



Disertación de Master en
Ingeniería Informática– Computación Móvil

***Procesos de Desarrollo de Software
en Aplicaciones Móviles***

Lizeth Paola Chandi Argoti

Leiria, septiembre 2017

Esta página fue dejada en blanco intencionalmente



Disertación de Master en
Ingeniería Informática – Computación Móvil

***Procesos de Desarrollo de Software
en Aplicaciones Móviles***

Lizeth Paola Chandi Argoti

Disertación de master realizado con la orientación de la Dra. Catarina Silva, Profesora de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria, y la co-orientación de la Dra. Tatiana Gualotuña y el MSc. Danilo Martinez, Docentes de la “Universidad de las Fuerzas Armadas – ESPE”.

Leiria, *septiembre 2017*

Esta página fue dejada en blanco intencionalmente

Agradecimientos

Agradezco a Dios por todo lo que me ha dado a lo largo de mi vida, y que hoy en día me permite estar culminando esta etapa que me enriqueció tanto personal como profesionalmente.

De todo corazón agradezco a mi familia, especialmente a mis padres y hermana; a su incondicional apoyo y sobre todo el amor que me dan día a día. Espero recompensar todo lo que han sabido brindarme.

Quedo totalmente agradecida con mis tutores, Catarina, Tatiana y Danilo; que me brindaron su tiempo, apoyo y conocimiento, para lograr obtener este gratificante trabajo.

Gracias a las diferentes empresas y desarrolladores independientes que dieron apertura a la realización de encuestas y entrevistas y lograr la realización del presente trabajo.

Agradezco a la Universidad de las Fuerzas Armadas ESPE, que me forjó como ingeniera, y ahora me permite dar el siguiente paso en mi carrera profesional.

Me quedo con los más lindos recuerdos de IPL, más que una universidad fue un hogar, por varios meses. Agradecer infinitamente todo lo que me aportaron cada uno de sus docentes, y en general, todos quienes forman parte del instituto.

Gracias a las becas otorgadas por el estado, que permiten adquirir conocimiento a cientos de jóvenes, quienes con gusto retribuiremos lo recibido.

Es grato poder lograr una meta, y más aún cuando a lo largo de ella, pudiste enaltecer el nombre de tu país, conocer nuevas culturas y tener el respaldo de quienes forman parte de tu vida.

Esta página fue dejada en blanco intencionalmente

Resumen

Hoy en día los dispositivos móviles, más allá de lograr la comunicación, proporcionan soluciones para diversas situaciones en la vida diaria de quienes los poseen. Es así que la tecnología móvil incrementa de manera exponencial, a tal punto que para el 2021 se estima que el número de usuarios móviles será de 9 billones aproximadamente [1].

Las aplicaciones móviles permiten realizar diferentes actividades de manera fácil y oportuna, estableciéndose como una opción robusta para proveer y buscar información en tiempo real. Gracias a sus usuarios, quienes generalmente portan consigo los dispositivos móviles, logran realizar acciones sin importar su ubicación. Los usuarios de apps móviles notan la facilidad de uso que brindan las aplicaciones móviles; por tanto, gran número de las apps móviles no son únicamente de fácil acceso, sino también de fácil uso.

Si bien el mercado de las aplicaciones móviles ha crecido de manera evidente y exponencial, un problema visible es la deficiencia de documentación científica que refleje los procesos de desarrollo centrados en aplicaciones móviles o las actividades que se realicen.

El presente trabajo consiste en el estudio y caracterización de los procesos de desarrollo de aplicaciones móviles actuales más frecuentes aplicados a nivel de software que se ejecutan en dispositivos móviles tales como teléfonos inteligentes o tabletas; el análisis de estudios que han abordado los desafíos relacionados con el estudio de los procesos de desarrollo de aplicaciones móviles, y posteriormente, se presentan casos de estudios que permiten visualizar los procesos de desarrollo de aplicaciones móviles utilizados por empresas de desarrollo de software como por desarrolladores independientes.

El resultado es la caracterización y selección de los procesos de desarrollo de SW mayormente utilizados en la actualidad para finalmente, brindar una guía en la selección de

un proceso de desarrollo de SW que sea el más adecuado para un proyecto tras cumplir criterios propios del proyecto.

Palabras-Clave: procesos de desarrollo, aplicaciones móviles, dispositivos móviles.

Abstract

Nowadays, technology allows individuals to use information as best as they want and they see needed. It is estimated that by 2021 the number of mobile devices users will be of about 9 trillion [1]. Mobile devices are a communication tool that offers solutions for different situations in user's daily life.

Mobile applications allow performing different activities in an easy and timely manner, hence constituting a robust option to provide and search for information in real time. Thanks to their users, who generally carry their mobile devices with them most of the time, they manage to take action regardless of their location. On the other hand, mobile application users notice the ease of use of mobile applications. Therefore, much of mobile applications are not only easily accessible, but also user-friendly.

While the market for mobile applications has grown exponentially and clearly, a visible problem is the lack of scientific documentation that reflects the development processes focused on mobile applications or the activities that are carried out.

The present work consists in the study and characterization of the development processes of current mobile applications applied at the level of software that executes in mobile devices, such as, smartphones and tablets. In addition, the analysis of studies that have addressed the challenges related to the study of the processes of development of mobile applications is part of the present work, and later, case studies are presented that allow visualizing the development processes of mobile applications used by software development companies as independent developers.

The result is the characterization and selection of the most commonly used software development processes to finally provide a guide in the selection of the software development process most appropriate for a project after meeting project criteria.

Keywords: mobile application development, software engineering.

Esta página fue dejada en blanco intencionalmente

Lista de Figuras

Figura 1. Aplicaciones nativas: APIs Bajo y Alto Nivel.	7
Figura 2. Construcción App. Web: Tecnologías esenciales	9
Figura 3. Aplicación Híbrida: Composición	11
Figura 6. Unidades de análisis: Nombres de Aplicaciones Móviles	30
Figura 7. Nivel de rigurosidad en la aplicación de la metodología por parte de las UA	41
Figura 8. Equipos de desarrollo de UA: Núm. de integrantes por género	44
Figura 9. Equipos de desarrollo de UA: Núm. de integrantes por rangos de edad	44
Figura 10. Plataformas de las UA.	47
Figura 11. Tipo de desarrollo: Número UA.	47
Figura 12. Tipos de empresas de UA industriales: porcentajes	48
Figura 13. Nivel de experiencia: Número de encuestados por UA.	48
Figura 14. Tipos de metodologías: Número de UA.	48
Figura 15. Metodologías de desarrollo de SW: número utilizado por UA.	48

Esta página fue dejada en blanco intencionalmente

Lista de Tablas

Tabla 1. Sistemas Operativos Móviles: Herramientas	6
Tabla 2. Aplicaciones móviles nativas: ventajas y desventajas.	8
Tabla 3. Aplicaciones móviles web: ventajas y desventajas.	10
Tabla 4. Página web de aplicación híbrida: Ventajas y desventajas según su locación.	12
Tabla 5. Apps. Móviles Híbridas: Frameworks de Desarrollo	12
Tabla 6. Comparación de los tipos de aplicaciones móviles: Características	14
Tabla 7. Aplicaciones móviles: escenarios favorecedores	15
Tabla 8. Posibles respuestas para las sub preguntas de investigación	33
Tabla 9. Número de Integrantes por Género: UA	34
Tabla 10. Rango de edades en los equipos de desarrollo: UA	34
Tabla 11. Nivel de experiencia del encuestado del equipo de desarrollo	35
Tabla 12. Plataformas de desarrollo: Unidades de análisis	36
Tabla 13. Tipos de aplicaciones móviles: unidades de análisis	36
Tabla 14. Tiempo de desarrollo, tipos de empresas desarrollando apps	37
Tabla 15. Tipos de proceso de desarrollo sw: UA	37
Tabla 16. Proceso / Metodologías Actividades: UA	39
Tabla 17. Tiempo de desarrollo de las apps y sus costos	41
Tabla 18. Desafíos con la metodología durante el proceso de desarrollo: UA	42

Tabla 19. Desafíos con la aplicación durante el proceso de desarrollo: UA	43
Tabla 20. Metodologías de desarrollo de SW y criterios de evaluación	56
Tabla 21. Criterios cuantitativos en relación a las Metodologías	58
Tabla 22. Criterios evaluativos del proyecto a desarrollarse	59
Tabla 23. Criterios Evaluativos: caso aplicativo	62
Tabla 24. Matriz Resultante: Recomendación de Metodologías de Desarrollo	63

Acrónimos y Siglas

<i>SIGLA</i>	<i>SIGNIFICADO</i>
<i>API</i>	Application Programming Interface (Interfaz de programación de aplicaciones)
<i>AUP</i>	Agile Unified Process (Proceso Unificado ágil)
<i>CSS</i>	Cascading Style Sheets (Hojas de Estilo en Cascada)
<i>HTML</i>	HyperText Markup Language (Lenguaje de marcas de hipertexto)
<i>MVC</i>	Model, View, Controller (Modelo Vista Controlador)
<i>RAD</i>	Rapid Application Development (Desarrollo rápido de aplicaciones)
<i>OS</i>	Operating System (Sistema Operativo)
<i>SDK</i>	Software Development Kit (Kit de Desarrollo de Software)
<i>SW</i>	SoftWare
<i>UI</i>	User Interface (Diseño de Interfaz de usuario)
<i>UX</i>	User eXperience (Experiencia del usuario)

Esta página fue dejada en blanco intencionalmente

Referencia Artículo

Lizeth Chandi, Catarina Silva, Danilo Martinez, Tatiana Gualotuña, “Mobile application development process: A practical experience”, 12th Iberian Conference on Information Systems and Technologies (CISTI), IEEE, DOI: 10.23919/CISTI.2017.7975825, 2017.

Lizeth Chandi, Catarina Silva, Danilo Martinez, Tatiana Gualotuña, “Model for Selecting Software Development Methodology”, ICITS'18 - The 2018 International Conference on Information Technology & Systems (submitted).

Esta página fue dejada en blanco intencionalmente

Contenido

PROCESOS DE DESARROLLO DE SOFTWARE EN APLICACIONES MÓVILES	I
PROCESOS DE DESARROLLO DE SOFTWARE EN APLICACIONES MÓVILES	III
AGRADECIMIENTOS	V
RESUMEN	VII
ABSTRACT	IX
LISTA DE FIGURAS	XI
LISTA DE TABLAS	XIII
ACRÓNIMOS Y SIGLAS.....	XV
CONTENIDO	XIX
CAPÍTULO 1.....	1
INTRODUCCIÓN	1
1.1 MOTIVACIÓN	2
1.2 OBJETIVOS	3
1.3 ESTRUCTURA DEL DOCUMENTO	4
CAPÍTULO 2.....	5
ANTECEDENTES Y TRABAJOS RELACIONADOS.....	5
2.1 ENFOQUES DE DESARROLLO DE APLICACIONES MÓVILES.....	5
2.1.1 <i>Desarrollo de Aplicaciones Móviles Nativas</i>	5
2.1.2 <i>Desarrollo de Aplicaciones Móviles Web</i>	8
2.1.3 <i>Desarrollo de Aplicaciones Móviles Híbridas</i>	10
2.1.4 <i>Comparación de los enfoques de aplicación móvil y sus posibles escenarios</i>	14
2.2 MODELOS Y METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES.....	16
2.2.1 <i>Modelos y Metodologías No Ágiles de Desarrollo de Aplicaciones Móviles</i>	17
2.2.2 <i>Modelos y Metodologías Ágiles de Desarrollo de Aplicaciones Móviles</i>	21
2.3 TRABAJOS RELACIONADOS: ESTUDIOS DE PROCESOS DE DESARROLLO DE APLICACIONES MÓVILES	25
2.4 CONCLUSIÓN.....	27
CAPÍTULO 3.....	29
PROPUESTA DE ENFOQUE DE INVESTIGACIÓN.....	29
3.1 PROCESOS DE DESARROLLO DE APLICACIONES MÓVILES: CASOS DE ESTUDIO	29
3.2 METODOLOGÍA.....	30
3.2.1 <i>Pregunta de Investigación</i>	31
3.2.2 <i>Método de recolección de datos y estrategia de selección</i>	32
3.2.3 <i>Resultados</i>	32
3.3 DISCUSIÓN DE RESULTADOS	44
A. HALLAZGOS	44
B. LIMITACIONES	50
C. IMPLICACIONES PARA INVESTIGACIONES FUTURAS	50
3.4 CONCLUSIONES.....	51
CAPÍTULO 4.....	52

PROPUESTA DE MODELO PARA SELECCIÓN DE METODOLOGÍA DE DESARROLLO DE SOFTWARE	53
4.1 MODELO EVALUATIVO PROPUESTO	53
4.1.1 DEFINICIÓN DE CRITERIOS.....	54
A. METODOLOGÍAS DE DESARROLLO	54
B. CRITERIOS DE SELECCIÓN.....	54
4.1.2 DESARROLLO DE CRITERIOS POR METODOLOGÍA DE DESARROLLO DE SW	56
4.1.3 CUANTIFICACIÓN DE CRITERIOS POR METODOLOGÍA DE DESARROLLO DE SW	58
4.1.4 EVALUACIÓN DE LOS REQUERIMIENTOS GENERALES DEL PROYECTO	59
4.1.5 PRODUCTO DE MATRICES.....	59
4.1.6 OBTENCIÓN DE RESULTADOS	60
4.2 CASO APLICATIVO: PROPUESTA MODELO EVALUATIVO PARA SELECCIÓN DE METODOLOGÍA DE DESARROLLO DE SW.	61
4.3 CONCLUSIONES	64
CAPÍTULO 5	65
CONCLUSIONES Y TRABAJO FUTURO	65
REFERENCIAS.....	71
APÉNDICES Y ANEXOS	75
ANEXO 1.....	77

Capítulo 1

Introducción

Actualmente, millones de aplicaciones están disponibles en diferentes tiendas en línea para los usuarios de teléfonos inteligentes. Las aplicaciones móviles más exitosas se han descargado más de cien millones de veces [2] y cada día se lanzan nuevas aplicaciones al mercado móvil. Es tal el alcance de las apps móviles, que hoy en día, emprendedores, consultores independientes, organizaciones y empresas de todo tipo, integran en su plan estratégico la implementación de aplicaciones, conscientes de que gran parte del acceso a las mismas se hace a través de dispositivos móviles.

El crecimiento de las apps móviles ha dado lugar a que sus desarrolladores adapten metodologías de desarrollo o surjan nuevas propuestas que permitan cumplir con las particularidades que implica el desarrollo de apps móviles. Dichas particularidades pueden ser: ciclos de desarrollos cortos y frecuentes, cambios tecnológicos (plataformas, sistemas operativos, sensores, etc.), documentación limitada, requisitos y recursos específicos del equipo de desarrollo y del cliente o usuario final, entre otros; siendo todos estos sujetos de innovación constante [3].

A pesar de evidenciar la influencia de los diferentes factores mencionados anteriormente para la elección de un proceso de desarrollo de software en particular, se conoce poco de los procesos de desarrollo para aplicaciones móviles que existen, sus características y cómo o por cual decidir a la hora de su implementación.

En la actualidad, las fases que se adoptan para el desarrollo de aplicaciones móviles difieren según el proyecto o tecnologías implantadas, sin embargo, dichas fases tienden generalmente a iniciar con la concepción de la idea y finalizar con la comercialización del producto final. Contar con un proceso de desarrollo de aplicaciones móviles es fundamental, así el equipo de trabajo llega a identificar sus tareas, tiempos, recursos, objetivos, etc., logrando la elaboración eficaz y eficiente del producto, y cumpliendo con los requisitos del sistema de la mejor manera posible [4].

El producto final de software es el resultado de las varias actividades llevadas a cabo como lo son: asignación de roles al grupo implicado, definición de objetivos y actividades a cada integrante, ejecución de un conjunto de buenas prácticas para trabajar colaborativamente y en equipo, establecimiento del calendario de actividades, gestión de riesgos, entre otros. Al final, son lineamientos que deben adaptarse y estar acorde a los recursos disponibles y a los requerimientos del cliente [5] [6].

Además, es necesario recalcar que el desarrollo de aplicaciones móviles ha transformado diferentes mercados, por tanto, se le exige mejoras y adaptaciones continuas para cubrir nuevas necesidades y cambios tecnológicos. Esto representa retos significativos para el campo de las aplicaciones móviles, como lo es el diseño de interfaces de usuario para los diferentes tamaños de pantallas de los dispositivos móviles, el estrecho lazo entre la experiencia de usuario y las capacidades de los dispositivos, los métodos de interacción de los usuarios que brindan las plataformas móviles, arquitecturas, etc. [7].

Esta necesidad de velocidad de adaptación es una de las causas de la limitada documentación formal y científica en el campo del desarrollo de aplicaciones móviles. En la actualidad, se carece de documentación científica que refleje procesos de desarrollo centrados en las aplicaciones móviles y sus actividades [8].

1.1 Motivación

Los procesos de desarrollo de aplicaciones móviles a nivel de software son levemente establecidos por la industria actual. Un problema visible es identificar qué y cómo llevar a cabo un proceso que permita cumplir con todo lo determinado en cuanto al producto final: aplicaciones móviles.

La motivación principal de este trabajo surge al evidenciar que para el desarrollo de aplicaciones móviles existen insuficientes procesos específicos y que en la mayoría de casos son poco conocidos. Deficiencia que debe ser cubierta, aportando procesos de desarrollo para aplicaciones móviles que permitan a sus desarrolladores obtener productos de calidad y con recursos mínimos. Además, las metodologías propias del desarrollo de apps móviles, al ser poco conocidas o explotadas, se presencia una deficiente documentación de las mismas, que no permite la toma de decisión entre una u otra metodología que se acople correctamente a las necesidades de un proyecto.

1.2 Objetivos

Tras la comprensión de los procesos de desarrollo de software, los tipos de aplicaciones móviles existentes en el mercado mundial, y la compleja identificación de los procesos de desarrollo de aplicaciones móviles en la actualidad (carencia de documentación formal acerca de los mismos); se plantea identificar y caracterizar los procesos de mayor relevancia para aplicaciones móviles y las posibles razones que permitan inclinarse hacia uno u otro proceso.

Mediante casos de estudio de la industria y academia, se pretende establecer los procesos de desarrollo de aplicaciones móviles actuales aplicados de manera recurrente a nivel de software. Entrevistas y encuestas focalizadas permitirán la recopilación de información relevante para lograr este objetivo.

Por otra parte, como resultado del estudio de los modelos y metodologías de desarrollo de aplicaciones móviles, se procura brindar un modelo para la selección de una metodología de desarrollo de SW, de acuerdo a recursos disponibles y requisitos deseados dentro de un proyecto de desarrollo de SW.

Es así que los objetivos de la presente investigación planteados son los siguientes:

- Identificar los procesos y modelos de desarrollo de software aplicados a nivel de software en aplicaciones móviles como también los enfoques de desarrollo de aplicaciones móviles, a través de la búsqueda, selección y análisis de artículos científicos referentes al tema.
- Establecer cómo se desarrollan las aplicaciones móviles actualmente en la industria y academia, mediante el planteamiento de casos de estudio que proyecten los procesos y modelos de desarrollo de software utilizados en los escenarios móviles.
- Guiar en la selección de una metodología de desarrollo de SW oportuna, en base a diferentes parámetros/criterios como recursos disponibles y requisitos identificados de un proyecto de SW.

1.3 Estructura del documento

De acuerdo con los objetivos definidos para el presente proyecto, la primera fase consiste en el análisis de la importancia del mercado de dispositivos móviles, aplicaciones móviles y los procesos de desarrollo de este tipo de software.

El Capítulo 2 trata los antecedentes y los trabajos relacionados al presente trabajo. En antecedentes, se abarca el desarrollo de aplicaciones móviles, considerando que el tipo de aplicación móvil determina qué tipo de desarrollo se debe llevar a cabo. De igual manera, se identifican los procesos de desarrollo de software, tanto de metodologías no ágiles como ágiles, centrándose en los procesos de aplicaciones móviles. Y en cuanto a trabajos relacionados, se analizan los estudios prácticos sobre el desarrollo de aplicaciones móviles.

Los casos de estudio permiten identificar los procesos que se utilizan en la academia e industria actualmente, poder seguir de cerca las fases llevadas a cabo y los problemas que surgen en los mismos, motivo por el cual en el Capítulo 3, se recopila y analiza la información sobre los casos de estudio, gracias a resultados tomados por encuestas, entrevistas y otros métodos/técnicas de primer nivel.

Tras la investigación de los Capítulos 2 y 3, se toman las metodologías resultantes del caso de estudio, y en el Capítulo 4, se propone un modelo para la selección de una metodología de desarrollo de software, basándose en criterios propios del proyecto, a fin de brindar una guía a quienes requieran la selección de un proceso de desarrollo de SW según ciertos recursos y requerimientos.

Por último, el Capítulo 5, muestra las conclusiones del presente trabajo, estableciendo los logros alcanzados y lo que se podría incrementar a futuro o mejorar de la presente investigación

Capítulo 2

Antecedentes y trabajos relacionados

En este capítulo se presentan los enfoques de desarrollo de aplicaciones móviles, seguidos de los modelos, procesos y metodologías de desarrollo de aplicaciones móviles identificados en artículos científicos, y de mayor relevancia y frecuencia de uso. Finalmente, se analizan estudios sobre los procesos de desarrollo de aplicaciones móviles.

2.1 Enfoques de desarrollo de aplicaciones móviles

Las aplicaciones móviles son adoptadas con mayor frecuencia por empresas, quienes se ven en la necesidad de decidir el tipo de aplicación móvil a desarrollarse, pues mucho dependerá de la funcionalidad y alcance que establezcan y requieran de la app para seleccionar el enfoque de la misma.

Las aplicaciones nativas (construidas específicamente para un SO en particular), web para móviles (aplicaciones web con formato para dispositivos móviles), e híbridas (la combinación de los tipos de aplicaciones mencionados anteriormente) son los tres tipos de aplicaciones actualmente existentes, y presentan beneficios como limitaciones, por lo que ningún enfoque o tipo supera otro. Sin embargo, existen ventajas e inconvenientes que son visibles para cada uno según el escenario en el que se desee adaptar [9].

