



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA**

**VICERRECTORADO DE INVESTIGACIÓN,  
INNOVACIÓN Y TRANSFERENCIA  
DE TECNOLOGÍA**

**CENTRO DE POSGRADOS**

**MAESTRÍA EN GERENCIA DE SISTEMAS**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE MAGISTER EN GERENCIA DE SISTEMAS**

**TEMA:**

**“DISEÑO DE UNA GUÍA PARA EL DESARROLLO DE  
APLICACIONES MÓVILES”**

**AUTOR: LUDEÑA CASTILLO, HERNÁN SIGEL**

**DIRECTOR: ING. PARRA MOREIRA, PABLO XAVIER**

**SANGOLQUÍ**

**2018**



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA**

**CENTRO DE POSGRADOS**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “DISEÑO DE UNA GUÍA PARA EL DESARROLLO DE APLICACIONES MÓVILES” fue realizado por el señor **Ludeña Castillo, Hernán Sigel** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 5 de mayo del 2018

**Ing. Pablo Parra M. MSG**

**C.C.: 1802733012**



## VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

### CENTRO DE POSGRADOS

### AUTORIA DE RESPONSABILIDAD

Yo, **Ludeña Castillo, Hernán Sigel** con cédula de ciudadanía n°1104011786, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño de una guía para el desarrollo de aplicaciones móviles** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas. Consecuentemente el contenido de la investigación mencionada es veraz

**Sangolquí, 5 de mayo del 2018.**

-----  
**ing. Hernán Sigel Ludeña Castillo**

**C.C 1104011786**



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y  
TRANSFERENCIA DE TECNOLOGÍA  
CENTRO DE POSGRADOS

AUTORIZACIÓN

Yo, **Ludeña Castillo, Hernán Sigel** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño de una guía para el desarrollo de aplicaciones móviles** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad

Sangolquí, 5 de mayo del 2018



ing. Hernán Sigel Ludeña Castillo

C.C 1104011786

## **DEDICATORIA**

A mi abuelita **Carmen**

A mis padres **Blanca y Sigel**

A mis hermanos **Jose, Ligia, Ana y Marcelo**

A mis sobrinos **Melissa y Michelle**

A mi esposa **Diana**

Y en especial a mi hija **Daniela**

**Hernán**

## **AGRADECIMIENTO**

Quiero expresar mi más sincero agradecimiento al director de Tesis, Ing. Pablo Parra, por su asesoramiento y confianza brindada en las diferentes etapas de la investigación, y al Coordinador de la Maestría Ing. Geovanny Ninahualpa, por su gestión y apoyo, en la consecución exitosa de este proyecto.

**Hernán**

## ÍNDICE

<b>CERTIFICADO DEL DIRECTOR .....</b>	<b>i</b>
<b>AUTORÍA DE RESPONSABILIDAD .....</b>	<b>ii</b>
<b>AUTORIZACIÓN .....</b>	<b>iii</b>
<b>DEDICATORIA.....</b>	<b>iv</b>
<b>AGRADECIMIENTO .....</b>	<b>v</b>
<b>ÍNDICE .....</b>	<b>vi</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>xii</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>xiv</b>
<b>RESUMEN.....</b>	<b>xv</b>
<b>ABSTRACT.....</b>	<b>xvi</b>
<b>CAPÍTULO I PLAN DE TESIS .....</b>	<b>1</b>
<b>1.1 Motivación y Contexto .....</b>	<b>1</b>
<b>1.2 Planteamiento del Problema.....</b>	<b>1</b>
<b>1.3 Formulación del Problema.....</b>	<b>2</b>
<b>1.4 Justificación e Importancia.....</b>	<b>3</b>
<b>1.5 Objetivo General .....</b>	<b>4</b>
<b>1.6 Objetivos Específicos .....</b>	<b>5</b>
<b>1.7 Metodología de la Investigación.....</b>	<b>5</b>
<b>1.7.1 Metodología .....</b>	<b>5</b>
<b>1.7.2 Técnicas .....</b>	<b>5</b>
<b>1.7.3 Herramientas .....</b>	<b>6</b>
<b>CAPÍTULO II PROYECTOS DE DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES .....</b>	<b>7</b>
<b>2.1 Introducción.....</b>	<b>7</b>
<b>2.2 Situación actual de los proyectos de desarrollo para dispositivos móviles..</b>	<b>7</b>
<b>2.3 Prácticas tradicionales y ágiles en proyectos de desarrollo para dispositivos   móviles.....</b>	<b>9</b>
<b>2.4 Convivencia entre prácticas tradicionales y ágiles .....</b>	<b>10</b>
<b>2.5 Porque fracasan los proyectos de desarrollo de aplicaciones para   dispositivos móviles.....</b>	<b>11</b>

<b>CAPÍTULO III MARCO TEÓRICO</b> .....	13
<b>3.1 La Era Móvil</b> .....	13
<b>3.2 Dispositivos Móviles</b> .....	14
<b>3.2.1 Concepto de Movilidad</b> .....	14
<b>3.2.2 Dispositivos Móviles</b> .....	14
<b>3.2.3 Evolución de los dispositivos móviles (Generaciones)</b> .....	15
<b>3.2.3.1 La Era Brick</b> .....	15
<b>3.2.3.2 La Era Candy Bar</b> .....	16
<b>3.2.3.3 La Era Feature Phone</b> .....	17
<b>3.2.3.4 La Era del Smartphone</b> .....	19
<b>3.2.3.5 La Era Touch</b> .....	20
<b>3.2.3.6 La Era del Futuro</b> .....	22
<b>3.2.4 Evolución de las tecnologías de red para teléfonos móviles</b> .....	25
<b>3.2.5 Características hardware y software de los dispositivos móviles</b> .....	26
<b>3.2.5.1 Características Hardware</b> .....	26
<b>3.2.5.2 Características Software</b> .....	27
<b>3.2.6 Sistemas Operativos</b> .....	28
<b>3.2.6.1 Windows Phone</b> .....	29
<b>3.2.6.2 Apple IOS</b> .....	29
<b>3.2.6.3 Android</b> .....	31
<b>3.2.6.4 Blackberry OS</b> .....	32
<b>3.2.6.5 Symbian OS</b> .....	33
<b>3.2.7 Sistemas Operativos móviles con menor popularidad</b> .....	33
<b>3.3 Aplicaciones Móviles</b> .....	34
<b>3.4 Aplicaciones Transaccionales</b> .....	36
<b>3.5 Tipos de Aplicaciones</b> .....	36
<b>3.5.1 Aplicaciones Web Móviles</b> .....	37
<b>3.5.2 Aplicaciones web progresivas o progressive web app (PWAs)</b> .....	40
<b>3.5.3 Aplicaciones híbridas</b> .....	43
<b>3.5.4 Aplicaciones híbridas – mixtas</b> .....	45
<b>3.5.5 Aplicaciones nativas</b> .....	47



<b>3.6 Elección entre nativo, web app o híbrido</b> .....	49
<b>3.7 Plataformas y frameworks de desarrollo móvil</b> .....	54
<b>3.7.1 Plataformas de desarrollo nativo</b> .....	54
3.7.1.1 Microsoft Xamarin .....	55
3.7.1.2 Adobe Flex .....	56
3.7.1.3 RubyMotion .....	56
<b>3.7.2 Plataformas de desarrollo híbrido</b> .....	56
<b>3.7.3 Adobe PhoneGap</b> .....	57
<b>3.7.4 Appcelerator Titanium</b> .....	57
<b>3.7.5 Ionic</b> .....	58
<b>3.7.6 Apache Cordova</b> .....	58
<b>3.7.7 React Native</b> .....	59
<b>3.8 Consideraciones y limitaciones</b> .....	59
<b>3.9 Ecosistema móvil</b> .....	63
<b>3.10 Fragmentación</b> .....	64
<b>3.11 Experiencia de usuario (UX)</b> .....	65
<b>3.12 Consideraciones en el diseño UX</b> .....	69
<b>3.13 Look and Feel</b> .....	71
<b>3.14 Design Thinking</b> .....	71
<b>3.15 Mobile First</b> .....	73
<b>3.16 Prototipos</b> .....	74
<b>3.17 Seguridad</b> .....	79
3.17.1 Panorama actual de las Metodologías de desarrollo .....	81
<b>3.18 Metodología para el desarrollo de aplicaciones para dispositivos móviles</b> .....	84
<b>3.19 Características de una metodología para el desarrollo de aplicaciones móviles</b> .....	85
3.19.1 Métodos tradicionales .....	87
3.19.1.1 Ciclo de vida .....	88
3.19.1.2 Modelo Waterfall (en cascada) .....	90
3.19.2 Métodos Agiles .....	91
3.19.3 Ventajas .....	93

3.19.4	Desventajas .....	96
3.2.1	XP.....	98
3.2.2	Kanban .....	100
3.2.3	Lean.....	101
3.2.3.1	LeanUX.....	103
3.2.4	Scrum .....	107
3.3	<b>Metodologías en el desarrollo de aplicaciones para dispositivos móviles</b>	112
3.3.1	Mobile-D .....	113
3.3.2	Masam .....	116
3.3.3	Hdm .....	119
3.3.4	SLeSS .....	122
3.4	<b>Propuestas para el desarrollo de aplicaciones para dispositivos móviles</b>	125
3.4.1	Un proceso integrado de desarrollo de software en móviles.....	125
3.4.2	Modelo de procedimientos “M-business” .....	127
3.4.3	Modelo de procesos MP–móvil .....	128
3.4.4	El modelo AGILEMAD .....	130
3.4.5	El modelo MDAM .....	131
3.5	Métodos Híbridos .....	134
3.6	Comparación de los métodos de desarrollo para móviles .....	135
3.7	Desarrollo de software ágil vs. Desarrollo de software tradicional .....	138
3.8	Criterios para la selección de buenas prácticas.....	141
	<b>CAPÍTULO IV GUÍA METODOLÓGICA PARA EL DESARROLLO DE APLICACIONES MÓVILES TRANSACCIONALES</b>	146
4.1	Introducción.....	146
4.2	Visión General.....	147
4.3	Principios de la guía.....	150
4.4	Objetivos de la guía.....	153
4.5	Características .....	154
4.6	Alcance.....	155
4.7	Público Objetivo .....	155
4.8	Controversia.....	156

4.8.1	Mejores prácticas de desarrollo.....	156
4.9	Descripción de los pasos de la guía .....	159
4.12.1	Fase I. Generación de la Idea .....	159
4.12.2	Fase II. Gobierno .....	165
4.12.3	Fase III Diseño .....	179
4.12.4	Fase IV. Prototipado.....	189
4.12.5	Fase V. Construcción.....	194
4.12.6	Fase VI. Integración.....	200
4.12.7	Fase VII. Verificación.....	206
4.12.8	Fase VIII. Lanzamiento y Monitoreo .....	213
4.10	Artefactos .....	221
4.11	Roles .....	221
4.14.1	Diseñador de Experiencia de Usuario (UX).....	222
4.14.2	Desarrollador Frontend.....	222
4.14.3	Desarrollador Backend .....	223
4.14.4	Equipo de Desarrollo (Team).....	223
4.14.5	Arquitecto .....	223
4.14.6	Scrum master .....	223
4.14.7	Analista en Diseño de Soluciones/ Dueño del Producto .....	223
4.14.8	Tester.....	224
4.14.9	Gerente de Proyecto/Responsable de proyecto .....	224
4.14.10	Arquitecto de Seguridad.....	224
4.14.11	Interesados/Stakeholders.....	224
4.14.12	Cliente .....	225
4.14.13	Usuario.....	225
4.12	Encuesta.....	225
4.15.1	Resultados .....	226
<b>CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>227</b>
4.15	Conclusiones .....	227
4.16	Recomendaciones .....	228
<b>ANEXOS .....</b>		<b>231</b>

<b>Anexo 1. Estudios de las metodologías de desarrollo de aplicaciones móviles .</b>	<b>231</b>
<b>Anexo 2. El modelo de madurez para el aseguramiento de software.....</b>	<b>234</b>
<b>Anexo 3. OWASP Top 10 Application Security Risks - 2017Risks – 2017 .....</b>	<b>235</b>
<b>Anexo 4. OWASP Mobile Security Top 10 2016 .....</b>	<b>235</b>
<b>Anexo 5. Formato de encuesta.....</b>	<b>236</b>
<b>Anexo 6. Resultados de la encuesta.....</b>	<b>250</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>287</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Motorola DynaTAC 8000X.....	16
<b>Figura 2.</b> Teléfono Nokia.....	17
<b>Figura 3.</b> Motorola V3 RAZR.....	18
<b>Figura 4.</b> Smartphone .....	20
<b>Figura 5.</b> iPhone SE .....	22
<b>Figura 6.</b> Cómo repercutirá en la vida diaria 5G .....	24
<b>Figura 7.</b> Sistemas operativos en teléfonos celulares.....	28
<b>Figura 8.</b> Logotipo del Sistema Operativo Windows 10 Mobile.....	29
<b>Figura 9.</b> Logotipo del Sistema Operativo iOS.....	30
<b>Figura 10.</b> Logotipo del Sistema Operativo Android .....	32
<b>Figura 11.</b> Logotipo del Sistema Operativo Blackberry OS.....	32
<b>Figura 12.</b> Logotipo del Sistema Operativo Symbian OS.....	33
<b>Figura 13.</b> Tipos de Aplicaciones.....	37
<b>Figura 14.</b> Las cinco fases del Design Thinking.....	72
<b>Figura 15.</b> Sketch.....	76
<b>Figura 16.</b> Wireframe .....	77
<b>Figura 17.</b> Mockup .....	78
<b>Figura 18.</b> Prototipo .....	79
<b>Figura 19.</b> Ciclo de Desarrollo Genérico .....	89
<b>Figura 20.</b> Ejemplo de fases del modelo waterfall .....	91
<b>Figura 21.</b> Scrum .....	108
<b>Figura 22.</b> Ciclo de desarrollo de Mobile-D.....	114
<b>Figura 23.</b> Procesos integrados y su relación en el modelo MP-Móvil.....	128
<b>Figura 24.</b> Prácticas ágiles para el desarrollo de apps .....	131
<b>Figura 25.</b> Fases de la metodología MDAM.....	132
<b>Figura 26.</b> Metodologías de Desarrollo .....	140
<b>Figura 27.</b> Ciclo iterativo de desarrollo. ....	155
<b>Figura 28.</b> Métodos usados en la guía metodológica .....	157
<b>Figura 29.</b> Prácticas para el desarrollo de soluciones móviles transaccionales.....	158
<b>Figura 30.</b> Obtención de un artefacto .....	221
<b>Figura 31.</b> Principales metodologías de desarrollo .....	250
<b>Figura 32.</b> Metodologías adaptables.....	252
<b>Figura 33.</b> Frecuencia de problemas .....	253
<b>Figura 34.</b> Proceso de la Guía .....	255
<b>Figura 35.</b> Valoración Ciclo Desarrollo .....	256
<b>Figura 36.</b> Ciclo Desarrollo .....	257
<b>Figura 37.</b> Pasos de la Guía .....	259
<b>Figura 38.</b> Prácticas propuestas .....	261
<b>Figura 39.</b> Necesidades del cliente.....	262
<b>Figura 40.</b> Enfoque ágil.....	264

