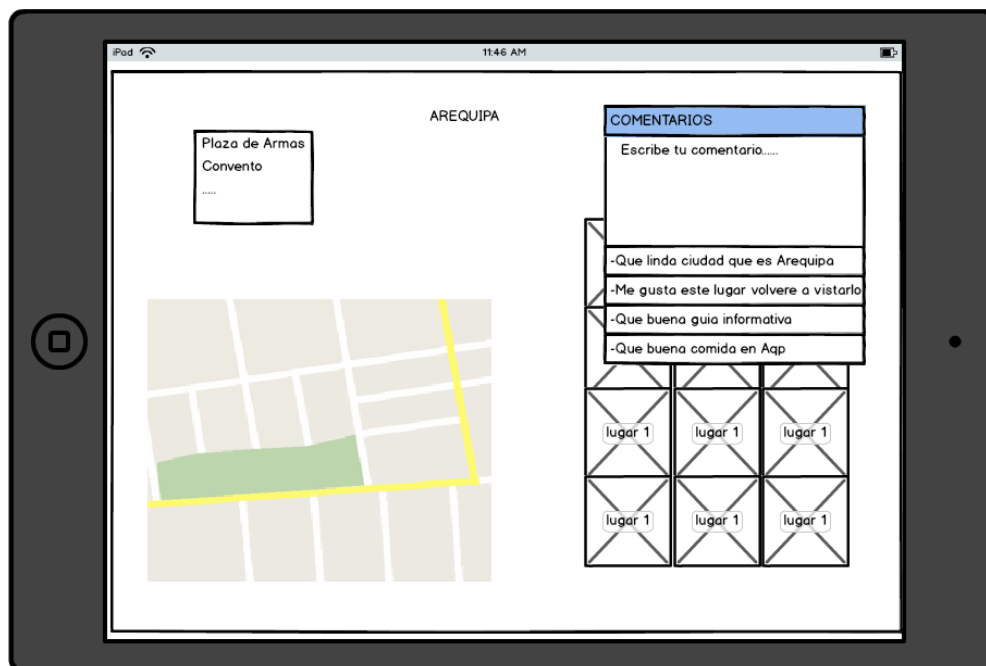


Figura 32: Wireframe de la vista de comentarios
Fuente: Elaboración propia



Figura 33: Wireframe de la vista de mapas
Fuente: Elaboración propia



Figura 34: Wireframe de la vista principal de comentarios

Fuente: Elaboración propia

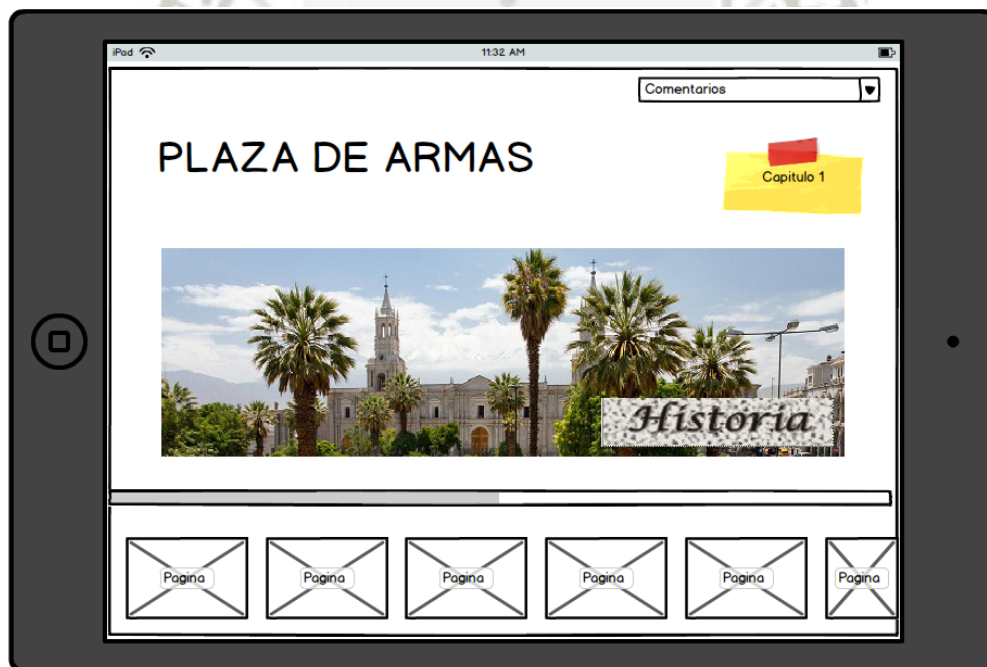


Figura 35: Wireframe de la vista de lectura

Fuente: Elaboración propia

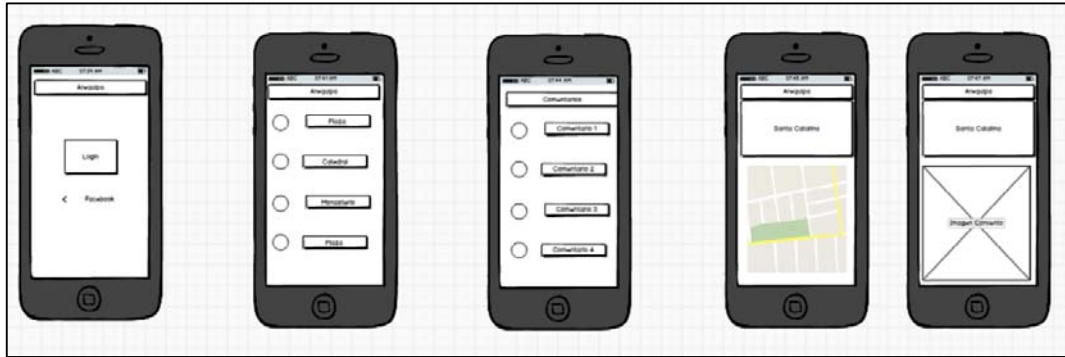


Figura 36: Equivalentes para telefonos
Fuente: Elaboración propia

3.7. Arquitectura

La aplicación a desarrollar tiene una arquitectura basada en niveles, se va a emplear el patrón *Model View ViewModel (MVVM)*, consta de tres partes. La razón más importante por la cual se utilizará el patrón *MVVM* es por la infraestructura de enlace de datos (*Data Binding*). Al enlazar los datos de una vista al modelo de vista, se consigue bajo acoplamiento de ambas capas, lo cual permite actualizar o dar mantenimiento a las diferentes capas de la arquitectura sin que afecte los cambios individuales por cada nivel; haciendo que la factibilidad de características importantes como mantenibilidad, extensibilidad y capacidad de prueba sean mayores frente a otras arquitecturas similares.

3.7.1. Aplicación del Patrón de Diseño MVVM

a. Modelo

El modelo representa los datos o información con la que trabajamos y por eso suele llamársele el objeto del dominio. Un ejemplo de modelo puede ser un contacto (con su nombre, número de teléfono, dirección y demás) o la descripción de un punto de publicación para un medio audiovisual transmitido en vivo.

La capa de modelo incluye todo el código que implementa la lógica principal de la aplicación y define los tipos requeridos para modelar el dominio de la aplicación. Esta capa es completamente independiente de las capas de vista y modelo de vista.

La clave para recordar el modelo es que contiene la información, pero no las acciones o servicios que la manipulan. No es responsable de darle forma para que se vea bien en la pantalla, o de obtener una lista de elementos de un servidor remoto (de hecho, en tal lista cada elemento sería a su vez un modelo). La lógica de la aplicación o “reglas empresariales” son generalmente mantenidas en clases separadas del modelo y actúan en él. Aunque no siempre es cierto, a veces el modelo puede validar la información.

b. Vista

La capa de vista define la interfaz de usuario que utiliza marcado declarativo. El marcado de enlace de datos define la conexión entre componentes específicos de la interfaz de usuario y diversos miembros de modelo de vista (y, en ocasiones, de modelo).

En el patrón MVVM la vista es activa. A diferencia de una vista pasiva sin conocimiento del modelo, y bajo el manejo total de un controlador o presentador, la vista en MVVM contiene comportamientos, eventos y enlaces a datos que, hasta cierto punto, necesitan saber sobre el modelo subyacente y el modelo de vista. Aunque tales eventos y comportamientos son asociados a propiedades, métodos y comandos, la vista es aún responsable de manejar sus propios eventos y no pasa esta tarea al modelo de vista.

c. Modelo de Vista

La capa de modelo de vista proporciona destinos de enlace de datos para la vista. En muchos casos, el modelo de vista expone el modelo directamente o proporciona miembros que encapsulan miembros de modelo específicos. El modelo de vista también puede definir miembros para realizar un seguimiento de los datos que son relevantes para la interfaz de usuario pero no para el modelo, como el orden de visualización de una lista de elementos.

El modelo de vista también hace disponibles métodos, comandos y otros puntos de acceso que ayudan a mantener el estado de la vista, manipular el modelo en respuesta a acciones de la vista y disparar eventos en la misma.