2.1.1 Desarrollo de Aplicaciones Móviles Nativas

Una aplicación nativa es diseñada específicamente para un sistema operativo [9]. Las aplicaciones nativas están disponibles en las principales tiendas, como *App Store* de Apple, *Google Play* de Android, *App World* de BlackBerry o *Windows Phone Marketplace* de Windows Phone, entre otras. Una vez descargada la aplicación, ésta se instala en el dispositivo móvil. Tras la instalación y primera ejecución realizada por el usuario, sin necesidad de contenedor o intermediario alguno, la aplicación se conecta con el sistema operativo del dispositivo móvil [10].

El desarrollo de aplicaciones nativas implica que sus desarrolladores, tras la escritura del código fuente, crean el ejecutable en formato binario que se pueda empaquetar junto con el resto de los recursos, como audio, imágenes, videos, y diferentes tipos de archivos declarados para el OS. Para todo esto, se requiere de herramientas, archivos y suministros específicos provistos por el distribuidor del OS, conocidos en conjunto como el *Software Development Kit* (SDK), conjunto de herramientas que hace posible compilar el código fuente para la plataforma de destino [11], del OS. Es decir, el SDK es propio de una plataforma, y cada OS presenta herramientas propias. Apple iOS cuenta con Xcode, siendo sus lenguajes Swift, C y C++; Android presenta su IDE, Android SDK, y su lenguaje es Java; Blackberry OS presenta el mismo lenguaje, Java, y su herramienta es el plug in BB Java Eclipse. Finalmente, Windows Phone cuenta con Visual Studio o Windows Phone como IDEs, y entre sus lenguajes principales C# y VB.NET (Tabla 1). Una debilidad para el desarrollo de las aplicaciones nativas, es el hecho de que el código que ha sido desarrollado para una plataforma no será compatible con otra. Por lo tanto, el desarrollo de aplicaciones nativas para múltiples OS requiere mayor esfuerzo y es más costoso [12].

Tabla 1. Sistemas Operativos Móviles: Herramientas

	<i>Apple iOS</i>	<i>Android</i>	<i>Blackberry OS</i>	<i>Windows Phone</i>
<i>Lenguajes</i>	Objective-C (recientemente Swift), C, C++	Java (algunos C, C++)	Java	C#, VB. NET, etc.
<i>Herramientas</i>	Xcode	Android SDK	BB Java Eclipse Plug-in	Visual Studio, Windows Phone
<i>Formato</i>	.app	.apk	.cod	.xap
<i>Tiendas</i>	Apple App Store	Google Play	BlackBerry App World	Windows Phone Marketplace

Fuente: IBM Software: El desarrollo de aplicaciones móviles nativas, web o híbridas

Además, con las aplicaciones nativas y su desarrollo es esencial señalar que este tipo de apps acceden libremente a las APIs, interfaz para un componente de software que puede ser invocado a una distancia a través de una red de comunicaciones utilizando tecnologías basadas en estándares [13], disponibles por parte del OS, logrando hacer uso de funciones propias del SO. Existen dos tipos API, alto nivel y bajo nivel. El API de alto nivel ofrece servicios mediante procesos, tales como: navegación por internet, gestión de calendario, fotos,

etc., envío y recepción de mensajes, recepción de notificaciones, entre otros. El API de bajo nivel, mediante acciones, interacciona con el HW, permitiendo: mostrar imágenes, procesar audio, reproducir sonidos, capturar videos y fotos, mostrar ubicación, etc. Es así, que se puede visualizar (Figura 1) como la interacción entre las APIs y el dispositivo móvil, da lugar a diferentes servicios, según su tipo [10].

Las aplicaciones nativas presentan ventajas y desventajas, como aprovechar las características propias del hardware de dispositivo móvil (GPS, cámara, sensores, etc.), o requerir mayor uso de recursos si se requiriera la app para diferentes plataformas. Estos y más pros y contras se detallan en la tabla 2 [9].

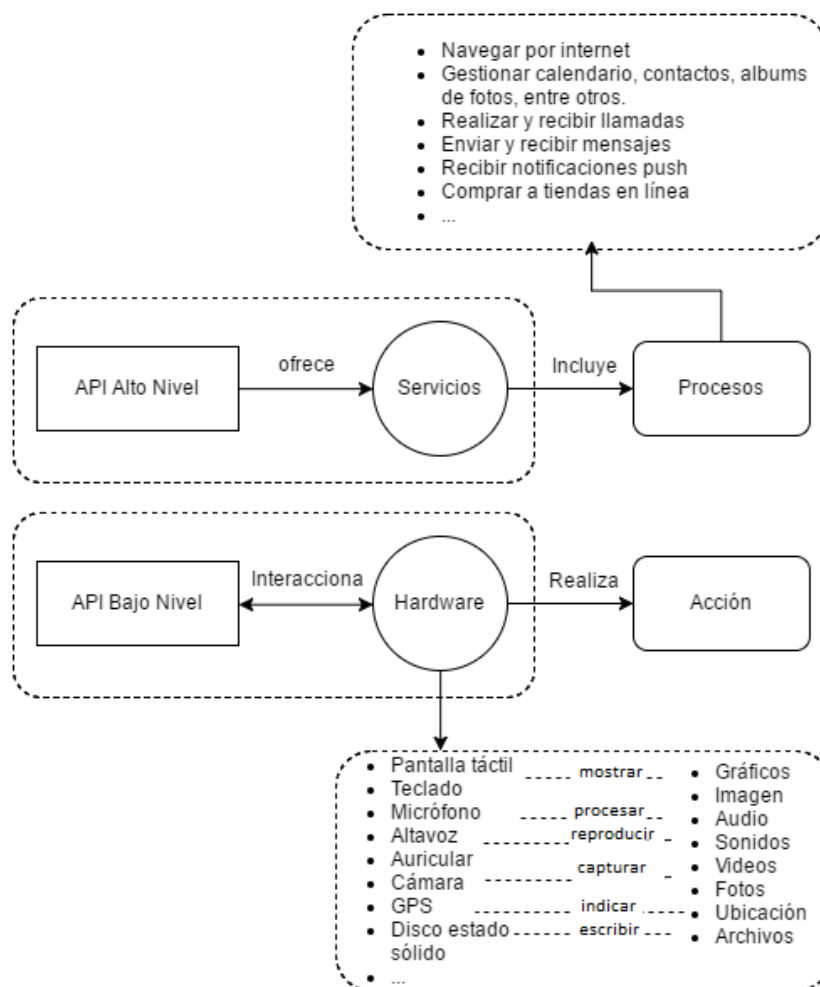


Figura 1. Aplicaciones nativas: APIs Bajo y Alto Nivel.

2.1.2 Desarrollo de Aplicaciones Móviles Web

Las aplicaciones móviles web se ejecutan dentro de un navegador del dispositivo móvil y se construyen en base a tres tecnologías esenciales: HTML, que permite visualizar el contenido, tanto imágenes como texto; CSS, que permite definir el estilo con el que contará la primera tecnología mencionada, y finalmente JavaScript, que se encarga de lograr realizar y presentar animaciones y la interacción misma del usuario con la aplicación móvil web (Figura 2).

Tabla 2. Aplicaciones móviles nativas: ventajas y desventajas.

<i>Aplicaciones Móviles Nativas</i>		
<i>Ventajas</i>	Experiencia de usuario máxima	Apps nativas aprovechan las capacidades del dispositivo móvil, el hardware incorporado (GPS, cámara, etc.) y el software (administrador de archivos, calendario, contactos, galería de imágenes, etc.)
	Facilidad de monetización	Se establece un precio por la app en una tienda de aplicaciones y, cuando los usuarios la compran, se gana dinero inmediatamente (menos comisión).
	Ejecución sin conexión	Al estar instalada la app en el dispositivo, no requiere conexión a Internet. La transferencia de datos puede continuar cuando se restaura una conexión de datos.
	Visibilidad constante	Las apps nativas colocan un logotipo en la pantalla de aplicaciones del usuario. Además, las tiendas de apps recomiendan a los usuarios actualizarlas con cierta frecuencia.
<i>Desventajas</i>	Mantenimiento y/o actualización.	Se requiere desarrollo, pruebas y distribución para diferentes plataformas, lo que significa uso de recursos por parte de la empresa en reiteradas ocasiones (costos, recursos humanos, tiempo, entre otros).
	Información de suscriptores	Los editores de contenidos comparten la información sobre sus suscriptores con la tienda de aplicaciones.

Actualmente, los navegadores vienen por defecto en teléfonos inteligentes y tabletas, y brindan soporte a nuevas funcionalidades como CSS3¹, HTML5¹ (ofrece disponibilidad offline, servicios de geolocalización, eventos táctiles, etc.), entre otros [9].

¹ CSS3: lenguaje para definir estilo o apariencia de páginas web, escritas con HTML o de los documentos XML. CSS3 cubre diseño, estilos web, forma y movimiento [49].

Al ingresar a sitios web que cuenta con una versión web, el sitio reconoce que el acceso se realiza desde un dispositivo, por lo que presentan páginas HTML diseñadas para una mejor experiencia en pantallas de varios tamaños, que generalmente hoy en día son táctiles. Y es gracias a herramientas JavaScript como Sencha Touch o jQuery Mobile, que los desarrolladores logran construir interfaces de usuario similares a las aplicaciones nativas [9].

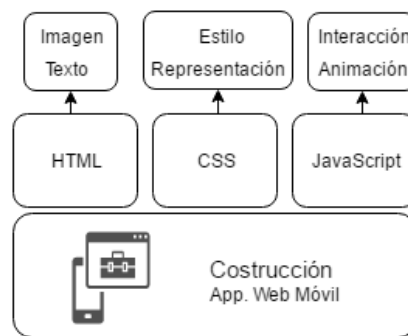


Figura 2. Construcción App. Web: Tecnologías esenciales

Existen desarrolladores que logran asemejar una app web a una app nativa a tal punto de presentar accesos directos para las aplicaciones webs, dejando atrás la desventaja que solían presentar las aplicaciones web, como era el no contar con logotipo (acceso directo) en la pantalla de aplicaciones del usuario.

Por otra parte, las aplicaciones web permiten el soporte para múltiples plataformas, hecho que se refleja en el bajo coste del desarrollo, convirtiéndose éste en su punto de robustez. Sin embargo, este punto se ve algo opacado si nos referimos a APIs. Las APIs libres y disponibles para las aplicaciones web son escasas, por lo que varias de sus funcionalidades son parciales o no accesibles para las apps web. Esto podría cambiar a futuro gracias a los avances que presente HTML.

Las aplicaciones móviles web presentan ventajas y desventajas; una gran ventaja es ser multiplataforma, por lo que pueden llegar a todo tipo de dispositivo, y por tanto a más usuarios; y un limitante fuerte es requerir conexión para su total funcionamiento. Existen

¹ HTML5: *HyperText Markup Language 5*, provee tres características: estructura, estilo y funcionalidad. Incluso cuando algunas APIs y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript [50].

ventajas adicionales, como requerir menor uso de recursos para su desarrollo, y que el mantenimiento sea de bajo costo y se lo pueda realizar de inmediato, sin embargo, esto se ve opacado con los navegadores para dispositivos móviles que son limitados y presentan restricciones con el HW (funciones de cámara, GPS, etc., que no están disponibles o no son compatibles) (Tabla 3) [12].

2.1.3 Desarrollo de Aplicaciones Móviles Híbridas

La combinación de las apps web y apps nativas, da como resultado las aplicaciones híbridas, que presentan código nativo y web, y hacen uso de APIs propios del dispositivo, tal como lo muestra la figura 3. Para los desarrolladores esto implica menos trabajo, puesto que no necesitan escribir una aplicación para cada plataforma móvil, al reutilizar desde diferentes dispositivos móviles parte de código HTML, CSS y JavaScript. Y todo esto sin que el usuario note la diferencia con una aplicación nativa, puesto que una app híbrida está disponible en tiendas de aplicaciones para su posterior descarga y almacenamiento en el dispositivo móvil [10] [12].

Tabla 3. Aplicaciones móviles web: ventajas y desventajas.

<i>Aplicaciones Móviles Web</i>		
<i>Ventajas</i>	Compatibilidad entre plataformas	Logran alcanzar un público más extenso. Siendo estandarizados los navegadores web móviles, es más fácil crear una app web móvil universal que una app nativa.
	Menor uso de recursos	Crear una app web móvil resulta de menor costo, más fácil e incluso menor tiempo por tratarse de un único desarrollo para todas las plataformas.
	Acceso simple	Usuarios no descargan la app, simplemente acceden a la URL a través del navegador del dispositivo móvil, entregando la aplicación más actualizada.
	Mantenimiento inmediato y de bajo costo	Mantenimiento multiplataforma, es decir, un solo mantenimiento que cubre a todas las plataformas, reduciendo costos.
<i>Desventajas</i>	Requiere conexión	Por la conexión a internet, la app web puede ser lenta o no estar disponible. Sus usuarios pagan por el tiempo de conexión a los operadores de red al no contar con cobertura Wi-Fi. Sin embargo, HTML5 resolvió estos inconvenientes al permitir la disponibilidad del contenido de la aplicación web en modo sin conexión.
	Navegadores móviles limitados	No son robustos como lo son los navegadores de escritorio, dando un aspecto torpe a las apps web.
	Acceso a HW y SW restringidos y/o limitados	Funciones de la cámara, GPS, o juegos e imágenes de alta resolución que no están disponibles o no son compatibles, descartarán las aplicaciones web inmediatamente.

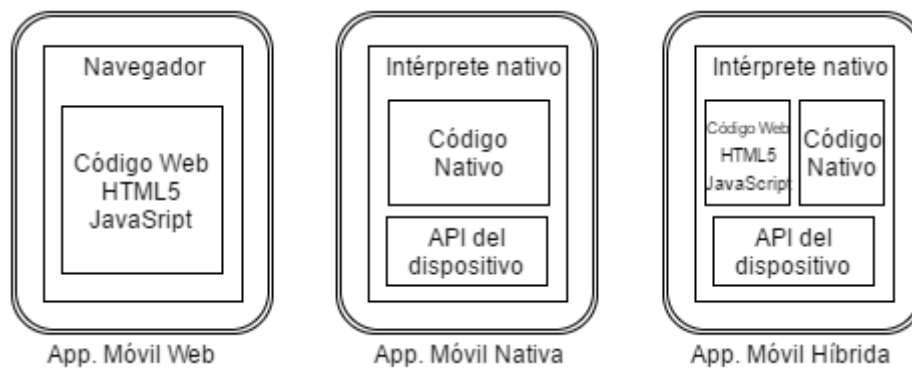


Figura 3. Aplicaciones Móviles: Composición

Una app híbrida presenta en su totalidad o gran porcentaje la interfaz de usuario en el navegador, complementándose con código nativo que proporcione acceso a las APIs del dispositivo [10]. Las APIs de los sistemas operativos a las cuales accede la parte nativa de una app híbrida, permite conectar al navegador y las APIs del dispositivo móvil a través de un motor de búsqueda HTML.

La aplicación híbrida cuenta con una página web, conjunto de archivos HTML, JavaScript, CSS y recursos audiovisuales; ya sea que resida en un servidor o incorporados en el código de la aplicación. Las dos opciones se muestran en la tabla 4, mostrando cómo favorece o no el alojamiento de los recursos HTML en un servidor web o en el dispositivo móvil [9] [10].

Debido a las grandes ventajas de este tipo de aplicaciones, los desarrolladores de apps híbridas pueden optar por codificar su puente, entre la parte web y nativa; o bien aprovechar soluciones ya construidas, conocidas como *frameworks* (Tabla 5).

Existen varios marcos de trabajo para el desarrollo de aplicaciones móviles híbridas, y es importante identificar que tecnologías se usarán, si este es de código abierto, si se desea que la apariencia de la app se asemeje a una app nativa o más bien a una página web, el rendimiento que brinde o la plataforma en la que se desea desplegar la app; y a partir de las necesidades identificadas, discernir entre los *frameworks* existentes.

Tabla 4. Página web de aplicación híbrida: Ventajas y desventajas según su locación.

Página Web de Aplicación híbrida		
Locación	Ventajas	Desventajas
Reside en Servidor	Los desarrolladores actualizan en la aplicación sin seguir el proceso de ciertas tiendas de aplicaciones: entrega y aprobación de la app.	El contenido no es accesible cuando el dispositivo no está conectado a la red: elimina la disponibilidad offline.
Incorporada en código de app.	Mejora el desempeño y la accesibilidad de la aplicación.	Los desarrolladores no pueden realizar actualizaciones remotas.

Tabla 5. Apps. Móviles Híbridas: Frameworks de Desarrollo

Frameworks de Desarrollo de Aplicaciones Móviles Híbridas		
N.	Framework	Descripción
1	Ionic	<i>Framework</i> de código abierto para apps móviles HTML5 que puede implementarse para desarrollo de apps móviles híbridas. Está construido sobre Angular y requiere de Cordova para su funcionamiento en móviles. Utiliza componentes JavaScript, CSS, Cordova y Angular (permite arquitectura MVC). Soporta HTML5, CSS3, versiones iOS7 plus y Android 4.1 plus con ayuda de <i>CordovaBlessed</i> y otros complementos. Para un desarrollador, en Ionic una aplicación de inicio presenta plantillas, lo cual facilita su uso [14].
2	Mobile Angular UI	<i>Framework</i> que permite crear aplicaciones móviles híbridas HTML5 y aplicaciones de escritorio con twitter <i>Bootstrap</i> y Angular JS. No tiene dependencia de jQuery, gracias a bibliotecas como fastclick.js y overthrow.js que brindan una mejor experiencia móvil [15].
3	Intel XDK	<i>Intel Cross Platform Development Kit</i> , Intel XDK, siendo un IDE (kit de desarrollo integrado) libre, provee facilidades para crear aplicaciones <i>Cordova</i> para plataformas iOS, Android y Windows. Intel XDK está disponible para los principales sistemas operativos de escritorio que son OS X, Windows y Linux. Además, proporciona facilidades como la edición de código, emulador incorporado, depuración y creación de nubes de aplicaciones <i>Cordova</i> . Intel XDK genera la aplicación en sus servidores de nube en lugar de construirlo localmente [16].
4	Titanium Mobile	Marco de código abierto creado y licenciado por <i>Appcelerator</i> bajo la licencia Apache 2, basado en un entorno de ejecución autónomo, compatible con iOS y Android únicamente. Las aplicaciones construidas con Titanium Mobile tienen la apariencia de las aplicaciones nativas con ciertas limitaciones en cuanto a rendimiento, perjudicando la experiencia del usuario. Estas aplicaciones se escriben en JavaScript. Titanium permite al desarrollador construir aplicaciones multiplataforma con una sola base de código [10].
5	Sencha Touch 2	<i>Framework</i> diseñado especialmente para construir aplicaciones móviles para dispositivos con pantalla táctil. Siendo una completa plataforma, entre sus características relevantes

		presenta: widget para biblioteca de interfaz de usuario, aplicación de la arquitectura MVC y sistema de compilación de línea de comandos [17].
6	Kendo UI	Kendo UI, de <i>Telerik</i> , es un framework HTML 5 que permite crear aplicaciones móviles multiplataforma. Kendo UI depende en gran medida de <i>jQuery</i> y tiene una serie de widgets basados en el mismo. Para los desarrolladores que se encuentran familiarizados con <i>jQuery</i> , Kendo UI resulta fácil de aprender. Kendo UI ha abierto la mayoría de las herramientas y funciones de JavaScript; pero gran parte de los widgets de uso común están bajo licencia comercial [18].
7	PhoneGap	Distribuido por Apache <i>Cordova</i> , <i>PhoneGap</i> ofrece acceso genérico a las funciones de plataformas específicas en todas las principales plataformas móviles. En cuanto a las aplicaciones construidas con <i>PhoneGap</i> , su apariencia visual y, en menor medida, el comportamiento, no reflejan una apariencia nativa, sino más bien la de un sitio web [10].
8	Framework 7	<i>Framework</i> de código abierto bajo licencia MIT y gratuito, que permite a desarrolladores crear aplicaciones iOS con tan solo CSS, JavaScript y HTML, tecnologías ya conocidas por los desarrolladores web. Este framework ofrece características como: retroceso rápido, desplazamiento nativo, animación de alto rendimiento, entre otros, que logran una aproximación notoria a las aplicaciones nativas de iOS [19].
9	jQuery Mobile	Sistema de interfaz de usuario basado en HTML5 que permite crear aplicaciones web y aplicaciones móviles que no presenten problemas en su ejecución, sin centrarse en la apariencia, puesto que las aplicaciones con <i>jQuery Mobile</i> no se asimilan a las apps nativas de plataformas como iOS o Android. <i>jQuery Mobile</i> se basa en <i>jQuery</i> y <i>jQuery UI</i> , y ofrece navegación Ajax con transiciones de página, eventos táctiles y varios widgets [20].
10	Onsen UI	<i>Framework</i> de código abierto, disponible bajo licencia de Apache. Onsen UI proporciona interfaces de usuario y herramientas para la creación rápida de aplicaciones móviles híbridas basadas en <i>PhoneGap/Cordova</i> . Onsen UI permite escribir aplicaciones en HTML5 y JavaScript y pasar a <i>Phonegap & Cordova</i> para presentarlas como nativas [21].
11	React Native	<i>Framework</i> que permite crear aplicaciones móviles multiplataforma (Android e iOS) utilizando sólo JavaScript. Utiliza el mismo diseño que React, lo que le permite componer una interfaz de usuario móvil rica a partir de componentes declarativos. <i>React Native</i> brinda un rendimiento de tipo nativo convirtiéndose en una gran ventaja del framework [22] [23].
12	NativeScript	Permite escribir la funcionalidad de la aplicación móvil en JavaScript, implementar aplicaciones móviles nativas para iOS, Android y Windows desde una sola base de código. Este marco de trabajo requiere algunas habilidades de codificación, pero lo compensa con una extensa documentación en profundidad [24].