<b>Figura 41.</b> Experiencia de Usuario.....	266
<b>Figura 42.</b> Construcción de APIs .....	268
<b>Figura 43.</b> Inclusión de Prototipado .....	270
<b>Figura 44.</b> Creación de código.....	271
<b>Figura 45.</b> Enfoque DevSecOps .....	273
<b>Figura 46.</b> Pruebas de calidad .....	275
<b>Figura 47.</b> Calidad del Producto .....	277
<b>Figura 48.</b> Monitoreo de App .....	279
<b>Figura 49.</b> Calificación de la guía.....	281
<b>Figura 50.</b> Nivel de Satisfacción de la guía.....	282
<b>Figura 51.</b> Recomendación de la guía .....	284

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	<i>Evolución de las tecnologías de red para teléfonos móviles</i>	25
<b>Tabla 2</b>	<i>Sistemas Operativos móviles con menor popularidad</i>	34
<b>Tabla 3</b>	<i>Tecnologías de las PWA</i>	42
<b>Tabla 4</b>	<i>Metodologías más usadas en estudios seleccionados</i>	83
<b>Tabla 5</b>	<i>Métodos ágiles</i>	93
<b>Tabla 6</b>	<i>Metodologías para el desarrollo de aplicaciones móviles</i>	112
<b>Tabla 7</b>	<i>Fases de MASAM</i>	118
<b>Tabla 8</b>	<i>Fases de Hybrid Methodology Design</i>	120
<b>Tabla 9</b>	<i>Fases del Enfoque SLeSS</i>	124
<b>Tabla 10</b>	<i>Comparativa de los métodos de desarrollo para aplicaciones móviles</i>	136
<b>Tabla 11</b>	<i>Fases de la Guía Metodológica</i>	149
<b>Tabla 12</b>	<i>Principales metodologías de desarrollo</i>	251
<b>Tabla 13</b>	<i>Metodologías adaptables</i>	252
<b>Tabla 14</b>	<i>Frecuencia de problemas</i>	254
<b>Tabla 15</b>	<i>Proceso de la Guía</i>	255
<b>Tabla 16</b>	<i>Valoración Ciclo Desarrollo</i>	256
<b>Tabla 17</b>	<i>Ciclo Desarrollo</i>	258
<b>Tabla 18</b>	<i>Pasos de la Guía</i>	260
<b>Tabla 19</b>	<i>Prácticas propuestas</i>	261
<b>Tabla 20</b>	<i>Necesidades del cliente</i>	263
<b>Tabla 21</b>	<i>Enfoque ágil</i>	265
<b>Tabla 22</b>	<i>Experiencia de Usuario</i>	267
<b>Tabla 23</b>	<i>Construcción de APIs</i>	269
<b>Tabla 24</b>	<i>Inclusión de Prototipado</i>	270
<b>Tabla 25</b>	<i>Creación de código</i>	272
<b>Tabla 26</b>	<i>Enfoque DevSecOps</i>	274
<b>Tabla 27</b>	<i>Pruebas de calidad</i>	276
<b>Tabla 28</b>	<i>Calidad del Producto</i>	277
<b>Tabla 29</b>	<i>Monitoreo de App</i>	280
<b>Tabla 30</b>	<i>Calificación de la guía</i>	281
<b>Tabla 31</b>	<i>Nivel de Satisfacción de la guía</i>	283
<b>Tabla 33</b>	<i>Recomendación de la guía</i>	284

## **RESUMEN**

En la última década, los dispositivos móviles han revolucionado las actividades que realizamos a diario, convirtiéndolos en herramientas de uso cotidiano y negocios, en gran medida por su fiabilidad, utilidad, portabilidad y en muchos casos por su seguridad. En sus inicios, eran exclusivos para transmitir voz, actualmente sus servicios se han expandido, con capacidades de transmitir audio, video, datos y ejecutar un sinnúmero de aplicaciones. Las aplicaciones móviles transaccionales han aumentado la demanda de desarrollo de proyectos, cuyos requisitos tienen ciertas particularidades que necesitan ser abordados por métodos más adaptables a su entorno. Al no existir un lineamiento de desarrollo apegado a la realidad se pone en duda el desarrollo, calidad, disponibilidad sin interrupciones y estabilidad de las aplicaciones móviles transaccionales, pudiendo incluso ser consideradas como enemigos por ser la puerta de entrada a vulnerabilidades. En base a la recopilación de información de tesis y artículos científicos sobre metodologías para el desarrollo de aplicaciones móviles, el presente trabajo enfoca sus esfuerzos en describir los pasos para mejorar el proceso de desarrollo sobre entornos móviles empresariales, siendo completamente diferente al proceso de desarrollo de software tradicional. El objetivo de la investigación es obtener una guía metodológica con las mejores prácticas de desarrollo de aplicaciones móviles transaccionales. La guía contribuirá con una mejor comprensión del proceso de desarrollo de aplicaciones móviles transaccionales, garantizará el cumplimiento de requerimientos del cliente, el aseguramiento de la calidad y la entrega de las soluciones móviles.

### **PALABRAS CLAVE:**

- **APLICACIONES**
- **MÓVILES**
- **MÉTODOS**
- **PRÁCTICAS**
- **GUÍA**



## **ABSTRACT**

In the last decade, mobile devices have revolutionized the activities we do every day, making them everyday tools and business, largely because of their reliability, usefulness, portability and in many cases their safety. In its beginnings, they were exclusive to transmit voice, now its services have been expanded, with capacities to transmit audio, video, data and to run countless applications. Transactional mobile applications have increased the demand for project development, whose requirements have certain peculiarities that need to be addressed by methods more adaptable to their environment. In the absence of a developmental framework attached to reality, the development, quality, uninterrupted availability and stability of transactional mobile applications is questioned and may even be considered enemies as the gateway to vulnerabilities. Based on the collection of thesis information and scientific articles on methodologies for the development of mobile applications, the present work focuses its efforts on describing the steps to improve the development process on business mobile environments, being completely different from the development process of traditional software. The objective of the research is to obtain a methodological guide with the best practices of mobile transactional applications development. The guide will contribute to a better understanding of the mobile transactional application development process, ensure compliance with customer requirements, quality assurance and delivery of mobile solutions.

### **KEY WORDS:**

- **APPLICATIONS**
- **MOBILE PHONES**
- **METHODS**
- **PRACTICES**
- **GUIDE**

# CAPÍTULO I

## PLAN DE TESIS

### 1.1 Motivación y Contexto

En la actualidad la existencia de muchas empresas TI (Tecnologías de información) dependen de la entrega de productos de alta calidad, con clientes satisfechos y proyectos terminados con éxito. Por tal motivo existe un alto interés en asegurar que los proyectos no fracasen, buscando para ello nuevas alternativas en cuanto a herramientas, guías o metodologías de desarrollo que permitan mejorar los procesos.

El software evoluciona muy rápidamente, dada la demanda y las oportunidades que ofrece el mercado en forma de necesidades no cubiertas. Al no existir lineamientos para el desarrollo de aplicaciones móviles transaccionales apegadas a nuestra realidad, las empresas improvisan utilizando diversas metodologías, ya sean tradicionales o ágiles. Estas metodologías no logran adaptarse totalmente al entorno móvil empresarial, debido a ciertas particularidades y restricciones especiales, que necesitan ser abordadas y priorizadas en una guía metodológica.

### 1.2 Planteamiento del Problema

Los proyectos de desarrollo de aplicaciones móviles tienen un elevado porcentaje de fracaso, por diversas razones, entre las más importantes tenemos, los errores en

estimación de costos, errores en la medición del riesgo o el desarrollo basado en programadores estrella.

El entorno de desarrollo actual no responde de manera eficiente a los cambios constantes de los requerimientos del producto en desarrollo, además al tratarse de una tecnología que está al filo de la vanguardia se requieren desarrolladores especializados que provocan aumento de costos.

Los entornos móviles empresariales, han creado un contexto de negocio altamente volátil, donde los requerimientos expresados hoy, ya no son válidos unos meses más tarde. Bajo esta nueva realidad, se requiere una guía metodológica que permita responder satisfactoriamente a los cambios del entorno. (Alaimo, 2013)

### **1.3 Formulación del Problema**

Se plantean las siguientes preguntas de investigación:

- ¿Qué problemas, restricciones y posibilidades existen en el desarrollo de aplicaciones móviles transaccionales?
- ¿Qué ciclo de desarrollo se aplica sobre las aplicaciones móviles transaccionales?
- ¿Cómo se implementa las prácticas ágiles en el desarrollo de aplicaciones móviles transaccionales?

- ¿Qué prácticas de las metodologías deben implementarse en la guía metodológica propuesta que permita mejorar el proceso de desarrollo de aplicaciones móviles transaccionales?

#### **1.4 Justificación e Importancia**

Hoy en día las empresas TI buscan mejorar el proceso de desarrollo de aplicaciones móviles, a través de la inclusión de nuevos métodos que permitirán abordar ciertos aspectos relevantes del proyecto, evitando de esta manera que los costos se eleven y los proyectos fallen.

La tendencia por el uso de los dispositivos móviles ha provocado que muchas empresas enfoquen sus esfuerzos en mejorar el estilo de vida de las personas, creando para ello nuevas y mejores aplicaciones móviles que resuelvan estas necesidades.

Tenemos el caso de Supermaxi y Megamaxi del Ecuador que poseen un asistente virtual, móvil, gratuito para teléfonos inteligentes, cuya función es crear una lista de compras y adquirir productos de supermercado para sus clientes. Esta aplicación cumple en parte las expectativas del usuario, ya que los problemas presentados luego de su lanzamiento se evidencian en los comentarios realizados por parte del usuario y como se van resolviendo a través de la retroalimentación de los usuarios. Estas tareas post producción se pueden evitar o disminuir si se incluye prácticas de participación del usuario en todo el ciclo de desarrollo de la aplicación móvil (Universo, 2014)

A pesar de la enorme cantidad de aplicaciones que surgen para acceder a nuevos servicios, en Ecuador existe aún mucho por explorar en el campo de las soluciones móviles. A ser un área muy poco explorada los errores son propensos de cometer, resultando sumamente importante que las empresas TI se pregunten sobre qué soluciones se necesitan en realidad para mejorar los procesos. Por tal motivo los proyectos móviles requieren de un análisis minucioso para que lleguen a ser rentables y exitosos.

El presente proyecto pretende entregar una guía metodológica que contribuya a mejorar los procesos de desarrollo de aplicaciones móviles transaccionales, y por ende su consecución exitosa. Aportando para que se conviertan en proyectos sólidos, de calidad, que brinden seguridad y que cumplan con todas las expectativas de sus usuarios. Ayudando a las empresas a estar por encima de la competencia, obtener más clientes y fortalecer su marca. La guía presentará los pasos para una correcta planeación de los proyectos móviles, diseño centrado en la experiencia de usuario, desarrollo seguro, programación, pruebas y monitoreo.

### **1.5 Objetivo General**

- Formular una guía metodológica para el desarrollo de aplicaciones móviles, a través de la implementación de características ágiles y elementos de mejores prácticas, para asegurar la calidad, contribuir en el cumplimiento de fechas de entrega y garantizar el cumplimiento de los requerimientos del cliente.

## 1.6 Objetivos Específicos

- Realizar el estado del arte de las metodologías y métodos propuestos para el desarrollo de aplicaciones móviles.
- Identificar y describir las principales prácticas que se aplican en la gestión y desarrollo de aplicaciones móviles.
- Documentar en una guía metodológica las fases y las buenas prácticas de ciclo de desarrollo de aplicaciones móviles.
- Validar la aceptación de guía metodológica con gerentes y líderes de proyecto.

## 1.7 Metodología de la Investigación

### 1.7.1 Metodología

- **Método Analítico:** utilizado para identificar los problemas existentes en la utilización de las metodologías de desarrollo de aplicaciones móviles.
- **Método descriptivo:** utilizado para describir y comparar las características de las metodologías de desarrollo de aplicaciones móviles, en cuanto a sus artefactos, roles, procesos y prácticas.

### 1.7.2 Técnicas

- **Técnica de Investigación Bibliográfica:** utilizada para recopilar la información referente a metodologías y métodos propuestos de desarrollo de aplicaciones

móviles.

- **Técnica de Encuestas:** utilizada sobre una muestra de gerentes y líderes de proyecto para validar la aceptación de la guía metodológica propuesta a través de los resultados obtenidos.

### 1.7.3 Herramientas

Para llevar a cabo la revisión bibliográfica, se realizó la búsqueda de literatura relevante a través de los siguientes instrumentos de investigación:

- Repositorios Digitales: Google Académico, ProQuest, Scopus, IEEE Explorer.
- Tesis de maestría del Repositorio de la Universidad de Fuerzas Armadas ESPE
- Encuestas y evaluación de resultados de aceptación.

## **CAPÍTULO II**

### **PROYECTOS DE DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES**

#### **2.1 Introducción**

El desarrollo de aplicaciones móviles es el proceso que permite crear software para realizar tareas específicas sobre un dispositivo móvil. El uso de estas aplicaciones ha crecido en los últimos años debido a la expansión de los dispositivos móviles y plataformas de desarrollo, a tal punto que son consideradas como el cambio tecnológico de mayor importancia en la automatización de procesos.

Los proyectos de aplicaciones móviles tienen ciertas consideraciones importantes que se deben tomar en cuenta a la hora de gestionarlos, como la existencia de una amplia gama de dispositivos, los múltiples sistemas operativos con sus versiones, y el apareamiento de un nuevo paradigma conocido como experiencia de usuario, que se origina a partir del cambio en la interfaz de usuario (IU), de ventanas, iconos, menús, punteros al uso de interfaces con tacto, gesto, voz y video. (Petty, 2011)

Las empresas TI buscan desarrollar aplicaciones móviles tomando en cuenta las consideraciones antes mencionadas, y con la ayuda de metodologías que les permitan llegar a tener parámetros aceptables de calidad, tiempo y costo.

#### **2.2 Situación actual de los proyectos de desarrollo para dispositivos móviles**



Los proyectos móviles han logrado satisfacer muchas necesidades de los usuarios, a través de aplicaciones que han cubierto la mayoría de los servicios de comunicaciones y entretenimiento. En términos de productividad, las empresas necesitan de aplicaciones móviles que permitan automatizar diferentes procesos transaccionales, administrativos, logísticos, ventas, etc. La demanda de nuevas y mejores aplicaciones es alta para el mercado, donde la experiencia de usuario UX (User Experience) juega un papel muy importante, ya que el usuario es el principal protagonista en la interacción con la aplicación.

A principios de la década, analistas informáticos incluyeron los proyectos de tecnología móvil como una de las tendencias más importantes de la industria. En la actualidad son considerados como una estrategia para la mayoría de las empresas, por tal motivo están invirtiendo y asumiendo riesgos para adoptar la tecnología móvil de manera inmediata.