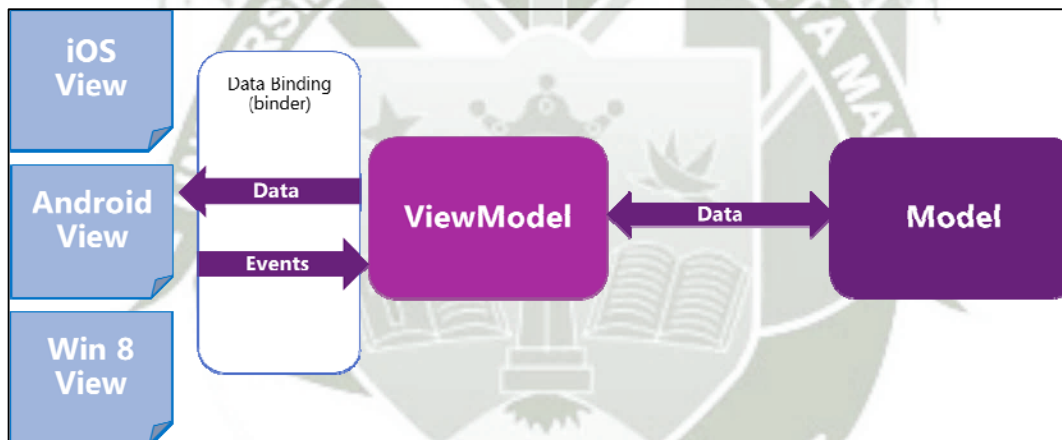


Figura 37: Arquitectura del patrón Model View ViewModel

Fuente: Marin (2012)

3.7.2. Estrategia de Organización del Proyecto

a. Librerías de clase portables o PCL

Las librerías de clase portables o *PCL*, han estado disponibles desde la versión .NET 4.0 y ahora han sido adoptadas por Xamarin para su aplicación a desarrollo de aplicaciones móviles multiplataforma. Se pueden utilizar para crear librerías dinámicas

portables que se pueden utilizar en varias plataformas, incluyendo Windows 8.1, *Windows Phone* 8.1, iOS y *Android*.

En el patrón *MVVM* la capa de Modelo y el Modelo de Vista son completamente desconocidas para la capa de Vista (interfaz de usuario). No entienden nada de la vista, así como para el control que está utilizando la vista o lo cual es la naturaleza de la vista y así sucesivamente. Aquí la vista sólo entiende el Miembro Vista-Modelo a través del Contexto de datos de las propiedades de dependencia.

Cuando se crea un proyecto de aplicación "regular" o un Proyecto de Biblioteca, el archivo *DLL* como resultado se limita a trabajar en la plataforma específica que se crea para. Esto le impide escribir una asamblea para una aplicación de *Windows Phone*, y luego volver a usarlo en *Xamarin.iOS* y *Xamarin.Android*.

Al crear una biblioteca de clases portable, sin embargo, se puede elegir una combinación de plataformas que desea que su código se ejecute. En las opciones de compatibilidad que realice al crear una biblioteca de clases portable se traducen en un identificador de "Perfil", que describe que las plataformas de los soportes de la biblioteca.

3.7.3. Ventajas y Desventajas de Utilizar una Librería de Clases Portable

a. Ventajas

- Intercambio centralizado de código - de escritura y código de prueba en un solo proyecto que puede ser consumido por otras bibliotecas o aplicaciones.

- Operaciones de refactorización afectarán a todo el código cargado en la solución (la biblioteca de clases portátiles y los proyectos específicos de cada plataforma).
- El proyecto PCL puede ser sencillo para los otros proyectos en una solución, o el conjunto de salida puede ser compartida por los demás para hacer referencia en sus soluciones.

b. Desventajas

- Debido a que la misma biblioteca de clases portable es compartida entre múltiples aplicaciones, bibliotecas específicas de la plataforma no se puede hacer referencia a algunos tipos de datos.
- El subconjunto de biblioteca de clases portable no puede incluir las clases que de otro modo estar disponibles tanto en *MonoTouch* y *Mono para Android*.

3.7.4. Proyecto Compartido

A diferencia de la mayoría de otros tipos de proyectos de un proyecto compartido no tiene ninguna salida (en forma de *DLL*), en lugar del código se compila en cada proyecto que hace referencia a ella. Esto se ilustra en el siguiente diagrama - conceptualmente todo el contenido del Proyecto Común al cual se hace referencia cada proyecto referencias y compilados como si fuera una parte de ellos, de tal forma que la estructura de la aplicación queda como se muestra a continuación.

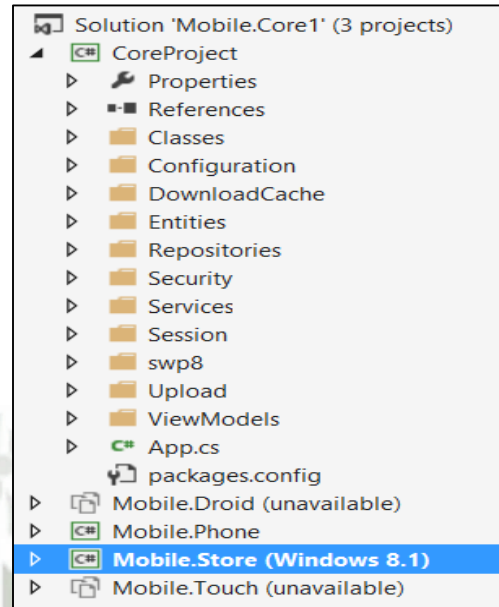


Figura 38: Proyecto compartido
Fuente: Elaboración propia

El código en un proyecto compartido puede contener directivas de compilación que activar o desactivar las secciones de código dependiendo de qué proyecto de aplicación está utilizando el código, que es sugerido por las cajas de plataforma de color en el diagrama.

Un proyecto compartido no consigue compila por sí mismo, que existe sólo como un conjunto de archivos de código fuente que se puede incluir en otros proyectos. Cuando se hace referencia por otro proyecto, el código se compila de manera efectiva como parte de ese proyecto. Proyectos compartidos no puede hacer referencia a cualquier otro tipo de proyecto (incluyendo otros proyectos compartidos).

3.7.5. Enlace de Datos entre Capas

El enlace de datos juega un papel muy importante en el patrón *MVVM*. Su ViewModel (Modelo de Vista) y las clases de la capa de Modelo deben ser diseñados para apoyar el enlace de datos. Normalmente, esto significa que deben implementar las *interfaces* de comunicación.

En efecto, el patrón *MVVM* gira en torno al concepto del enlace de datos, o *Data Binding* por su nombre en inglés. El enlace de datos es una conexión entre la interfaz de usuario y un objeto de datos que permite que los datos fluyan entre ellos. Cuando se establece un enlace y los datos cambian, los elementos de la interfaz de usuario enlazados a los datos pueden mostrar los cambios automáticamente. De manera similar, los cambios que realiza el usuario en un elemento de la interfaz de usuario se pueden guardar en el objeto de datos.


Se puede hablar largo y tendido del concepto de *Data Binding*, pero de momento solamente intentaremos resumir brevemente la idea que nos permita poder seguir el hilo de explicación del patrón *MVVM*, a continuación mostramos como es que se realiza el enlace de datos en cada una de las plataformas:



```
<TextBlock
    Text="{Binding FirstName}" />
```



```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    local:MvxBind="Text FirstName" />
```



```
var label = new UILabel();
Add(label);
this.CreateBinding(label)
    .To<FirstViewModel>(vm => vm.FirstName)
    .Apply();
```

Figura 39: Código base para los enlaces de datos

Fuente: Elaboración propia

Todos los enlaces incluyen lo siguiente:

- Un origen de enlace, que es un objeto con los datos que se desean representar.
- Un destino de enlace, que es una propiedad del elemento de la *UI* que usamos para representar los datos.
- Un objeto *Binding*, que mueve los datos entre el origen y el destino y que puede cambiar su formato a través de un convertidor de valores opcional.

Los enlaces pueden tener varias direcciones:

- Los enlaces *OneTime* actualizan el destino con los datos del origen cuando se crea el enlace.
- Los enlaces *OneWay* actualizan el destino con los datos de origen cuando se crea el enlace y cada vez que cambian los datos. Este es el modo predeterminado.
- Los enlaces *TwoWay* actualizan tanto el destino como el origen cuando alguno de ellos cambia.

3.7.6. Interfaces Importantes

A continuación describimos las *interfaces* de comunicación entre las capas del patrón *MVVM*, las cuales juegan un papel muy importante en el comportamiento del mismo.

- *Interface* *INotifyPropertyChanged*

La principal característica de esta *interface* es notificar el cambio de estado de una propiedad del lado del Modelo y/o Modelo de Vista Hacia una capa superior. Implementar la interfaz *INotifyPropertyChanged* permite proporcionar notificaciones de

cambio a todos los controles enlazados a datos en la capa de Vista cuando haya cambios en el valor de la propiedad.

```
namespace System.ComponentModel
{
    public interface INotifyPropertyChanged
    {
        event PropertyChangedEventHandler PropertyChanged;
    }
}
```

Figura 40: Código de interface namespace

Fuente: Elaboración propia

- *Interface* Inotify Collection Changed

Al igual que la *interface* Inotify Property Changed notifica cambios en los estados de las propiedades, en este caso colecciones de objetos. Los datos en el Modelo de vista representan una colección de objetos que van a ser representados visualmente en la capa de vista.

- ICommand

Esta *interface* brinda la posibilidad de comunicación a través de Comandos, los cuales proporcionan una manera conveniente de representar acciones u operaciones que pueden ser fácilmente unidos a los controles en la interfaz de usuario.

```
namespace System.Windows.Input
{
    public interface ICommand
    {
        bool CanExecute(object parameter);
        void Execute(object parameter);
        event EventHandler CanExecuteChanged;
    }
}
```

Figura 41: código de interface namespace

Fuente: Elaboración propia

3.7.7. Definición de la Arquitectura

El aplicativo se basa en una arquitectura simple, se puede mantener esta capa de acceso a datos utilizando *Windows Azure*, de esta forma el consumo de servicios y la comunicación en cada plataforma es indiferente y sencilla como se puede observar en la siguiente figura.