Tomando únicamente tres puntos de referencia, apariencia, rendimiento y plataforma se presenta una variedad de pros y contras con los marcos de trabajo como: PhoneGap da a sus aplicaciones la apariencia de un sitio web, Titanium Mobile brinda la apariencia de aplicación nativa, pero con limitaciones en cuanto a rendimiento, mientras que React Native y Framework 7, permite obtener una aplicación con apariencia y rendimiento de tipo nativo, para Android e iOS y únicamente para iOS respectivamente. De esta manera se puede visualizar, que los *frameworks* presentan diferencias tanto en tecnologías como en el resultado de las aplicaciones desarrolladas, y depende estrictamente de las características en cuanto al desarrollo a llevarse a cabo como en el producto de software que se desea obtener para la selección de un *framework* u otro.

2.1.4 Comparación de los enfoques de aplicación móvil y sus posibles escenarios

Para seleccionar un tipo de aplicación móvil que vaya acorde a lo que le conviene o necesita un cliente, se recomienda tomar en cuenta las características de los tipos de aplicaciones móviles. Las aplicaciones web pueden presentar un lenguaje nativo, como web y nativo simultáneamente; las aplicaciones híbridas son multiplataforma, de mayor portabilidad, flexibilidad, y reutilizan código existente, mientras que las aplicaciones nativas utilizan de la mejor manera los recursos de los dispositivos, siendo a su vez una limitación en cuanto a portabilidad. La comparación de las características relevantes de los tipos de aplicaciones móviles se presenta en la Tabla 6 [12].

Tabla 6. Comparación de los tipos de aplicaciones móviles: Características

<i>Comparación de los tipos de Aplicaciones Móviles</i>			
<i>Característica</i>	<i>Aplicación nativa</i>	<i>Aplicación Web</i>	<i>Aplicación Híbrida</i>
Lenguaje	Nativo	Nativo y web, o nativo únicamente	Web
Portabilidad	Bajo	Alto	Alto
Gráficos avanzados	Alto	Mediano	Mediano
Flexible para actualización	Bajo	Mediano	Alto
Experiencia de instalación	Alta	Alta	Mediana
Uso conocimiento existente	Bajo	Alto	Alto
Soporte múltiple plataforma	Nulo	Medio	Alto

Además de las características propias de cada tipo de aplicación móvil (Tabla 6), es de suma importancia identificar el escenario en el que la aplicación móvil se desarrollará. En la tabla 7 se pueden apreciar los escenarios y el tipo de aplicación que se ajusten a los mismos [9] [10].

Tabla 7. Aplicaciones móviles: escenarios favorecedores

<i>Aplicaciones móviles y sus escenarios favorecedores</i>		
<i>Tipo</i>	<i>Escenario</i>	<i>Descripción</i>
Aplicación móvil nativa	Único SO móvil	Si el alcance de la aplicación móvil es limitado y se decide llegar a un mercado específico que presenta un tipo de dispositivo específico; esto implica que lograr una cobertura para múltiples plataformas no sea una prioridad y desarrollar una única aplicación nativa permite ahorro de recursos, siendo ideal este tipo de aplicación móvil.
	Funcionalidad nativa	Una aplicación puede basarse en una funcionalidad específica, funcionalidad que puede involucrar sensores o protocolos que requieran un desarrollo nativo. Ejemplificando, Skype hace uso del protocolo voz sobre IP y requiere el acceso a los contactos del usuario, y tomando en cuenta las tecnologías disponibles actualmente, el caso ideal es una aplicación móvil nativa. Por otro lado, en cuanto a la capacidad de respuesta en tiempo real, las tecnologías Web no ofrecen una solución adecuada por el momento. Ante esta necesidad, a los desarrolladores les conviene el desarrollo nativo.
Aplicación móvil web	Distribución y actualización directa	Evitar las tiendas de aplicaciones descarta solicitar aprobación o distribución de las apps por su parte, permitiendo a la organización responsable del desarrollo, el control de la distribución y actualización (periodos de actualización a libre decisión) de la app. Esto es posible con las aplicaciones móviles web.
	Aplicación piloto	Crear una aplicación móvil web es de menor costo y tiempo con respecto a la aplicación móvil nativa. Optar por un desarrollo móvil web permite obtener una versión piloto que, tras un estudio de mercado, puede mejorarse o expandirse; y de principio es ideal como táctica de mercado. Incluso, si se viera necesario una app móvil híbrida, se puede hacer uso del código existente.
	Visibilidad	Si bien una aplicación móvil puede exponerse a través de las tiendas de aplicaciones, los resultados de los buscadores logran un mayor alcance, que permiten la visualización de las aplicaciones móviles web a una audiencia más grande.

Aplicación móvil híbrida	Menos uso de recursos	El proceso de desarrollo y de mantenimiento continuo de una aplicación móvil híbrida se lo realiza en menor tiempo y costo, debido a que, gracias al puente nativo, los desarrolladores aprovechan al máximo todas las características y funcionalidades de los dispositivos móviles y, por otra parte, las porciones del código con lenguajes Web se logran compartir entre las distintas plataformas móviles.
	Conocimiento previo	El desarrollo web involucra habilidades generalmente comunes y sencillas que presentan amplia documentación. Al decidirse por una aplicación móvil híbrida, con la solución correcta, los desarrolladores web son capaces de construir aplicaciones únicamente con habilidades web, como HTML, CSS y JavaScript, además de ofrecer al usuario una experiencia similar a la nativa.
	Expectativas futuras	Al contar con una gran parte desarrollada en HTML en una aplicación móvil híbrida, los desarrolladores se aseguran de que lo realizado hoy no será obsoleto para mañana, puesto que HTML así lo permite, gracias a su aumento rápido en cuanto a disponibilidad, funcionalidad y sofisticación.

Es por demás visible que los tipos de aplicaciones presentan sus pros y contras, pero vale resaltar las características sobresalientes de cada uno. En cuanto a las aplicaciones móviles nativas su mayor ventaja es el acceso a las funcionalidades y/o componentes propios del dispositivo. En el caso de las aplicaciones móviles web, éstas logran ser actualizadas de manera inmediata y desde una única interfaz, siendo esta su mayor fortaleza. Y, por último, las aplicaciones móviles híbridas son robustas al ser multiplataforma, por tanto, no requieren varios desarrollos según el número de sistemas operativos.

2.2 Modelos y Metodologías de Desarrollo de Aplicaciones Móviles

Para llevar a cabo la realización de un sistema, es indispensable contar con una serie de pasos, que permita obtener un excelente producto o el mejor, si fuera posible. Estos pasos en conjunto ordenados son denominados proceso de software [12]. Un proceso de software se ajusta según las necesidades y características del proyecto a llevarse a cabo, por los ingenieros de software o líderes de proyecto. Al final, este proceso amoldado será la referencia de todo el equipo de trabajo, incluyendo a quienes han solicitado el sistema.

Vale mencionar la importancia de contar con un proceso de software, puesto que, tras su definición y constante seguimiento, brinda organización, control y estabilidad a todas las actividades a realizar. En caso de no contar con un proceso de software, bien se podría presentar caos en la elaboración del producto software, en cualquiera o todas sus actividades [25].

Si bien todo proceso se puede modificar a fin de lo que se requiera, existen modelos que funcionan como representaciones o plantillas, de modo que se pueda generalizar y establecer buenas prácticas de la ingeniería de software, conocidos como modelos de proceso del software [9]. Sommerville [25] define modelo de proceso de software como “Una representación simplificada de un proceso de software, representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto, un modelo de procesos del software es una abstracción de un proceso real.”

Por otra parte, un proceso de software detallado y completo se denomina “Metodología”. Las metodologías se basan en una combinación de los modelos (cascada, evolutivo, incremental, etc.), con la definición precisa de los roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías para uso de herramientas de apoyo, entre otros [12].

2.2.1 Modelos y Metodologías No Ágiles de Desarrollo de Aplicaciones Móviles

Los métodos no ágiles se enfocan en la planificación global del proyecto, y una vez que se ha especificado el proyecto, inicia el ciclo de desarrollo del software. Estos métodos se centran en el control del proceso mediante el establecimiento de roles, actividades, herramientas, y artefactos como modelado y documentación. Estos esquemas "tradicionales" han demostrado abordar correctamente el desarrollo de software que requiere amplia cantidad de tiempo y recursos [5].

Sin embargo, estos métodos no resultan adecuados a proyectos que, deseando mantener un proceso de alta calidad, cuentan con entornos volubles y limitaciones drásticas para los tiempos de desarrollo. Esta falta de flexibilidad y restricción de recursos que presentan las metodologías no ágiles conllevan dificultades a los equipos de desarrollo [8].

A continuación, los procesos o modelos de desarrollo de apps móviles que se guían en base a metodologías no ágiles o conocidas también como metodologías tradicionales.

Modelo en Cascada

El modelo en cascada, *waterfall*, se basa principalmente en una secuencia de fases: análisis de requisitos, diseño del sistema, diseño del software, codificación, pruebas, implementación del SW, y su mantenimiento [5].

A continuación, la ampliación de sus fases:

1. Análisis y definición de requisitos: se realizan diferentes actividades, como *focus groups*, a fin de recolectar la información necesaria acerca del sistema realizarse, de manera que se pueda contar con requerimientos.
2. Diseño del sistema y del software: se establece la arquitectura completa del sistema, describiendo lo esencial del sistema SW y sus relaciones.
3. Implementación y prueba de unidades: tras la implementación es necesario realizar una serie de pruebas que permitan verificar el cumplimiento de cada una según lo esperado y previamente especificado.
4. Integración y prueba del sistema: Tras la integración de las unidades que conforman el SW en sí, se requiere asegurar que en conjunto cumplan con los requerimientos del mismo. Una vez que el sistema ha sido probado y los resultados satisfactorios, se entrega el sistema software al cliente.
5. Funcionamiento y mantenimiento: Es en esta fase en la que se instala el sistema. Si hubiera errores previamente no identificados, en esta fase se deberán corregir.

Vale mencionar, que en cada fase se genera la documentación necesaria, lo cual puede convertirse en una ventaja de la metodología; sin embargo, el punto fuerte a considerar para optar por esta metodología es la definición de los requerimientos, puesto que se deberá tener certeza que estos no cambiarán radicalmente en el proceso, caso contrario, las iteraciones habrán consumido varios recursos, y se requerirá repetir etapas previas del proceso [25].

Desarrollo Rápido de Aplicaciones (*RAD - Rapid Application Development*)

Desarrollo rápido de aplicaciones es una metodología de desarrollo de software que utiliza una planificación mínima para obtener un prototipo, siendo este un modelo de trabajo que es funcionalmente equivalente a un componente del producto [26].

Los proyectos RAD siguen un modelo iterativo e incremental que cuentan con pequeños equipos compuestos por desarrolladores, expertos en dominios, representantes de clientes y otros recursos de TI que trabajan progresivamente en su prototipo. El aspecto más importante para que este modelo tenga éxito es asegurarse de que los prototipos desarrollados son reutilizables [26].

RAD enfatiza la participación de los usuarios y la construcción de prototipos para probar técnicas y refinar los requisitos. Una vez definido el alcance del proyecto en la fase de planificación de necesidades, los usuarios interactúan con los analistas del sistema para desarrollar prototipos en la fase de diseño del usuario. Los usuarios aportan una retroalimentación directa durante la fase de construcción. Finalmente, como última fase los usuarios finales son entrenados para usar el producto tal como está desplegado [26].

El éxito del desarrollo rápido de aplicaciones depende de la participación de personas con las habilidades y talentos adecuados [26].

Modelo Incremental

El modelo incremental, al final de cada iteración, entrega un software completamente operacional. Este modelo inicia con el desarrollo de la arquitectura completa del sistema (necesario conocer los requerimientos completos desde la primera fase), seguido de incrementos funcionales sucesivos. Cada incremento tiene su ciclo de vida propio y se basa en el ciclo anterior. Tras cada ciclo, al obtener un incremento, en este se realiza exclusivamente corrección de errores, más no cambios [5].

El modelo incremental del marco de trabajo previamente mencionado, presenta tres ciclos: modelar, evaluar, e implementar.

La fase del modelado consiste en comprender las metas deseadas; definir qué parámetros se pueden utilizar para cuantificar y comunicar qué tan bien se están cumpliendo los objetivos; y

mediante la calendarización de los objetivos y establecimiento de métricas, generar informes para la visualización de los resultados obtenidos [27].

La fase de implementación involucra la configuración de la base de datos, el cálculo de métricas y fechas definidas e identificadas en el modelo; la configuración de formularios e informes para la interfaz de usuario de la aplicación móvil; y el mapeo sistemático de la aplicación móvil como tal [5] [27].

La fase de evaluación permite probar la usabilidad de la mediante la observación y manejo de la app por parte de los usuarios; y mediante la definición de ingeniería y listas de control de dominio que se pueden utilizar para cuantificar sistemáticamente la medida en que la aplicación está cumpliendo con los criterios previamente establecidos. Un análisis de las sesiones “pensar en voz alta” y listas de control se utiliza para entender cómo mejorar el modelo y la aplicación en la siguiente iteración del desarrollo [27].

Prototipado

El modelo prototipado o *prototype*, permite a los desarrolladores crear partes de la solución para demostrar funcionalidades y llevar a cabo mejoras necesarias antes de desarrollar la solución definitiva [28].

Este modelo se aplica generalmente cuando el cliente conoce y define los objetivos generales para el software que se desea desarrollar, sin embargo, no tiene la certeza o manera de afirmar que este funcionará, ya sea porque desconoce la eficacia de un algoritmo, de la adaptabilidad del sistema o de la forma en que interactúa el hombre y la máquina, entre otros posibles motivos. El prototipo logra que el grupo de desarrollo y el cliente comprendan de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos [29].

Resumen

En resumen, si bien existen más modelos y metodologías no ágiles, las expuestas anteriormente se han aplicado en el desarrollo de apps móviles y sobresalen ciertas características de cada modelo o metodología. El modelo en cascada ordena las etapas del proceso para el desarrollo de software de manera rigurosa, promoviendo una metodología de trabajo ordenada y efectiva, puesto que para que una etapa inicie, habrá que dar por

finalizada la etapa previa, y su característica relevante es que es aplicable el modelo en cascada siempre y cuando los requisitos estén establecidos totalmente.

En cuanto a la metodología RAD, si bien se asemeja a cascada debido al proceso lineal secuencial, se diferencia por su ciclo de desarrollo extremadamente corto. Por otra parte, el modelo incremental utiliza el principio de trabajo en cadena o Pipeline, involucrando al cliente quien acepta o rechaza elementos al final de cada incremento a fin de que el software se adapte mejor a sus necesidades. Y finalmente se tiene el modelo prototipado, mismo que permite a los usuarios evaluar las propuestas de desarrolladores y probarlas previo a su implementación. También ayuda a comprender los requisitos que son específicos del usuario y no pueden haber sido considerados por el desarrollador durante el diseño del producto.

De esta manera, es visible como cada modelo o metodología presenta su singularidad que puede favorecer o no al desarrollo de SW, sin embargo, las metodologías no ágiles no presentan características que limiten o especifiquen que tipo de software se puede desarrollar bajo sus lineamientos, permitiendo el desarrollo de todo tipo de SW, incluyendo apps móviles. Sin embargo, esta apertura puede no favorecer en el caso de desarrollo de aplicaciones móviles, pues se podría dejar de lado muchos puntos que se deben considerar al desarrollar aplicaciones para dispositivos móviles, como el contar con sensores propios del dispositivo, el tipo y tamaño de las pantallas, diseño de la interfaz y la selección del tipo de aplicación (web, nativa, etc.), entre otros.

2.2.2 Modelos y Metodologías Ágiles de Desarrollo de Aplicaciones Móviles

Los métodos ágiles nacen como respuesta a potenciar el desarrollo de software, en lugar de su planificación. Estos métodos dan mayor importancia a la capacidad de respuesta, en lugar del seguimiento estricto de un plan. Estas propuestas nacieron puesto que se requiere en muchos proyectos de mayor flexibilidad, de permitir cambios sin que implique un costo excesivo [30].

Las metodologías ágiles ofrecen un desarrollo más ligero y rápido. Han logrado un alto nivel de popularidad y el éxito en la industria de software reconociendo que una buena relación cliente / desarrollador es crucial para el éxito en el desarrollo de software [31].

A continuación, los procesos, modelos y metodologías de desarrollo de apps móviles que se guían en base a metodologías ágiles.

Programación Extrema (XP – *eXtreme Programming*)

La programación extrema, siendo una metodología ágil, se caracteriza por la aceptación de cambios en los requisitos sobre la marcha del desarrollo de un software, considerado como un aspecto natural por quienes defienden esta metodología. La adaptación que permite XP es considerada una aproximación a la reducción de esfuerzos y recursos para definir los requisitos de un proyecto y re definirlos como se requiera a lo largo de su trayecto [30].

El ciclo de vida de XP consiste de seis fases: Exploración, Planificación, Iteraciones por entrega, Producción, Mantenimiento y Muerte [26] [30].

En la fase de Exploración, los clientes realizan las *story cards* que se requiere para la primera entrega, en donde se describen las funcionalidades que se desea del SW. A su vez, el equipo de desarrollo se relaciona con las herramientas, la tecnología y las prácticas a ser utilizadas durante el proyecto [26].

En la segunda fase, Planificación, se establece la prioridad de cada una de las *stories* y se define el contenido de la primera entrega. Se estima el esfuerzo de cada *story*, de esta manera se define un cronograma. En la siguiente fase, Iteraciones por entrega, se realizan iteraciones del SW antes de la entrega del primer *release*. El cliente decide que *stories* van a ser implementadas para cada iteración. Además, se realizan test funcionales, llevadas a cabo por el cliente al final de cada iteración [26] [30].

En cuanto a la fase Producción, en esta se requiere realizar pruebas antes que el SW sea entregado al cliente. En esta fase, si aparecieran cambios se decide si se incorporarán o no en la presente entrega. Las ideas y sugerencias propuestas son documentadas para ser implementadas posteriormente. Tras la creación y entrega del primer *release*, el proyecto debe mantener el SW en producción, mientras se trabaja en iteraciones nuevas [26].

La siguiente fase corresponde al Mantenimiento, en la cual los programadores deben satisfacer los requerimientos del cliente, exigiendo un poco más al equipo de desarrollo, o resultando en la necesidad de contratar más personas que apoyen con esta fase.

Finalmente, en la fase Muerte el cliente no tiene ninguna *story* a ser implementada. Esta es la etapa en la cual no hay más cambios en la arquitectura, el diseño o el código y aquí es cuando se realiza la documentación correspondiente [26].

Proceso Unificado ágil (AUP - *Agile Unified Process*)

AUP es una versión simplificada de RUP combinada con Agile Method (AM). AUP presenta iteraciones, en las cuales se define las actividades que los miembros del equipo de desarrollo realizan para construir, validar y entregar el software que satisfaga las necesidades de los usuarios [32].

Las cuatro fases de AUP, son: 1) Iniciación, siendo su objetivo el identificar el alcance inicial del proyecto, una arquitectura potencial, y obtener el financiamiento inicial del proyecto y la aceptación de las partes interesadas. 2) Elaboración, permite definir la arquitectura del sistema. 3) Construcción, como su nombre lo indica, se logra la construcción del sistema, mismo que deberá satisfacer las necesidades de las partes interesadas. 4) Transición, de manera que se valide e integre el sistema con el entorno de uso [32].

Scrum

Jeff Sutherland obtuvo el nombre de “Scrum” del artículo “*The New New Product Development Game*”, publicado por Takeuchi y Nonaka en la revista *Harvard Business* en 1986. En el artículo escriben: “Cada vez más, las empresas se dan cuenta de que el viejo y secuencial enfoque para desarrollar nuevos productos simplemente no va a hacer el trabajo. En su lugar, las empresas en Japón y los Estados Unidos están usando un método holístico - como en el rugby, la bola se pasa dentro del equipo, ya que se mueve como una unidad en el campo” [33].

Scrum, siendo una metodología ágil y flexible, tiene como objetivo maximizar el retorno de la inversión para el cliente, priorizando la construcción de la funcionalidad de mayor valor para el mismo y en los principios de adaptación, inspección continua, auto gestión e innovación. Scrum es una metodología conocida por sus iteraciones, (1) planeación, (2) facilitar y monitorear, y (3) reflejar, en las cuales se encuentran: *product backlog*, *sprint planning*, *sprint*, *sprint backlog*, *daily sprint meeting*, demo y retrospectiva; y cuenta con roles definidos, como son: *scrum master*, *product owner* (PO), y el equipo en sí. [30].

La metodología Scrum da lugar a iteraciones con la periodicidad de tiempo que se establezca, que en resumen consisten en el diseño, desarrollo y pruebas del piloto/producto de software. Al final de cada iteración, es de suma importancia la recepción de comentarios por parte de los responsables de cada una [34].

En Scrum, se realizan varias pruebas y reuniones con usuarios, y aunque no es mandatorio, en algunos casos estas se realizan en escenarios reales, logrando el aumento de la flexibilidad y la calidad de la aplicación prototipo, gracias a la detección y corrección de errores. Todas las actividades anteriores, se desarrollan de forma iterativa [34].

MobileD

Las características de los dispositivos móviles, terminales y el entorno de red dan lugar a limitaciones en la aplicación de cualquiera de los métodos ágiles de desarrollo de software existentes. Es por esto, que nació MobileD, siendo un enfoque específico para móviles. MobileD, desarrollado en colaboración con tres empresas que desarrollan productos y servicios de software para móviles, es un enfoque de desarrollo ágil basado en la programación extrema (prácticas de desarrollo), metodologías de cristal (método de escalabilidad), y RUP (*Rational Unified Process*) [35].