La liberación frecuente de nuevas versiones de sistemas operativos y nuevos dispositivos móviles conduce a tomar medidas necesarias para asegurar un producto móvil. Resulta necesario incluir un análisis continuo de supervisión de desarrollo para evitar basarse en suposiciones y mantenerse centrado en términos de funcionalidad de la aplicación.

Mientras el número de proyectos de desarrollo de aplicaciones móviles está incrementando, el número de nuevas iniciativas de investigación con relación al proceso o ciclo de desarrollo ha sido escaso. La falta de lineamientos en los estándares de

desarrollo y procesos exponen al dispositivo móvil a ataques potenciales que necesitan ser tratados con prontitud. Estas deficiencias en el proceso de desarrollo han introducido nuevas preocupaciones a nivel informático y empresarial, ya que afectan directamente a la calidad y a la seguridad de aplicaciones móviles transaccionales lanzadas al mercado.

El desarrollo de aplicaciones móviles transaccionales está en un proceso de madurez, donde se está aprendiendo día a día aún como diseñar, construir y desplegar grandes aplicaciones móviles transaccionales complejas.

### **2.3 Prácticas tradicionales y ágiles en proyectos de desarrollo para dispositivos móviles**

El elevado crecimiento de las soluciones móviles es aprovechado por las empresas para ampliar sus servicios, posicionar su marca, e interactuar de manera eficiente con los clientes a través de notificaciones y otras funciones a través de las aplicaciones. A medida que las empresas incrementan sus procesos de negocio y ofrecen nuevos productos aumenta la presión de las empresas de Tecnologías de información (TI) en ofrecer nuevas aplicaciones móviles que satisfagan a las necesidades del mercado. Esta demanda requiere de un enfoque que se adapte al ritmo acelerado de cambios del mercado. (Amaya, 2013)

Las prácticas tradicionales de desarrollo utilizan un enfoque en cascada para desarrollar aplicaciones de escritorio, que no permiten adaptarse totalmente al desarrollo

de aplicaciones móviles. Por naturaleza las aplicaciones móviles requieren de un tratamiento más flexible como el enfoque de desarrollo ágil, debido a diversos factores propios del ecosistema móvil, como requisitos cambiantes, diversidad de dispositivos, conectividad de red, entre otras características específicas.

Los cambios vertiginosos en el mercado móvil presionan a los equipos de desarrollo y operaciones a adoptar prácticas ágiles de desarrollo que permitan desplegar constantemente nuevos productos. Por tal motivo resulta imprescindible emplear el enfoque ágil en el desarrollo de proyectos de aplicaciones móviles. (Lee, 2014)

## **2.4 Convivencia entre prácticas tradicionales y ágiles**

Debido a la creciente demanda de desarrollo de software para dispositivos móviles, resulta complejo y altamente costoso utilizar un solo método de desarrollo, mucho más si es tradicional.

Las empresas han optado por utilizar varios enfoques, dirigiendo sus esfuerzos en la entrega de prototipos funcionales para obtener una muy buena experiencia de usuario. Priorizan factores, como la comunicación entre los miembros del equipo más que a la documentación, así como la participación constante del cliente. También, omiten ciertos procesos que no aportan a la entrega de un producto y que solo producen retrasos.

Con todo lo anterior expuesto encuentran un balance entre los métodos tradicionales y ágiles con el único objetivo de lograr que un proyecto alcance el alto grado de satisfacción del cliente. (Leiva & Villalobos, 2015)

## **2.5 Porque fracasan los proyectos de desarrollo de aplicaciones para dispositivos móviles**

Las empresas TI se mueven muy rápidamente para abordar la demanda de aplicaciones que solicita el mercado. Sin embargo, existe un riesgo al moverse tan rápido. Las empresas pueden fracasar al lanzar rápidamente aplicaciones inseguras y con defectos al mercado, ya que pueden ser vulneradas o atacadas. En vista de ello las empresas se ven obligadas a liberar actualizaciones tan pronto obtengan la retroalimentación del usuario final.

Por tal motivo las aplicaciones pueden hundirse muy rápidamente luego del lanzamiento si durante su ciclo de vida de desarrollo no se realiza una retroalimentación de usuarios reales con los programadores y diseñadores.

Uno de los principales problemas en las empresas es no saber lo que realmente desean, y por lo tanto cometen muchos errores. Comienzan omitiendo la creación de la hoja de ruta donde se detalla cada paso y costos que asumirán. No toman conciencia sobre los costos de diseño y desarrollo que pueden ir variando durante la vida del proyecto. No consiguen retroalimentación de los consumidores, ni tampoco analizan lo que la competencia está haciendo.

Los proyectos pueden fracasar si no se cuenta con un equipo de desarrollo interno o freelance asignado al proyecto a tiempo completo, que permita abordar toda la complejidad del software para móviles. Las empresas son conscientes de lo muy complejo que resulta encontrar desarrolladores con habilidades de diseño y programación para manejar todo un proyecto, es por ello que cometen errores al contratar programadores o pasantes con muy poca experiencia en móviles.

La elección incorrecta de una persona encargada de gestionar el proyecto, es decir sin experiencia en proyectos de tecnologías móviles puede ser crucial a la hora de abordar un proyecto. Debido a que muchos gerentes de proyectos que trabajan bajo el enfoque tradicional no consideran prioritario contratar personal con habilidades de desarrollo móvil, incrementando aún más las probabilidades del fracaso. También asumen que los desarrolladores internos resolverán muchos problemas, omitiendo la posibilidad de contratar a empresa externa especializada con amplia experiencia demostrada en software para móviles. El problema puede verse agrandado si contratan una empresa externa sin solicitar evidencia de experiencia.

Las empresas deben ser conscientes que ponen mucho en juego cuando un proyecto móvil fracasa, ya que provocan una percepción negativa en el mercado, tendrán clientes insatisfechos y pérdidas por las inversiones realizadas. (Santa María, 2015)

## **CAPÍTULO III**

### **MARCO TEÓRICO**

#### **3.1 La Era Móvil**

En la actualidad las empresas TI se encuentran enfocadas en adoptar nuevas formas de interactuar con las aplicaciones móviles para mantener la fidelidad de los usuarios, en esta nueva era las aplicaciones móviles se convierten en la puerta de entrada al negocio digital.

Las empresas se enfrentan a nuevos desafíos en términos de desarrollo de aplicaciones móviles, ya que tendrán que actualizar no solo sus habilidades de desarrollo, sino también su comprensión de cómo entregar aplicaciones móviles más rápidamente.

La creciente demanda de aplicaciones obliga a las empresas TI utilizar nuevos enfoques para descentralizar, acelerar el desarrollo y lanzar rápidamente al mercado. Están provocando cambios en los sistemas empresariales de backend para permitir una mayor colaboración entre los equipos de desarrollo y operaciones. También se están incluyendo nuevas técnicas de mejora de experiencia de usuario UX, integración con asistentes personales virtuales (VPA) y el uso herramientas de desarrollo, que permitirán descubrir nuevas oportunidades de innovación.

Las aplicaciones van cambiando a medida que la tecnología avanza, por tal motivo resulta importante revisar las tendencias tecnológicas, la arquitectura, el marco y las herramientas móviles, para llegar a comprender si siguen siendo las mejores alternativas para la estrategia del negocio.

## **3.2 Dispositivos Móviles**

Para comprender el ecosistema móvil, se detallan los conceptos de movilidad, dispositivo móvil y se analiza la evolución de los dispositivos móviles.

### **3.2.1 Concepto de Movilidad**

La definición va más allá de un concepto de portabilidad, se relaciona todo un ecosistema donde se incluyen desarrolladores, herramientas y tecnologías disponibles, así como las prioridades de la empresa.

También podría considerarse como la capacidad de acceder a toda información en cualquier momento mediante un dispositivo.

### **3.2.2 Dispositivos Móviles**

Los dispositivos móviles son dispositivos computacionales de tamaño reducido, con capacidades especiales de memoria y procesamiento, relacionadas con movilidad, portabilidad de datos, aplicaciones, comunicación desde cualquier parte, y cuya posesión, así como su operación se asocia al uso personal. (Guevara, 2017)

Una de sus características importantes es la movilidad, al ser pequeños son portables y fácilmente empleados cuando se los transporta. Algunos ejemplos de dispositivos móviles son los teléfonos celulares, smartphones, smartwatch, tablets, PDA, entre otros.

### **3.2.3 Evolución de los dispositivos móviles (Generaciones)**

La tecnología móvil ha ido evolucionando de muchas maneras según las necesidades de los usuarios finales. En la industria a menudo se las conoce como generaciones o simplemente "G", haciendo referencia a la madurez y capacidad de las actuales redes móviles. Estas redes móviles son solo una parte de un complejo ecosistema móvil.

En (Fling, 2009) se señala cinco hitos para segmentar la historia de la tecnología móvil y las denomina eras de los dispositivos. Cada era hace referencia a las generaciones de la red celular. Estas eras se describen a continuación.

#### **3.2.3.1 La Era Brick**

En la primera era 1G (1973 - 1988), aparecen los primeros modelos de teléfonos celulares. Su costo en el mercado era relativamente alto, así como su tamaño debido a las baterías de gran tamaño.

Se caracterizó por utilizar tecnología analógica y por transmitir únicamente voz. La transmisión era inexacta y con mala calidad de sonido, baja capacidad de almacenamiento y seguridad completamente nula. La tecnología dominante fue el Sistema Avanzado de Telefonía Móvil AMPS (Advanced Mobile Phone System).



El teléfono más representativo sin duda fue el Motorola DynaTAC 8000X que se lo puede observar en la Figura 1.



**Figura 1.** Motorola DynaTAC 8000X

Fuente: (Fling, 2009)

### 3.2.3.2 La Era Candy Bar

La segunda era 2G (1988 - 1998), a diferencia de la anterior se caracterizó por usar comunicaciones digitales. Se popularizaron los servicios transferencia de datos y los servicios de mensajería corta SMS (Short Message Service) que estaban limitados por 140 caracteres. Además, estos dispositivos permitían conectarse con mayor rapidez a la señal de las redes.

Los teléfonos celulares reducen su tamaño, son más avanzados, y su costo aumenta para presentarnos nuevas funcionalidades. Utilizan protocolos de codificación más complejos y avanzados. Los mensajes SMS trabajan con la tecnología CDMA (Code Division Multiple Access) y el Sistema Global para Comunicaciones Móviles (GSM), estándar popularmente usando hasta la actualidad. La forma rectangular de estos teléfonos es lo más representativo, como se puede observar en la Figura 2.



**Figura 2.** Teléfono Nokia

Fuente: (Fling, 2009)

### 3.2.3.3 La Era Feature Phone

La era 2.5G (1998 - 2008), tuvo las evoluciones más importantes. Hasta este momento los teléfonos celulares realizaban tres actividades principales: llamadas de voz, enviar mensajes y jugar el clásico snake.

Aumentaron las velocidades de transferencia, se incluyeron nuevos servicios y el uso aplicaciones con acceso a Internet. También permitían escuchar música, y tomar fotografías. La demanda de celulares aumentó al introducirse la cámara y los modelos con forma slim. Aparecen nuevas tecnologías, destacándose el servicio de mensajería multimedia (MMS), entre otros protocolos como el WAP (Wireless Application Protocol), EDGE (Enhanced Data Rates for Global Evolution) y GPRS (General Packet Radio System). El intercambio de paquetes de datos se realiza a través de la red móvil (GPRS, CDMA, TDMA). En la Figura 3 se observa el teléfono más representativo de esta era.



**Figura 3.** Motorola V3 RAZR

**Fuente:** (Fling, 2009)

#### **3.2.3.4 La Era del Smartphone**

La tercera era 3G (2002 - Presente), se caracterizó por presentar la convergencia de datos y voz. Apareció la conexión inalámbrica a Internet (Wi-Fi) para dar soporte a las altas velocidades de acceso y a las altas transmisiones de datos. Se incluye los videos e imágenes en movimiento, video conferencia, descarga de archivos, y las aplicaciones multimedia.

Aparece el Smartphone que se destaca por el uso de sistemas operativos en común, con aplicaciones desarrolladas por terceros, pantallas muy largas y teclado de tipo QWERTY. Se destacan las tecnologías GPS (Sistema de Posicionamiento Global, UMTS (Universal Mobile Telephone Service) y HSDPA (High Speed Downlink Packet Access). El teléfono Smartphone se puede observar en la Figura 4.



**Figura 4.** Smartphone

Fuente: (Fling, 2009)

### 3.2.3.5 La Era Touch

En la cuarta era 4G (2007 - Presente), los smartphones tienen una interfaz más enriquecida con pantallas táctiles. El hardware y software evoluciona para mejorar el rendimiento. Aparecen nuevos dispositivos móviles conocidos como tablets y la televisión en alta definición (HD).

Se integraron nuevas tecnologías: GPS, USB, Wifi, bluetooth, touchscreen. Las velocidades de transmisión eran mayores a 301 Mbps y con técnicas de avanzado

rendimiento como MIMO y OFDM. Tenían una elevada calidad de Servicio (QoS). Aparecieron nuevas formas de interactuar y comprender la información.

En esta era se fueron integrando muchas tecnologías que permitieron mejorar el intercambio de información y la compatibilidad entre diferentes marcas y modelos. Las principales tecnologías fueron High Speed Packet Access (HSPA+), Long Term Evolution (LTE) y el Estándar IEEE 802.16.

Con el aumento de usuarios a nivel mundial se crean servicios para satisfacer nuevas necesidades y dar solución a problemas en diversos sectores de tipo empresarial, académico, salud o social. (Gasca, Camargo, & Medina, 2014)

Steve Jobs presentó al mundo el iPhone, que se observa en la Figura 5.



**Figura 5.** iPhone SE

Fuente: (Apple, 2017)

### 3.2.3.6 La Era del Futuro

La quinta era 5G tiene prevista su liberación para el 2018/2020. Actualmente está en etapa de desarrollo, sin estandarizar y con prototipos en desarrollo. Se dará origen a la tecnología inalámbrica ultrarrápida que se expandirá a todos los dispositivos y maquinas alrededor del mundo.

El internet de las cosas permitirá que todos los dispositivos estén conectados a una velocidad promedio de 5 a 10 gigabytes con tiempo de respuesta de 10 milisegundos, constituyéndose como un salto cuantitativo de conexión total. Los dispositivos serán capaces de aprovechar la localización, movimiento y el conocimiento para revolucionar la vida cotidiana de las personas.

El ancho de banda ultrarrápido permitirá conectar cosas que no se han conectado antes, como datos de estacionamientos, semáforos o máquinas industriales. Los automóviles podrán intercambiar datos y cloud computing podrá estar más cerca de los usuarios. El Internet de las Cosas (IoT) y el Big Data permitirán abrir nuevos servicios. Toda esta distribución de inteligencia por la red permitirá mejorar los servicios y la seguridad a nivel global.