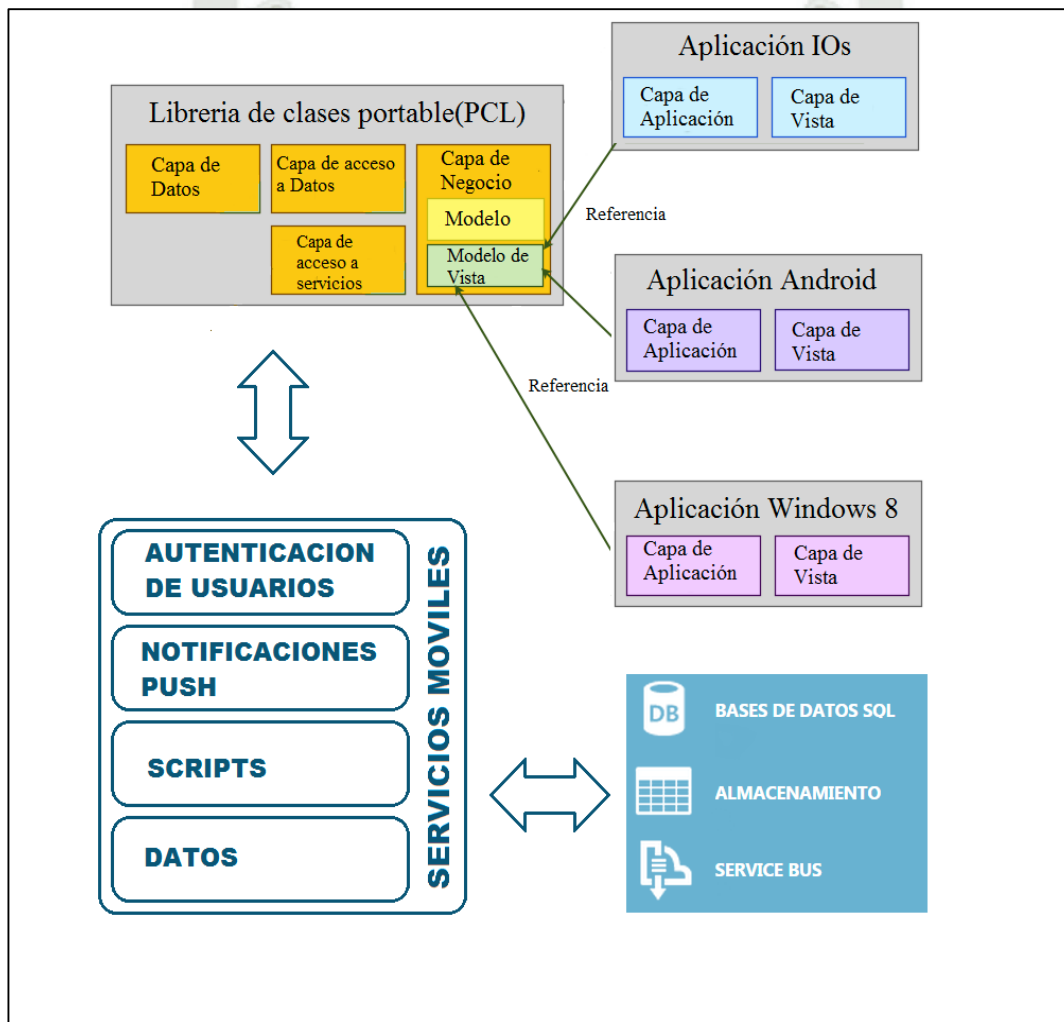


Figura 42: Arquitectura propuesta para la aplicación Móvil multiplataforma
Fuente: Elaboración propia

Un punto clave de la creación de aplicaciones multiplataforma es crear una arquitectura que se presta a un uso compartido máximo de código a través de plataformas. La adhesión a los siguientes principios de programación orientada a objetos ayuda a construir una aplicación con arquitectura multiplataforma:

Encapsulación: Asegurar que las clases e incluso capas de arquitectura sólo exponen una *API* mínima que realiza sus funciones requeridas, y oculta los detalles de implementación. A nivel de clase, esto significa que los objetos se comportan como Cajas Negras y que el consumo de código no necesita saber cómo lograr sus tareas. A nivel arquitectónico, que significa la implementación de patrones como de fachadas que fomenten una *API* simplificada que orquesta más interacciones complejas en nombre del código en capas más abstractas. Esto significa que el código de la interfaz de usuario sólo debe ser responsable de mostrar pantallas y aceptar la entrada de usuario; y nunca interactuar con la base de datos directamente. Del mismo modo el código de acceso a datos sólo debe leer y escribir en la base de datos, pero nunca de interactuar directamente con los botones o etiquetas.

Separación de Responsabilidades: Asegúrese de que cada componente (tanto a nivel arquitectónico y de clase) tiene un propósito claro y bien definido. Cada componente debe realizar sólo las tareas definidas y exponer esa funcionalidad a través de una *API* que es accesible a las otras clases que necesitan para utilizarlo.

Polimorfismo: Programación de una interfaz (o clase abstracta) que soporta múltiples implementaciones, significa que el código del núcleo puede ser escrito y compartido a través de plataformas, al tiempo que interactúan con las características específicas de la plataforma.

3.7.8. Capas de la Aplicación

A lo largo de este documento y los estudios de caso nos referimos a los siguientes seis capas de aplicación:

Capa de datos: la persistencia de datos no volátil, es probable que sea una base de datos SQLite, pero podría ser implementado con archivos *XML* o cualquier otro mecanismo adecuado.

Capa de Acceso a Datos: Envoltura alrededor de la capa de datos que ofrece crear, leer, actualizar, eliminar (*CRUD*) el acceso a los datos sin exponer a los detalles de implementación de la persona que llama. Por ejemplo, el DAL puede contener sentencias SQL para consultar o actualizar los datos, pero el código de referencia no tendría que saber esto.

Capa de Negocios: A veces llamada la capa de lógica de negocios, contiene definiciones de entidades de negocios (el modelo de datos) y la lógica de negocio. Candidato para el patrón de negocios de fachada.

Capa de Acceso a servicios: Se utiliza para acceder a los servicios en la nube, desde servicios *web* complejos (REST, JSON, WCF) a la simple recuperación de los datos e imágenes de los servidores remotos. Encapsula el comportamiento de redes y proporciona una *API* simple para ser consumido por las capas de aplicación y la interfaz de usuario.

Capa de aplicación: Código que es normalmente específica de la plataforma (no se comparte en general en todas las plataformas) o código que es específico de la aplicación (por lo general no reutilizable). Una buena prueba de si se debe colocar el

código en la capa de aplicación en comparación con la capa de interfaz de usuario es (a) para determinar si la clase tiene ningún tipo de control de visualizaciones reales o (b) si se puede compartir entre múltiples pantallas o dispositivos (por ejemplo. El *iPhone* y el *iPad*).

Interfaz de usuario (*UI*) Capa - La capa de usuario, e-contiene pantallas, *widgets* y los controladores que los gestionan.



CAPÍTULO IV: PRUEBAS Y RESULTADOS

La aplicación de promoción del destino turístico Arequipa para dispositivos móviles, hace uso de la interactividad y multifuncionalidad que estos dispositivos nos permiten. A través de contenido escrito, gráfico y audiovisual, presentamos el centro histórico de Arequipa y sus principales empresas turísticas. La revolución que están produciendo estos dispositivos los hace de uso global permitiéndonos llegar a todo el mundo. Estamos interesados en más turistas para nuestros destinos y en más turistas para la empresa y los emprendimientos en turismo.

4.1. Capturas de Pantalla de la Aplicación

4.1.1. Registro e Inicio de Sesión

A continuación mostramos el proceso de registro en inicio de sesión, el cual es realizado utilizando el sistema de autenticación de la red social *Facebook*, esto a su vez permite compartir nuestros comentarios, sobre el aplicativo y el destino turístico que es objeto de este trabajo. El proceso visual de registro e inicio de sesión es el mismo, ya que para poder registrarse el usuario deberá contar con una cuenta en la red social *Facebook*, que es de donde se extraerán los datos la primera vez que inicie sesión, posteriormente el proceso ya no será necesario ya que los datos de usuario se guardan y el inicio de sesión será automático.



El Login y Registro es
automatico a travez del
sistema de autentificacion de
Facebook

Figura 43: Vista de login y registro
Fuente: Elaboración propia

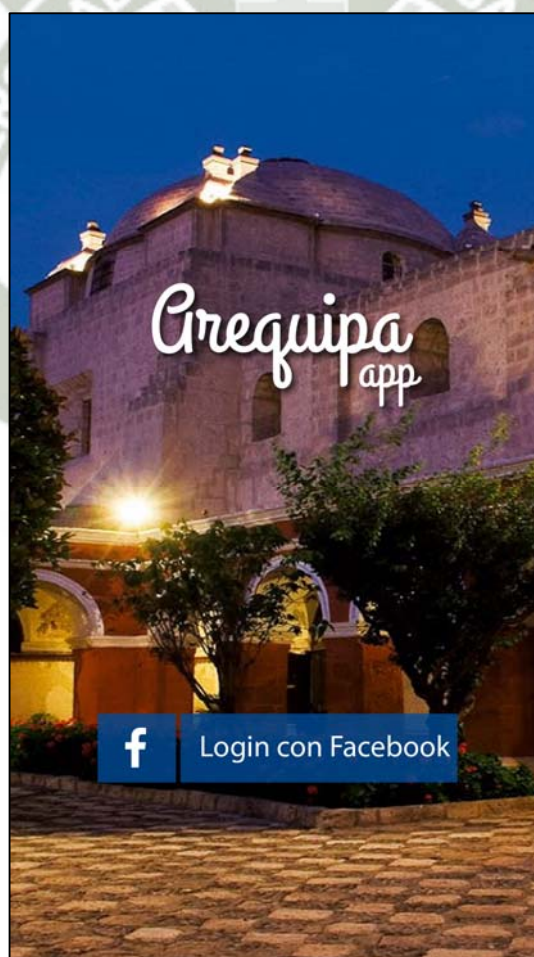


Figura 44(a): Inicio de sesión y registro
Fuente: Elaboración propia

En la siguiente vista se muestra el proceso de registro del dispositivo móvil en el servidor en la nube, esto es para el envío de las notificaciones *Push*, este proceso se realiza la primera vez que inicia el aplicativo luego de ser instalado.