El método MobileD se optimiza para un equipo de menos de diez desarrolladores, quienes trabajen en un espacio compartido, con el objetivo de ofrecer una aplicación móvil completamente funcional en breve tiempo (aproximadamente 10 semanas). MobileD presenta las siguientes fases: puesta en marcha, núcleo, Core2, estabilización, y baja temporal. Cada fase consta de tres diferentes tipos de días de desarrollo: día de planificación, jornada de trabajo, y el día de lanzamiento. Además, las prácticas de las diferentes fases constan de nueve elementos principales: (1) escalonamiento y estimulación, (2) línea de arquitectura, (3) desarrollo móvil basado en pruebas, (4) integración continua, (5) programación a la par, (6) métricas, (7) mejora de procesos ágiles de software, (8) fuera del sitio del cliente, y (9) enfoque centrado en el usuario [35].

Resumen

En resumen, si bien existen más modelos y metodologías ágiles que las previamente descritas, estas están entre las más conocidas. XP se caracteriza por dar prioridad a la adaptabilidad, al permitir cambios de requisitos en cualquier fase del proyecto. AUP adopta muchas técnicas ágiles como XP, pero se diferencia de esta última metodología al conservar cierta formalidad que presenta y adopta de RUP. En cuanto a Scrum, es un conjunto simple de roles, responsabilidades y reuniones que no cambian. Al eliminar la imprevisibilidad innecesaria, el equipo de un proyecto está mejor capacitado para hacer frente a la impredecibilidad del desarrollo de todo proyecto y el aprendizaje continuo que se requiere. Por otra parte, MobileD se caracteriza por presentar ciclos de desarrollos muy rápidos en equipos muy pequeños, logrando conseguir productos funcionales en su totalidad en corto tiempo. Ante todas estas características, en conclusión, es decisión del líder de proyecto y todos los involucrados, decidir que metodología o método adoptar, y poder contar con un proceso a lo largo del desarrollo de un producto de software que se alinee a las características y necesidades requeridas.

Finalmente, es importante desatacar que existen más metodologías de desarrollo de SW, tanto ágiles como no, que las expuestas anteriormente y pueden no ser específicamente para desarrollo de aplicaciones móviles, y es por esta razón que nacen propuestas. Estas propuestas se pueden denominar *ad hoc*, considerando que este término se utiliza para referirse a una nueva teoría científica propuesta, la cual carece de un mecanismo explicativo esencial [36]. En el ámbito del presente estudio, las propuestas *ad hoc*, constituyen modelos basados en las prácticas tomadas por los responsables del desarrollo de un producto software. Estas prácticas bien pueden ser de otros modelos como propuestas propias de sus autores, puestas en marcha en conjunto en el desarrollo de un sistema.

2.3 Trabajos relacionados: estudios de procesos de desarrollo de aplicaciones móviles

En esta sección se muestran los estudios que han abordado los desafíos relacionados con el estudio de los procesos de desarrollo de aplicaciones móviles.

Usando técnicas de primer grado para recolectar datos, los investigadores han identificado problemas o desafíos en el desarrollo de SW móvil. Flora et al. [8] investiga los desafíos enfrentados durante los procesos de desarrollo de aplicaciones móviles a través de una encuesta de la comunidad móvil de investigación y desarrollo. Algunas empresas restringen su enfoque a una única plataforma que reduce el alcance de sus aplicaciones, pocos participantes expresaron su preocupación por la falta de recursos humanos informados y experimentados para desarrollar aplicaciones móviles, evaluando el esfuerzo requerido al inicio del ciclo y observando que el equipo desarrollador gasta mucho tiempo entendiendo y analizando los requisitos del cliente aumentando su carga de trabajo con el desarrollo, la integración y la prueba de las aplicaciones [8]

Wasserman [37] a través de una encuesta para desarrolladores de aplicaciones móviles, concluyó que la mayoría de las aplicaciones móviles son relativamente pequeñas, con uno o dos desarrolladores responsables de diseñar e implementar toda la aplicación, usando rara vez procesos de desarrollo formal y con muy poco desarrollo, bajo pocos parámetros. Además, varios estudios coinciden en que la mayoría de los desarrolladores de aplicaciones móviles usan enfoques ágiles o aproximaciones ad hoc, dado que, como los teléfonos móviles están evolucionando tan rápidamente, se requiere corto tiempo de desarrollo y planificación y los ciclos de desarrollo pueden ser el camino [8] [38].

Ingeniería de software para aplicaciones móviles tiene un fuerte énfasis en el diseño de interfaz de usuario (UI). Flora et al. [8] establecen que los desarrolladores de dispositivos móviles encuentran difícil realizar el mejor uso posible del espacio de pantalla limitado, y el diseño de interfaz de usuario adquiere mayor importancia que nunca. Este hecho está relacionado con la experiencia del usuario (UX), donde los desarrolladores indican que una pantalla más pequeña, diseños de pantalla y diferentes estilos de interacción del usuario tienen un impacto importante en el diseño de aplicaciones móviles, lo que eventualmente se traduce en baja adopción y uso de la aplicación.

A pesar de que los desarrolladores pueden controlar la mayoría de los aspectos de la UX, las aplicaciones móviles a menudo comparten elementos comunes de la interfaz de usuario con otras aplicaciones y, por lo tanto, deben adherirse a las directrices de interfaz de usuario desarrolladas externamente, muchas de las cuales se implementan en kits de desarrollo de software que forman parte de la plataforma [37]. Además, los desarrolladores de aplicaciones

móviles deben anticipar los dispositivos de destino para el diseño de interfaz de usuario. Específicamente, si se apunta a una única plataforma, los desarrolladores pueden decidir construir una única aplicación para todas las plataformas con el riesgo de algunas incoherencias funcionales o, en su lugar, considerar la creación de múltiples versiones orientadas a cada plataforma de hardware [38].

Además, en [37] se concluye que a pesar de la existencia de un gran número de aplicaciones móviles en la actualidad, los procesos de desarrollo de SW no están totalmente adaptados a los proyectos de desarrollo de aplicaciones móviles y son complejos de seguir, ya que las características de las aplicaciones móviles en funcionamiento y los entornos reales en los que se implementarán, derivan un nuevo conjunto de temas de investigación con diferentes enfoques.

2.4 Conclusión

Seleccionar un tipo de desarrollo de aplicación móvil requiere del previo análisis de los objetivos de la aplicación, del escenario en el que se desarrollará y desenvolverá, e incluso el tipo de usuarios finales a los cuales se desea llegar; todo esto a fin de que se seleccione un tipo de aplicación acorde a los recursos con los que se cuenta y con la funcionalidad requerida.

Sin embargo, afirmar que un tipo de aplicación es mejor que otro no es factible. No hay un tipo de app superior a otro, pero si existen características que los diferencian, que a su vez logran que se acoplen a escenarios distintos y variados. Es por demás visible que los tipos de aplicaciones presentan sus pros y contras, es así que identificar las características sobresalientes de cada uno es necesario para poder seleccionar el tipo de app que vaya acorde a las necesidades del proyecto a realizarse.

Adicional a esto, es importante tomar en cuenta que, sin importar el tipo de aplicación móvil, un grupo de desarrollo que trabaje con lenguajes específicos y estos se utilicen en el proceso de desarrollo de una aplicación, puede no comprometer inversiones significativas (incluso nulas) para capacitaciones, ya que se pueden valer del conocimiento previo, las habilidades propias de los desarrolladores y de aptitudes existentes, reduciendo el uso de recursos humanos como materiales, significativamente.

En sí la selección de un tipo de aplicación móvil, influirá en el modelo o método de desarrollo a seguir para llevar a cabo la elaboración de dicha app. Existen escenarios que permiten el mejor desempeño de un tipo de aplicación frente a otro, mismos que fueron expuestos en el presente capítulo, así como también modelos o métodos que, en base a las características del proyecto como sus recursos, se logran adaptar y guiar en el transcurso/ciclo de vida del desarrollo del SW.

Por otra parte, es visible que en los estudios sobre metodologías de desarrollo de aplicaciones móviles abordadas se presentan técnicas de investigación, como encuestas genéricas que usualmente no se enfocan en un desarrollo de aplicaciones específico, resultando en interpretaciones relativamente abstractas y generales.

En el siguiente capítulo, se presenta una metodología de investigación que permite el análisis de aplicaciones móviles específicas en los contextos industrial y académico.

Capítulo 3

Propuesta de enfoque de investigación

Para describir el escenario e identificar los procesos de desarrollo de SW móviles que se desenvuelven en la práctica dentro de los contextos de la academia e industria, se propone un caso de estudio siguiendo las guías de Runeson et al. [39], con el objetivo de identificar y caracterizar los procesos y metodologías de desarrollo de aplicaciones móviles que se aplican actualmente en el campo empresarial y académico.

Se logra obtener la información requerida mediante encuestas y entrevistas a los equipos de desarrollo. Las empresas y desarrolladores independientes seleccionados cuentan con la experiencia en el desarrollo de al menos una aplicación móvil.

Las Unidades de Análisis (UA), las técnicas de recolección de información, la metodología misma, y los resultados obtenidos son expuestas a lo largo del presente capítulo.

3.1 Procesos de desarrollo de aplicaciones móviles: Casos de estudio

El estudio se presenta en dos contextos, utilizando un diseño de múltiples casos (*multiple-case design*) compuesto por dos estudios únicos embebidos (*single embedded study*) [39] [40], como se muestra en la Figura 6. El primer estudio único contiene cinco Unidades de Análisis (UA) que corresponden a los procesos de desarrollo que se aplican en el campo académico. La primera UA es el desarrollado por un estudiante de Ingeniería Informática de la Universidad Politécnica de Madrid. En tanto que la segunda unidad de análisis es una aplicación desarrollada por estudiantes de la carrera de Ingeniería en Sistemas Informáticos y Computación de la Escuela Politécnica Nacional de Quito. Mientras que las tres unidades de análisis restantes son aplicaciones desarrolladas por estudiantes del Instituto Politécnico de Leiria.

Por otra parte, el segundo estudio único corresponde a los procesos de desarrollo que aplican ocho industrias que tienen como actividad principal el desarrollo de SW, pertenecientes a diferentes países. Así se tiene seis empresas ecuatorianas: Kruger Corporation (<http://www.krugercorporation.com>), Sofya Systems SA, Devsu SW (<http://devsu.com/>), Sgainnovar (<http://www.sgainnovar.com>), Alterbios Soluciones de Software (www.alterbios.com) y Oshyn Inc. (www.oshyn.com) que tienen como sede principal la ciudad de Quito. La séptima empresa, VOIDsteam, Lda. (<http://void.software>) se encuentra en Leria, Portugal. Finalmente, la octava empresa es Nanosystems Tecnología Móvil S.A. de C.V. (<http://www.nanosystems.com.mx>) pertenece a la ciudad de México D.F, México; misma que provee SW a Arca Continental (<http://www.arcacontal.com>) que siendo multinacional presenta áreas de mantenimiento de aplicaciones móviles en diferentes países como Ecuador, Argentina, Perú, entre otros. Todas las industrias mencionadas fueron seleccionadas dada la apertura que tenían al estudio, a las cuales se agradece encarecidamente su colaboración y tiempo brindado.

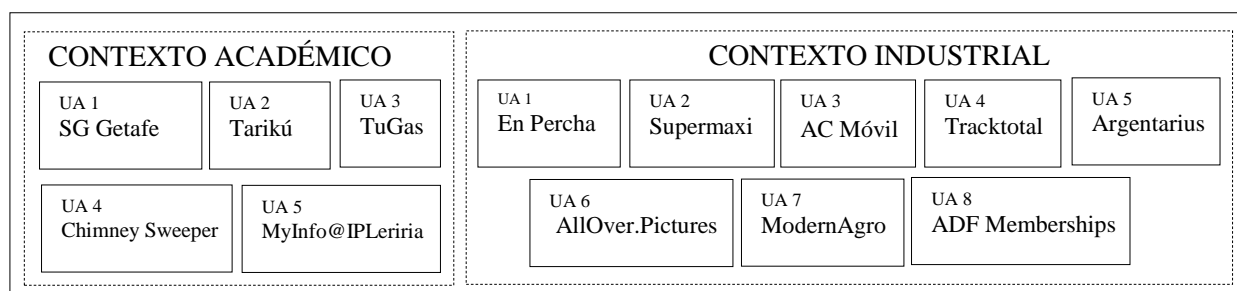


Figura 4. Unidades de análisis: Nombres de Aplicaciones Móviles

3.2 Metodología

La metodología que se lleva a cabo, se basa en directrices de Runeson et al. [39]. Esta se centra en una pregunta de investigación, que al poder presentar varias aristas y abarcar un sin número de temas relacionados, se recurre a sub preguntas de investigación que, al contar con un enfoque y lineamientos, permiten dar respuesta y complementar a la pregunta inicial. Una vez que se cuenta con las preguntas de investigación, es fundamental plantear técnicas para recolectar la información requerida, para lo cual, se utilizaron encuestas y entrevistas. A continuación, se define lo mencionado.

3.2.1 Pregunta de Investigación

Con el objetivo de identificar y caracterizar los procesos y metodologías en desarrollo de aplicaciones móviles que se aplican actualmente en la industria, se plantea la siguiente pregunta principal de investigación:

PI: "¿Cómo se llevan a cabo los procesos de desarrollo de aplicaciones móviles en la industria y academia?"

Dado el punto de vista general de la cuestión, se recurrió a sub preguntas de investigación, para lograr resultados más tangibles:

- P1. ¿Cuántas mujeres y hombres conforman los equipos de desarrollo en la industria y academia en nuestro escenario?
- P2. ¿Cuál es el rango de edad de los equipos de desarrollo en la industria y academia en nuestro escenario?
- P3. ¿Cuál es el nivel de experiencia del equipo de desarrollo en la industria y academia?
- P4. ¿Cuáles son las plataformas de desarrollo que la industria y la academia eligen con más frecuencia?
- P5. ¿Qué tipo de aplicaciones móviles desarrolla la industria y academia?
- P6. ¿Qué tipo de empresas desarrollan SW en nuestro escenario?
- P7. ¿Qué tipos de procesos de desarrollo de SW son aplicados por la industria y academia?
- P8. ¿Qué procesos o metodologías específicas de desarrollo se aplican?
- P9. ¿Cuáles son las actividades específicas que conforman las metodologías o procesos aplicados por la industria y la academia para el desarrollo de aplicaciones móviles?
- P10. ¿Qué período se requiere para el desarrollo de aplicaciones móviles?
- P11. ¿Cuáles son los costos aproximados del desarrollo de las aplicaciones móviles?
- P12. ¿A qué nivel de rigor se aplican las metodologías o procesos de desarrollo de aplicaciones móviles en la industria y academia?
- P13. ¿Qué problemas aparecen durante el proceso de desarrollo de aplicaciones móviles en la industria y academia?

3.2.2 Método de recolección de datos y estrategia de selección

Para la recopilación de datos, se utilizan técnicas de primer grado [11], como entrevistas y cuestionarios, que permiten el contacto directo con las fuentes de información. Para la recolección de información, se definió un cuestionario para cada estudio con preguntas abiertas y cerradas que intentan responder a las sub-preguntas de investigación.

La información recolectada se ha organizado de acuerdo con las sub preguntas. La estrategia aplicada se basa en proporcionar un conjunto de respuestas posibles a cada una de las sub-preguntas de investigación, como se presenta en la Tabla 8.

En cuanto a la encuesta llevada a cabo, se contó con diferentes preguntas que permitan dar respuesta a las sub-preguntas planteadas anteriormente, y por tanto a la pregunta principal. (Ver Anexo 1).

Al inicio de la investigación se realizó una encuesta general, sin embargo, tras la tabulación de los primeros datos obtenidos, se pudo evidenciar la necesidad de contar con dos encuestas, que puedan enfocarse de mejor manera a los dos contextos de estudio: academia e industria. Por tanto, la estructura de las encuestas es similar, sin embargo, en el caso de la industria, se presentan preguntas dirigidas a la empresa en la que se está llevando a cabo la misma.

3.2.3 Resultados

Esta sección presenta los resultados obtenidos a partir de los estudios de caso, organizados de acuerdo con las sub preguntas de investigación.

P1. Integrantes del equipo de desarrollo por género

En la Tabla 9 se puede evidenciar notoriamente que el número de hombres que conforman los equipos de desarrollo de las UA en su totalidad sobrepasan al número de mujeres en los mismos. Únicamente cuatro de las UA (1 de la academia, 3 de la industria) cuentan con mujeres en sus equipos de desarrollo. Por otra parte, ninguna de las UA presenta un número superior de mujeres que al número de hombres. Todos estos antecedentes se han evidenciado, como lo menciona T. Perry en su artículo [41]: las mujeres tienen una participación menor al de hombres en las ingenierías, y más aún en aquellas relacionadas a la tecnología, lo corrobora también Koslowski en [42]. Además, las actividades que realiza el género

femenino suelen ser diferentes a las de los hombres, siendo esta una posible raíz del problema que surge aún antes del campo laboral, en el proceso educativo [43].

TABLA 8. POSIBLES RESPUESTAS PARA LAS SUB PREGUNTAS DE INVESTIGACIÓN

#	Opciones de respuesta	Observaciones
P1	#Hombres, #Mujeres	Ninguna
P2	(años) Menor a 25, entre 25 y 35, entre 35 y 40, más de 40	Ninguna
P3	Sin experiencia, Novato, Intermedio, Experto	De acuerdo a [44]:(años de trabajo) Sin experiencia: 1-2, Novato: 2-5, Intermedio: 5-9, Experto: >= 10
P4	IOS, Android, Windows Phone, otro.	Ninguna
P5	App Nativa, app Web, app Híbrida	Ninguna
P6	Microempresa, Pequeña empresa, Mediana empresa, Empresa grande	De acuerdo a [45]: (empleados) Microempresa: 1-9, Pequeña empresa: 10-49, Mediana empresa: 50-199, Empresa Grande >=200
P7	Ágil, No Ágil	Ninguna
P8	Nombre del Proceso / Método/ Metodología	Si no se denomina con ningún nombre se clasifica como AD- HOC
P9	Actividades de proceso	Lista de las actividades del proceso
P10	(meses) 1-3, 3-6, 6-12, más de 12.	Ninguna
P11	Valor en dólares.	Se puede no contar con este dato en algunos casos de estudio debido a confidencialidad
P12	Bajo, Medio, Alto	Bajo: 0-3, Medio:4-6, Alto: 7-10
P13	Desafíos con los procesos de desarrollo	Descripción de los inconvenientes con los procesos de desarrollo
P14	Desafíos con la aplicación	Descripción de los inconvenientes con las aplicaciones móviles

TABLA 9. NÚMERO DE INTEGRANTES POR GÉNERO: UA

	Universidad / Industria	Aplicación Móvil	Mujeres	Hombres	Total # Personas
Academia	UPM	SG Getafe	0	1	1
	EPN	TARIKÚ	0	4	4
	IPL	TuGas	0	3	3
	IPL	Chimney Sweeper	0	2	2
	IPL	MyInfo@IPLeriria	1	0	1
Industria	Sofya Systems S.A	Argentarius	0	2	2
	Kruger Corporation	En Percha	0	7	7
	Devsu SW	Supermaxi	0	5	5
	Sgainnovar	Tracktotal	0	3	3
	Alterbios Soluciones	ModernAgro	0	7	7
	NanoSystems	ACMovil	2	10	12
	VOIDsteam	AllOver.Pictures	2	9	11
	Oshyn	ADF Memberships	1	8	9

P2. Rangos de edad en los equipos de desarrollo

En la tabla 10, se observa que, en las UA gran parte de los integrantes de los equipos de desarrollo tienen entre 25 y 35 años, seguidos por integrantes menores a 25 años. Con menos personas, se encuentra posicionado el rango de entre 35 y 40 años. Finalmente, el rango de edad que menos integrantes presenta corresponde a las personas con más de 40 años. Por otro lado, únicamente las UA académicas, cuentan integrantes que no superan los 25 años.

TABLA 10. RANGO DE EDADES EN LOS EQUIPOS DE DESARROLLO: UA

	Universidad / Industria	Aplicación Móvil	Menor a 25	Entre 25 y 35	Entre 35 y 40	Más de 40	Total # Personas
Academia	UPM	SG Getafe	1	0	0	0	1
	EPN	TARIKÚ	4	0	0	0	4
	IPL	TuGas	3	0	0	0	3
	IPL	Chimney Sweeper	2	0	0	0	2
	IPL	MyInfo@IPLeriria	1	0	0	0	1
Industria	Sofya Systems S.A	Argentarius	1	1	0	0	2
	Kruger Corporation	En Percha	0	7	0	0	7
	Devsu SW	Supermaxi	0	5	0	0	5
	Sgainnovar	Tracktotal	0	2	1	0	3
	Alterbios Soluciones	ModernAgro	1	6	0	0	7
	NanoSystems	ACMovil	2	7	2	1	12
	VOIDsteam	AllOver.Pictures	3	4	3	1	11
	Oshyn	ADF Memberships	2	4	2	1	9

P3. Nivel de experiencia del encuestado en el equipo de desarrollo

Dada la escala en [44], los encuestados de los equipos de desarrollo de 7 UA (2 de la industria, 5 académicos) no tienen experiencia, los desarrolladores presentan como máximo un año de trabajo. En las restantes UA, los encuestados de los equipos de desarrollo de las aplicaciones Argentarius, ModernAgro y AllOver.Pictures están calificados como novatos, con desarrolladores que presentan de 2 a 5 años de trabajo. En cuanto a los encuestados de las aplicaciones En Percha y Tracktotal, su nivel de experiencia corresponde a intermedio, con 9 años de trabajo como máximo. Finalmente, AC Móvil presenta en su grupo de desarrollo a un experto, con 10 años de trabajo en el campo de desarrollo de aplicaciones móviles.