La vida diaria de las personas mejorará con la conexión de los dispositivos a las redes 5G. A continuación, los ámbitos en que repercutirá:

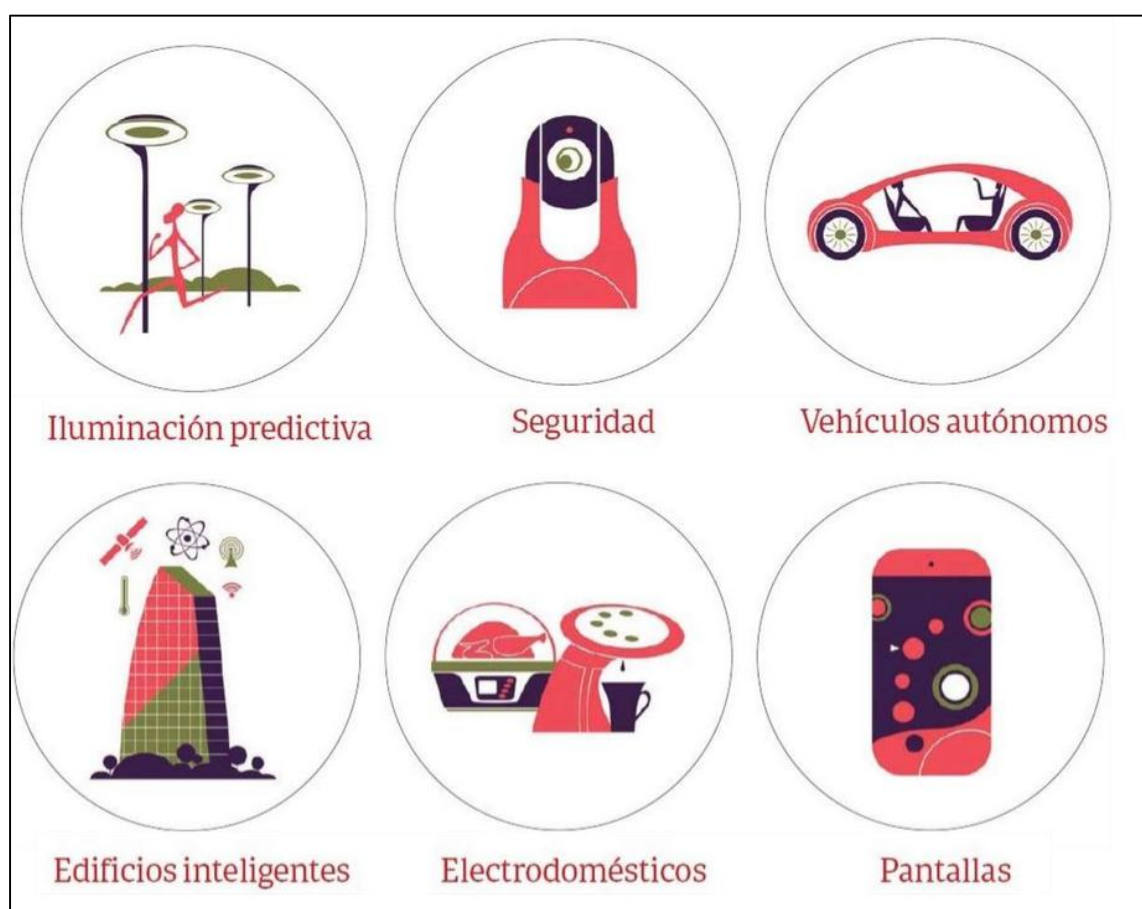
- **Edificios inteligentes.** Desde el teléfono móvil se podrá encender el aire acondicionado o encender las luces.
- **Electrodomésticos.** Una heladera podrá realizar el pedido del supermercado, conocer alimentos que están por vencer y que comida preparar con los alimentos disponibles.
- **Pantalla.** Los teléfonos inteligentes se convertirán en controles automáticos universales.
- **Iluminación predictiva.** A través de sensores se podrá apagar y reactivar la iluminación.
- **Seguridad.** A través del celular se podrá monitorear lo que ocurre en la oficina, el hogar o estacionamiento.



- **Vehículos autónomos:** Podrán realizar maniobras milimétricas.

La compañía china Huawei anunció la firma de un acuerdo con la operadora móvil rusa Megafon para estandarizar y desarrollar redes 5G de prueba, en vistas a la Copa Mundial de Fútbol de 2018. (Belucci, 2017)

En la Figura 6 se observa los ámbitos donde intervendrá las redes 5G, según lo explicado con anterioridad.



**Figura 6.** Cómo repercutirá en la vida diaria 5G

Fuente: (Belucci, 2017)

### 3.2.4 Evolución de las tecnologías de red para teléfonos móviles

En el estudio realizado por (Sonawane, 2013) se presenta una categorización realizada a partir de la velocidad de las tecnologías de red, la evolución se muestra en la Tabla 1.

**Tabla 1**

*Evolución de las tecnologías de red para teléfonos móviles*

Era	Velocidad de la red	Características
<b>1G</b>	No aplicable	<ul style="list-style-type: none"> <li>• Redes analógicas</li> <li>• Sólo servicio de voz</li> <li>• No hay servicio de datos</li> </ul>
<b>2G</b>	Hasta 20 kbps	<ul style="list-style-type: none"> <li>• Servicio de voz digital</li> <li>• Servicio de mensajes cortos (SMS)</li> <li>• Llamada de conferencia</li> <li>• Identificador de llamadas</li> <li>• Mensaje de voz</li> <li>• Correo electrónico y navegación web</li> </ul>
<b>2.5G</b>	Hasta 380 kbps (depende de la naturaleza de protocolo)	Todos los servicios 2G aplicables con: <ul style="list-style-type: none"> <li>• Mensaje multimedia Servicio (MMS)</li> <li>• Ubicación en tiempo real</li> </ul>
<b>3G</b>	Hasta 2.4 Mbps (depende de la naturaleza de protocolo)	Soporta todas las funciones 2G y 2.5G y las siguientes características: <ul style="list-style-type: none"> <li>• Vídeo de movimiento completo</li> <li>• Transmisión de video</li> <li>• Juegos en 3D</li> <li>• Navegación web más rápida</li> </ul>
<b>4G</b>	Hasta 100 Mbps (depende de la naturaleza de protocolo)	Soporta todas las funciones de red anteriores con las siguientes características especiales: <ul style="list-style-type: none"> <li>• Conferencia</li> <li>• Vídeo de transmisión de alta calidad</li> <li>• Alta calidad de voz sobre IP</li> </ul>
<b>5G</b>	5 y 10 gigabytes	<ul style="list-style-type: none"> <li>• Tiempo de respuesta de 10 milisegundos.</li> <li>• Conexión total de todos los objetos</li> <li>• Cloud computing</li> <li>• Internet de las Cosas (IoT) y el Big Data.</li> </ul>

### **3.2.5 Características hardware y software de los dispositivos móviles**

La evolución de los teléfonos inteligentes o Smartphone ha ocasionado que los teléfonos integren más características de mejora de experiencia de usuario, considerándolos hoy en día como microcomputadoras ya que hacen posible realizar muchas más tareas que antes solo las computadoras las podían realizar.

Los dispositivos móviles poseen ciertas particularidades propias que los diferencian de las computadoras personales. Se puede encontrar un conjunto de características y limitaciones hardware, así como una variedad de sistemas operativos y plataformas móviles.

#### **3.2.5.1 Características Hardware**

Las características hardware de los teléfonos los celulares se agrupan en dos tipos, de gama baja (menores requerimientos) y de gama alta.

Los dispositivos de gama baja presentan pantallas de diferentes dimensiones y resoluciones. Se distinguen dos tipos: táctiles y no táctiles (tradicionales). La memoria y su procesador son de capacidad menor respecto a las computadoras. El ingreso de datos se realiza de varias formas, tales como pantalla táctil, teclas softkey, tecla home, pad direccional, teclados numéricos y teclados de tipo QWERTY. Existe una amplia gama de sensores como acelerómetro, giroscopio, micrófono, o cámara. Además, soportan diferentes tipos de conectividad a redes 2G (voz digital), 2.5G, 3G (voz y datos), 4G

(multimedia), tecnologías Wi-Fi y bluetooth en sus múltiples versiones. La batería de poder es considerada baja dependiendo del tamaño del dispositivo computadoras.

Los dispositivos de gama alta presentan pantallas de alta definición con resoluciones HD, Full HD y 4K con tamaño de más de 5 pulgadas, procesador de hasta 8 núcleos, mientras más núcleos más rápido se ejecutan las tareas. Brindan una capacidad de escalamiento de hardware y clientes de correo. Utilizan tecnología force touch/ 3d touch en algunos teléfonos, que permite detectar los niveles de presión de pantalla para realizar acciones, como hacer zoom, abrir un menú o abrir una aplicación. Tiene capacidad de almacenamiento superior hasta los 256 GB, con memorias internas de 32 y 128 GB, y tarjetas de memoria microSD para duplicar la capacidad. Poseen cámaras de alta calidad, con estabilizadores óptico de imagen, tecnologías de luminosidad, y lentes que mejoran la calidad de las imágenes. Brindan un diseño estilizado con estructura metalizada, brillantes, con bordes redondeados y suaves. Las baterías son potentes y de gran duración, con soporte de largas horas de consumo de energía. También tienen parlantes estéreo con potente sonido y alta calidad de audio, protección contra el agua y polvo y lectores biométricos de huellas que mejoran la seguridad. (Luzardo, 2016)

### **3.2.5.2 Características Software**

Existe una diversidad de sistemas operativos y versiones como Android, iOS, Windows Phone, BlackBerry OS, Symbian o Linux Mobile. La creación de aplicaciones es específica para cada sistema operativo. Para el desarrollo de aplicaciones debe considerarse el impacto tanto en teléfonos de gama baja y media. Se presentan

plataformas de desarrollo para cada sistema operativo específico, siendo la incompatibilidad uno de los principales problemas. En la Figura 7 se observan los principales sistemas operativos para dispositivos móviles.



**Figura 7.** Sistemas operativos en teléfonos celulares

Fuente: *(Tecnología, s.f)*

### **3.2.6 Sistemas Operativos**

Los sistemas operativos fueron creados originalmente para computadores de escritorio y portátiles. Con el paso del tiempo los dispositivos móviles adoptaron sus propios sistemas operativos. En el caso de dispositivos móviles como teléfonos o tabletas los sistemas operativos se consideran más simples, con orientación a la conectividad inalámbrica y a necesidades específicas.

Entre los sistemas operativos móviles más usados en la actualidad tenemos: Windows Phone, Apple iOS y Android. Los sistemas operativos que también fueron muy usados son Symbian OS y Blackberry.

### 3.2.6.1 Windows Phone

Conocido también como Windows Mobile, es desarrollado por Microsoft y creado en base al núcleo de Windows CE. Es un sistema operativo que posee un sinnúmero de aplicaciones básicas propias y de terceros, su versión actual es la 10. Diseñado para ser similar a la versión escritorio de Windows. Posee un Marketplace llamado Microsoft Store donde se puede adquirir aplicaciones. Su logo se muestra en la Figura 8.



**Figura 8.** Logotipo del Sistema Operativo Windows 10 Mobile

Fuente: (Microsoft, 2017)

### 3.2.6.2 Apple IOS

Es un sistema operativo desarrollado por Apple, anteriormente denominado iPhone OS, era exclusivo para el teléfono iPhone, luego fue introducido en su línea de productos multimedia como el iPad y el iPod Touch. Creado a partir de Mac OS X el sistema operativo del portátil Mac.

Sistema exclusivo de Apple que no permite la instalación en hardware de terceros. Aunque, a través de jailbreak, los usuarios pueden acceder al sistema de archivos permitiendo instalar aplicaciones de terceros, considerado legal solo en los Estados Unidos.

Es una combinación perfecta entre software y hardware, para el manejo de la pantalla multitáctil. Su última versión es la 10 liberada en el 2017. Tiene una tienda de distribución de aplicaciones llamada App Store y una plataforma de desarrollo de aplicaciones llamada XCODE donde se puede crear aplicaciones con los lenguajes de programación Objective-C y Swift. Para poder compilar Se necesita de un computador con sistema operativo MacOS. El logo se muestra en la Figura 9.



**Figura 9.** Logotipo del Sistema Operativo iOS

Fuente: (Apple, 2017)

### 3.2.6.3 Android

Es un Sistema operativo basado en el núcleo de Linux, desarrollado por Google y actualmente líder del mercado, se encuentra diseñado para dispositivos móviles como smartphones, tablets, relojes inteligentes, televisores y automóviles.

Google posee una tienda para la descarga de aplicaciones llamada Google Play. Su última versión liberada es la 7.1.2 (Nougat). En marzo del 2017 Google presentó Android Go un sistema operativo más ligero para teléfonos de gama baja o con pocos recursos. (Gisbert, 2017)

El IDE o plataforma de desarrollo de aplicaciones es Android Studio, aunque anteriormente se usaba Eclipse. Tiene librerías de compatibilidad para usar características nuevas en dispositivos antiguos. Las pantallas se diseñan en formato XMLDocument. Se pueden crear recursos para pantallas con distintos tamaños. El lenguaje de programación de aplicaciones es Java y Kotlin, aunque a mediados del 2017 se anunció a Kotlin como principal lenguaje de programación. Su logo se muestra en la Figura 10.





**Figura 10.** Logotipo del Sistema Operativo Android

Fuente: (Google, 2014)

#### 3.2.6.4 Blackberry OS

Es un sistema operativo desarrollado por la empresa canadiense BlackBerry, antiguamente llamada RIM (canadiense). Es un sistema multitarea que posee sus propios métodos de entrada, el touchpad y las pantallas táctiles. Su última versión liberada fue la 7.1 en el 2014. Su predecesora es BlackBerry 10, que es un sistema operativo para teléfonos inteligentes Blackberry. Su logo se muestra en la Figura 11.



**Figura 11.** Logotipo del Sistema Operativo Blackberry OS.

Fuente: (BlackBerry, 2017)

### 3.2.6.5 Symbian OS

Fue un producto Open Source de la alianza entre Nokia y varias empresas de telefonía móvil como Sony Ericsson, Samsung, Siemens, Benq, Fujitsu, Lenovo, LG y Motorola. Este sistema operativo fue uno de los pioneros y más usados en su momento.

Actualmente existen pocos dispositivos con este sistema debido a la pérdida acelerada de usuarios, su última versión estable conocida fue la versión 10.1 lanzada en 2012. Su logo se muestra en la Figura 12.



**Figura 12.** Logotipo del Sistema Operativo Symbian OS

Fuente: *(Symbian, s.f)*

### 3.2.7 Sistemas Operativos móviles con menor popularidad

Entre los sistemas operativos móviles menos populares tenemos: Tizen, Firefox OS, Ubuntu y WebOS, como podemos observar en la Tabla 2.

**Tabla 2***Sistemas Operativos móviles con menor popularidad*

SISTEMA OPERATIVO	DESCRIPCIÓN
<b>Tizen</b>	Anteriormente llamado Bada, es un sistema operativo móvil basado en Linux y desarrollado por Samsung Electronics. Posee interfaces creadas en base a estándares web como HTML5 y diseñado para ser usado en dispositivos inteligentes como, teléfonos, televisores, tablets, netbooks y sistemas integrados de entretenimiento.
<b>Firefox OS</b>	Es un sistema operativo basado en Linux y HTML5, de código abierto, desarrollado por Mozilla Corporation, creado para smartphones y tabletas de gama baja. Su última versión estable fue 2.5.0 liberada en el 2015.
<b>Ubuntu Touch</b>	Sistema operativo basado en Linux y creado por la empresa Canonical, desarrollada para ordenadores de sobremesa, portátiles, netbooks, tabletas y teléfonos inteligentes.
<b>WebOS</b>	Sistema Operativo multitarea desarrollo por la empresa Palm, Inc., y basado en Linux para dispositivos inteligentes como televisores, relojes, móviles y tabletas.