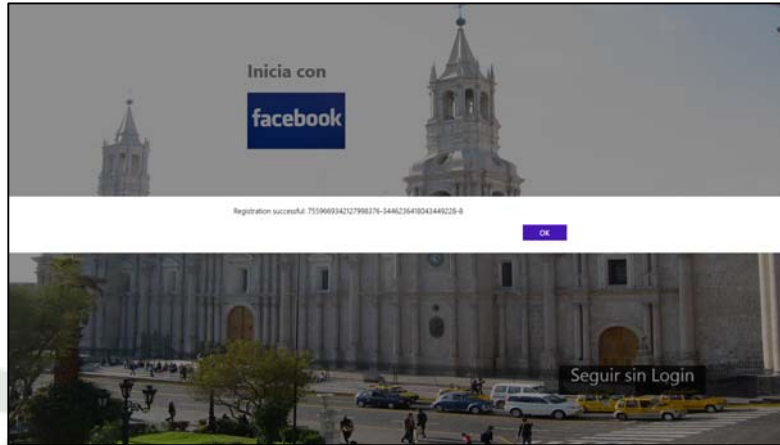


Figura 44(b): Inicio de sesión y registro
Fuente: Elaboración propia

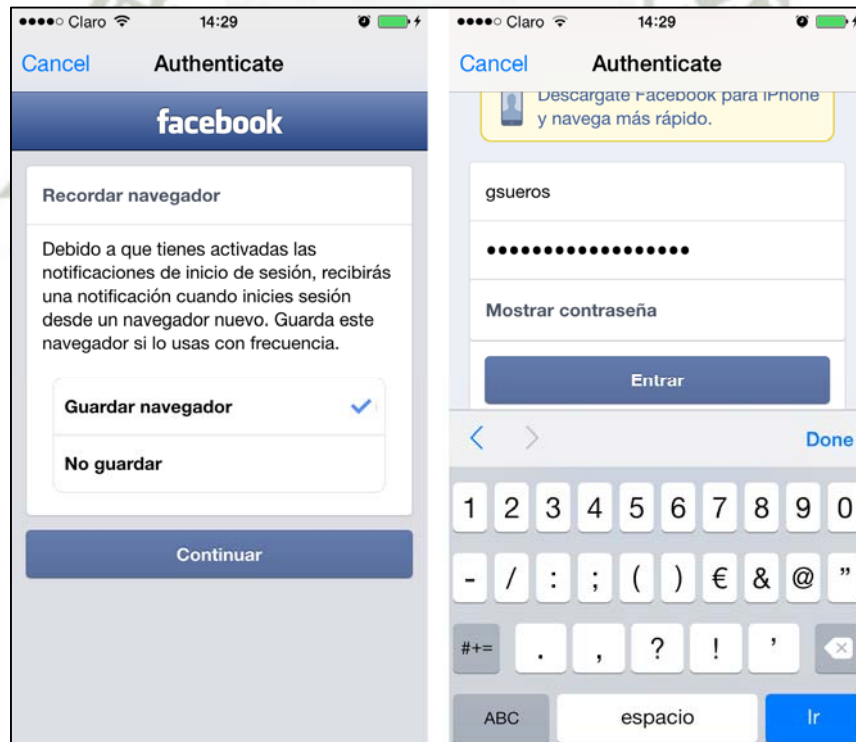


Figura 45: Login y registro con Facebook en teléfonos inteligentes
Fuente: Elaboración propia

4.1.2. Escribir un Comentario

A continuación mostramos el proceso de postear un comentario, como se observa la caja de comentarios es una lista delegable para que no interfiera con el uso interactivo del aplicativo, permitiéndole al usuario un uso adecuado del aplicativo.

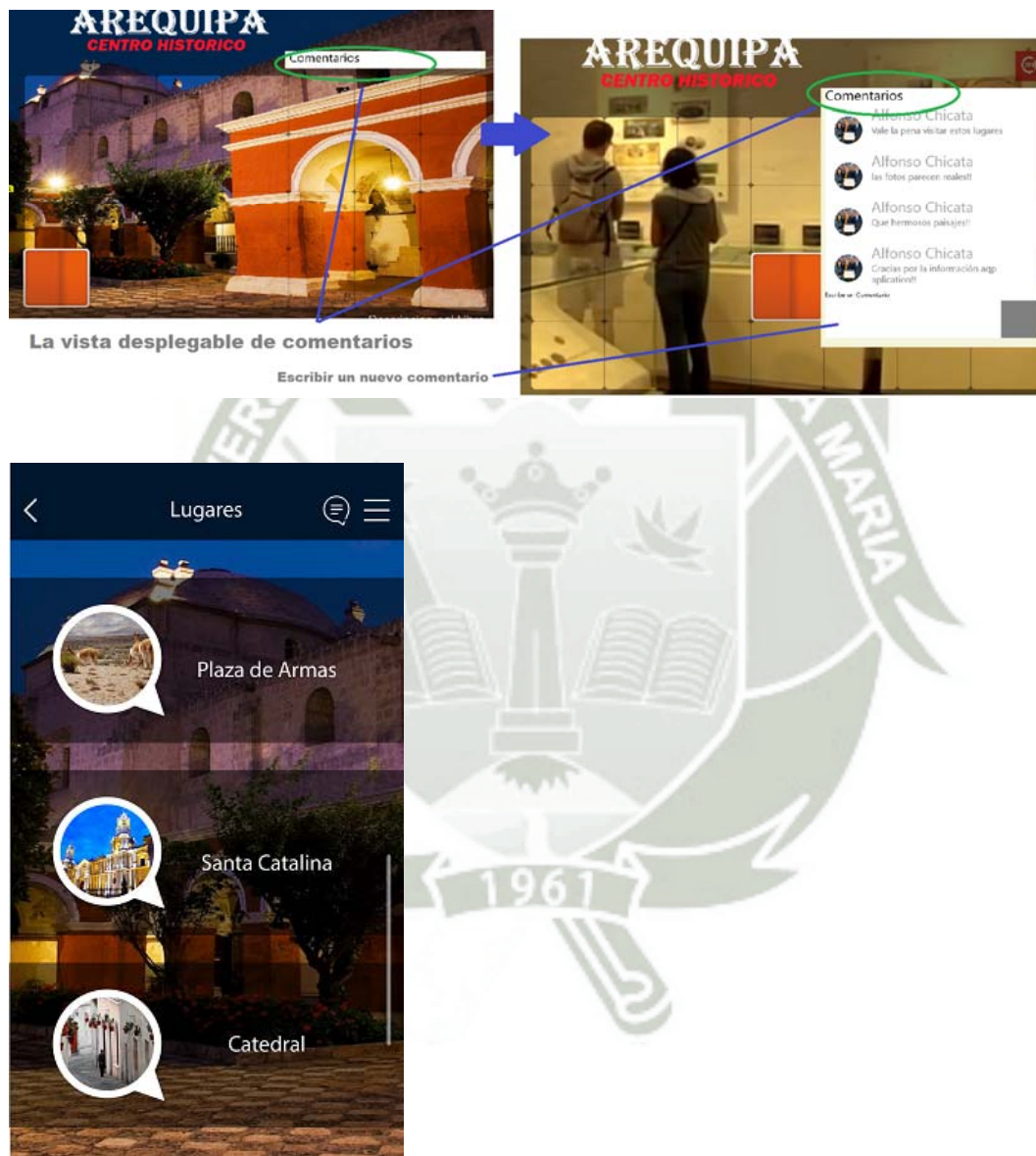


Figura 46: Vista multimedia principal
Fuente: Elaboración propia

4.2. Usabilidad e Interactividad del Aplicativo

La aplicación tiene contenido que apunta a generar el interés por el destino Arequipa, los conceptos de promoción son novedosos como lo son sus mapas y fotografías y videos que buscan fijar experiencias. Combinando lo tradicional con la avanzada tecnología, esta *App*, es capaz de ofrecer una experiencia totalmente nueva en la presentación de contenidos digitales, más rico que un libro tradicional y un nivel de animación e interacción similar al de una película o un video juego. La aplicación reúne el resultado de la investigación en tecnología innovadora para las ilustraciones, animaciones y una interfaz de usuario multi-táctil.



Figura 47: Usabilidad e interactividad del aplicativo
Fuente: Elaboración propia

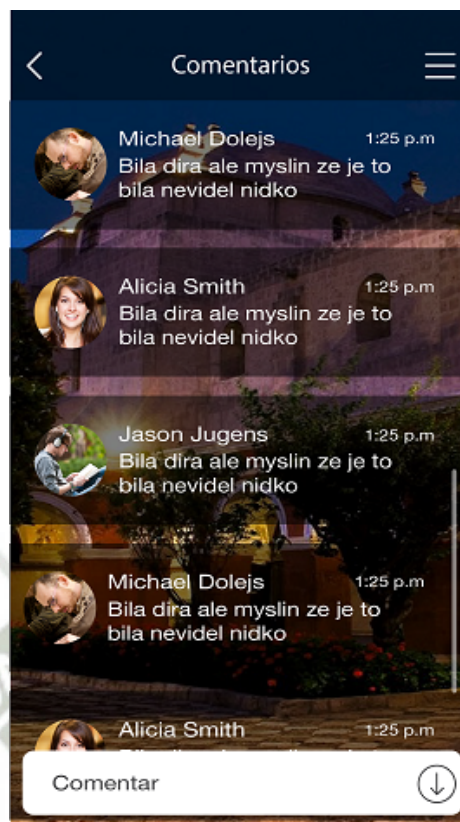


Figura 48: Funcionalidad de la interactividad de la aplicación.

Fuente: Elaboración propia



Figura 49: Funcionalidad de la interactividad de la aplicación.