TABLA 11. NIVEL DE EXPERIENCIA DEL ENCUESTADO DEL EQUIPO DE DESARROLLO

	Universidad/ Industria	Aplicación Móvil	Años trabajo	Nivel de experiencia
ACADEMIA	UPM	SG Getafe	1	Sin experiencia
	EPN	Tarikú	2	Sin experiencia
	IPL	TuGas	0	Sin experiencia
	IPL	Chimney Sweeper	0	Sin experiencia
	IPL	MyInfo@IPLeriria	0	Sin experiencia
INDUSTRIA	Sofya Systems	Argentarius	3	Novato
	Kruger Corporation	En Percha	9	Intermedio
	Devsu SW	Supermaxi	2	Sin experiencia
	Sgainnovar	Tracktotal	6	Intermedio
	Alterbios Soluciones	ModernAgro	3	Novato
	NanoSystems	AC Móvil	10	Experto
	VOIDsteam	AllOver.Pictures	3	Novato
	Oshyn	ADF Memberships	2	Sin experiencia

P4. Plataformas de Desarrollo

Como se presenta en la Tabla 12, 5 UA (1 académica, 3 de la industria) desarrollan aplicaciones sólo para Android. Una UA, de la academia, desarrolla una aplicación únicamente para iOS, mientras que las restantes siete UA (2 académicas, 5 de la industria) han desarrollado las aplicaciones móviles para Android e iOS.

P5. Tipos de aplicaciones móviles

La Tabla 13 muestra como 10 UA (4 de la academia, 6 de la industria) desarrollaron aplicaciones móviles nativas. Tres de las UA (1 de la academia, 2 de la industria) desarrollan aplicaciones móviles híbridas. Por otra parte, ninguna de las UA presenta una aplicación móvil web.

TABLA 12. PLATAFORMAS DE DESARROLLO: UNIDADES DE ANÁLISIS

	Universidad / Industria	Aplicación Móvil	Plataforma		Dispositivo		
			iOS	Android	Smartphone	Tablet	HH
Academia	UPM	SG Getafe		X	X	X	
	EPN	TARIKÚ	X	X	X		
	IPL	TuGas	X		X		
	IPL	Chimney Sweeper		X	X		
	IPL	MyInfo@IPLeriria	X	X	X	X	
Industria	Sofya Systems S.A	Argentarius	X	X	X	X	
	Kruger Corporation	En Percha	X	X	X		
	Devsu SW	Supermaxi	X	X	X	X	
	Sgainnovar	Tracktotal		X	X	X	
	Alterbios Soluciones	ModernAgro		X	X	X	
	NanoSystems	ACMovil		X	X		X
	VOIDsteam	AllOver.Pictures	X	X	X	X	
	Oshyn	ADF Memberships	X	X	X	X	

TABLA 13. TIPOS DE APLICACIONES MÓVILES: UNIDADES DE ANÁLISIS

	Universidad / Industria	Aplicación móvil	Tipo de app. móvil		
			Nativa	Web	Híbrida
Academia	UPM	SG Getafe	X		
	EPN	TARIKÚ	X		
	IPL	TuGas	X		
	IPL	Chimney Sweeper	X		
	IPL	MyInfo@IPLeriria			X
Industria	Sofya Systems	Argentarius			X
	Kruger Corporation	En Percha	X		
	Devsu SW	Supermaxi	X		
	Sgainnovar	Tracktotal			X
	Alterbios Soluciones	ModernAgro	X		
	NanoSystems	ACMovil	X		
	VOIDsteam	AllOver.Pictures	X		
	Oshyn	ADF Memberships	X		

P6. Tipos de empresas que desarrollan aplicaciones en nuestro escenario

Según la escala en [45] únicamente se cuenta con una empresa grande, Kruguer Corporation, con 248 empleados. Las 7 UA restantes son pequeñas empresas, siendo parte de esta cifra Sofya Systems SA: 11 empleados, Devsu SW: 20 empleados, Sgainnovar: 10 empleados, Alterbios Soluciones: 12 empleados, NanoSystems: 12 empleados, VoidSteam: 20 empleados, y Oshyn: 30 empleados. (Ver Tabla 14).

TABLA 14. TIEMPO DE DESARROLLO, TIPOS DE EMPRESAS DESARROLLANDO APPS

	Universidad/ Industria	Aplicación móvil	Tipo de empresa
INDUSTRIA	Sofya Systems	Argentarius	Pequeña empresa
	Kruger Corporation	En Percha	Empresa grande
	Devsu SW	Supermaxi	Pequeña empresa
	Sgainnovar	Tracktotal	Pequeña empresa
	Alterbios Soluciones	ModernAgro	Pequeña empresa
	NanoSystems	AC Móvil	Pequeña empresa
	VOIDsteam	AllOver.Pictures	Pequeña empresa
	Oshyn	ADF Memberships	Pequeña empresa

P7. Tipos de Procesos de Desarrollo de SW

Las unidades de análisis (UA) se dividen en dos tipos de procesos de desarrollo: ágil y no ágil, con cada una de sus metodologías o procesos (Tabla 15). Se observa que en gran mayoría (92,3%) las UA optaron por metodologías ágiles.

TABLA 15. TIPOS DE PROCESO DE DESARROLLO SW: UA

	Universidad / Industria	Aplicación Móvil	Tipos de Procesos de desarrollo de SW.		Procesos de Desarrollo de SW / Metodología
			Ágil	No Ágil	
Academia	UPM	SG Getafe		X	ADHOC
	EPN	TARIKÚ	X		ADHOC
	IPL	TuGas	X		XP
	IPL	Chimney Sweeper	X		XP
	IPL	MyInfo@IPLeriria	X		ADHOC
Industria	Sofya Systems	Argentarius	X		AUP
	Kruger Corporation	En Percha	X		SCRUM
	Devsu SW	Supermaxi	X		SCRUM
	Sgainnovar	Tracktotal		X	RAD
	Alterbios Soluciones	ModernAgro	X		SCRUM
	NanoSystems	ACMóvil	X		SCRUM
	VOIDsteam	AllOver.Pictures	X		SCRUM
	Oshyn	ADF Memberships	X		SCRUM

P8. Procesos, Métodos y Metodologías aplicadas

Dos UA, TuGas y Chimney Sweeper, utiliza los principios de XP como metodología. Además, una UA presenta el modelo AUP para el desarrollo de la aplicación Argentarius. Una UA aplicó la metodología RAD para el desarrollo de la aplicación Tracktotal. Por otra parte, las aplicaciones AC Móvil, Supermaxi, En Percha, ModernAgro, AllOver.Pictures y ADF Memberships, presentan Scrum como metodología. Y, por último, las restantes tres UA, SG

Getafe, Tarikú y MyInfo@IPLeriria, sin ninguna metodología específica adoptan el término ADHOC. Esto no implica que sea la misma metodología o las mismas actividades. (Véanse tablas 15 y 16)

P9. Actividades concretas llevadas a cabo en los procesos /metodologías de desarrollo

En la recolección de las actividades concretas aplicadas en el desarrollo de las aplicaciones móviles en las diferentes UA, se observa el buen uso del levantamiento/análisis de requerimientos, etapa inicial fundamental para establecer y definir las características del SW a desarrollarse. Lo que no sucede con el diseño de interfaces o experiencia del usuario, etapas que no son aplicadas en algunas UA o remotamente descritas en otras. Adicional, las actividades que se describen de manera muy similar entre una UA y otra, son aquellas de Scrum. Por otra parte, solo una UA no reportó actividades, más esto no implica que no hagan uso de actividades propias de su modelo propuesto y adaptado a su proyecto. Todas las actividades llevadas a cabo son detalladas en la Tabla 16.

P10. Tiempo de desarrollo de la aplicación móvil

Tres UA (2 académicas, 1 de la industria) desarrollaron sus aplicaciones móviles en un período de 1-3 meses. Cuatro UA (3 de la industria, 1 académica) desarrollaron las aplicaciones móviles en un período de 3-6 meses. Dos UA (1 de la industria, 1 académica) les tomó de entre 6 a 12 meses el desarrollo de las aplicaciones móviles. Finalmente, cuatro UA (1 académica, 3 de la industria) les tomó más de 12 meses el desarrollo de las aplicaciones móviles.

P11. Costo de desarrollo de app. Móvil

El costo aproximado del desarrollo de las aplicaciones móviles en el contexto académico no presenta valor alguno, puesto que sus desarrollos naces con fines educativos. En cuanto al contexto industrial, algunas empresas no presentaron valor alguno debido a la confidencialidad de los mismos. De los datos obtenidos, se observa como el desarrollo de la aplicación móvil Tracktotal viene a ser el valor más bajo con \$4000. Y el valor más alto presenta el desarrollo de la aplicación Supermaxi con \$20000. (Tabla 17).

TABLA 16. PROCESO / METODOLOGÍAS ACTIVIDADES: UA

	Aplicación Móvil	P8	ACTIVIDADES
ACADEMIA	SG Getafe	ADHOC	<ol style="list-style-type: none"> 1. Estudio tecnológico 2. Requisitos 3. Prototipo de alta fidelidad, 4. Pruebas de prototipo de alta fidelidad 5. Implementación de la primera versión completa 6. Evaluación de usabilidad 7. Documentación
	TARIKÚ	ADHOC	<ol style="list-style-type: none"> 1. Análisis de requerimientos 2. Diseño UI 3. Desarrollo de app piloto 4. Pruebas unitarias 5. Corrección de errores 6. Proceso de pilotaje: App obtenida usada por habitantes de la zona
	TuGas	XP	<ol style="list-style-type: none"> 1. Reuniones diarias: Hablamos de lo que debemos hacer 2. Pair programming: La programación y el código de revisión por pares 3. Comunicados semanales: Lanzamiento versiones funcionales para pruebas
	Chimney Sweeper	XP	<ol style="list-style-type: none"> 1. Reuniones diarias: Hablamos de lo que debemos hacer 2. Pair programming: La programación y el código de revisión por pares 3. Comunicados semanales: Lanzamiento versiones funcionales para pruebas 4. Investigación sobre plataforma: investigación sobre Unity para aprender cómo implementar nuevas características
	MyInfo@IPLeriria	ADHOC	No reporta
INDUSTRIA	Argentarius	AUP	<ol style="list-style-type: none"> 1. Especificación de requerimientos: Se definen los requerimientos tanto en la aplicación móvil como en el backend. 2. Diseño de la solución: Se define la arquitectura de la solución informática y protocolos de comunicación entre frontend y backend. 3. Diseño de la aplicación móvil: Se realiza el diseño de la aplicación móvil a través de maquetas y se define la experiencia de usuario. 4. Diseño del backend: Se definen los recursos que consumirá la aplicación móvil. 5. Construcción: Se codifica la aplicación tanto en backend como en frontend. 6. Pruebas e integración: Se realizan pruebas unitarias tanto al frontend como al backend y se integran estas dos capas. 7. Implantación: Ejecución de la versión final de la aplicación en el ambiente de producción. 8. Publicación: Se publica la aplicación en las tiendas respectivas. 9. Mejora continua: Se corrigen errores y se aumentan prestaciones a la aplicación móvil.
	En Percha	SCRUM	<ol style="list-style-type: none"> 1. Sprint Planning: Planificación de la iteración 2. Sprint: Ejecución de la iteración 3. Scrum Daily Meeting: Reunión diaria de sincronización del equipo 4. Sprint Review: Demostración de los requisitos completados 5. Sprint Retrospective: Retrospectiva 6. Refinamiento: Refinamiento de la lista de requisitos y cambios en el proyecto

	Supermaxi	SCRUM	<ol style="list-style-type: none"> 1. Sprint Planning: Planificación de las tareas de la iteración 2. Sprint: Ejecución de las tareas 3. Scrum Daily Meeting: Reunión diaria de sincronización del equipo 4. Sprint Review: Demostración de los requisitos completados 5. Sprint Retrospective: Análisis del sprint realizado 6. Adjustment: Refinamiento de la lista de requisitos y cambios en el proyecto
	Tracktotal	RAD	<ol style="list-style-type: none"> 1. Requisitos: Revisión y aprobación del documento de requerimientos 2. Desarrollo del aplicativo 3. Verificación: Pruebas del aplicativo 4. Implementación: Instalación en dispositivos de los usuarios
	ModernAgro	SCRUM	<ol style="list-style-type: none"> 1. Reunión Backlog: Se describen todas las tareas a realizarse para el sistema 2. Definición de sprints y sus tareas: Se distribuyen las tareas en los sprints o iteraciones que sean pertinentes, así mismo la duración de cada sprint 3. Asignación de tiempos y responsables: A cada tarea se asignará un programador responsable y una estimación de tiempo pertinente 4. Reunión diaria: El equipo de desarrollo se reúne para presentar el informe del avance de las tareas y para la retroalimentación. 5. Reunión de fin de sprint: El equipo se reunirá al final de cada sprint para un informe de todas las tareas realizadas 6. Definir roles: Scrum master, product owner, etc.
	AC Móvil	SCRUM	<ol style="list-style-type: none"> 1. Inicio: Esta fase incluye los procesos relacionados con la iniciación de un proyecto. 2. Plan y estimación: Esta fase consiste en procesos relacionados con la planificación y estimación de tareas, que incluyen: 3. Implementación: Esta fase está relacionada con la ejecución de las tareas y actividades para crear el producto de un proyecto. 4. Revisión y Retrospectiva: Esta fase se ocupa de revisar las entregas y el trabajo que se ha hecho y de determinar maneras de mejorar las prácticas y los métodos usados para hacer el trabajo del proyecto. 5. Lanzamiento (Release): Esta fase enfatiza en la entrega de los productos aceptados al cliente y la identificación, documentación e internalización de las lecciones aprendidas durante el proyecto.
	AllOver.Pictures	SCRUM	<ol style="list-style-type: none"> 1. Backlog Grooming: El proceso en el que el equipo de desarrollo con el propietario del producto mejorar las especificaciones de cada una de las tareas (Stories) presentes en el Backlog. 2. Sprint Planning: reunión de planificación de Sprint y ubicada en Sprint Backlog. 3. Sprint: periodo de 2 semanas en el que los Stories se desarrollan en el Sprint Backlog. 4. Sprint Review: Reunión realizada en el último día de Sprint, que toma el trabajo durante el sprint por lo general un Demo "público". 5. Sprint Retrospective: Reunión del equipo de desarrollo con el fin de detectar puntos de mejora en el proceso.
	ADF Memberships	SCRUM	<ol style="list-style-type: none"> 1. Daily standup: Reunión de pie diaria para saber el estado de las tareas 2. Estimaciones: Se estima las tareas en conjunto 3. Iteraciones: Se trabajaba en iteraciones de 1 mes

TABLA 17. TIEMPO DE DESARROLLO DE LAS APPS Y SUS COSTOS

	Universidad/ Industria	Aplicación móvil	Tiempo de desarrollo App móvil (meses)	Costo desarrollo App móvil (USD)
ACADEMIA	UPM	SG Getafe	6-12	\$0
	EPN	Tarikú	3-6	\$0
	IPL	TuGas	1-3	\$0
	IPL	Chimney Sweeper	1-3	\$0
	IPL	MyInfo@IPLeriria	Más de 12	\$0
INDUSTRIA	Sofya Systems	Argentarius	6-12	\$11.000
	Kruger Corporation	En Percha	Más de 12	No reporta
	Devsu SW	Supermaxi	Más de 12	\$ 20.000
	Sgainnovar	Tracktotal	3-6	\$ 4.000
	Alterbios Soluciones	ModernAgro	1-3	\$10.000
	NanoSystems	AC Móvil	3-6	No reporta
	VOIDsteam	AllOver.Pictures	3-6	No reporta
	Oshyn	ADF Memberships	Más de 12	No reporta

P12. Nivel de rigurosidad en la aplicación de la metodología

Tres UA de la industria presentan un nivel medio de rigor al aplicar la metodología de desarrollo de aplicaciones móviles, y las restantes 10 UA presentan un alto nivel, presentando valores de entre 7 y 10 (Figura 7). Los motivos de esta ponderación se mencionan a continuación, y se analizan en la sección *Discusión de resultados*.

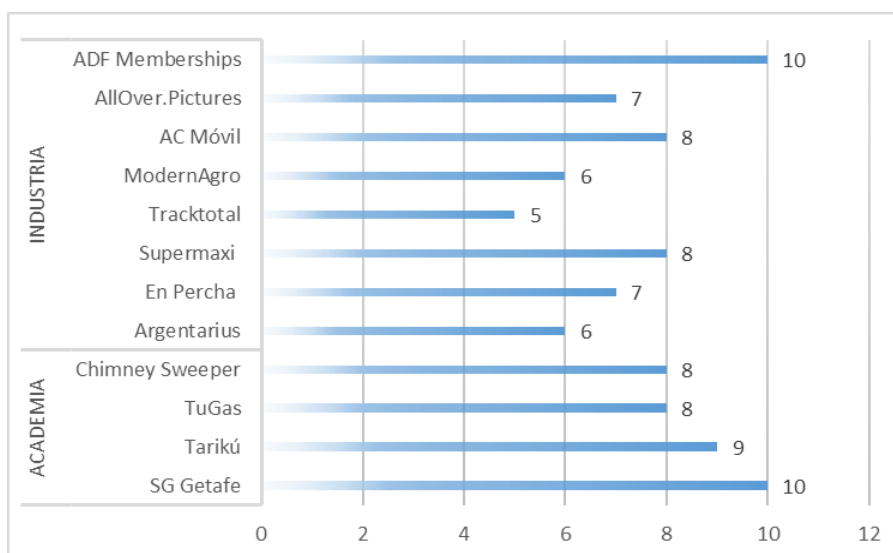


Figura 5. Nivel de rigurosidad en la aplicación de la metodología por parte de las UA

P13. Problemas identificados durante el proceso de desarrollo con respecto a la metodología

Ciertas UA presentan algún tipo de dificultades durante y con el proceso de desarrollo. Los desafíos que se informan suelen referirse a requisitos insuficientemente planteados, cambiados en el proceso y la falta de conocimiento o herramientas para el desarrollo de la aplicación. En la Tabla 18, se muestran los retos reportados por los encuestados.

P14. Problemas identificados durante el proceso de desarrollo con respecto a la aplicación.

Gran parte de las UA presentan algún tipo de dificultades durante el proceso de desarrollo con respecto a la aplicación misma. Los inconvenientes que se informan suelen referirse a la falta de conocimiento o herramientas para el desarrollo de la aplicación. En la Tabla 19, se muestran los retos reportados por los encuestados.

TABLA 18. DESAFÍOS CON LA METODOLOGÍA DURANTE EL PROCESO DE DESARROLLO: UA

	Universidad/ Industria	Aplicación móvil	Desafíos con respecto a la metodología
ACADEMIA	UPM	SG Getafe	Poco tiempo libre para enfocarse más en el desarrollo
	EPN	Tarikú	No se encontraron problemas directos con la metodología
	IPL	TuGas	No se encontraron problemas directos con la metodología
	IPL	Chimney Sweeper	No se encontraron problemas directos con la metodología
	IPL	MyInfo@IPLeriria	Durante el desarrollo, nunca se cerró el alcance del proyecto, por lo que a menudo surgían nuevas necesidades.
INDUSTRIA	Sofya Systems	Argentarius	Los requerimientos de la aplicación son muy volátiles, por lo que el modelo en cascada de AUP tuvo que ser violado y adaptado en su lugar un modelo en espiral
	Kruger Corporation	En Percha	Demasiadas reuniones
	Devsu SW	Supermaxi	No se encontraron problemas directos con la metodología
	Sgainnovar	Tracktotal	Los requerimientos variaron en la fase de implementación.
	Alterbios Soluciones	ModernAgro	La metodología no se puede aplicar al pie de la letra por que los miembros del equipo están en diferentes proyectos y con diferentes actividades que interfieren entre sí. Además, existe una falta de comprensión entre los miembros del equipo por la mala o nula comunicación de errores o mejoras realizadas durante el desarrollo de la app.
	NanoSystems	AC Móvil	No se encontraron problemas directos con la metodología
	VOIDsteam	AllOver.Pictures	Debido a la madurez del equipo en relación con la metodología utilizada, los problemas fueron mínimos y los existentes fueron eliminados o superados inmediatamente por el propio equipo.
Oshyn	ADF Memberships	No se encontraron problemas directos con la metodología	

TABLA 19. DESAFÍOS CON LA APLICACIÓN DURANTE EL PROCESO DE DESARROLLO: UA

	Universidad/ Industria	Aplicación móvil	Desafíos con respecto a la aplicación
ACADEMIA	UPM	SG Getafe	Problemas con tipos de datos y codificación.
	EPN	Tarikú	El universo de dispositivos Android, únicamente se probó la app en nuestros dispositivos (equipo de desarrollo) y no tomábamos en cuenta como se vería o funcionaría en los demás, afectó en el momento de lanzamiento beta, cuando muchos de los usuarios reportaban fallas
	IPL	TuGas	No reporta
	IPL	Chimney Sweeper	No reporta
	IPL	MyInfo@IPLeria	Una de las restricciones de desarrollo móvil que afectó en gran medida el proceso de desarrollo de la aplicación está relacionada con los sistemas operativos iOS, es decir, la limitación de que usted tiene que desarrollar para esta plataforma, lo que implica que tiene equipos de Apple e incluso una cuenta de desarrollador. Otra de las restricciones fue que si desea tener acceso a la aplicación, incluso sin conexión a Internet, lo que implicaba un aprendizaje para mantener la información en el almacenamiento móvil.
INDUSTRIA	Sofya Systems	Argentarius	Desconocimiento del framework utilizado. Optimización de peticiones al backend. Comportamiento del frontend que afecta los recursos disponibles en el backend
	Kruger Corporation	En Percha	Cambios del usuario Soporte versiones anteriores Pruebas en distintos dispositivos y versiones de S.O.
	Devsu SW	Supermaxi	La no implementación de pruebas automatizadas desde el inicio del desarrollo causó retrasos por errores
	Sgainnovar	Tracktotal	Problemas de compatibilidad con IDEs de desarrollo y SDK de Android.
	Alterbios Soluciones	ModernAgro	Requerimientos no definidos correctamente
	NanoSystems	AC Móvil	El constante cambio en el modelo de datos, o la falta de ellos (datos).
	VOIDsteam	AllOver.Pictures	La decisión de desarrollar aplicaciones móviles nativas siempre trae el desafío de armonizar la funcionalidad y la experiencia del uso de las plataformas Android iOS /. En el caso de AllOver.Pictures, el equipo de desarrollo trató de mantener a lo largo de la aplicación la "experiencia del usuario" nativa de cada plataforma. Esta decisión tuvo un impacto en el Departamento de Diseño VOID una vez que el diseño inicial tuvo que ser adaptado a las nuevas necesidades de cada plataforma.
Oshyn	ADF Memberships	Ninguno	

3.3 Discusión de resultados

En esta sección, se discuten los principales hallazgos del estudio. También se destacan las limitaciones que pueden amenazar la validez del estudio y se examina las implicaciones para la investigación en este campo.