### 3.3 Aplicaciones Móviles

Las aplicaciones móviles llamadas “apps” son software que cumplen funciones específicas, se encuentran presentes en los dispositivos móviles, e incluidas desde un principio en los sistemas operativos para móviles. Las aplicaciones móviles permiten a los usuarios realizar actividades diarias a través de un dispositivo móvil.

Existe una diferencia entre aplicación web y móvil. Las aplicaciones móviles son descargadas e instaladas antes de su uso, mientras que una aplicación web puede

accederse desde un navegador usando el internet y adaptándose a la pantalla del dispositivo móvil. (web responsive).

Se pueden clasificar dependiendo de los servicios que ofrecen, tales como: bancarios/financieros, compras/ventas al detalle, restaurantes mapas/navegación, productividad, educación/aprendizaje, clima, comunicación (voz IP), salud, viajes, comida/bebida, bienes raíces, turismo/hoteles, guías temáticas, estacionamientos, bienestar social, redes sociales, juegos, música, noticias, entreteniéndose/ocio, video/películas/imágenes o deportes.

Dependiendo de su conexión se distinguen dos tipos, aplicaciones auto-contenidas y aplicaciones con conexión a internet.

Las aplicaciones auto-contenidas poseen contenido estático como imágenes y menús que nunca cambian en el tiempo, el contenido se encuentra alojado en la aplicación. Son de tipo nativo y no requieren una conexión a internet, ni alojamiento en hosting. La app calculadora es un ejemplo de este tipo de aplicación.

Las aplicaciones con conexión a internet presentan contenido alojado en un servidor o mejor conocido como sistemas de backend. Generalmente manejan una base de datos, cuya información es accedida a través de servicios web de tipo REST o SOAP. Los cambios en la aplicación se pueden realizar a través de una consola administrativa. Su

desarrollo es considerado mucho más complejo que las aplicaciones auto-contenidas. Ejemplos de estas aplicaciones tenemos facebook, twitter y mensajería instantánea.

### **3.4 Aplicaciones Transaccionales**

Los sistemas transaccionales permiten automatizar procesos operativos dentro de una organización, su función primordial es procesar transacciones tales como pagos, cobros, entradas, salidas, etc. De estos sistemas se desprenden las aplicaciones transaccionales, que ejecutan tareas especificadas y son creadas en base a las necesidades de la organización.

Bajo esta perspectiva las aplicaciones móviles pueden considerarse de tipo transaccional cuando facilitan las tareas, atienden necesidades, optimizan procesos internos o permiten acceder a servicios. Entre las principales áreas donde se manejan aplicaciones transaccionales tenemos el sector financiero (mobile-banking/e-commerce), comercial (m-commerce), empresarial (m-business), académico (m-learning), o salud (m-health).

### **3.5 Tipos de Aplicaciones**

En el desarrollo de aplicaciones móviles se identifican cuatro tipos:

- Aplicaciones web móviles
- Aplicaciones híbridas -web
- Aplicaciones mixtas

- Aplicaciones nativas

Dentro de las aplicaciones web móviles se distingue:

- Aplicaciones Web Progresivas (PWAs)

En la Figura 13 se muestran los tipos de aplicaciones:



**Figura 13.** Tipos de Aplicaciones

Fuente: (Amrhein, 2013)

### 3.5.1 Aplicaciones Web Móviles

Son conocidas como webapps, no son aplicaciones reales, son sitios web que se ven y sienten como aplicaciones nativas, pero no se implementan como tal.

Utilizan el desarrollo web regular, pero con características optimizadas para poder ser visualizadas desde dispositivos móviles. Son ejecutadas por el navegador que viene por defecto dentro de los dispositivos. Su acceso se lo realiza escribiendo la url de página web y dependiendo del tamaño se ajustará a la pantalla, este comportamiento también es conocido como diseño web responsive o adaptativo.

Las principales características de las aplicaciones web móviles se describen a continuación:

- Su desarrollo se realiza dentro de un browser utilizando lenguajes de desarrollo web HTML5, CSS3, Javascript y un framework web móvil como por ejemplo jqueryMobile, KendoUI, AngularJS o Ionic.
- Para acceder solo se requiere un navegador desde un dispositivo móvil.
- Todos los dispositivos desplegaran la página de la misma forma.
- Son aplicaciones rápidas y baratas de desarrollar, pero son menos potentes que las aplicaciones nativas.
- Son utilizadas para brindar información desde cualquier dispositivo, independientemente del sistema operativo, inclusive desde una computadora.
- Se pueden programar de forma independiente al sistema operativo. No emplean ningún “software development kit” o SDK (kit de herramientas de desarrollo).

Las principales ventajas de las aplicaciones web móviles se describen a continuación:

- La compatibilidad es una de sus principales ventajas. Se utiliza un solo código fuente para todos los sistemas operativos.
- Son utilizadas desde el navegador de cualquier dispositivo independientemente del sistema operativo.
- No requieren ninguna aprobación en las tiendas de aplicaciones para su publicación, son aplicaciones que residen en servidores y solo con acceder a una URL los usuarios pueden utilizar la aplicación.
- No necesitan instalarse, se comercializan y promocionan de forma independiente.
- Al tratarse de aplicaciones que funcionan sobre la web el usuario siempre va a estar viendo la última versión, en vista de ello no es necesario que el usuario ejecute actualizaciones, ya que son inmediatas y no requieren de la autorización.

Las principales desventajas de las aplicaciones web móviles se describen a continuación:

- No pueden ser publicadas en plataformas para su distribución.
- El usuario tiene que digitar la dirección URL en el navegador.
- Tienen restricciones e inconvenientes en factores importantes como gestión de memoria.



- No pueden hacer uso de los componentes hardware de un teléfono, como GPS, cámara web, o acelerómetro. Al ser accedidas desde un navegador no tienen disponibilidad de ciertos privilegios para manipular componentes.
- Deben adaptarse a las medidas del dispositivo móvil siendo en algunos casos imprecisa.
- En la mayoría de los casos no se pueden usar en modo offline. Esto no aplica para iOS ya que implementa un WebStorage de HTML5 y Offline Application Cache.
- Al tener una interfaz más genérica e independiente, la experiencia de usuario respecto a los elementos de navegación e interacción son de menor calidad que las nativas.

Ejemplos de aplicaciones móviles web tenemos Facebook Móvil o LinkedIn Touch.

### **3.5.2 Aplicaciones web progresivas o progressive web app (PWAs)**

Conocidas como PWA, término utilizado para nombrar a una nueva generación de aplicaciones. Utilizan las últimas tecnologías preexistentes, técnicas y patrones del desarrollo web para presentar una experiencia y comportamiento en móviles similar a una aplicación nativa.

Las principales características de las aplicaciones web progresivas se detallan a continuación:

- Son páginas web que se comportan como aplicaciones nativas.
- Aumentan su funcionalidad dependiendo del dispositivo donde se ejecutan.
- Son construidas usando estándares de desarrollo web, HTML5, CSS y un API de Javascript como AngularJS o Ionic.
- Se comportan como aplicaciones web nativas, pero usando tecnologías web.
- La Interfaz de usuario es lo más parecida a una nativa.
- Pueden trabajar en modo offline o sin conexión
- Poseen el mayor rendimiento posible en móviles y se cargan en forma instantánea.
- No utilizan ningún envoltorio como Apache Cordova/Phonegap sobre aplicaciones web, como lo hacen las aplicaciones híbridas para convertirlas en nativas.
- Las PWA aparecen con la nueva generación de estándares web y navegadores.
- Tienen la capacidad de enviar notificaciones a los usuarios como una aplicación nativa.

Las tecnologías con las que trabajan las PWAs, se observan en la Tabla 3.

**Tabla 3***Tecnologías de las PWA*

TECNOLOGÍA	CARACTERÍSTICAS
Responsive Web Design	<ul style="list-style-type: none"> <li>• Es una técnica que permite a las aplicaciones adaptarse al tamaño de pantalla.</li> <li>• Utilizan frameworks específicos y animaciones CSS para crear interfaces de usuario.</li> <li>• Se pueden usar técnicas como Mobile First (primero se diseña el contenido en el móvil y luego se incrementa conforme la pantalla aumenta) y las MediaQueries (reglas de CSS que se aplican a ciertos tamaños de pantalla)</li> </ul>
Service Workers	<ul style="list-style-type: none"> <li>• Permiten ejecutar servicios en segundo plano en los navegadores de manera independiente a la aplicación.</li> <li>• Es un proxy entre el navegador y el servidor, que permite el acceso offline a la página y la carga mucho más rápida que una aplicación nativa.</li> <li>• Es soportado por los navegadores Firefox y Chrome.</li> <li>• Poseen capacidades avanzadas como, trabajar sin conexión, enviar notificaciones, descargas en segundo plano, cacheado de información e interceptación de comunicaciones.</li> </ul>
App Shell	<ul style="list-style-type: none"> <li>• Es un patrón que permite dividir la aplicación para cargar por separado la funcionalidad y el contenido.</li> </ul>
Manifiesto de aplicación	<ul style="list-style-type: none"> <li>• Brinda la posibilidad de anclar una página web desde el navegador hacia inicio, para posteriormente acceder a ella. (Alarcón, 2016)</li> <li>• Android y Chrome usan el archivo "manifiesto".</li> <li>• Permite controlar el aspecto del icono utilizando diversas técnicas.</li> </ul>

Las principales **ventajas** de las aplicaciones web progresivas se detallan a continuación:

- **Seguridad:** funcionan a través del protocolo HTTPs, canal seguro donde la información no puede ser comprometida

- **Velocidad:** ofrecen una interacción rápida y responsiva, cercana o igual a la experiencia de las aplicaciones nativas.
- **No dependen de la conexión:** Pueden ser accedidas sin internet, o con una conexión lenta de internet.
- **Actualizadas:** a través de actualizaciones en el background y websockets, el usuario puede recibir la última información en tiempo real.
- **Son como apps:** el usuario no siente la diferencia ya que se ven como aplicaciones nativas.
- **Linkeables:** la web está basada en links, al no necesitar instalar nada, se puede descubrir nuevas páginas navegando y linkeando entre sí.

La principal desventaja de las aplicaciones web progresivas es:

- Su audiencia es limitada al estar disponibles para versiones recientes de Chrome y Android.

### 3.5.3 Aplicaciones híbridas

Este tipo de aplicaciones es una combinación de las aplicaciones tipo web y nativas, se desarrollan usando lenguajes de desarrollo web, y un framework dedicado como Apache Cordova/Phonegap; Xamarin Studio o Titanium Appcelerator. Los frameworks dedicados sirven para crear o empaquetar las aplicaciones híbridas a una aplicación de tipo nativa. (Otero, Martínez, & Díaz, s.f)

Las principales características de las aplicaciones híbridas se detallan a continuación:

- No tienen un entorno de desarrollo específico, y usan herramientas gratuitas.
- Generalmente su desarrollo consiste en tomar el desarrollo de Web Móvil creado en HTML5, CC3, JavaScript y empaquetarlo en forma de aplicación.
- La lógica de la aplicación no reside en el servidor sino en el dispositivo.
- La comunicación con los componentes del teléfono es vía comunicadores como Apache Cordova/PhoneGap, ejecutándose en Shell Nativo y con infraestructura de tipo web.
- Todas las ventanas son nativas, y funcionan como contenedores de vistas web.
- Se puede reutilizar las vistas y/o la lógica.
- Al igual que las aplicaciones web, se basan en HTML de un navegador, con la advertencia de que el navegador está incrustado en la aplicación.
- Recibe información al instante desde una página web o sistema web.
- Tienen un diseño visual que no se identifica con el sistema operativo, pero puede usar controles y botones nativos de cada plataforma.

Las principales ventajas de las aplicaciones híbridas se detallan a continuación:

- El uso de los recursos del dispositivo y del sistema operativo se realiza en forma directa, permitiendo el acceso a buena parte de los componentes y sensores del teléfono.
- El costo de desarrollo es bajo en comparación al desarrollo nativo.
- Se puede crear aplicaciones multiplataforma y, por lo tanto, reducir significativamente los costes de desarrollo
- Al igual que las aplicaciones nativas, se alojan en una tienda de aplicaciones para su distribución.

Las principales desventajas de las aplicaciones híbridas se detallan a continuación:

- La documentación en internet es limitada y desordenada.
- Las plataformas de empaquetamiento tienen un costo.
- Empaquetar el mismo código de una plataforma a otra, no siempre resulta ser transparente, debiendo hacerse modificaciones en la capa de presentación.

#### **3.5.4 Aplicaciones híbridas – mixtas**

Las aplicaciones mixtas han tenido más auge en los últimos tiempos. Son consideradas mixtas cuando el 40% del desarrollo es nativo.

Las principales **características** de las aplicaciones híbridas-mixtas se detallan a continuación:

- Tiene su interface, UI o frontend dividida en dos partes, una parte web en un servidor o en la aplicación, y una parte en la capa de presentación de forma nativa.
- Se debe utilizar el desarrollo nativo cuando sea necesario, por ejemplo, para usar la cámara, o al utilizar desarrollo web en partes más complejas o muy repetitivas.
- Son implementadas por grandes empresas como LinkedIn e Instagram.
- Cuando las vistas son web se utiliza un Webview o componente a nivel de desarrollo nativo, es aquí donde se alojará la parte HTML, haciendo que luzca como una aplicación nativa.
- El contenedor de las aplicaciones es nativo (menú, botones, opciones, entre otras).
- Es útil para ventanas de mayor complejidad y que necesiten de mayor flexibilidad.
- Se puede utilizar HTML directo con una ventana nativa que contiene componentes propios de la plataforma.
- Se utiliza para ventanas que cambian su contenido muy frecuentemente.
- Se evita crear nuevas versiones, ya que solo al cambiar el contenido a nivel de servidor podremos visualizar la nueva funcionalidad.
- Todas las ventanas que necesitan funcionalidad nativa del teléfono como la cámara utilizan los componentes de forma nativa.
- Se debe planificar desde un principio qué componentes pueden crearse nativos y cuales web.
- Son la mejor opción para aplicaciones grandes que cambian a menudo.

### 3.5.5 Aplicaciones nativas

Las aplicaciones móviles nativas son aquellas que se encuentran instaladas en el sistema de archivos de cada dispositivo, y son distribuidas por las tiendas de aplicaciones, por ejemplo, el App Store de iOS o Play Store de Android.