Fuente: Elaboración propia

4.3. El Contenido del Aplicativo

Esta aplicación cambia la forma en que se conoce la cultura y los recursos turísticos del Centro Histórico de Arequipa, pues presenta un diseño de interfaz de usuario único que le permite una navegación rápida a través del contenido. Así, podrá navegar a través de los diferentes temas deslizándose a través de una tabla de contenidos visual o navegar a través de un capítulo por medio de las páginas. Para empezar a leer, puede usar los dedos para que aparezcan los contenidos. Para volver a la tabla de contenidos solo.



Figura 50: Vista de Contenido informativo brindado por el aplicativo
Fuente: Elaboración propia



Figura 51: Vista de Mapa del Centro Histórico
Fuente: Elaboración propia

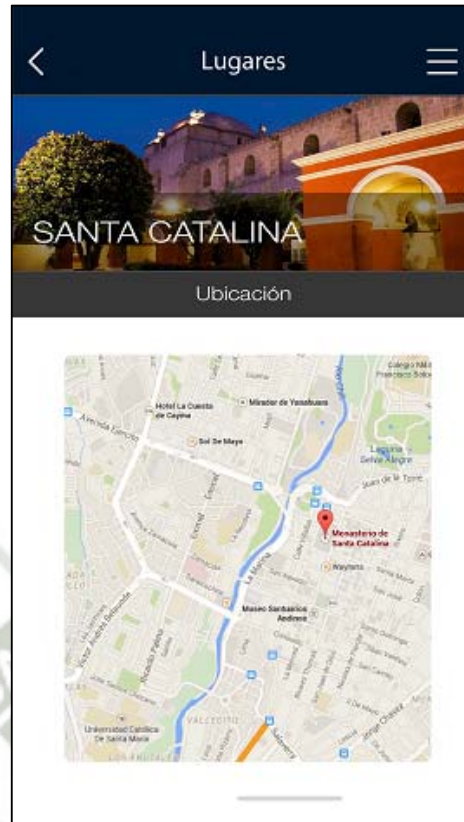


Figura 52: Vista de Contenido informativo
brindado por el aplicativo
Fuente: Elaboración propia

4.4. Análisis de los Resultados

Se tuvo la participación de 2 especialistas en el área de desarrollo de aplicaciones móviles, docentes de la Maestría en Informática de la Universidad Nacional de San Agustín, Auspiciada por el Concytec, y 28 alumnos del curso de Desarrollo de Móvil que se dicta en la Maestría, dando un total de 30 encuestados, los cuales contestaron una encuesta de manera electrónica, la cual cuenta con preguntas de opción múltiple con escala del 1 al 5, donde 1 es el indicador más bajo y 5 es indicador más alto, estos datos se representan en gráficas de barras con su respectivo análisis donde se interpretan los resultados de cada pregunta realizada a los encuestados. La conclusión es un juicio

razonado, basado en la síntesis de los resultados, sustentado por el análisis de los datos y las definiciones sobre las cuales se trabajó durante el desarrollo de esta tesis.

A continuación mostramos los resultados de las encuestas realizadas y las conclusiones que se obtienen a partir de los mismos, tomando como punto de referencia lo desarrollado en los capítulos anteriores.

¿La aplicación móvil multiplataforma cumple con lo propuesto, para promocionar un destino turístico?

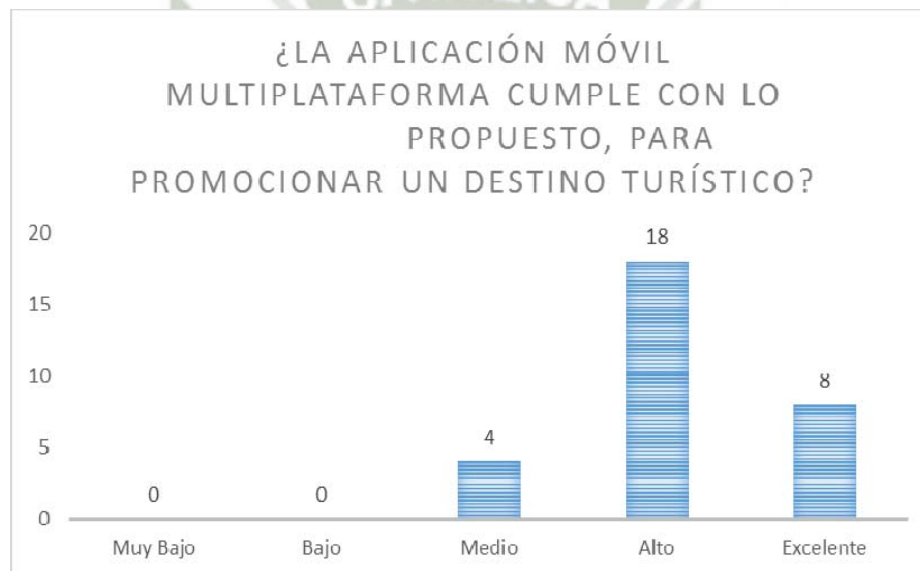


Figura 53: ¿La aplicación móvil multiplataforma cumple con lo propuesto, para promocionar un destino turístico?

Fuente: Elaboración propia

Como se observa en los resultados de la encuesta la apreciación de los encuestados no indica que el aplicativo cumple con los objetivos para el cual fue diseñado, desde este punto de vista podemos afirmar que se tuvo buen número de aciertos con lo planteado en los objetivos.

¿Cómo evalúa el diseño de interfaz de usuario de la aplicación?

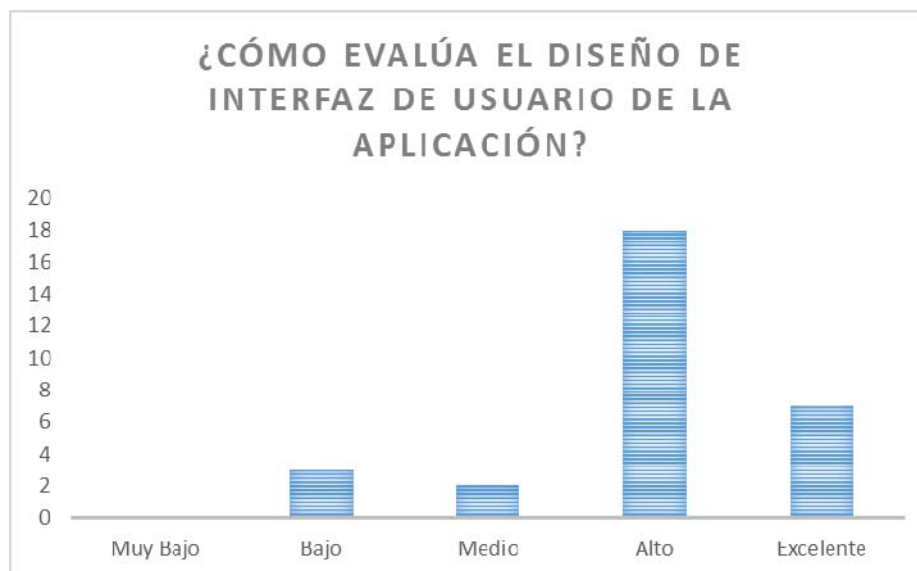


Figura 54: ¿Cómo evalúa el diseño de interfaz de usuario de la aplicación?

Fuente: Elaboración propia

¿Cómo evalúa el desempeño de la aplicación en un dispositivo real?

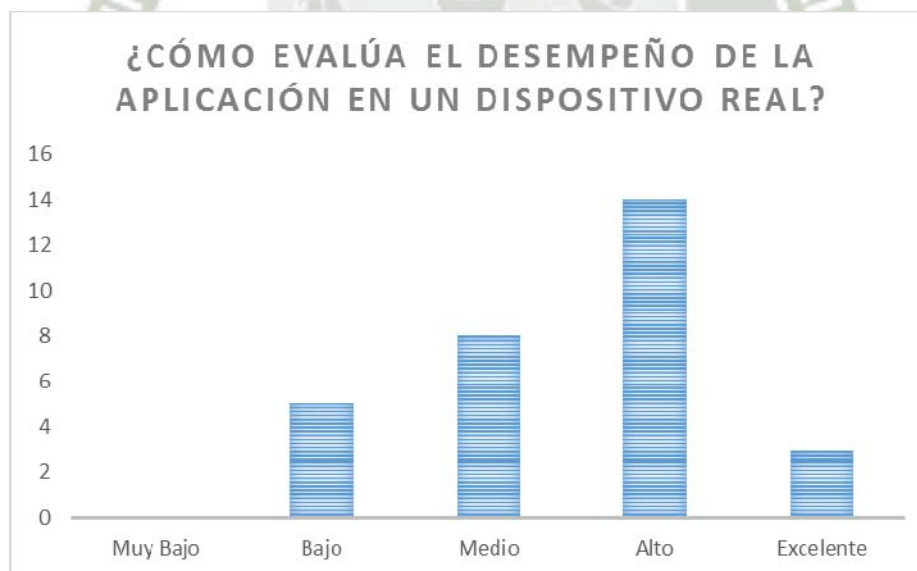


Figura 55: ¿Cómo evalúa el desempeño de la aplicación en un dispositivo real?

Fuente: Elaboración propia

A diferencia de pruebas hechas en los simuladores respectivos de cada plataforma, en cuyo caso el desempeño del aplicativo depende de las características de la computadora en que corre el simulador; las pruebas en dispositivos reales, es primordial para evaluar el desempeño del aplicativo en un ambiente real. Tal y como muestran los resultados, por lo general el desempeño es bueno, y los picos bajos pueden justificarse en la capacidad de hardware de los dispositivos o las versiones de los sistemas operativos instalados.

¿Cómo evalúa la usabilidad de la aplicación en una tableta?