A. Hallazgos

Los principales hallazgos de la presente investigación son:

- En los equipos de desarrollo, se visualiza como en las UA académicas existen menos integrantes en relación a las UA de la industria. Un punto a resaltar importante de los equipos de desarrollo, sin importar el contexto de las UA, es que se conforman en gran número por el género masculino; siendo la cantidad de mujeres mínima, y en algunas UA incluso nulas, dejando visible la participación minoritaria del género femenino en proyectos de tecnología [41]. Asimismo, es notable como los integrantes en gran parte presentan de entre 25 y 35 años; pero según el contexto sobresalen ciertos rangos de edad: en el académico, los integrantes de los equipos de desarrollo son menores a 25 años; y únicamente en el industrial se presentan integrantes mayores de 35 años. Vale mencionar que existe una relación entre la edad de los integrantes/encuestados y su nivel de experiencia. Novatos, intermedios y expertos sobrepasan de los 35 años y son parte de industrias; mientras que, con menos de 25 años se encuentran los desarrolladores independientes que no cuentan con experiencia de acuerdo a [44].

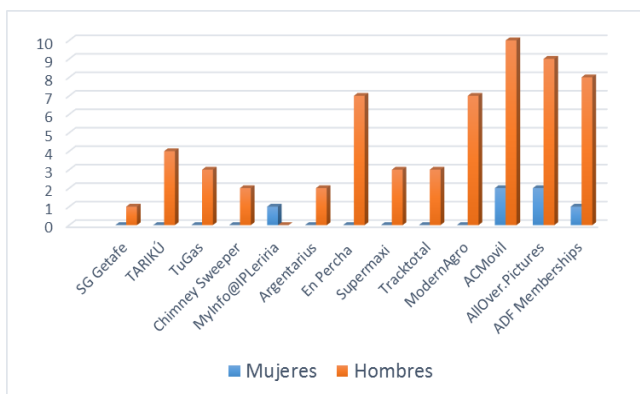


Figura 6. Equipos de desarrollo de UA: Núm. de integrantes por género

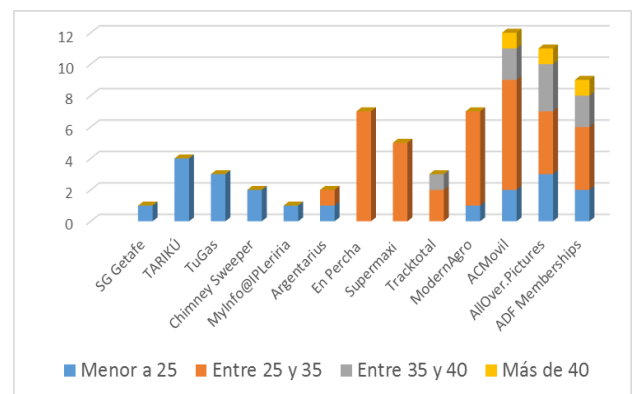


Figura 7. Equipos de desarrollo de UA: Núm. de integrantes por rangos de edad

- Todas las aplicaciones, en un 100%, fueron desarrolladas para teléfonos inteligentes. La gran mayoría, 12 de las 13 aplicaciones (92,3%) son desarrolladas para Android, y de estas 13

aplicaciones móviles, el 53,8% son desarrolladas para las plataformas Android e iOS simultáneamente, mientras que con el 76,9% son aplicaciones nativas, dejando el porcentaje restante a las aplicaciones híbridas. Las aplicaciones se desarrollaron bajo el enfoque nativo por razones tales como el rendimiento, el uso de sensores o la proyección a futuro (crecimiento exponencial) de la aplicación móvil (Ver figuras 10 y 11).

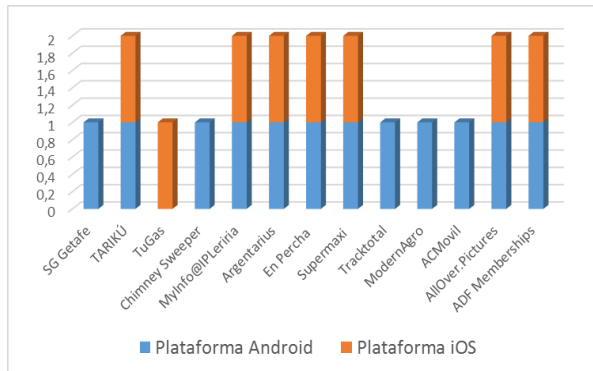


Figura 8. Plataformas de las UA.

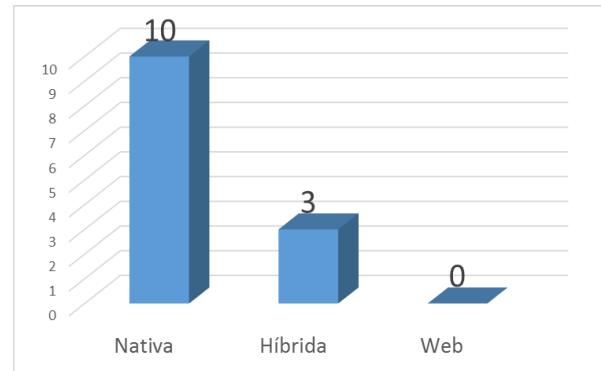


Figura 9. Tipo de desarrollo: Número UA.

- En cuanto a la industria, el 87,5% de la UA son empresas pequeñas y el 12,5% restante está constituido por una empresa grande. Entre las empresas, los encuestados resultan ser en mayor número novatos (75%), y tan sólo un experto (25%); siendo visible la falta de conocimiento de los miembros del equipo de desarrollo, como menciona [8]: la falta de recursos (equipo de desarrollo) informados y experimentados para desarrollar aplicaciones móviles, siendo esta una realidad que afecta a lo largo del proceso de desarrollo de aplicaciones móviles. Por otro lado, en una situación menos favorable, se encuentran las cinco UA académicas que presentan encuestados con poca o ninguna experiencia en su totalidad, quienes además de generalmente estar en formación, desarrollan apps móviles como parte de su proceso de aprendizaje (Ver figuras 12 y 13).
- La mayoría de las UA (69,2%) invirtieron 12 meses o menos para el desarrollo de aplicaciones, haciendo visible el corto tiempo de desarrollo y planificación requerido, como menciona [8] [38]. Este factor contribuye varias veces a la selección de metodologías ágiles, como sucede con todas las UA de la industria, y con la mayoría de las UA académicas (Figura 14). Las UA en el contexto industrial, hicieron uso de métodos ágiles, recurriendo a SCRUM y AUP en un 75% y 12,5% respectivamente, mientras que, con métodos no ágiles, hicieron uso únicamente de RAD en un 12,5%. En cuanto a la academia, las UA utilizan metodologías ágiles en su gran mayoría, y un solo caso con

metodología tradicional. Dos UA hacen uso de XP, una UA utiliza AUP y tres de las metodologías utilizadas, son propuestas (ADHOC). El uso de ADHOC, ya sea con directrices ágiles o no ágiles, evidencia la afirmación: "Los desarrolladores usaron raramente procesos formales de desarrollo" [37], siendo una posible razón para crear nuevas propuestas tras la complejidad para el equipo de desarrollo de seguir procesos de desarrollo SW no adaptados totalmente a proyectos de desarrollo de aplicaciones móvil es [37]. Si bien ADHOC puede alinearse a metodologías conocidas, deben ser tratados como propuestas a procesos diferentes (Figura 15).

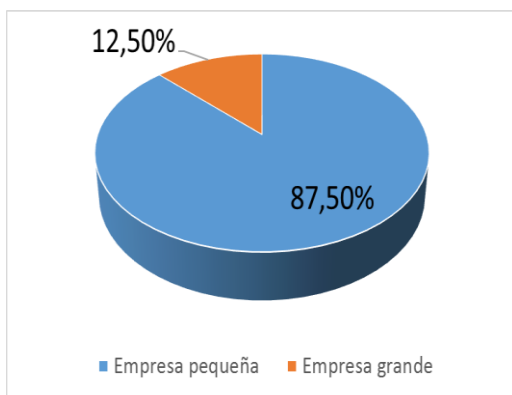


Figura 10. Tipos de empresas de UA industriales: porcentajes



Figura 11. Nivel de experiencia: Número de encuestados por UA.

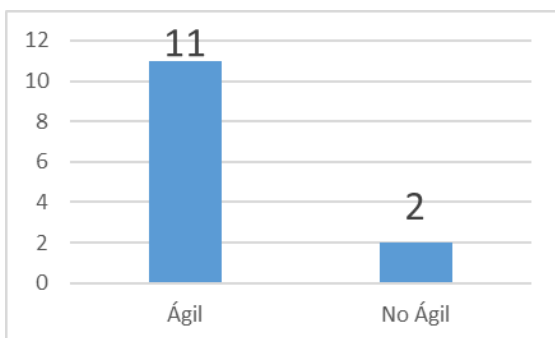


Figura 12. Tipos de metodologías: Número de UA.

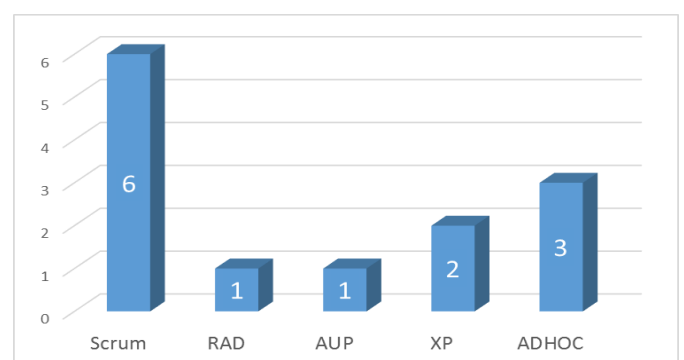


Figura 13. Metodologías de desarrollo de SW: número utilizado por UA.

- En las UA, varios procesos comienzan con actividades que implican requisitos, seguidos del diseño y desarrollo de la aplicación y finalmente la evaluación. Sin embargo, algunas actividades que comprenden los procesos o métodos identificados son diversas o fuera de lo común, como es el caso de una UA que establece como actividad la investigación de la plataforma Unity. Por otro lado, sólo un UA académico utiliza requisitos específicos para cubrir los estándares de diseño para mejorar el UX. Esto deja en evidencia la adopción mínima o nula de estándares que cubren la variedad de tamaños de pantalla, cámaras y el uso de sensores: GPS, acelerómetro, giroscopio, entre otros, lo que significa que las funcionalidades de los dispositivos móviles no se explotan en su totalidad o de la manera más adecuada. La posible razón de la pérdida de recursos en los dispositivos móviles puede ser la falta de conocimiento de la existencia de estos estándares o evitar estas directrices con el fin de reducir el tiempo para el desarrollo de aplicaciones [8].
- La gran mayoría de los desarrolladores de UA establece actividades relacionadas con la usabilidad con el objetivo de mejorar la interacción del usuario y UX, especialmente cuando el tamaño de la pantalla es una gran limitación. Sin embargo, en las UA se reportan problemas de interfaz, que limitan la funcionalidad de la aplicación. Además, el incumplimiento del tiempo necesario para probar diferentes modelos de dispositivos móviles. Las posibles razones para tener estos inconvenientes es dejar de lado las directrices específicas para aplicaciones en dispositivos móviles o la falta de conocimiento del desarrollador, como se mencionó en puntos anteriores.
- Ciertos retos en los procesos o metodologías de software móvil generalmente reflejan retrasos en el desarrollo de la aplicación móvil. El esfuerzo requerido al comienzo del proceso de desarrollo en el que el equipo pasa mucho tiempo evaluando los requisitos del cliente aumenta su carga de trabajo con el desarrollo, la integración y la prueba de las aplicaciones [8]. Es así que las UA se ven afectadas negativamente en actividades que tardan mucho tiempo en esperar, al igual que las relacionadas con los requisitos: esperar la aprobación o cambios por parte de los usuarios, no dar tiempo suficiente para la correcta elevación de los requisitos o no tener en cuenta las necesidades del usuario final a tiempo. Estas cuestiones combinadas con el tiempo limitado necesario para el desarrollo de aplicaciones móviles significan que el equipo de desarrollo no identifica o evita factores en las primeras etapas de desarrollo, trayendo dificultades a largo plazo.

B. Limitaciones

Las principales limitaciones del estudio están relacionadas con la baja experiencia del encuestado del equipo de desarrollo, la ubicación de la gran mayoría de empresas en la misma posición geográfica y la posible ambigüedad en las preguntas de la encuesta. Además, una importante limitación y amenaza al estudio es que se recibieron respuestas de personas que estaban dispuestas a ser encuestadas. Existe el riesgo de recibir respuestas equivocadas o sesgadas, ya que varias preguntas captan opiniones, no hechos. Los sesgos comunitarios pueden reflejarse en los resultados y tener pocos UA significa que la muestra de la encuesta no representa el mundo real y el resultado puede basarse en opiniones y no necesariamente en hechos probados.

Adicional, el uso de técnicas investigativas de primer nivel, dieron resultados favorables cuando se tuvo la disposición y colaboración del encuestado, sin embargo, existieron unidades de análisis potenciales, cuyas personas de contacto dieron respuestas cortas, incoherentes e incluso nulas en varias preguntas, por lo cual se obviaron para la presente investigación.

C. Implicaciones para investigaciones futuras

Los hallazgos mencionados anteriormente pueden considerarse como un posible tema de investigación futura y, mejor, si la recopilación de datos se mejora mediante diferentes versiones de encuestas y con estudios de casos geográficamente variados.

La primera implicación ambiciosa es la formalización del proceso de desarrollo de las aplicaciones móviles, que puede comenzar con el establecimiento de reglas o directrices, para lograr metodologías propias del desarrollo de aplicaciones móviles de manera que el resultado permita contar con un guía fiel a las características tan propias de las apps móviles. Estas directrices permitirán inclinarse hacia un proceso u otro, teniendo en cuenta diferentes parámetros como el número de recursos en el equipo de desarrollo, el tiempo establecido para el desarrollo de la aplicación, el nivel de conocimiento de los miembros del equipo de desarrollo, entre otros.

Una segunda implicación es el estudio y la propuesta para el uso de normas de diseño de aplicaciones móviles. Para que en el proceso que se lleva a cabo, se tengan en cuenta estas particularidades que permiten la mejora de la UX, sin dejar de lado y aprovechando las

características y funcionalidades de los dispositivos móviles como el tamaño de la pantalla, sensores, GPS, otros.

3.4 Conclusión

En este trabajo, hemos identificado la escasa documentación formal sobre metodologías o procesos de desarrollo de aplicaciones móviles aplicadas por la industria y el mundo académico, a pesar de las miles de aplicaciones móviles existentes en el mercado. Dado el mayor grado de interés por el rápido desarrollo de las aplicaciones, compartir sus experiencias de desarrollo con la comunidad científica ha pasado a segundo plano. Por lo tanto, el objetivo principal de esta investigación es contribuir a una mayor comprensión de los procesos de desarrollo de software móvil o metodologías aplicadas por la industria y el mundo académico para el desarrollo de aplicaciones móviles. Se ha propuesto un enfoque práctico, basado en estudios de casos que se conforman de trece unidades de análisis (8 de la industria y 5 de la academia).

Durante el análisis para identificar y caracterizar los procesos y metodologías en el desarrollo de aplicaciones móviles, se han planteado 13 sub-preguntas de investigación que se deben responder en el contexto de cada caso de estudio, llegando así a los siguientes hallazgos principales. Los desarrolladores de aplicaciones móviles tienen pocos procesos definidos específicamente para el dominio móvil. Sin embargo, entre los procesos identificados, Scrum se convierte en una referencia, ya que se utiliza principalmente en la industria. Los procesos identificados se basan en métodos y metodologías ágiles y no ágiles que apenas integran actividades concretas del campo móvil, sin poder explotar adecuadamente las funcionalidades y características de los dispositivos móviles. Además, se verifica que un inconveniente latente en el proceso de desarrollo de aplicaciones móviles es la falta de recursos experimentados para desarrollar aplicaciones móviles.

Estos resultados muestran la necesidad de compartir y proponer métodos específicos para el desarrollo de aplicaciones móviles que integren características y restricciones de dispositivos móviles, así como técnicas de usabilidad para mejorar la experiencia del usuario al emplear una aplicación móvil. Otras investigaciones abordarán estas necesidades e incluirán una experiencia práctica más amplia, tanto en términos de unidades de análisis como de preguntas de investigación.

(Inicia en número de página impar)

Capítulo 4

Propuesta de Modelo para Selección de Metodología de Desarrollo de Software

En este capítulo se establecen variables y criterios a considerar para el desarrollo de un modelo propuesto, a fin de seleccionar una metodología de desarrollo de SW determinada. Es importante reconocer que, si bien muchos factores pueden influenciar en la selección de una metodología de desarrollo de SW, existen puntos claves que se deben considerar, tales como recursos disponibles, requerimientos del proyecto, entre otros.

A continuación, se realizará la descripción del modelo evaluativo propuesto.

4.1 Modelo evaluativo propuesto

Se propone un modelo que guíe en la toma de decisión en cuanto a la adopción de una metodología de desarrollo de SW u otra. La propuesta de modelo se sustenta en el producto de matrices, operación intensiva de cálculo fundamental utilizada en muchos algoritmos en computación científica [46].

Dentro de diferentes áreas, se hace uso ampliamente de operaciones con matrices, y específicamente en la computación e informática, se presentan en codificación, procesamiento de señales digitales, procesamiento de imágenes, gráficos, robótica, diversos algoritmos y modelos de control, entre otros. Las matrices y las operaciones que se pueden realizar con las mismas, son el motor central de varios algoritmos [46].

El modelo propuesto consiste inicialmente en la definición de criterios (características de las metodologías de desarrollo de SW) cuantificados por metodología, convirtiéndose en la primera matriz. La segunda matriz es el resultado de la evaluación de varios criterios de acuerdo a los requerimientos/necesidades del proyecto a llevarse a cabo, por lo que es recomendable llevar a cabo esta evaluación con el líder del proyecto o jefe del mismo.

Finalmente, se realiza el producto de las dos matrices previamente obtenidas. Una vez que se cuenta con estos valores, se lleva a cabo la sumatoria vertical total de los valores (por columna). Los totales obtenidos de la sumatoria por cada columna, permiten visualizar las metodologías que se acogen de mejor manera a lo que requiere el proyecto, siendo el mayor puntaje el recomendable principalmente.

A continuación, se lleva a cabo la definición de criterios que permitan posteriormente la calificación cuantitativa por metodología de desarrollo de SW.

4.1.1 Definición de criterios

En la presente sección se lleva a cabo el análisis de los criterios a considerar para la selección de una metodología de desarrollo, que permita dar el mejor uso de los recursos disponibles para la realización de un proyecto de SW [47].

A. Metodologías de desarrollo

La selección de las metodologías a ser evaluadas se realizó bajo los resultados obtenidos en el caso de estudio realizado en el capítulo anterior, en el cual se observó el uso recurrente, tanto en UA académicas como de la industria, de las siguientes metodologías: RUP, RAD, SCRUM, y XP.

B. Criterios de selección

Una vez definidas las metodologías a ser evaluadas, se ponen en consideración los siguientes criterios:

a) Presupuesto

El presupuesto disponible con el que cuente un proyecto para su desarrollo, influye directamente en la metodología a seleccionarse, puesto que aquellos proyectos de extensa dimensión, requieren mayor número de recursos. Lo ideal de este criterio de selección es que no demande gastos excesivos.

b) Tiempo de entrega

Lo ideal es que todo proyecto se realice y entregue lo más pronto posible; sin embargo, existen actividades propias de las metodologías, como entrega de documentación

formal, realización de reuniones periódicas, entre otras, que pueden extender o acortar tiempos. Es por esto, que el tiempo de entrega del proyecto es decisivo en cuanto a la metodología a seleccionarse. Las metodologías ágiles, como su mismo nombre lo indica, son conocidas por presentar ciclos o fases rápidas, por lo cual son consideradas ideales a la hora de llevar a cabo un proyecto que presenta un tiempo limitado.

c) Documentación

Puede generarse documentación formal tanto en metodologías ágiles como en aquellas tradicionales, sin embargo, para estas últimas, la rigurosidad o la extensión de los documentos sobrepasan a aquellos que se presentan en las metodologías ágiles. Si bien, la documentación puede ser omitida para ciertos proyectos, según su equipo de desarrollo y las condiciones acordadas con el cliente, otros no pueden dejar de lado el contar con sustentos documentados.

d) Recurso humano

Además del número de personas que conformen un equipo de desarrollo, es importante considerar el rol de cada uno de los integrantes. Y dentro de un proyecto, según la metodología, puede no solo ser necesario la presencia del equipo de desarrollo sino también del cliente e incluso de usuarios finales. Por tanto, es recomendable identificar la disponibilidad del personal que se requiere a corto y largo plazo, cuán importante es y de esta manera poder seleccionar que metodología permite cubrir estas particularidades.

e) Dimensión del proyecto

Identificar la dimensión de un proyecto de manera inmediata puede no ser una tarea sencilla, sin embargo, es de vital importancia conocer su dimensión, de manera que esto pueda alinear varios criterios, como los recursos que se requieren tanto humano como en tiempo, a su vez el presupuesto, entre otros; y así optar por la metodología que vaya acorde a lo identificado. En este criterio, a mayor dimensión del proyecto, mayor necesidad de recursos, por lo que lo “ideal” son proyectos de mediana o corta dimensión.

f) Adaptabilidad

Es importante mencionar que, en el transcurso del desarrollo de un proyecto pueden aparecer cambios significativos, y mucho depende del proyecto para que se considere importante la adaptabilidad que posea una metodología a dichos cambios. Si bien, todo proyecto es susceptible a mejoras, ciertos proyectos dentro de su desarrollo presentan cambios constantes, para lo cual es de suma importancia la flexibilidad que la metodología pueda brindar.