Las principales características de las aplicaciones nativas se detallan a continuación:

- Tienen acceso y pueden aprovechar al máximo todos los recursos hardware y software del dispositivo. Recursos como cámara, GPS, acelerómetro, brújula o lista de contactos.
- Son desarrolladas en un lenguaje de programación, entorno de desarrollo y plataforma específica.
- Tienen un funcionamiento fluido, estable y con alto rendimiento.
- Son consideradas de costo elevado y con tiempos de desarrollo extensos.
- Ofrecen la posibilidad de alta experiencia de usuario.
- Son descargadas desde una tienda de aplicaciones como Play Store de Google o App Store de iOS
- Son creadas de manera individual para cada sistema operativo, por ejemplo, una para Android y otra para IOS.
- Son desarrolladas con lenguajes de programación propios de cada sistema operativo, como java/kotlin para Android u Objective-C/Swift para iOS.
- El usuario debe descargar actualizaciones para obtener la última versión.



- En cada nueva actualización se corrige errores o se añade mejoras.
- Pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación.

Las principales ventajas de las aplicaciones nativas se detallan a continuación:

- Los recursos hardware se pueden utilizar de forma directa, característica que solo las aplicaciones nativas pueden brindar.
- Son más fluidas y con mejor desempeño que las aplicaciones web.
- Pueden ser publicadas y puestas a la venta en tiendas de aplicaciones.
- Generalmente no necesitan una conexión internet, pueden trabajar sin conexión o modo offline.
- Los cambios son mínimos entre diferentes versiones de sistema operativo.
- Utilizan el SDK (kit de desarrollo) que es provisto por el fabricante para la visualización correcta de los componentes.
- A nivel de diseño, tienen una interfaz basada en los estándares del sistema operativo, por tal motivo tienen mayor coherencia y consistencia que el resto de las aplicaciones.
- Brindan mejoras de usabilidad al entregar interfaces familiares.

Las principales desventajas de las aplicaciones nativas se detallan a continuación:

- Solo pueden ser utilizadas por un dispositivo que cuente con el sistema operativo para el que fue desarrollada.
- Existe poca reutilización de elementos entre plataformas.
- Por lo general el costo en tiempo y dinero es más elevado.
- Tienen mayores costos de mantenimiento al tener varias versiones.
- La gestión de actualizaciones y aprobaciones por parte de las empresas que publican son tediosas y con un tiempo prudencial.
- Una aplicación debe desarrollarse para cada sistema operativo.

### **3.6 Elección entre nativo, web app o híbrido**

La elección del tipo de aplicación a desarrollar dependerá exclusivamente de los requisitos y particularidades del proyecto. Mientras en algunos proyectos puede favorecer la ubicuidad y la portabilidad de una aplicación web móvil, a otros les favorecen la optimización y la explotación de dispositivos que ofrece el enfoque híbrido o nativo. Cabe indicar que el tipo de aplicación a elegir tendrá impacto directo en la experiencia de usuario.

La tendencia apunta a que la mayoría de las aplicaciones tengan un cierto nivel de híbrido, debido a que este enfoque es la única forma de ofrecer beneficios tanto a la web móvil como a las aplicaciones nativas. Con un enfoque híbrido la velocidad de publicación en el mercado es más rápida, los costos de desarrollo y mantenimiento son más bajos, y

el público objetivo es más amplio. Cualquiera que sea el tipo de aplicación a elegir lo más importante siempre será la calidad del desarrollo y del producto final. (Redwood, 2016)

A continuación, las principales diferencias a tomar en cuenta para la elección del tipo aplicación a desarrollar en base a ciertas características móviles.

### **Características y API del Dispositivo**

Aunque las aplicaciones web tienen acceso limitado a ciertas características nativas, las aplicaciones nativas (y los componentes nativos de las aplicaciones híbridas) tienen acceso a todos los recursos hardware del dispositivo, como cámara, gestos, notificaciones push, widgets del panel, GPS y los contactos que se exponen a través del API del dispositivo para poder interactuar.

### **Modo Sin Conexión (Offline)**

Una aplicación nativa puede funcionar sin una conexión a internet, dándole una ventaja con respecto a los híbridos y web que tienen almacenamiento en caché del navegador con acceso limitado.

### **Visibilidad**

Las aplicaciones web presentan el contenido mucho más visible en un navegador que en una aplicación. Las personas al tener una necesidad de información ingresan a un motor de búsqueda, escriben una consulta y elijen una página de los resultados. No se

dirigen a una tienda de aplicaciones, para buscar una aplicación, descargarla e intentar encontrar su respuesta en la aplicación nativa instalada. La realidad es que la mayoría de los usuarios no mantienen demasiadas aplicaciones instaladas, debido a pueden perder espacio en su dispositivo, y prefieren tener aplicaciones que se usen con frecuencia.

### **Velocidad**

Las aplicaciones nativas son mucho más rápidas que las aplicaciones híbridas y web, ya que por definición se ejecutan a velocidad nativa. Su capacidad de respuesta es la clave de usabilidad de la mayoría de las aplicaciones. Las aplicaciones híbridas y web se ejecutan sobre capas adicionales que consumen recursos informáticos que disminuyen la velocidad de la aplicación. Un punto por tomar en cuenta, si se está desarrollando un juego intensivo de gráficos, la velocidad no debería ser una preocupación importante para las aplicaciones web o híbridas.

### **Mantenimiento**

Mantener una aplicación nativa puede resultar muy complicado tanto para los usuarios, como para desarrolladores que tienen que tratar con múltiples versiones de la misma información en diferentes plataformas, además los cambios deben ser empaquetados en una nueva versión y publicados en la tienda de aplicaciones. Por otra parte, el mantenimiento de una aplicación web y una aplicación híbrida es mucho más sencillo al tratarse de mantener una página web, haciéndose con tanta frecuencia como sea necesario.

### **Independencia de la plataforma**

Las aplicaciones web e híbridas tienen una ventaja en cuanto a las aplicaciones nativas. Los navegadores brindan la posibilidad de admitir diferentes versiones de HTML5, así como reutilizar parte del código para crear nuevas aplicaciones.

### **Restricciones de contenido, proceso de aprobación y tarifas**

Cumplir con las políticas de un tercero sobre su contenido y diseño puede ser tan engorroso e impactar en tiempo y dinero. Las aplicaciones nativas e híbridas deben pasar por los procesos de aprobación y las restricciones de contenido impuestas por las tiendas de aplicaciones, en cambio para las aplicaciones la web todo es gratuito.

### **Costo**

Las aplicaciones híbridas son más baratas a desarrollar, debido a sus características web más comunes, que requieren habilidades acumuladas en base a la experiencia previa en la web. Los desarrolladores web suelen tener salarios más bajos que el de sus colegas de iOS y Android considerados como talento más especializado. Las actualizaciones son una ventaja porque se realizan en muchos casos una vez para luego desplegarse en todas las plataformas. Todo esto beneficiando el balance de una compañía. Aunque las aplicaciones nativas son más costosas de desarrollar, en muchos casos son mejor revisadas en las tiendas de aplicaciones.

### **Despliegue en tiendas**

La velocidad y facilidad de despliegue en las tiendas es una consideración por tomar en cuenta, mientras las aplicaciones nativas tardan un tiempo en desarrollarse y requieren pruebas por cada plataforma, las aplicaciones híbridas tienen la ventaja ya que son más rápidas de construir, probar e implementar. El uso de un marco de interfaz de usuario de HTML5 como jQuery Mobile facilita el prototipado y los desarrollos rápidos de aplicaciones web e híbridas que acelera las cosas, en contraposición de las nativas que se desarrollan como aplicaciones independientes para cada plataforma.

## **Rendimiento**

Las aplicaciones híbridas tienen una enorme desventaja en cuanto a rendimiento, en gran medida debido a la naturaleza de su arquitectura, es por ello que sus velocidades pueden verse comprometidas con cuellos de botella en el DOM (Document Object Model). Sin embargo, se han mejorado con la inclusión de nuevas soluciones, con el uso de frameworks HTML5 / AngularJS y React Native que ofrecen calidad y el rendimiento. En cambio, las aplicaciones nativas que tienen acceso a los componentes del dispositivo se desarrollan de forma más fácil y efectiva a través del uso de tecnologías y lenguajes nativos (Objective C / Swift y Android)

## **Experiencia de Usuario (UX)**

Una de las principales prioridades es proporcionar una experiencia de usuario consistente con el sistema operativo y con la mayoría de las aplicaciones disponibles. La ventaja la toma las aplicaciones nativas, pero no significa que una aplicación web o una

aplicación híbrida no la pueda ofrecer, lo que cambian son los gráficos y las imágenes que los usuarios están acostumbrados a visualizar. La aplicación Flipboard es un excelente ejemplo de un producto con UX de calidad para la web móvil, que tiene un nivel de interacción y rendimiento que compite con aplicaciones nativas.

## **Seguridad**

Aunque las aplicaciones híbridas están creciendo rápidamente, las aplicaciones nativas se consideran más seguras desde la perspectiva de una empresa. La desventaja de las aplicaciones híbridas o web móvil es que son más vulnerables, ya que pueden ser atacadas fácilmente con técnicas como “Man In the Middle” o “Code Injection”. La ventaja de las aplicaciones nativas es que incorporan varias características de seguridad, como la cadena de arranque seguro de Apple o el aislamiento de procesos (SandBox) de Android para ofrecer seguridad a los datos de los clientes.

### **3.7 Plataformas y frameworks de desarrollo móvil**

El mercado cuenta con una importante cantidad de plataformas de desarrollo de aplicaciones, agrupadas según el tipo nativo e híbrido. En el siguiente apartado se describen las principales plataformas.

#### **3.7.1 Plataformas de desarrollo nativo**

Las aplicaciones móviles nativas son desarrolladas para ser instaladas en un sistema operativo específico y utilizando un kit de desarrollo de software (SDK).

En iOS (Apple), la plataforma de desarrollo para Iphone y Ipad es Xcode, que trabaja con los lenguajes de programación Objective-C y Swift. En Android (Google), se desarrolla con los IDEs de Eclipse e IntelliJ IDEA, utilizando los lenguajes de programación Java y Kotlin. Windows Phone usa los lenguajes C# y Visual Basic .NET, y mientras que BlackBerry usa el lenguaje C++. Existen plataformas de desarrollo con menor popularidad como Microsoft Xamarin, Adobe Flex y RubyMotion que se describen en el siguiente apartado.

#### **3.7.1.1 Microsoft Xamarin**

Es una plataforma que permite crear apps nativas para iOS, Android y Windows 10. Su característica principal es su versatilidad para desarrollar en multiplataforma, siendo una ventaja frente a Apple y Google. En los últimos años es considerada como la plataforma favorita de los desarrolladores.

Es una buena alternativa para programar iOS en Windows sin ninguna computadora Mac. Brinda un panel de información interactiva en tiempo real llamado Xamarin Insights que brinda datos de usuarios en línea. Tiene la capacidad para compartir código en múltiples plataformas centradas en el lenguaje C#. Es posible realizar automatización de pruebas de rendimiento y funcionamiento a través de la herramienta Xamarin Test Cloud. Trabaja de manera similar a Apache Cordova. La versión gratuita para Windows y Mac es Xamarin Studio Community, y la versión de pago es Visual Studio Professional.



### **3.7.1.2 Adobe Flex**

Es una gran solución para programar iOS en Windows. Trabaja con Adobe Air y desde los IDEs Eclipse o Flash Builder. Brinda la posibilidad de no tener Xcode instalado para compilar y firmar una aplicación en la tienda, pero resulta necesario tener un computador Mac para subir las apps en el App Store de iOS.

### **3.7.1.3 RubyMotion**

Es una herramienta multiplataforma para el desarrollo de aplicaciones nativas para iOS y Android a través del lenguaje de programación Ruby. Las aplicaciones se pueden crear desde la línea de comandos del terminal o desde cualquier editor. Permite escribir código en una consola interactiva, con cambios en tiempo real, integración de bibliotecas de terceros y un marco integrado automatizado de pruebas.

## **3.7.2 Plataformas de desarrollo híbrido**

Existe una gran variedad de frameworks móviles que ofrecen opciones para crear aplicaciones compatibles con Android e iOS, y que no requieren conocer sus lenguajes específicos. La mayoría permiten crear aplicaciones móviles haciendo uso de los lenguajes conocidos: HTML, CSS y JavaScript. En la siguiente sección se describen las plataformas populares: Adobe PhoneGap, Apache Cordova, Ionic, Appcelerator Titanium y React Native.

### **3.7.3 Adobe PhoneGap**

Es una poderosa herramienta multiplataforma para la creación de webapps o aplicaciones multiplataforma basadas en la tecnología web de JavaScript, CSS y HTML. Es la plataforma de distribución open source de Apache Cordova y de desarrollo de Adobe. Entre de su principal ventaja es que los desarrolladores no necesitan de conocimientos de los lenguajes nativos como iOS o Android.

Tiene una serie de APIs que permiten controlar características de los dispositivos como el acelerómetro, la cámara, las notificaciones o la geolocalización. Ofrece una experiencia de usuario completa de app. Ofrece un servicio en la nube llamado Adobe PhoneGap Build y Adobe Creative Cloud que permite construir rápidamente apps móviles y compilarlas con facilidad sin necesidad de SDKs, compiladores o hardware específico. Permite crear aplicaciones para: iPhone, Android, Windows Phone, BlackBerry, Blackberry 10, WebOS, Symbian y Bada.

### **3.7.4 Appcelerator Titanium**

Es un entorno de desarrollo no gratuito basado en Eclipse el framework de código abierto para el desarrollo de aplicaciones. No maneja archivos HTML ni CSS, como lo hace Apache Cordova y PhoneGap, a menos que se desee crear una aplicación que use interfaces de usuario nativas y basadas en HTML.

Tiene acceso a las APIs nativas que dan soporte al hardware, para crear aplicaciones nativas basándose en Javascript y compatibles con múltiples dispositivos y sistemas

operativos. Utiliza estándares abiertos y tecnología web. Posee un conjunto de herramientas móvil para emular en una plataforma real, en lugar del navegador.

Al ejecutar una aplicación en el dispositivo, no se envuelve en una vista web, sino es interpretada por un motor Javascript (JavaScriptCore en iOS o Rhino en Android). Es ampliamente compatible con sistemas de Bases de Datos como MySQL y MongoDB. Posee un sistema de estadísticas online de desarrollos realizados sobre la plataforma y permite la depuración de errores de funcionamiento a través de verificaciones automatizadas.

### **3.7.5 Ionic**

Ionic es un framework open source que permite crear aplicaciones multiplataforma híbridas utilizando HTML5, CSS (generado por SASS) y componentes Javascript muy ligados del lenguaje AngularJS.

Es un ecosistema pues posee una serie de herramientas que facilitan el proceso de desarrollo de aplicaciones híbridas. Utiliza la parte CSS del framework para crear increíbles diseños que parezcan nativos. Posee una interfaz de línea de comandos con emuladores integrados y una app packager basado en Apache Cordova.

### **3.7.6 Apache Cordova**

Es un framework propiedad de Adobe Systems que utiliza herramientas web genéricas para el desarrollo de aplicaciones móviles híbridas, como JavaScript, HTML5 y CSS3. Al

combinar todos los elementos web se crea un componente llamado webview que se envuelve dentro del Apache Córdoba.

### **3.7.7 React Native**

Es un framework que permite desarrollar aplicaciones multiplataforma. Su propósito es crear apps nativas, en lugar de crear aplicaciones híbridas que solamente se ejecuten en un navegador. El desarrollo se realiza a través de los lenguajes Javascript y React.