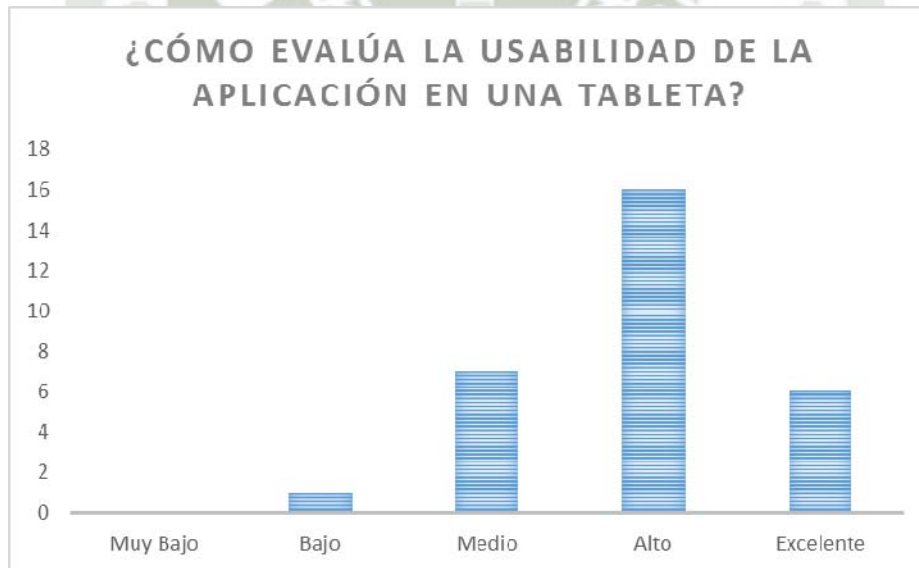


Figura 56: ¿Cómo evalúa la usabilidad de la aplicación en una tableta?

Fuente: Elaboración propia

En este caso los resultados nos permiten verificar la usabilidad del aplicativo en una tableta, obteniendo resultados buenos en cuanto a la usabilidad.

¿Cómo evalúa la usabilidad de la aplicación en un teléfono inteligente?

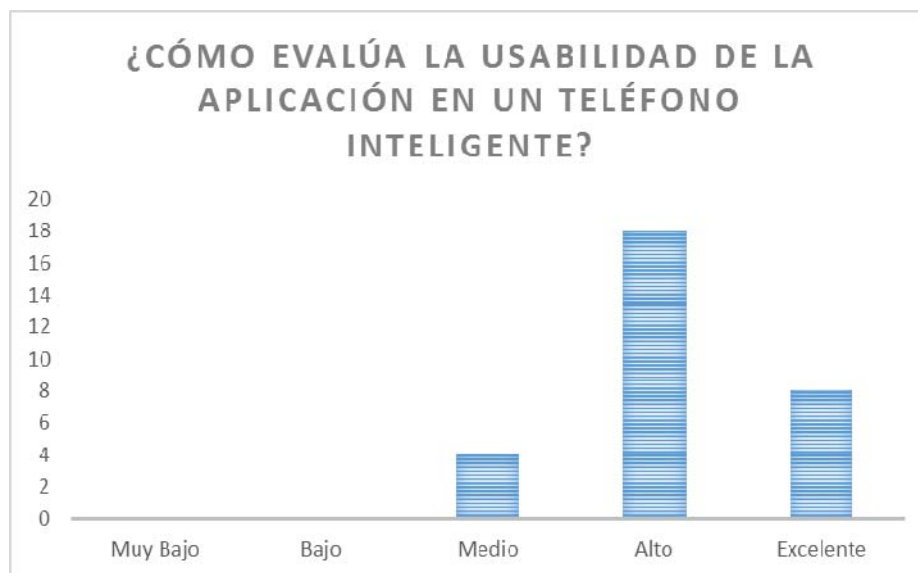


Figura 57: ¿Cómo evalúa la usabilidad de la aplicación en un teléfono inteligente?

Fuente: Elaboración propia

Como se observa los resultados a cerca de la usabilidad en teléfonos inteligentes en aun mejor que la de tabletas, esto nos da indicios sobre las diferencias entre una aplicación diseñada para tabletas y teléfonos inteligentes. Los diseños deben permitir que la aplicación brinde los mismos servicios con usabilidad similar, aunque no igual.

¿Usarías la aplicación en tus viajes? Evalúa el tema de portabilidad multiplataforma y multi-resolución.

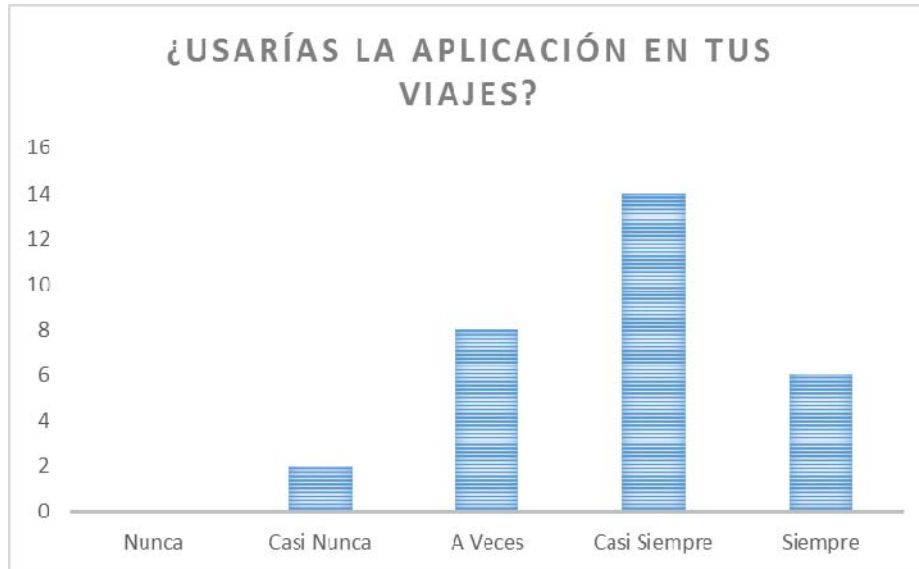


Figura 58: ¿Usarías la aplicación en tus viajes?
Fuente: Elaboración propia

El tema de cuantos usuarios van utilizar la aplicación durante los viaje tiene que ver con tipo de dispositivo, ya sea un tableta o un teléfono inteligente, tal y como ya se mencionó la cantidad de información mostrada, la facilidad de uso y todas las características propias de una aplicación nativa.

4.4.1. Rendimiento

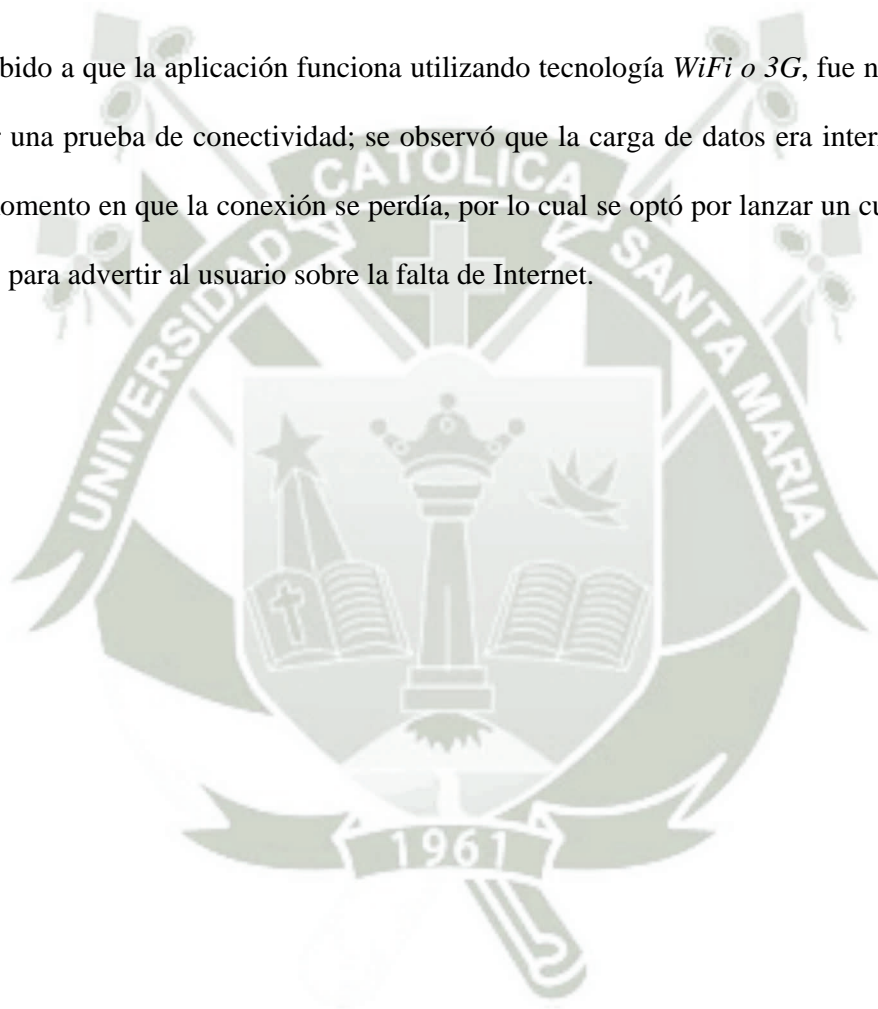
En esta prueba se estudió el tiempo en que carga la aplicación y el tiempo de carga de datos, esto es, cuánto tarda en realizar el análisis de XML e interpretarlo. Se observó que en la primera que se abría la aplicación, inmediatamente después de instalar, el tiempo de espera fue mayor, mientras que las siguientes ocasiones la carga se realizaba en un tiempo mínimo. Se consideró que el rendimiento de la aplicación era el óptimo.

4.4.2. Puntos Crash

En esta prueba se navegó por la aplicación, realizando un completo uso de todas las características de las mismas para detectar algún fallo o punto en que el proceso se detuviera, sin embargo el resultado fue favorable, debido a que no ocurrió algún error.