4.1.2 Desarrollo de criterios por metodología de desarrollo de SW

Las metodologías seleccionadas presentan múltiples características que al desplegarse bajo diferentes factores pueden convertirse en ventajas como en desventajas. A continuación, se lleva a cabo el desarrollo de una matriz (Tabla 20) que logre vislumbrar las diferencias que resaltan en cada una de las metodologías de desarrollo de acuerdo a criterios previamente definidos [47].

TABLA 20. METODOLOGÍAS DE DESARROLLO DE SW Y CRITERIOS DE EVALUACIÓN

CRITERIO	METODOLOGÍAS DE DESARROLLO DE SW			
	RAD	AUP	SCRUM	XP
Presupuesto disponible	Requiere un equipo disciplinado en manejo de costos	Demanda bajos costos	No demanda gastos en cuestiones de personal y recursos para el desarrollo de los proyectos	No demanda gastos excesivos en cuestiones de personal y recursos para el desarrollo de los proyectos
Tiempo de entrega	El tiempo de entrega, si bien no se le permite que sea extenso, se acorta en la medida de lo posible	Al ser una metodología ágil, el tiempo de entrega se no será extenso, sin embargo se lo establece sin cortes drásticos de tiempo	Resalta por sus drásticas reducciones de los tiempos de desarrollo	El tiempo de entrega se acorta lo mayormente posible, a fin de cumplir con una de las características de las metodologías ágiles
<i>Timeboxing</i>	Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario	Las funciones secundarias son eliminadas únicamente si se requiere cumplir con el calendario.	Las funciones secundarias son eliminadas como sea necesario para cumplir con el calendario	Las funciones secundarias son eliminadas únicamente si se requiere cumplir con el calendario
Documentación requerida	Produce la documentación necesaria para	Todo se describe concisamente utilizando poca	La poca documentación que esta metodología	Carece del manejo de una documentación formal para el desarrollo

	facilitar el futuro desarrollo y mantenimiento	documentación	ofrece es el código resultado de las diferentes iteraciones	de los proyectos, la escasa documentación que esta metodología presenta es generalmente código fuente
Recurso humano necesario	Equipos pequeños de profesionales, en caso de proyectos medianos. Proyectos grandes requerirán de mayor número de RRHH. Fundamental el involucramiento de los usuarios/clientes	Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto	Es la metodología que menos personal necesita ya que posee muy pocos roles y su tamaño crece dependiendo del grupo de programación que se recomienda debe ser conformado por 5 a 10 personas. Como metodología ágil tiene al cliente como parte del equipo llamándolo <i>Product Owner</i> , el cual participa constantemente de las correcciones y observaciones en las diferentes entregas de los <i>Sprints</i>	Cuenta con numerosos roles para el control de los procesos en la diferentes iteraciones y el número de personas puede aumentar dependiendo del tamaño del grupo de programadores, aun así, el total de miembros no se recomienda sobrepase de los 15. XP como parte de su estructura de desarrollo posee en sus prácticas como fundamental la participación del cliente no solamente como apoyo a los desarrolladores, sino formando parte del grupo. Dar más valor a la colaboración con el cliente que a la negociación contractual.
Dimensión del proyecto	Pueden ser proyectos de dimensión media o grande, sin embargo, cuando se trata de proyectos extensos, requiere recursos humanos suficientes para crear número de equipos correctos	Principalmente proyectos de mediana dimensión o pequeños	Orientada principalmente a proyectos no demasiado extensos	Orientada principalmente a proyectos no demasiado extensos
Adaptabilidad, respuesta a cambios	No todo tipo de aplicación es apropiada para RAD. El sistema debe ser modular, caso contrario construir los componentes para RAD traerá dificultades	Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo	Flexibilidad en respuesta a los cambios. Resalta por su gran adaptabilidad frente a cambios	Flexibilidad en respuesta a los cambios. Resalta por su gran adaptabilidad frente a cambios. Da más valor a la respuesta al cambio que al seguimiento de un plan

En la presente sección se caracterizó a cada metodología según criterios previamente establecidos, y en base a estas características cualitativas, se obtendrán las cuantitativas que dará lugar a la primera matriz que se requiere.

A continuación, partiendo del comportamiento que cada metodología presenta frente a cada criterio, se llevará a cabo su respectiva cuantificación.

4.1.3 Cuantificación de criterios por metodología de desarrollo de SW

En la presente sección, la matriz que se obtiene permite visualizar el grado de cumplimiento de cada metodología respecto a cada criterio, siendo la calificación valores de entre 1 y 5. El menor valor (mín. 1) representa un cumplimiento escaso o no deseado de cierto criterio, mientras el mayor valor (máx. 5) constituye el mejor o ideal cumplimiento de dicho criterio (Tabla 21).

TABLA 21. CRITERIOS CUANTITATIVOS EN RELACIÓN A LAS METODOLOGÍAS

#	CRITERIO	METODOLOGÍAS DE DESARROLLO DE SW			
		RAD	AUP	SCRUM	XP
1	Presupuesto disponible	3	4	5	4
2	Dimensión del proyecto	5	4	3	3
3	Tiempo de entrega	4	4	5	5
4	<i>Timeboxing</i>	5	4	5	4
5	Documentación requerida	3	3	1	2
6	Recurso humano necesario:				
6.1	Desarrolladores	3	3	1	2
6.2	Clientes	4	3	4	3
7	Adaptabilidad, respuesta a cambios	1	4	5	5
8	Iterativo	5	5	5	5

4.1.4 Evaluación de los requerimientos generales del proyecto

En esta fase del modelo propuesto, es fundamental la participación del líder del proyecto, quien será el encargado de evaluar características del proyecto de SW a llevar a cabo, mismo para el que se requiere una metodología de desarrollo de SW, con respecto a los criterios establecidos. Ante las preguntas que abarcan los criterios a ser evaluados, se obtienen respuestas afirmativas y negativas, asignándose valor de 1 y 0 respectivamente, según las necesidades que el proyecto presente (Tabla 22).

TABLA 22. CRITERIOS EVALUATIVOS DEL PROYECTO A DESARROLLARSE

#	Criterios evaluativos	SI (1) NO (0)
1	¿El presupuesto para el proyecto es limitado?	-
2	¿La dimensión del proyecto puede considerarse extenso, grande?	-
3	¿Se requiere entregar el proyecto en un corto periodo de tiempo, en relación con la dimensión del mismo?	-
4	¿Están dispuestos a eliminar funciones secundarias del SW a fin de cumplir con el calendario del proyecto?	-
5	¿Es necesario para el proyecto generar documentación robusta?	-
6.1	¿Se requiere en el proyecto un equipo de desarrollo robusto y amplio?	-
6.2	¿Se requiere en el proyecto el acompañamiento del cliente en el proceso o varias etapas del mismo?	-
7	¿El proyecto a desarrollar puede presentar cambios significativos en cualquier fase o etapa del proyecto?	-
8	¿Se requiere que en el desarrollo del proyecto se realicen iteraciones?	-

4.1.5 Producto de matrices

Las matrices son instrumentos del algebra que facilitan el ordenamiento y manejo de datos, y aparecen en situaciones propias de las Ciencias Sociales, Económicas y Biológicas [48]. En el modelo propuesto, una vez que se obtienen los datos necesarios de las dos matrices, se realiza un producto de dichas matrices, entre los resultados numéricos de las tablas 21 y 22, *Criterios cuantitativos* y *Criterios evaluativos* (matrices A y B), respectivamente.

$$A, B \in M_{m \times n}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \\ a_{71} & a_{72} & a_{73} & a_{74} \\ a_{81} & a_{82} & a_{83} & a_{84} \\ a_{91} & a_{92} & a_{93} & a_{94} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{51} \\ b_{61} \\ b_{71} \\ b_{81} \\ b_{91} \end{pmatrix}$$

$$A \in M_{9 \times 4}$$

$$B \in M_{9 \times 1}$$

$$A \times B = \begin{pmatrix} a_{11} \cdot b_{11} & a_{12} \cdot b_{11} & a_{13} \cdot b_{11} & a_{14} \cdot b_{11} \\ a_{21} \cdot b_{21} & a_{22} \cdot b_{21} & a_{23} \cdot b_{21} & a_{24} \cdot b_{21} \\ \vdots & \vdots & \vdots & \vdots \\ a_{81} \cdot b_{81} & a_{82} \cdot b_{81} & a_{83} \cdot b_{81} & a_{84} \cdot b_{81} \\ a_{91} \cdot b_{91} & a_{92} \cdot b_{91} & a_{93} \cdot b_{91} & a_{94} \cdot b_{91} \end{pmatrix}$$

$$A \times B = C = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ \vdots & \vdots & \vdots & \vdots \\ c_{91} & c_{92} & c_{93} & c_{94} \end{pmatrix}$$

$$C \in M_{9 \times 4}$$

4.1.6 Obtención de resultados

Una vez que se realiza el producto de matrices descrita en el punto anterior, se procede a realizar la sumatoria de los valores resultantes, columna a columna; obteniendo de esta manera la puntuación de cada metodología de desarrollo de software en base a los criterios establecidos por el líder de proyecto o jefe de proyecto.

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ \vdots & \vdots & \vdots & \vdots \\ c_{91} & c_{92} & c_{93} & c_{94} \end{pmatrix}$$

$$\sum c_{m1}, \sum c_{m2}, \sum c_{m3}, \sum c_{m4}$$

Con los valores obtenidos, ordenados de manera ascendente, se puede visualizar que metodología presenta el mayor valor, siendo esta la que se adapte de mejor manera al

proyecto a llevarse a cabo y para el cual se busca dicha metodología. Además, si no se pudiera optar por la metodología mejor puntuada, se cuenta con los valores siguientes, que a mayor valor mejor adaptación presentará.

De esta manera, se puede guiar en la toma de decisión de una metodología u otra de desarrollo de software para la ejecución de un proyecto en específico, en base a características básicas del mismo.

4.2 Caso aplicativo: Propuesta modelo evaluativo para selección de metodología de desarrollo de SW.

A continuación, se desarrolla el modelo propuesto dentro de un posible escenario.

Para dar inicio al desarrollo del modelo, se contactó a un líder de proyecto que llevará a cabo el desarrollo de una aplicación móvil, quien, al responder preguntas sobre su proyecto, permitió definir los criterios evaluativos del producto de software, que vienen a ser los requerimientos iniciales o básicos del mismo.

El líder de proyecto desarrollará una aplicación móvil que permita realizar compras en línea (*e-commerce*¹), con las siguientes particularidades. El proyecto en sí es extenso, cuenta con un presupuesto limitado, y el tiempo de entrega del mismo no es limitado. Sin embargo, estarían dispuestos a dejar atrás el desarrollo de funcionalidades extras o complementarias no primordiales para cumplir con los tiempos/calendario del proyecto. Documentación robusta, la presencia del cliente, y la adaptabilidad de la metodología a adoptarse en este proyecto son totalmente requeridas y necesarias. Mientras que, no se necesita un equipo de desarrollo robusto y grande, y tampoco es fundamental con iteraciones en el proceso de desarrollo de la app móvil. (Tabla 23).

¹ E-commerce: El comercio electrónico (Electronic Commerce, E-Commerce) consiste en la distribución, compra, venta, mercadotecnia y suministro de información complementaria para productos o servicios a través de redes informáticas como Internet u otras [51].

TABLA 23. CRITERIOS EVALUATIVOS: CASO APLICATIVO

#	Criterios evaluativos	SI (1) NO (0)
1	¿El presupuesto para el proyecto es limitado?	SI
2	¿La dimensión del proyecto puede considerarse extenso, grande?	SI
3	¿Se requiere entregar el proyecto en un corto periodo de tiempo, en relación con la dimensión del mismo?	NO
4	¿Están dispuestos a eliminar funciones secundarias del SW a fin de cumplir con el calendario del proyecto?	SI
5	¿Es necesario para el proyecto generar documentación robusta?	SI
6.1	¿Se requiere en el proyecto un equipo de desarrollo robusto y amplio?	NO
6.2	¿Se requiere en el proyecto el acompañamiento del cliente en el proceso o varias etapas del mismo?	SI
7	¿El proyecto a desarrollar puede presentar cambios significativos en cualquier fase o etapa del proyecto?	SI
8	¿Se requiere que en el desarrollo del proyecto se realicen iteraciones?	NO

Una vez que se obtienen los datos evaluativos del proyecto, se procede a realizar el producto de matrices, en base a los criterios cuantitativos definidos por metodología de desarrollo en el modelo propuesto y los criterios evaluativos (Tabla 21 y Tabla 23, matrices A y B respectivamente). Finalmente, tras el producto de matrices (matriz C) y la sumatoria vertical de los valores por metodología, se obtienen los valores resultantes (Tabla 24).

$$A = \begin{pmatrix} 3 & 4 & 5 & 4 \\ 5 & 4 & 3 & 3 \\ 4 & 4 & 3 & 5 \\ 5 & 4 & 5 & 4 \\ 3 & 3 & 1 & 2 \\ 3 & 3 & 1 & 2 \\ 4 & 3 & 4 & 3 \\ 1 & 4 & 5 & 5 \\ 5 & 5 & 5 & 5 \end{pmatrix} A \in M_{9 \times 4}$$

$$B = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} B \in M_{9 \times 1}$$

$$C = \begin{pmatrix} 3 & 4 & 5 & 4 \\ 5 & 4 & 3 & 3 \\ 0 & 0 & 0 & 0 \\ 5 & 4 & 5 & 4 \\ 3 & 3 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 4 & 3 & 4 & 3 \\ 1 & 4 & 5 & 5 \\ 0 & 0 & 0 & 0 \end{pmatrix} C \in M_{9 \times 4}$$

$$\text{TOTAL} = \begin{matrix} 21 & 22 & 23 & 18 \end{matrix}$$

TABLA 24. MATRIZ RESULTANTE: RECOMENDACIÓN DE METODOLOGÍAS DE DESARROLLO

#	CRITERIO	METODOLOGÍAS DE DESARROLLO DE SW			
		RAD	AUP	SCRUM	XP
1	Presupuesto disponible	3	4	5	4
2	Dimensión del proyecto	5	4	3	3
3	Tiempo de entrega	0	0	0	0
4	<i>Timeboxing</i>	5	4	5	4
5	Documentación requerida	3	3	1	2
6	Recurso humano necesario:				
6.1	Desarrolladores	0	0	0	0
6.2	Clientes	4	3	4	3
7	Adaptabilidad, respuesta a cambios	1	4	5	5
8	Iterativo	0	0	0	0
TOTAL		21	22	23	18

En el presente escenario, se tiene como resultado:

- ✓ RAD: 21
- ✓ AUP: 22
- ✓ SCRUM: 23
- ✓ XP: 18

Es así que se evidencia:

- Es cuestión del equipo de desarrollo y/o líder del proyecto, la selección de una metodología u otra, quien deberá tomar en cuenta que a mayor puntaje presente una metodología, mayor acoplamiento a los requisitos del proyecto tendrá. Por tanto, los totales de las metodologías funcionan como indicadores de acoplamiento de las mismas con respecto al proyecto evaluado.
- La metodología que mayormente cubre los requisitos del proyecto descrito (caso aplicativo) es SCRUM, y si bien, esta metodología puede o no ser adoptada por el equipo de desarrollo, adicional se brinda el listado del resto de metodologías con un

respectivo orden, permitiendo visualizar cuales se acoplan de mejor a no tan recomendable manera, siendo en este caso puntual: AUP, RAD y XP.

- Se pueden destacar los puntos fuertes de la metodología que se recomienda en primera instancia. Scrum permite un tiempo de entrega corto, no requiere gastos excesivos, es decir, se logra llevar a cabo con un presupuesto asequible para el proyecto. Adicional, Scrum realiza timeboxing, y se adapta a cambios en cualquier punto del proceso de desarrollo del SW, siendo flexible en su totalidad.

4.3 Conclusión

El modelo propuesto cumple con su objetivo principal, al presentar un listado de metodologías de desarrollo de software evaluadas en base a las necesidades de un proyecto en particular. Sin embargo, si bien la propuesta se basa en el producto de matrices, el uso de esta operación matemática para evaluación de metodologías de desarrollo de SW como tal, no se encuentra previamente aplicada.

En cuanto al modelo propuesto, se evidencia que los criterios cuantitativos (primera matriz) son resultantes de previos estudios, al ser valores que cuantifican características propias de las metodologías, y podrían incrementarse si así se deseara. Por otra parte, los criterios evaluativos (segunda matriz) van de la mano de los cuantitativos, puesto que son preguntas dirigidas al líder o jefe del proyecto, basadas en las características del proyecto al que se desea desarrollar.

Adicional, un desafío del modelo propuesto es la evaluación de un número limitado de metodologías, por lo que a futuro adicionar metodologías y/o métodos permitiría que se cuente con una variedad favorable para quien haga uso de este modelo propuesto, obteniendo un amplio abanico de recomendaciones.

Capítulo 5

Conclusiones y trabajo futuro

El objetivo inicial de este trabajo consiste en el estudio de los procesos de desarrollo de aplicaciones móviles actuales aplicados con mayor frecuencia a nivel de software, para finalmente, brindar una guía en la selección de una metodología de desarrollo de SW que sea la adecuada para un proyecto.

Para alcanzar los objetivos propuestos, al tratarse de un tema referente a desarrollo de aplicaciones móviles, fue indispensable analizar los enfoques de desarrollo de apps móviles nativas, web e híbridas. Se observó como la selección de un enfoque de aplicación depende mucho del escenario en el cual se implementará la app móvil. Sin contar, que cada tipo de app móvil presenta una fortaleza o característica propia: las apps móviles nativas hacen uso de las funcionalidades y componentes propios del dispositivo, las aplicaciones móviles web son actualizadas de forma inmediata requiriendo una única interfaz; y finalmente las aplicaciones móviles híbridas, al ser multiplataforma, no requieren de diferentes desarrollos para cada sistema operativo, minimizando el uso de recursos. Estas características relevantes permiten, a quien requiera desarrollar una app móvil, identificar qué enfoque de app móvil se adapta de mejor manera al entorno en el que se implementará y las necesidades que este SW deba cubrir.

Por otra parte, al estudiar las metodologías de desarrollo de SW que se han venido aplicando en el desarrollo de apps móviles, quedó en evidencia y se confirmó la falta de documentación formal sobre las mismas, problema inicial que dio lugar al presente trabajo. Para presentar las metodologías y modelos de desarrollo de software descritos en el capítulo 2, se establecieron dos grupos: los que se basan en metodologías ágiles y no ágiles. De esta manera, fue visible como cada modelo o metodología presenta su singularidad, sin embargo, las metodologías y modelos descritos no presentan características que limiten o especifiquen el tipo de software que se puede desarrollar bajo sus lineamientos, permitiendo el desarrollo apps móviles. Esta apertura puede no ser favorecedora en el caso de desarrollo de aplicaciones móviles, pues no

se cuenta con etapas que prioricen puntos relevantes a considerar en el desarrollo aplicaciones para dispositivos móviles, como el tipo y tamaño de las pantallas, diseño de la interfaz, la selección del tipo de aplicación (web, nativa, etc.), contar con sensores propios del dispositivo, y considerar que existen cientos de modelos de dispositivos móviles, entre otros.

Una vez analizadas las metodologías y modelos de desarrollo de SW, varios estudios dejan en evidencia los desafíos que se presentan en el uso de metodologías de desarrollo de apps móviles: gran parte de las aplicaciones móviles son relativamente pequeñas, usando rara vez procesos de desarrollo formal; los desarrolladores de aplicaciones móviles usan enfoques ágiles o aproximaciones ad hoc, ya que los teléfonos móviles evolucionan rápidamente, y requieren corto tiempo de desarrollo y planificación. Por otra parte, en cuanto al diseño de UI los desarrolladores de dispositivos móviles tienen como desafío dar el mejor uso posible del espacio de pantalla limitado, y el diseño de interfaz de usuario adquiere mayor importancia que nunca. Adicional, el enfoque de desarrollo de la app móvil viene a ser crucial de identificar, puesto que al decidir construir una única aplicación para todas las plataformas se corre el riesgo de incoherencias funcionales y si se llegara a considerar la creación de múltiples versiones orientadas a cada plataforma de hardware, será necesario mayor uso de recursos.

Y a pesar de los desafíos que actualmente existen con el desarrollo de apps móviles y el gran número de aplicaciones móviles en el mercado, los procesos de desarrollo de SW no están totalmente adaptados a los proyectos de desarrollo de aplicaciones móviles y son complejos de seguir, debido a que las características de los dispositivos móviles y los entornos reales en los que se implementarán las aplicaciones, derivan un nuevo conjunto de reglas. Adicional, se descubrió que gran parte de la documentación sobre desarrollo de apps móviles, se desarrollaba en un ámbito educativo, el cual presenta grandes diferencias en cuanto al ámbito empresarial. Son estos antecedentes que dieron lugar al siguiente cuestionamiento: ¿cómo se llevan a cabo los procesos de desarrollo de aplicaciones móviles en la industria y academia?, considerando dos ámbitos a diferenciarse.

Es así que se propone un enfoque de investigación, basado en directrices de Runeson et al. [39] cuya metodología se centra en una pregunta de investigación, a la cual se dará respuesta con sub preguntas que permiten tomar en cuenta a todas las aristas del tema a ser investigado. El caso de estudio elaborado está compuesto por unidades de análisis tanto de la industria como

la academia que, en base a técnicas investigativas de primer nivel, se logró identificar y caracterizar los modelos y metodologías de desarrollo de aplicaciones móviles utilizadas por empresas de desarrollo de software y desarrolladores independientes.

Para iniciar, uno de los hallazgos es que las aplicaciones móviles fueron desarrolladas en gran parte para Android, y en menor número para Android e iOS simultáneamente. Adicional, la mayoría de las aplicaciones móviles son nativas, seguidas por aplicaciones híbridas.