## **3.8 Consideraciones y limitaciones**

A continuación, se describen algunas limitaciones y consideraciones importantes a tomar en cuenta en el desarrollo de una aplicación.

### **Tipo de desarrollo**

El desarrollo de las aplicaciones para dispositivos móviles se puede realizar en forma web, híbrida, mixta o nativa, cada una con sus ventajas y desventajas. En el desarrollo multiplataforma se debe tomar en cuenta la creación y adaptación de la aplicación para varios sistemas operativos y dispositivos, además tienen la dificultad de no tener acceso a todos los recursos hardware y software del dispositivo, por lo tanto, tienen un menor rendimiento.

### **Tiendas de aplicaciones**

Las aplicaciones pueden publicarse en las llamadas tiendas cumpliendo con ciertos requisitos y políticas establecidas por cada plataforma como Play Store para Android,

App Store para iPhone o Microsoft Store para Windows. Se debe tomar en cuenta los tiempos de revisión que toman las tiendas para publicar, así como la gestión de cuentas de usuario para que las tiendas realicen las pruebas sobre las aplicaciones. Existen restricciones de Apple y Google en cuanto a los servicios que son consumidos por la aplicación, los servicios deben exponerse a través de certificados seguros emitidos por entidades certificadoras reconocidas, como por ejemplo Secure GlobalSign, Symantec o Trustwave, Geotrust. Es conveniente gestionar los requerimientos y restricciones de plataforma en etapas tempranas para que no existan retrasos en la salida a producción.

### **Actualización constante**

La evolución continua de la tecnología móvil provoca que las aplicaciones deban adaptarse y actualizarse para las nuevas tecnologías. Estas actualizaciones permiten asegurar el correcto funcionamiento. En cuanto al sistema operativo iOS existe una sola versión, pero en el caso de Android se debe instalar versiones betas como software base para empezar a desarrollar.

### **Tiempos de desarrollo**

Los proyectos de desarrollo de aplicaciones móviles se realizan en períodos cortos de tiempo, en comparación al desarrollo de software tradicional para computadoras de escritorio. El desarrollo nativo ocupa mayor tiempo y esfuerzo en comparación a las aplicaciones multiplataforma.

### **Experiencia de usuario**

El desarrollo de la interfaz de usuario requiere mucha más personalización en comparación a las computadoras de escritorio. La creación de prototipos de interfaz de usuario permite crear aplicaciones apegadas a las necesidades de los clientes, donde la experiencia de usuario juega un papel muy importante.

### **Portabilidad**

Existen una serie de limitaciones físicas relacionadas con el tamaño, forma de pantalla o teclado, así como la disposición y el limitado número de teclas.

### **Diseño**

La interrupción de tareas y la capacidad multitarea son limitaciones en los dispositivos móviles.

### **Movilidad**

Existen limitantes respecto a gestión de la información según la localización, siendo en muchas cosas variable e incompleta.

### **Canal**

Presentan variabilidad en el ancho de banda y en la disponibilidad debido a heterogeneidad de redes, así como desconexiones y riesgos de seguridad.

### **Usabilidad**

Grupos de usuarios exigen necesidades específicas, combinados con la diversidad de dispositivos y plataformas tecnológicas, elevan la complejidad al diseñar en entornos móviles.

### **Fragmentación de los sistemas operativos**

Existen en el mercado un sinnúmero de tecnologías, estándares, protocolos y sistemas operativos que añaden más complejidad al desarrollo móvil.

### **Capacidades limitadas de los terminales**

Los dispositivos de gama baja presentan capacidades bajas de almacenamiento, potencia gráfica baja, interfaces poco funcionales, riesgos en la integridad de datos, uso limitado de periféricos y limitada duración de baterías.

### **Time-to-market**

Los requisitos se imponen con tiempos estrictos para el lanzamiento, frente a una industria en movimiento.

### **Limitaciones de hardware**

Las limitaciones técnicas que presentan los dispositivos de gama baja obligan a enfocar esfuerzos en optimización y rendimiento de las aplicaciones.

### **Metodología de desarrollo**

Se necesitan nuevos enfoques o metodologías que se adapten a los cambios del entorno y requisitos del mercado.

### **Número elevado de aplicaciones**

El número elevado de aplicaciones para móviles disponibles en el mercado han ocasionado que los consumidores se sientan fatigados, y recurran a un conjunto básico de aplicaciones, dando como resultado que las empresas no puedan retener a los usuarios. Las empresas TI se preguntan cómo hacer para atraer y retener a los consumidores, debiendo transformar las capacidades de las aplicaciones para entregar valor o corren el riesgo de ser ignoradas.

### **3.9 Ecosistema móvil**

Un ecosistema móvil es el conjunto de actores que giran alrededor de las aplicaciones y dispositivos móviles, incluyen fabricantes de hardware, operadoras de telecomunicaciones y elementos software que permiten la ejecución una aplicación.

Un ecosistema está compuesto por el sistema operativo, navegadores web, métodos de entrada de información, usuarios, canales de distribución, tiendas de aplicaciones y las aplicaciones como tal.



### **3.10 Fragmentación**

La mejor manera que tiene un sistema operativo para crecer es a través de la variabilidad, también llamada fragmentación. La fragmentación es considerada como la causa y consecuencia de la expansión de un sistema.

La fragmentación es un conjunto de elementos y condiciones de una situación, que no permite compartir entre diferentes ecosistemas una misma aplicación. Para ello se requiere adaptar los ecosistemas para que las aplicaciones sean compatibles y puedan compartirse. La fragmentación trae consigo muchos riesgos que pueden conducir al fracaso o reducir la calidad del producto, entre ellos generar más errores, altos costos o alargar las fases de un proyecto. Para reducir el riesgo se debe implementar estrategias que permitan que una aplicación pueda ejecutarse en muchas versiones de la misma plataforma. La fragmentación se origina por diferentes factores que se describen a continuación.

#### **Hardware diferente**

Dispositivos con componentes de distintas características como tamaño de pantalla, teclado, sensores, procesador, etc.

#### **Software / Plataformas diferentes**

Múltiples sistemas operativos con sus versiones y frameworks.

#### **Variaciones de funcionalidades**

Aplicaciones con versiones limitadas y pagadas.

### **Preferencias de usuario**

Localización de la aplicación (idioma, etc.)

### **Diversidad del entorno**

Diferentes operadoras y sus API, problemas de cortafuegos, limitaciones en redes y roaming

### **3.11 Experiencia de usuario (UX)**

La experiencia de usuario (UX) es el proceso que lleva a cabo el usuario cuando interactúa con un producto o sistema, y cuyo resultado es la generación de un nivel de satisfacción. La percepción del producto no depende únicamente de factores de diseño, sino de aquellos relacionados con sentimientos, emociones, o confiabilidad del producto. El conjunto de factores y elementos permitirán obtener una percepción positiva o negativa de dicho producto, servicio, o dispositivo. Concluyendo si gusta o no a los usuarios. A continuación, se describen los principales factores que envuelven el desarrollo UX.

### **Accesibilidad**

Capacidad que tienen los usuarios para acceder a un sitio web, independientemente de las limitaciones, de modo que puedan entender, percibir, navegar e interactuar de manera exitosa.

## **Lean UX**

Es un proceso que maximiza la eficiencia del trabajo, centrado en mejorar la relación del equipo, elimina procesos innecesarios y mantiene un propósito claro. Se caracteriza por resultados de alta calidad sobre la experiencia del usuario, su forma eficiente de trabajo y la utilización correcta del tiempo. (Quer, 2016)

## **Agile UX**

Es el resultado de la combinación de metodologías de desarrollo de software ágil con UX, donde equipos de desarrollo y UX trabajan juntos.

## **Arquitectura de información**

La arquitectura permite que los usuarios puedan satisfacer sus necesidades de información a través de la organización, clasificación, estructuración y descripción de contenidos en un sitio web, con el menor esfuerzo posible.

## **Diseño de experiencia de usuario (UX)**

Es el proceso de mejora de grado de satisfacción que tiene un usuario al usar un producto, sitio o aplicación. Este concepto viene de la mano con una mejora constante de usabilidad y accesibilidad. El diseño UX tiene como finalidad generar placer entre la interacción entre cliente y producto, en pocas palabras que el usuario le agrade estar en el sitio o aplicación. Permite diseñar aplicaciones útiles, productos fáciles de usar, con ventajas competitivas para la empresa y valor agregado para el usuario.

## **Diseño de interfaz de usuario (UI)**

La interfaz de usuario nos permite tener el control de un sitio o aplicación. Este diseño se encuentra directamente relacionado con el Diseño Gráfico, es de tipo “look and feel”, y se centra en la presentación visual de un producto digital

## **Diseño de Interacción**

Se definen las acciones del usuario a realizar sobre la aplicación y cómo responderá sobre las mismas. Se puede decir que es la actividad y resultado de definir el comportamiento interactivo entre el usuario y el sitio web.

## **Mobile-First**

Es una técnica que permite centrarse en necesidades reales de los usuarios priorizando las tareas claves y diseño web en los dispositivos móviles. Creada debido a las limitaciones de las pantallas pequeñas, y para obtener sitios más fáciles de navegar y entender.

## **Diseño gráfico**

Es el proceso de diseñar el aspecto gráfico o visual de un sitio web. La expresividad del producto y de potenciar la identidad visual son parte del proceso de comunicación, que se consiguen gracias a un buen diseño gráfico

## **Test A/B**

A través de un objetivo definido, esta técnica permite añadir pequeños cambios a nivel de diseño para lograr identificar su efectividad en cada versión.

### **Target personas**

Es una técnica del diseño de interacción, que permite añadir características y objetivos importantes de las personas a los datos de usuario. Se realiza el estudio de los usuarios usando “personas”, que consisten en usuarios ficticios que representan objetivos, necesidades y patrones de conducta.

### **Wireframe**

Es una técnica que permite centrarse en el diseño de contenidos antes que, en el diseño visual. Se trata de una guía visual aproximada de la estructura de una página web, donde se define su distribución visual y contenido.

### **Usabilidad**

Es una técnica que nos permite evaluar la facilidad de uso de una aplicación. Define dos dimensiones, una objetiva que permite medir atributos de eficiencia, eficacia, y aprendizaje a través de la observación, y otra subjetiva basada en la satisfacción y percepción del usuario. Una vez identificados los factores que envuelven al desarrollo UX, es necesario complementarlos con ciertas consideraciones para el desarrollo de un proyecto de Experiencia de usuario.

### 3.12 Consideraciones en el diseño UX

El Diseño de Interfaz (UI) está relacionado al diseño gráfico y es menos técnico y analítico que el Diseño UX (User Experience). La posición intermedia entre todos los diseños es ocupada por el Diseño de Interacción (IxD), que relaciona aspectos visuales con funciones y operaciones. Los tres diseños se complementan para alcanzar una navegación fluida por parte del usuario sin la necesidad de pensar lógicamente en cada una de sus acciones

El diseño IU nos proporciona mejoras visuales con muy poca usabilidad, mientras que el diseño UX se enfoca en lo usable antes que, en lo estético. Lo ideal siempre será tener un equilibrio entre ambas disciplinas para cumplir con los objetivos del producto. Según, (Neoland, 2016), para el desarrollo de un proyecto de Experiencia de usuario, se debe seguir los pasos:

1. Crear los objetivos del proyecto.
2. Analizar el usuario.
3. Desarrollo gráfico de líneas visuales del producto.
4. Realizar el prototipado de las pantallas
5. Diseño de animación e interacción entre elementos visuales.
6. Enfocar el proyecto con todas las pantallas y sus dimensiones.
7. Desarrollo a nivel técnico de los elementos visuales.
8. Validar continuamente si el usuario se encuentra en el centro para validar las decisiones de los requerimientos de diseño

Para la ejecución exitosa de estos pasos se deben utilizar herramientas de diseño UX. Las principales herramientas se detallan a continuación.

### **Herramientas de diseño UX para el diseño web**

Permiten a los diseñadores crear prototipos y diseños web creativos. Ejemplo: Moqups, Wireframe, Pencil, Balsamiq, Power Mockup, UXPin, Solidify, Gliffy, POP

### **Herramientas de prueba A/B**

Permiten evaluar el rendimiento de una página web. Ejemplo: Desinion, Visual Website Optimizer, Optimizely

### **Herramientas de pruebas de usabilidad**

Permiten identificar las necesidades de los usuarios objetivos y su psicología. Se dividen en herramientas de pruebas de usabilidad remota, servicios DIY y las herramientas de contacto. Ejemplo: Loop11.

### **Pruebas de usabilidad**

Permiten analizar la página web desde la perspectiva de usuario, obteniendo una impresión directa de la gente que utilizando el producto. Ejemplo: Appsee, Attensee, UserVoice, MouseStats

### **Herramientas de colaboración**

Ayudan a los miembros del equipo a colaborar de manera efectiva y sin ningún esfuerzo. Ejemplo: Filesq, Notism, Red Pen, Memosort, Trello, Invision

### **3.13 Look and Feel**

El aspecto y comportamiento GUI se puede definir como el conjunto de propiedades y características que dan una identidad visual única y pueden ser percibidos de manera diferente por cada usuario. Es un concepto que se relaciona con la apariencia de una aplicación, y en cómo se presentan los diferentes componentes como botones, menús, campos de texto, combos o pestañas. El aspecto de estas interfaces se caracteriza principalmente por parámetros fundamentales como el tipo de letra, forma, color y disposición de los elementos. Lo que se puede “percibir” y “sentir” sobre estas interfaces está muy influido por la interacción con cada una de ellas.