4.4.3. Conectividad

Debido a que la aplicación funciona utilizando tecnología *WiFi* o *3G*, fue necesario realizar una prueba de conectividad; se observó que la carga de datos era interrumpida en el momento en que la conexión se perdía, por lo cual se optó por lanzar un cuadro de diálogo para advertir al usuario sobre la falta de Internet.



CONCLUSIONES

1. Se ha desarrollado un aplicativo móvil multiplataforma, bajo el Framework de desarrollo Xamarin y se utilizó la plataforma Windows Azure para los servicios móviles en la nube. Asimismo se obtuvo prototipos para tabletas y teléfonos inteligentes, con el objetivo de cubrir de forma adecuada las diferentes plataformas y tipos de dispositivos móviles. Se logró una versión para el sistema operativo IOs en su versión 8.1; por la otra parte, para tabletas se consiguió una versión para Android en su versión 4.2 y para Windows en su versión 8.1.
2. Se introdujo de manera práctica al patrón de diseño MVVM como se observa en la parte de desarrollo de esta tesis, cumpliendo con el objetivo de aplicar este patrón al desarrollo de una aplicación móvil multiplataforma. Al ser posible separar la lógica de negocio y el modelo de datos de las particularidades de la vista, se desarrolló un ViewModel lo suficientemente desacoplado de tal forma que se reutilizó el código del Backend y la lógica de negocio, para las diferentes plataformas.
3. Se analizó la comunicación entre la capa de Modelo de Vista y la capa de Vista, esto gracias a la implementación de la interfaz INotifyPropertyChanged que por medio de notificaciones se concreta la comunicación entre las dos capas.
4. Se comprobó la importancia del uso de la computación en la nube (Cloud Computing) para el aprovechamiento de servicios móviles ofrecidos por la nube, en el presente trabajo, Microsoft Windows Azure, ayuda con buenas prácticas a utilizar los diferentes servicios y soluciones que este servicio nos presenta, como es el caso de notificaciones Push, almacenamiento en la nube y escalabilidad de base de datos.

5. La usabilidad del aplicativo móvil está fuertemente influenciada por el contexto de uso, sobre esto existen aspectos que definen los posibles factores que pueden ser parte del contexto y que afectan la usabilidad de una aplicación. Para ello se utilizó una Tablet Ultrabook con Windows 8.1, una tableta con Android 4.2 y un iPhone 5 con sistema iOS 8.1.
6. Con la finalidad de entender cómo el usuario interactúa con el aplicativo y como es que el aplicativo se comporta y reacciona ante posibles fallos, se analizó tanto las interacciones directas del usuario con el dispositivo y la aplicación así como el contexto externo en el cual se utiliza el aplicativo móvil, de esta forma se observa que el usuario percibe durante su interacción por completo las características del aplicativo.
7. La mejor forma de obtener resultados sobre realizar pruebas de usabilidad es un entorno real de uso, debido a los requerimientos y tipo de proyecto. Para este caso los resultados fueron validados por una cantidad aceptable de usuarios, los cuales además tienen cierto grado de conocimiento en el área de desarrollo de aplicativos móviles; los resultados obtenidos fueron muy buenos, con algunas debilidades, las cuales también fueron analizadas, de tal forma que sirven de retroalimentación para mejoras y trabajos futuros.

RECOMENDACIONES

1. Utilizar adecuadamente las herramientas que aportan las nuevas tecnologías y sí también se aplican con la mayor eficiencia en la sociedad de la información. Las cifras mundiales demuestran que el Turismo se produce con mucha mayor magnitud en los países desarrollados, que en los que están en vías de desarrollo.
2. Las empresas del sector turístico pueden beneficiarse y darle un valor agregado a sus productos con el uso de las aplicaciones móviles: los “*smartphone*” permiten la comunicación directa y personal, en tiempo real, y genera una reacción inmediata de los usuarios. Además el uso de las plataformas móviles puede ser un soporte para nuevos productos y servicios basados en las experiencias, las emociones y el poder compartirlas.
3. A partir de este trabajo, se puede desarrollar otras versiones que vayan de la mano con la necesidad de los clientes. La interacción online es una oportunidad para seguir creciendo en este campo del desarrollo de aplicaciones móviles.
4. Se recomienda desarrollar este aplicativo en diferentes idiomas, principalmente en el idioma francés e inglés, ya que en la estadística mostrada en el desarrollo de este trabajo, se indica que la mayor cantidad de turistas proviene de Francia.
5. Para incentivar el desarrollo de este tipo de aplicaciones, a nivel regional, se recomienda que, el gobierno regional de Arequipa promueva el primer concurso regional de aplicaciones móviles multiplataforma dirigidas al turismo. Con el objetivo de incrementar la actividad turística en la ciudad de Arequipa.

6. La tecnología es una ciencia que avanza a una velocidad impresionante, por tal motivo, se recomienda que el desarrollo de las nuevas aplicaciones móviles estén a la vanguardia de los nuevos sistemas operativos que nos presentan las empresas que se encuentren líderes en ese momento.



REFERENCIAS BIBLIOGRÁFICAS

Acord, Christian G.; Murphy, Corey C. (2012) “Cross-platform mobile application development: a pattern-based approach”. Monterey, California: Naval Postgraduate School.

Aplicación turística Be your Guide- Toledo recuperado de <http://www.europapress.es/turismo/nacional/noticia-fitur-seis-aplicaciones-turisticas-premiadas-concurso-the-apptourism-awards-2014-20140124094501.html>

Akkiraju, Rama et al. (2011) “Toward the Development of Cross-Platform Business Applications via Model-Driven Transformations”. In SERVICES I, pages585–592.IEEE Computer Society.

Aklilu Redda, Yonathan. (2012) “Cross platform Mobile Applications Development. Noruega. Norwegian University of Science and Technology.

Allen, Sarah; Graupera, Vidal; Lundrigan, Lee. (2010) “ProSmart- phone Cross-Platform Development, Apress”.

Anglin, Todd Anglin. (2013) “The 5 hottest areas for mobile development tools. Guest Post at Telerik, a developer services company”. June 7, 2013.

Aranaz Tudela, Jaime. (2009) “Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma *android* de google”. Madrid, España, Universidad Carlos III de Madrid.

Baker, J. et al. (2011) "Megastore: Providing Scalable, Highly Available Storage for Interactive Services”. In Conf. on Innovative Data Systems Research.

Bigné, E., Font, X. & Andreu, L. (2000). Marketing de destinos turísticos: Análisis y estrategias de desarrollo. ISBN 84-7356 -262 -3. ESIC Editorial: Madrid, España.

Bruning, Mathieu. (2012) “Native Cross-platform Mobile Application Development Using Voind”. Delft, Holanda. Delft University of Technology.

Climent, J. (2003). Diccionario de la lengua española. Madrid, España: Espasa Calpe.

Diep, Chi-Kien; Tran, Quynh-Nhu; Tran, Minh-Triet. (2013) “Online Model-driven IDE to Design GUIs For Cross-platform Mobile Application. Advanced Program in Computer Science, University of Science”. Vietnam National University - Ho Chi Minh City.

Dimmick, John W., Sikand, Jaspreet y Patterson, Scott J. (1994): “The Gratifications of the Household Telephone: Sociability, Instrumentality and Rassurance”, en Communication Research, 21(5):643-663.

EUROPAPRESS, JULIO 2014. DE 125 'APPS' CANDIDATAS Fitur.- Seis aplicaciones turísticas premiadas en el concurso 'The AppTourism Awards 2014'. Recuperado de <http://www.europapress.es/turismo/nacional/noticia-fitur-seis-aplicaciones-turisticas-premiadas-concurso-the-apptourism-awards-2014-20140124094501.html>

Fernández, J. M. (2006). Tipos de dispositivos móviles. Consultado en octubre 11, 2011 en http://leo.ugr.es/J2ME/INTRO/intro_4.htm

Fidler, R., (1997): Mediamorphosis: Understanding New Media. Thousand Oaks (CA), Pine Forge Press.

Fombona, J. & Pascual, M. & Madeira, M. (2012). Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles. ISSN: 1133-8482. N° 41 Julio 2012 – pp.197-210. Pixel – Bit. Revista de medios y educación.

Glera Aransay, Cristina. (2013) Desarrollo de una guía para dispositivos móviles de establecimientos para celíacos en Logroño”. España: Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación.

Gómez Orts, Francisco Javier. (2013) “Metodología para implementar en la nube Aplicaciones Web basadas en Java”. Valencia, España. Universidad Politécnica de Valencia.

Greenberg, Albert et al. (2012), "VL2: A Scalable and Flexible Data Center Network". Communications of the ACM, vol. 54, no. 3, pp. 95-104.

Harriger, A.; Lutes, K.; Purdum, J. (2007) “Designing Curricula to Teach Concepts and Increase Em- ployability”. Proceedings of American Society for Engineering Education, 2007.

Huang, Kuo-Ying. (2010) “Challenges in Human-Computer Interaction Design for Mobile Devices”. WCECS 2009, pp. 236 – 241.

Icharts. (2012) “ComScore Reveals the Top *Smartphone* Platform Share”. Retrieved March 19, 2012, from: <http://www.icharts.net/chartchannel/chart/2012/comscore-reveals-top-smartphone-platform-share>

IDC. (2012) “IDC: *Android, Windows Phone* to lead smartphone market in 2015”.

Retrieved March 19th, 2012 from: <http://www.fiercemobilecontent.com/story/idc-android-windows-phone-lead-smartphone-market-2015/2011-06-10>

Jiménez, L & Jiménez, W. (2013). *TURISMO: Tendencias globales y planificación estratégica*. 1Ed.- Bogotá, Colombia: Ecoe Ediciones.