Se identificó la participación minoritaria del género femenino en proyectos de tecnología. En el contexto académico, los integrantes de los equipos de desarrollo son menores a 25 años; y en el industrial se presentan integrantes mayores de 35 años, siendo los más jóvenes inexpertos y los segundos, novatos, intermedios y mínimamente expertos.

En cuanto a la industria, la mayoría de empresas son pequeñas y una pequeña parte son empresas grandes. Los encuestados en el contexto empresarial, se constituyen en gran número por novatos y solo existe un experto, dejando en evidencia la falta de conocimiento de los miembros del equipo de desarrollo.

La mayoría de las UA invirtieron como máximo 12 meses para el desarrollo de aplicaciones, dejando en claro el corto tiempo de desarrollo y planificación requerido, siendo un factor para la selección de metodologías ágiles. Las UA en los contextos industrial y académico, hicieron mayor uso de métodos ágiles (SCRUM, XP y AUP). Sin embargo, en la academia, tres de las metodologías utilizadas son propuestas (ADHOC). El uso de ADHOC, ya sea con directrices ágiles o no ágiles, evidencia que raramente se hace uso de procesos formales de desarrollo, siendo este un motivo para crear nuevas propuestas, debido a la complejidad para el equipo de desarrollo de seguir procesos de desarrollo SW no adaptados totalmente a proyectos de desarrollo de aplicaciones móviles.

Adicional, fue visible la adopción mínima o nula de estándares que cubren la variedad de tamaños de pantalla, cámaras y el uso de sensores: GPS, acelerómetro, giroscopio, entre otros, lo que significa que las funcionalidades de los dispositivos móviles no se explotan en su totalidad o de la manera más adecuada. Además, la complejidad de contar con el suficiente tiempo para probar diferentes modelos de dispositivos móviles, son razones para dejar de lado las directrices específicas para aplicaciones en dispositivos móviles.

Estos resultados muestran la necesidad de compartir y proponer métodos específicos para el desarrollo de aplicaciones móviles que integren características y restricciones de dispositivos móviles. Y bajo esta necesidad surge el capítulo 4, propuesta de un modelo sencillo y rápido para la selección de una metodología de desarrollo de SW.

Las metodologías utilizadas para la propuesta de este modelo, son las identificadas en el capítulo 3: RAD, AUP, SCRUM y XP. El modelo consiste inicialmente, en definir características de las metodologías de desarrollo de SW, y en una primera matriz calificar a cada metodología en base a estas características. La segunda matriz es la evaluación de ciertos criterios (relacionados con las características de la primera matriz) de acuerdo a los requerimientos del proyecto a llevarse a cabo. Finalmente, al contar con estas dos matrices, se realiza un producto entre las mismas, y la sumatoria de los valores obtenidos, permite sugerir que metodologías mayormente se acogen a lo que requiere el proyecto evaluado.

Es así que se concluye que el modelo propuesto cumple con su objetivo principal, basándose en el producto de matrices. Sin embargo, el uso de esta operación matemática para evaluación de metodologías de desarrollo de SW como tal, no se encuentra previamente aplicada.

Tras todo lo expuesto, se evidencia la complejidad que abarca la selección de una metodología de desarrollo de una aplicación móvil, y las características propias, que tanto las aplicaciones móviles como los dispositivos en las que se desplegarán presentan. Si bien, seleccionar una metodología y un enfoque de desarrollo de una app móvil no es una tarea fácil, existen varios parámetros a considerar que pueden guiar en la toma de la decisión, sin embargo, esto no quita que ciertos desafíos se presenten en el transcurso del desarrollo del SW.

Trabajo Futuro

A futuro, hay varios avances que podrían darse en el presente trabajo. Como primera mejora, para el capítulo 3, se incrementarían UA y estas deberán ser de diferentes ubicaciones geográficas. Al contar con pocas UA y de mismas zonas geográficas, no se logra representar el mundo real. Adicional, las técnicas de investigación no deberán ser únicamente de primer nivel (encuestas y entrevistas), ya que el resultado puede basarse en opiniones y no en hechos.

Adicional, para establecer y formalizar el proceso de desarrollo de las aplicaciones móviles, se puede iniciar con definir reglas o directrices, que logren integrar las características y restricciones de los dispositivos móviles, así como técnicas de usabilidad para mejorar la experiencia del usuario al emplear una aplicación móvil. Otras investigaciones deberán abordar las necesidades expuestas e incluirán una experiencia práctica más amplia, tanto en términos de unidades de análisis como de preguntas de investigación.

En cuanto al capítulo 4, un trabajo a futuro en el modelo propuesto, es incrementar el número de metodologías para la evaluación de las mismas, de manera que se cuente con una variedad favorable para quien haga uso del modelo, obteniendo un mayor número de recomendaciones. De igual manera, pueden incrementarse los criterios o características a evaluarse por metodología.

Además, si bien el modelo propuesto permite la selección de una metodología de desarrollo de SW, un trabajo futuro sería la inclusión de criterios evaluativos que guíen en la selección del enfoque de desarrollo de la app móvil, es decir, si deberá ser una app móvil web, nativa o híbrida, según el escenario en el que se implementará y el mercado al que se planea llegar con la misma.

(Inicia en número de página impar)

Referencias

- [1] I. Update, O. N. The, P. Of, and T. H. E. Networked, “Mobility Report,” no. February, pp. 4–7, 2013.
- [2] Deloitte, “There’s no place like phone: Consumer usage patterns in the era of peak smartphone,” pp. 1–61, 2016.
- [3] R. Colomo-Palacios, J. Calvo-Manzano, A. De Amescua, and T. San Feliu, *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement*. 2014.
- [4] V. Rahimian and R. Ramsin, “Designing an Agile Methodology for Mobile Software Development : A Hybrid Method Engineering Approach,” pp. 351–356, 2007.
- [5] M. Stoica, M. Mircea, and B. Ghilic-Micu, “Software development: Agile vs. traditional,” *Inform. Econ.*, vol. 17, no. 4, pp. 64–76, 2013.
- [6] H. K. Flora, S. V Chande, and X. Wang, “Adopting an Agile Approach for the Development of Mobile Applications,” *Int. J. Comput. Appl.*, vol. 94, no. 17, pp. 975–8887, 2014.
- [7] W. McIver, “Software Engineering Processes for Mobile Applications Development,” *NSERC Mob. First, Frederict.*, vol. 1, no. 506, pp. 1–74, 2015.
- [8] H. K. Flora, X. Wang, and S. V.Chande, “An Investigation into Mobile Application Development Processes: Challenges and Best Practices,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 6, pp. 1–9, 2014.
- [9] IBM Software, “El desarrollo de aplicaciones móviles nativas, Web o híbridas,” 2012.
- [10] H. Heitkötter, S. Hanschke, and T. a Majchrzak, “Evaluating Cross-Platform Development Approaches for Mobile Applications,” *Web Inf. Syst. Technol.*, vol. 140, pp. 120–138, 2013.
- [11] J. W. & Sons, *Professional Android 4 Application Development*. 2012.
- [12] W. McIver, “Software Engineering Processes for Mobile Application Development,” 2015.
- [13] M. Reddy, *API Design for C++*, 1st ed. Morgan Kaufmann, 2011.
- [14] Ionic, “Welcome to Ionic,” *How to get the most out of Ionic.*, 2016. [Online]. Available: <http://ionicframework.com/docs/guide/preface.html>.
- [15] Angular, “Mobile Angular UI,” *Mobile Angular UI Documentation*. [Online]. Available: <http://mobileangularui.com/docs/>.
- [16] Intel, “Zona para desarrolladores Intel, XDK.” [Online]. Available: <https://software.intel.com/es-es/intel-xdk>.
- [17] A. Kosmaczewski, *Sencha Touch 2 Up and Running*, First Edit. Sebastopol: O’Reilly

- Media Inc, 2013.
- [18] S. Ganatra, *Kendo UI Cookbook*. Packt Publishing Ltd, 2015.
 - [19] Framework7, “Framework7,” *About Framework7*, 2016. [Online]. Available: <http://framework7.io/>.
 - [20] F. JQuery, “jQuery mobile,” *A Touch-Optimized Web Framework*, 2017. [Online]. Available: <http://jquerymobile.com/>.
 - [21] M. x O. U. Team, “Onsen UI,” 2017. [Online]. Available: <https://onsen.io/>.
 - [22] “React Native,” *LEARN ONCE, WRITE ANYWHERE: BUILD MOBILE APPS WITH REACT*, 2017. [Online]. Available: <https://facebook.github.io/react-native/>.
 - [23] B. Eisenman, *Learning React Native: Building Native Mobile Apps with JavaScript*. O’Reilly Media, Inc., 2015.
 - [24] P. S. Corporation, “NativeScript,” 2017. [Online]. Available: <https://www.nativescript.org/>.
 - [25] I. Sommerville, *Ingenieria del Software*, Séptima. PEARSON EDUCACION, SA., 2005.
 - [26] M. JAVANMARD and M. ALIAN, “Comparison between Agile and Traditional software development methodologies,” no. May. 2015.
 - [27] P. Mata, A. Chamney, G. Viner, D. Archibald, and L. Peyton, “A development framework for mobile healthcare monitoring apps,” *Pers. Ubiquitous Comput.*, vol. 19, no. 3, pp. 623–633, 2015.
 - [28] S. G. S. Scott E. Donaldson, *Successful Software Development*, First. 2001.
 - [29] P. Neumann, “Prototyping,” pp. 1–13, 2004.
 - [30] C. J. S. Roberth G. Figueroa and A. A. Cabrera, “METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES,” *Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación*, p. 9.
 - [31] H. K. Flora and S. V. Chande, “A Review and Analysis on Mobile Application Development Processes using Agile Methodologies,” *Int. J. Res. Comput. Sci.*, vol. 3, no. 4, pp. 8–18, 2013.
 - [32] R. Kalirungkut, “REVIEW ON THE EFFECTIVENESS OF AGILE UNIFIED PROCESS IN SOFTWARE DEVELOPMENT WITH VAGUE SYSTEM,” vol. 9, no. 10, pp. 1763–1768, 2014.
 - [33] K. Schwaber and Gustav Bergman, “The Scrum Story,” *SOFTHOUSE, LEAN Mag.*
 - [34] V. V., S. D.C., H. P., E. J.S., Q. M.E., and M. S., “Helping children eat well via mobile software technologies,” *2nd Int. Work. Mob. Dev. Lifecycle, MobileDeLi 2014*, pp. 9–16, 2014.
 - [35] A. P.a *et al.*, “Mobile-D: An agile approach for mobile application development,” *Proc. Conf. Object-Oriented Program. Syst. Lang. Appl. OOPSLA*, pp. 174–175, 2004.
 - [36] A. de A. de L. Española, “Real Academia Española,” *ad hoc*, 2017. [Online]. Available: <http://dle.rae.es/?id=0jmLlgY>.
 - [37] A. I. Wasserman, “Software engineering issues for mobile application development,” *Proc. FSE/SDP Work. Futur. Softw. Eng. Res. ACM, 2010.*, pp. 397–400, 2010.

- [38] J. Dehlinger and J. Dixon, “Mobile application software engineering: Challenges and research directions,” *Work. Mob. Softw. Eng.*, vol. 2, p. 2, 2011.
- [39] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.
- [40] R. K. Yin, *Case study research : design and methods*, 4rd ed., vol. 5. London, Sage, 2009.
- [41] T. S. Perry, “Are Women Exiting Engineering Because Men Have All the Fun?,” *IEEE Spectrum*, 2016. [Online]. Available: <http://spectrum.ieee.org/view-from-the-valley/at-work/tech-careers/are-women-being-pushed-out-of-engineering-because-men-have-all-the-fun>. [Accessed: 05-Jul-2017].
- [42] L. Kozlowski, “Women in Tech: Female Developers By the Numbers,” *Forbes*, 2012. [Online]. Available: <https://www.forbes.com/sites/lorikozlowski/2012/03/22/women-in-tech-female-developers-by-the-numbers/#2a74b42b1760>. [Accessed: 09-Jul-2017].
- [43] J. James, “IT gender gap: Where are the female programmers?,” *TechRepublic*, 2010. [Online]. Available: <http://www.techrepublic.com/blog/software-engineer/it-gender-gap-where-are-the-female-programmers/>. [Accessed: 07-Jul-2017].
- [44] O. Dieste *et al.*, “Empirical Evaluation of the Effects of Experience on Code Quality and Programmer Productivity : An Exploratory Study Empirical Evaluation of the Effects of Experience on Code Quality and Programmer Productivity : An Exploratory Study,” *Empir. Softw. Eng.*, 2016.
- [45] Superintendencia de Compañías, *Valores y Seguros, Resolución No. SC-INPA-UA-G-10-005*. Quito, Guayaquil, 2010.
- [46] A. Mishra, H. Yadav, S. Rani, and M. T. V. Design, “A Review of Different Methods for Matrix Multiplication Based on FPGA,” vol. 5, no. February, pp. 747–751, 2014.
- [47] T. Gómez, R. López, P. Pablo, S. Bacalla, U. Nacional, and M. De San, “Criterios de selección de metodologías de desarrollo de software,” *Ind. Data*, vol. 13, p. 74, 2010.
- [48] C. Dy, “Matrix Algebra and Applications,” pp. 173–256.
- [49] M. David, *Picture CSS3*, Second. 2013.
- [50] A. R. Shankar, *Pro HTML5 Games*, Third. 2013.
- [51] J. C. R. Licklider and R. Taylor, “Comercio Electrónico (E-Commerce).”

(Inicia en número de página impar)

Apéndices y anexos

Inicia en número de página impar)

Anexo 1

(Inicia en número de página impar)

ENCUESTA

Datos de la Empresa

1. Nombre:
2. Dirección:
3. Teléfono:
4. Ciudad:
5. Correo electrónico:
6. Web:
7. Número de empleados:
8. Se define como:
 - Micro empresa
 - Pequeña Empresa
 - Mediana Empresa
 - Gran Empresa
 - No lo conozco

Datos del encuestado

1. Nombre:
2. Formación Académica:
 - Secundario
 - Técnico / Tecnológico (Técnico Superior, Tecnólogo, Ingeniero Técnico)
 - Grado (Ingeniería, Licenciatura)
 - Posgrado (Master)
 - PhD
3. Función en la empresa/equipo de desarrollo:
4. Años de experiencia:
5. Número de Proyectos en los que ha participado:
App Móviles: Otros:

Datos del Equipo de desarrollo:

1. ¿Cuántas personas integran el equipo de desarrollo?
(# Personas)
 - De 1 a 5
 - De 5 a 10
 - De 10 a 15
 - Más de 15
2. ¿Cuántas mujeres y hombres forman parte del equipo de desarrollo?
 - a. Mujeres:
 - b. Hombres:
3. ¿Cuántas personas del equipo de desarrollo son parte de los siguientes rangos de edad?
(Años)
Menor a 25: #_____

- Entre 25 y 35: # _____
 Entre 35 y 40: # _____
 Más de 40: # _____
4. ¿Cuál es la formación académica del equipo de desarrollo?
 Secundario: # _____
 Técnico / Tecnológico (Técnico Superior, Tecnólogo, Ingeniero Técnico) : # _____
 Grado (Ingeniería, Licenciatura): # _____
 Posgrado (Master): # _____
 PhD: # _____
 5. ¿En cuántos proyectos han participado?
 a. Apps Móviles:
 b. Otros:
 6. ¿Cuántos años de experiencia tiene el equipo de desarrollo?
 a. Apps Móviles:
 b. Otros:

Datos de la Aplicación

1. ¿Cuál es el nombre de la aplicación?
2. Realice una breve explicación de su aplicación.
3. ¿Cuáles son los usuarios finales a los cuales va dirigida su aplicación?
4. ¿Para qué plataformas (sistema/s operativo/s) se desarrolló o se está desarrollando la aplicación móvil?
 iOS Android Windows Phone • Otro: _____
5. Se encuentra disponible para descargar
 Sí No ¿Por qué? _____
6. ¿De qué tipo, según el desarrollo, es su aplicación móvil? Seleccione.
 Nativa Web Híbrida
7. ¿Qué metodologías o procesos de desarrollo de software para aplicaciones móviles conoce?
8. ¿Se aplica o aplicó alguna metodología o proceso formal para el desarrollo de la aplicación móvil?
 Sí
 ¿Cuál es el nombre de la metodología o proceso aplicado?

 A su criterio la metodología o proceso aplicado se lo puede clasificar como:
 Ágil No Ágil
 No
 ¿Por qué?

9. Indique las actividades que compone la metodología o proceso que se está aplicando/se aplicó (Agradecemos que sea específica y detallada la respuesta en la medida de lo posible)

Act	Nombre	Descripción
1		
2		
...		
n		

10. Indique las razones del por qué eligió dicha metodología o proceso.
11. ¿Cuánto tiempo se empleó/empleará (estimado) en el desarrollo de la aplicación móvil?
 (# Meses)
 De 1 a 3
 De 3 a 6

De 6 a 12

Más de 12

12. ¿Cuál es el costo aproximado del desarrollo la aplicación? ¿Qué técnica de estimación se ha aplicado?
13. ¿A su criterio, en una escala de 0 a 10, donde 0 es bajo y 10 es alto, con qué nivel de rigurosidad se aplicaron todas las actividades de la metodología o proceso? ¿Por qué?
14. Indique cuáles son los problemas identificados durante el proceso de desarrollo con respecto a la aplicación.
15. Indique cuáles son los problemas identificados durante el proceso de desarrollo con respecto a la metodología o proceso que se siguió.
16. ¿Qué características y/o restricciones concretas del desarrollo de aplicaciones móviles cree que pueden haber afectado al proceso de desarrollo de la aplicación? ¿De qué manera?

Toda la información recopilada será utilizada para fines de investigación. Gracias

"I think it's a new feature. Don't tell anyone it was an accident. " - Larry Wall

(Inicia en número de página impar)

Anexo 2

INQUÉRITO PARA EMPRESAS

Formulário

Dados da empresa

1. Nome:
2. Morada:
3. Telefone:
4. Cidade:
5. E-mail:
6. Web:
7. Número de empregados:
8. Define-se como:
 - Micro Empresa
 - Pequena Empresa
 - Média Empresa
 - Grande Empresa
 - Desconheço
9. Número de trabalhadores (aproximado):

Dados pessoais

1. Nome:
2. Formação Académica:
 - Secundário
 - Técnico / Tecnológico
 - Licenciado
 - Pós-graduado (Mestre)
 - PhD
3. Função na empresa/equipa de desenvolvimento:
4. Anos de experiência:
5. Número de Projectos em que participou:
 - App Móveis: Outros:

Dados da equipa de desenvolvimento:

1. Quantas pessoas integram a equipa de desenvolvimento?
(# Pessoas)
 - De 1 a 5
 - De 5 a 10
 - De 10 a 15
 - Mais de 15
2. Quantas mulheres e homens integram a equipa de desenvolvimento?
 - a. Mulheres:
 - b. Homens:
3. Indique quantas pessoas da equipa de desenvolvimento fazem parte de cada gama de idades.

(Anos)

Menos de 25: # _____

Entre 25 e 35: # _____

Entre 35 e 40: # _____

Mais de 40: # _____

4. Qual a formação académica da equipa de desenvolvimento?
Secundário: # _____
Técnico / Tecnológico: # _____
Licenciado: # _____
Pós-graduado (Mestre): # _____
PhD: # _____
5. Em quantos projetos já participou a equipa de desenvolvimento?
 - a. Apps Móveis:
 - b. Outros:
6. Quantos anos de experiência tem a equipa de desenvolvimento?
 - a. Apps Móveis:
 - b. Outros:

Dados da aplicação

1. Qual o nome da aplicação?
2. Apresente uma breve descrição da aplicação.
3. Quais os utilizadores finais a que a aplicação é dirigida?
4. Para que plataformas (sistema/s operativo/s) se desenvolveu ou se está a desenvolver a aplicação móvel?
iOS Android Windows Phone • Outro: _____
5. Para que dispositivos móveis se desenvolveu ou está a desenvolver a app?
Smartphone Tablet Hand Held • Outro: _____
6. A app já está disponível na(s) store(s)?
Sim Não Porquê? _____
7. Qual o tipo da aplicação, de acordo com o tipo de desenvolvimento? Selecione.
Nativa Web Híbrida
8. Que metodologias ou processos de desenvolvimento de software para aplicações móveis conhece?
9. Aplica ou aplicou alguma metodologia ou processo formal para o desenvolvimento da aplicação móvel?
Sim
Qual o nome da metodologia ou processo aplicado?

De acordo com o seu critério, a metodologia ou processo aplicado pode classificar-se como:
Ágil Não Ágil
No
Porquê?
10. Indique as atividades que compõem a metodologia ou processo que está a aplicar ou aplicou. (Agradecemos que a resposta seja específica e detalhada na medida do possível.)

Act	Nome	Descrição
1		
2		
3		
4		
5		

6		
...		
n		

11. Indique as razões de ter escolhido essa metodologia ou processo.
12. Quanto tempo demorou ou vai demorar o desenvolvimento da aplicação?
(# Meses)
 - De 1 a 3
 - De 3 a 6
 - De 6 a 12
 - Mais de 12
13. Qual o custo aproximado da aplicação?
Que técnica de estimação usou?
14. De acordo com o seu critério, numa escala de 0 a 10, onde 0 é baixo e 10 é alto, qual o nível de rigor com que a metodologia ou processo foi aplicado a todas as atividades?
Porquê?
15. Indique quais os problemas identificados durante o processo de desenvolvimento com respeito à aplicação.
16. Indique quais os problemas identificados durante o processo de desenvolvimento com respeito à metodologia ou processo que se seguiu.
17. Que características e/ou restrições concretas do desenvolvimento de aplicações móveis que entende que possam ter afetado o processo de desenvolvimento da aplicação? De que forma?

Toda a informação compilada será utilizada apenas para fins de investigação. Agradecemos a colaboração.

"I think it's a new feature. Don't tell anyone it was an accident. " - Larry Wall