Marin, Cristian-Stefan. (2012) “Microsoft Windows Azure cloud application development and security concepts”. Tornio, Finlandia. Kemi-Tornio University of Applied Science.

Martínez González, Felipe Luis. (2011) “Aplicaciones para Dispositivos Móviles”. Valencia, España. Universidad Politécnica de Valencia.

McWherter, Jeff; Gowell, Scott (2012) “Professional Mobile Application Development”. John Wiley & Sons, Inc. 2012.

Mochón, F (2004). *La economía y el turismo*. Buenos Aires, Argentina: Ed. Mc Graw-Hill

Organización Mundial del Turismo. (2005-2007). *Entender el turismo: Glosario Básico*. Retrieved from Organización Mundial del Turismo OMT: Valls, J-F (2007). *Gestión de destinos turísticos sostenibles*. Córdoba: I Congreso Nacional de Planificación, Dinamización y Calidad en Destinos Turísticos. Diputación Provincial de Córdoba, Colombia.

Peppers, Jonathan. (2014) “Xamarin cross-platform Application Development - Develop production-ready applications for iOS and Android using Xamarin” Birmingham – Mumbai: Packt Publishing.

PERUNATURAL, Recuperado de

<http://www.sernanp.gob.pe/sernanp/noticia.jsp?ID=1929>

Ramadath, Srinivas; Collins, Maurice. (2012) “Mobile Application Development: Challenges and Best Practices. Accenture white paper, 2012”.

Saks, Mattias. (2012) “Cross-Platform Interaction Design”. Estonia, Tallinn. University Institute of Informatics.

Scott Densmore, Alex Homer, Masashi Narumoto, John Sharp, y Hanz. (Zhang. (2013) Building Hybrid Applications in the cloud on Windows Azure. EEUU: Microsoft

SEGITUR (2014). Concurso The AppTourism Awards 2014. Recuperado de <http://www.segittur.es/es/proyectos/proyecto-detalle/Concurso-The-AppTourism-Awards-2014#.U7iS8pR5OSp>

Singh, Inderjeet; Cicchetti, Antonio. (2012) “Comparison of Cross-Platform Mobile Development Tools”. 16th International Conference on Intelligence in Next Generation Networks.

Sommer, Andreas. (2012) “Comparison and evaluation of cross-platform frameworks for the development of mobile business applications”. Munich: Technische Universität München.

Souza, A. (2006): “Re-Conceptualizing the Mobile Phone – From Telephone to Collective *Interfaces*”, en *Australian Journal of Emerging Technologies and Society*, Vol 4 (2), 2006: 108-127.

Springer, Thomas. (2013) “Application Development for Mobile and Ubiquitous Computing”. Sajonia, Alemania. Technische Universität Dresden Chair of Computer Networks.

Valls, J-F (2007). Gestión de destinos turísticos sostenibles. Córdoba: I Congreso Nacional de Planificación, Dinamización y Calidad en Destinos Turísticos. Diputación Provincial de Córdoba, Colombia.

Wargo, Jhon M. (2012) “Building Cross-Platform Mobile Apps”. New York. Addison Wesley.

Williams, F., Dordick, H., & Jesuale, H. (1985) “Focus Group and Questionnaire Development for Exploring Attitudes Towards Telephone Service”, en WILLIAMS, F. (comps): *Social Research and the Telephone*. Los Ángeles (CA), Herbert Dordick and Associates.

Wu, Ling; Liang, Guangtai; Wang, Qianxiang. (2012) “Program Behavior Analysis and Control for Online IDE”. *International Conference on Computer Software and Applications Workshops, IEEE 2012*, pp. 182 – 187.



ANEXO N° 01

CUESTIONARIO SOBRE EL APLICATIVO MÓVIL DE AREQUIPA

Marque una opción de acuerdo a su experiencia con el aplicativo móvil.

- ¿La aplicación móvil multiplataforma cumple con lo propuesto, para promocionar un destino turístico?
 - Muy Bajo
 - Bajo
 - Medio
 - Alto
 - Excelente
- ¿Cómo evalúa el diseño de interfaz de usuario de la aplicación?
 - Muy Bajo
 - Bajo
 - Medio
 - Alto
 - Excelente
- ¿Cómo evalúa el desempeño de la aplicación en un dispositivo real?
 - Muy Bajo
 - Bajo
 - Medio
 - Alto
 - Excelente

- ¿Cómo evalúa la usabilidad de la aplicación en una tableta?
 - Muy Bajo
 - Bajo
 - Medio
 - Alto
 - Excelente

- ¿Cómo evalúa la usabilidad de la aplicación en un teléfono inteligente?
 - Muy Bajo
 - Bajo
 - Medio
 - Alto
 - Excelente

- ¿Usarías la aplicación en tus viajes? Evalúa el tema de portabilidad multiplataforma y multi-resolución?.
 - Nunca
 - Casi Nunca
 - A veces
 - Casi siempre
 - Siempre

GLOSARIO DE TÉRMINOS

Android.- es un sistema operativo diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas,

Android Market.- es la tienda virtual de las aplicaciones desarrolladas para el sistema operativo Android

API.- Interfaz de programación de aplicaciones

APN.- Notificaciones push de Apple

App.- El término *app* es una abreviatura de la palabra en inglés application. Es decir, una app es un programa. Pero con unas características especiales

Apple.- Empresa multinacional que produce equipos tecnológicos y software

App Store.- Tienda virtual de las aplicaciones del sistema operativo IOS

BlackBerry.- s una marca de teléfonos inteligentes (smartphones) desarrollada por la compañía canadiense BlackBerry

Bluetooth.- es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Chat.- El chat¹ (término proveniente del inglés que en español equivale a 'charla'), también conocido como cibercharla, designa una comunicación escrita realizada de manera instantánea mediante el uso de un software y a través de Internet entre dos, tres o más personas

Cluster.- El término clúster (del inglés cluster, "grupo" o "racimo") se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardwares comunes y que se comportan como si fuesen una única computadora.

E-Book.- Un libro electrónico,¹ libro digital, ciberlibro, también conocido como e-book o eBook, es una versión electrónica o digital de un libro

Facebook.- es un sitio web de redes sociales creado por Mark Zuckerberg

Firewall.- cortafuegos (firewall en inglés) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

FITUR.- Feria internacional del turismo

Frameworks.- es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

(GDSs) o Sistemas de Distribución Global

Google.- es una empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías. El principal producto de Google es el motor de búsqueda de contenido en Internet del mismo nombre.

Google Maps.- es un servidor de aplicaciones de mapas en la web que pertenece a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo

Google Play.- Google Play Store (anteriormente Android Market) es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google

GPS.- permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros, aunque lo habitual son unos pocos metros de precisión

Hardware.- se refiere a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

Hosting.- es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

HTML5.- hace referencia al lenguaje de marcado para la elaboración de páginas web.

HTTP / HTTPS.- es el protocolo usado en cada transacción de la World Wide Web.

Interface.- en informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles.

iOS.- es un sistema operativo móvil de la empresa Apple

IP.- Internet Protocol (en español 'Protocolo de Internet') o IP es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI.

iPhone.- iPhone es una línea de teléfonos inteligentes diseñada y comercializada por Apple Inc

iTunes.- es un reproductor de medios y tienda de contenidos multimedia desarrollado por Apple con el fin de reproducir, organizar y sincronizar iPods, iPhones, iPads y comprar música.

Javascript.- es un lenguaje de programación interpretado

Netbooks.- Netbook es una categoría de computadora portátil, de bajo costo y dimensiones reducidas, que aporta mayor movilidad y autonomía

OMT.- Organización mundial de turismo

PDA's.- e trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE o Windows Mobile de Microsoft entre otros, el cual le proporciona capacidades similares a los PC de escritorio.

PhoneGap.- es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems

Plugin.- es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

pocket-book.- libro de bolsillo

Pokec PC.- e trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE o Windows

Mobile de Microsoft entre otros, el cual le proporciona capacidades similares a los PC de escritorio.

PYMEs.- Pequeñas y medianas empresas

Ringtone.- Un tono de llamada, popularmente conocido por el anglicismo Ringtone, es el término con el cual se conoce el sonido hecho por un teléfono para indicar la llegada de un mensaje de texto o al recibir una llamada.

SEGITTUR.- La Sociedad Estatal para la Gestión de la Innovación y las Tecnologías Turísticas

Smartphone.- es un tipo teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.

Tablets.- es una computadora portátil de mayor tamaño que un teléfono inteligente o un PDA, integrada en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o un estilete (pasivo o activo), sin necesidad de teclado físico ni ratón

Tecnología WAP.- es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, p.ej. acceso a servicios de Internet desde un teléfono móvil.

TICS.- Tecnología de la información y de la comunicación

TI.- Tecnología de la información

UIT.-Unión Internacional de Telecomunicaciones

URIs.- El URL es una cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la Internet.

Widgets.- s una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine

Wi-Fi.- s un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica

Windows.- es el nombre de una familia de distribuciones para PC, smartphone, servidores y sistemas empotrados, desarrollados y vendidos por Microsoft, y disponibles para múltiples arquitecturas, tales como x86 y ARM

Windows Phone.- es un sistema operativo móvil desarrollado por Microsoft como sucesor de Windows Mobile.

Windows Store.- es la plataforma de distribución digital de software desarrollada por Microsoft como parte de Windows 8 para aplicaciones de estilo Modern UI.²

Wireframes.- es un algoritmo de renderización del que resulta una imagen semitransparente, de la cual solo se dibujan las aristas de la malla que constituye al objeto

Wireless.- La comunicación inalámbrica o sin cables es aquella en la que la comunicación (emisor/receptor) no se encuentra unida por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio

WNS.- Servicio de notificación de Windows

XML.- es un lenguaje de marcas desarrollado por el World Wide Web Consortium

(W3C) utilizado para almacenar datos en forma legible.