### **3.14 Design Thinking**

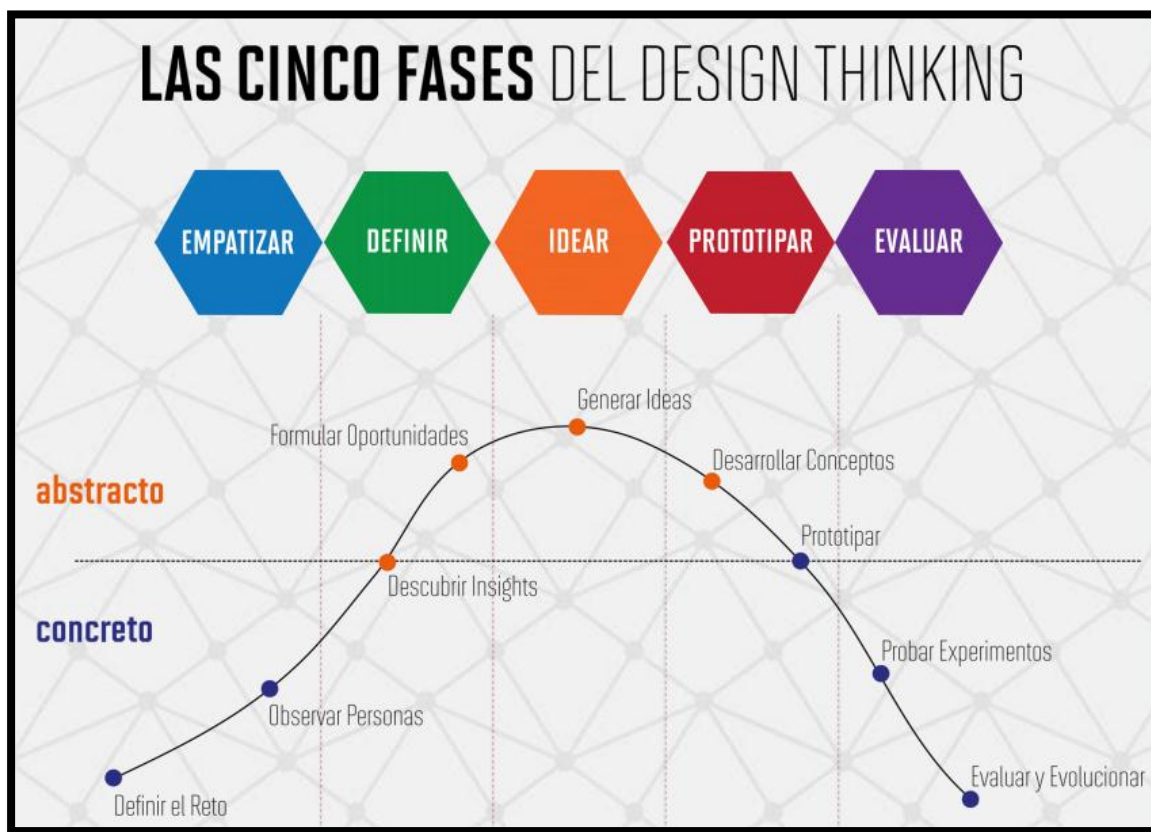
Es una metodología enfocada en fomentar la innovación de las organizaciones de una forma eficaz y exitosa. Se centra en el proceso de diseño, dejando en un segundo plano el producto final. No es diseñar, más bien es pensar como lo hace un diseñador. Integra enfoques de diferentes campos mediante la participación de equipos multidisciplinarios. (Institute, 2013)

Los equipos multidisciplinarios deben pretender observar al usuario, para desarrollar empatía. Deben adquirir conocimientos básicos sobre los usuarios del producto, y sobre el problema que afrontan. También deben definir el punto de vista a partir del cual se



debe desarrollar el diseño, generando para ello un usuario tipo, para el cual se diseña el producto. Finalmente deben construir prototipos a partir de las ideas más prometedoras, para que prueben el producto y obtengan retroalimentación en base a dicha interacción.

La metodología presenta cinco fases (Pizarro, 2017), que se pueden observar en la Figura 14.



**Figura 14.** Las cinco fases del Design Thinking

Fuente: (Kobayashi, 2015)

1. **Empatizar:** Esta etapa se encarga de entender a los usuarios, investigando las necesidades y hábitos del público objetivo para definir con claridad el problema de fondo. También debemos estudiar su entorno para identificar los problemas.
2. **Definir:** En esta etapa se analiza la información una vez recolectada, para filtrar todos los datos importantes que permitan crear conclusiones y especificar los problemas.
3. **Idear:** En esta etapa el equipo multidisciplinario debe idear soluciones desde su visión particular luego de haber definido el problema. Una técnica muy utilizada es la lluvia de ideas que permite trabajar con representaciones visuales, explorando nuevas alternativas para analizar y seleccionar la de mejor enfoque. Finalmente, se obtiene conclusiones y posibles soluciones.
4. **Prototipar:** En esta etapa se construye prototipos en base a ideas seleccionadas, que permitirán refinar detalles.
5. **Evaluar:** Esta etapa permite que a través de los prototipos se realice pruebas con usuarios finales, para verificar la eficiencia y las dificultades que experimentan las personas al interactuar con la solución.

### 3.15 Mobile First

Es una filosofía de experiencia de usuario para las aplicaciones de tipo web, una manera de encarar el trabajo y una forma de facilitar la labor durante el diseño responsive, comenzando siempre por los dispositivos con pantallas más pequeñas para ir adaptándolas a las más grandes.

Mobile First es en realidad un concepto bastante simple, consiste en diseñar primero para móviles, debemos comenzar siempre centrándonos en los dispositivos con pantallas más pequeñas y generalmente con menor ancho de banda disponible para la navegación.

Cuando se diseña un sitio web responsive se debe tener en cuenta una enorme cantidad de contextos donde el contenido va a ser consumido. El desarrollo se centra únicamente en los elementos y las acciones más importantes de un sitio web, creando una experiencia y una buena usabilidad para el usuario. Usando diferentes técnicas tales como detección de dispositivos, media queries, condicionales, servicios de diferentes imágenes en función del dispositivo realizamos una mejora progresiva del sitio web para poder adaptar la experiencia.

### **3.16 Prototipos**

Un prototipo consiste en transformar las ideas y conceptos recopilados de los usuarios en elementos tangibles que muestran una aproximación del producto final. Intervienen en el proceso de creación de modelos de un determinado producto y las sucesivas pruebas que se realizan sobre él.

El prototipo suele ser una forma rápida y barata de poner en escena conceptos que serán abordados con los usuarios e interesados. Permiten poder evaluar y mejorar la solución final antes de su implementación. Este proceso debe ir progresando conforme el proyecto va avanzando, hasta que el prototipo presente características funcionales,

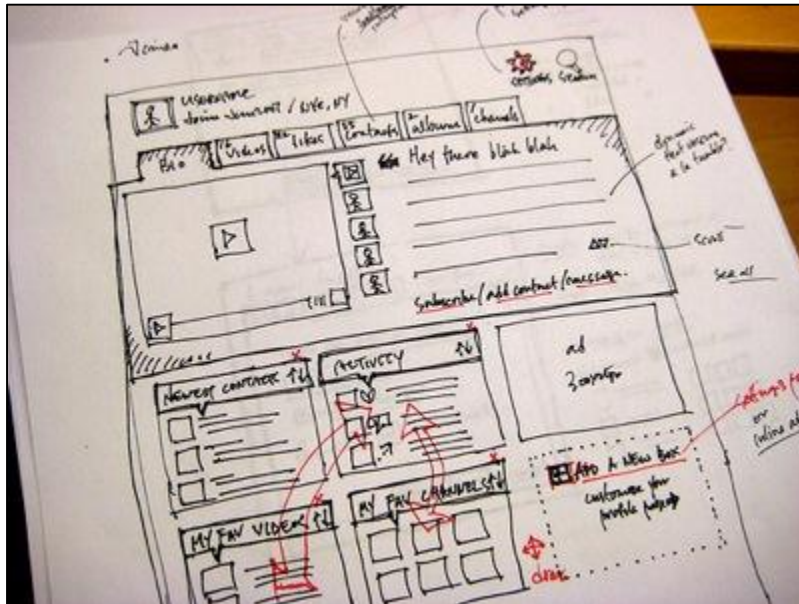
formales o de uso. Existen diferentes tipos y etapas de prototipado que permiten aproximarse cada vez más al producto final.

La incertidumbre en el diseño de aplicaciones móviles está llevando a la creación una nueva gama de herramientas para modelar prototipos. Estas herramientas proporcionan un espacio notable para el diseño, la colaboración, la evaluación y, por lo general, la creación de código. El diseño y el desarrollo se unen en un proceso continuo que conduce a la integración y despliegue continuo.

En el diseño de productos y/o aplicaciones se distinguen cuatro conceptos sketch, mockup, wireframe y prototipo.

### **Sketch**

Es un dibujo rápido que no tiene muchos detalles y que representa una idea, concepto, o generalidad de un proyecto. Es el primer boceto estático que se realiza en baja calidad de un proyecto digital, usualmente a mano y cuyo entregable es una imagen, o un papel. No tiene un trabajo conceptual muy extenso. Aquí prima la creatividad, la experiencia y el deseo del diseñador. En la Figura 15 podemos observar un ejemplo de sketch.



**Figura 15.** Sketch

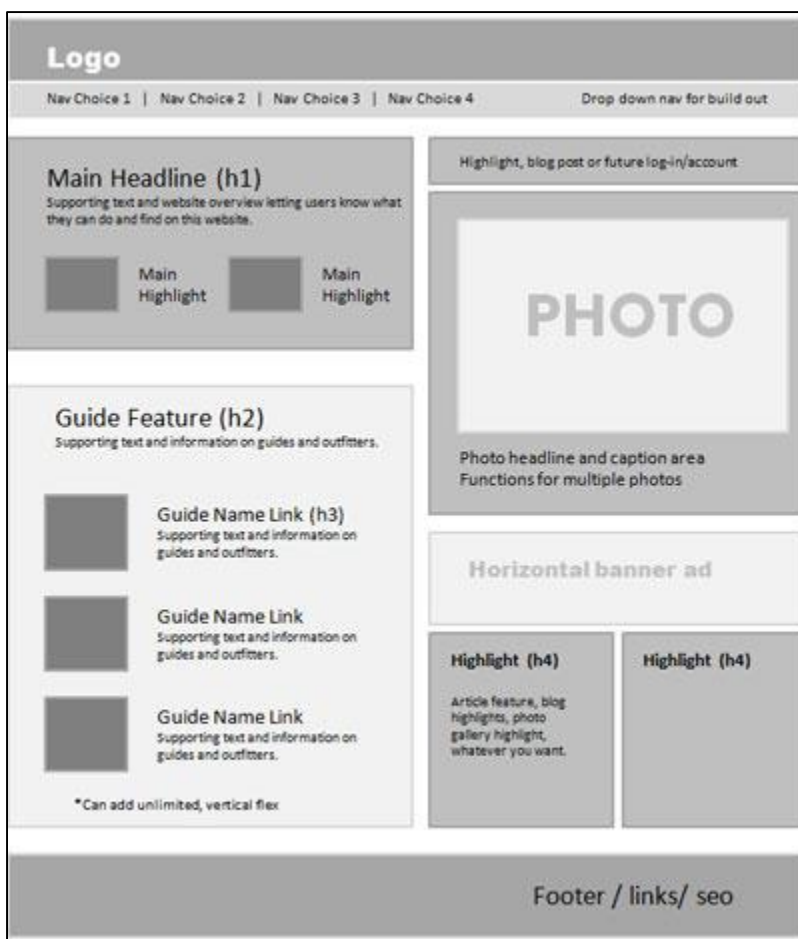
Fuente: (Otherwise, 2012)

## Wireframe

Es el siguiente paso en el diseño de un proyecto digital. Es una representación estática en escala de grises de baja calidad de un diseño, cuyo entregable es una imagen. En esta representación básica se define como estará estructurada la información, zonas de contenido, interacción de la interfaz, visualización y descripción básica del usuario. Es considerada como la parte más importante del diseño ya que contiene representaciones del producto final.

Un wireframe es un boceto que representa la estructura de una página o aplicación web. Se incluye la creación de planos y los dibujos de las pantallas resultantes de la conceptualización sin posibilidad de interacción. No se debe utilizar colores, tipografías ni cualquier elemento gráfico. Lo importante es centrarse en la funcionalidad del sitio y la

experiencia del usuario. Al ser representaciones tan simples, permitirán crear múltiples versiones de un mismo proyecto de una forma rápida, así como aplicar cambios o nuevas ideas que aparezcan a posterior. En la Figura 16 podemos observar un ejemplo de wireframe.



**Figura 16.** Wireframe

Fuente: (*Otherwise, 2012*)

## Mockup

Es una representación más avanzada y estática en calidad media-alta de un diseño gráfico y comunicativo del proyecto. Su entregable es una imagen. La parte visual real

del proyecto podrá ser revisada en esta representación. Las funcionalidades básicas, la visualización del contenido y la estructura de la información se muestran de una forma estática. En la Figura 17 podemos observar un ejemplo de mockup.



**Figura 17.** Mockup

Fuente: (Otherwise, 2012)

## Prototipo

Es una representación del producto final en alto detalle y navegable, cuya calidad puede ser mediana o alta. El entregable es formato navegable, PPT, HTML, o una animación. Permite simular la interacción de usuario con el contenido y la interfaz del producto como si se tratara del producto final. Los prototipos son una serie de wireframes

enlazados entre sí, plasmados en forma digital y con una interacción definida. En la Figura 18 podemos observar un ejemplo de prototipo:



**Figura 18.** Prototipo

Fuente: (Otherwise, 2012)

### 3.17 Seguridad

Las aplicaciones móviles traen grandes beneficios a nuestras vidas, pero pueden introducir grandes riesgos, si no tomamos medidas para mejorar la seguridad.

Las aplicaciones nativas se consideran más seguras que las aplicaciones híbridas, ya que son capaces de aprovechar las características de seguridad específicas de la plataforma. Para minimizar ataques existen varias técnicas, herramientas y bibliotecas



de terceros específicos de cada plataforma. Por lo contrario, las aplicaciones híbridas al depender de las vistas web son más propensas a ataques de inyección cuando se utilizan ciertas API.

Se puede obtener acceso no autorizado a dispositivos móviles cuando los atacantes roban o acceden a un dispositivo perdido. También existen aplicaciones malintencionadas que permiten obtener datos del propio teléfono, o de otras aplicaciones. Un atacante puede realizar ingeniería inversa sobre una aplicación para acceder a sus recursos o al código fuente. En el siguiente apartado se describen las principales vulnerabilidades identificadas tanto para las aplicaciones nativas como híbridas.

### **Seguridad en Aplicaciones Nativas**

Las vulnerabilidades son casi las mismas en cualquier plataforma, con diferencias en las herramientas y en las técnicas de explotación.

En Android, las aplicaciones nativas almacenan datos sensibles de forma local en el teléfono, esto puede ser aprovechado por los atacantes que podrían explotar poniendo un programa "Shell" en el dispositivo o simplemente haciendo una copia de seguridad de la aplicación. (Curran, 2016)

En iOS, si el teléfono está "jailbreak", los hackers son capaces de operar a través de un método similar al de Android. Los ataques comunes se originan principalmente por un

almacenamiento de datos local inseguro, una implementación de “SSL” débil, fugas de datos no deseadas o por una inyección de código.

### **Seguridad en Aplicaciones Híbridas**

Al igual que las nativas no están exentas de vulnerabilidades, especialmente si su código está mal escrito. Al hacer uso extensivo de las vistas web, pueden dejar la aplicación abierta a “exploits” potenciales si el código no se encuentra bien protegido.

Cualquier ataque específico de JavaScript, HTML, ingeniería inversa, ataques de “man-in-the-middle”, son considerados como una amenaza. Las vulnerabilidades relacionadas con el “backend API” también son muy comunes. Una forma de minimizar las amenazas es trabajar con marcos seguros que brinden controles de seguridad. Los ataques más comunes se originan por inyección de JavaScript (debido al uso de JS y HTML), implementación SSL débil (igual que nativa) y por problemas de caché.

En esta primera sección se abordó conceptos referentes a movilidad, en la siguiente se describirá conceptos relacionados al proceso de desarrollo de aplicaciones, iniciando con el análisis del panorama actual de las metodologías de desarrollo en el mercado.

#### **3.17.1 Panorama actual de las Metodologías de desarrollo**

Para conocer el estado actual de las metodologías de desarrollo de aplicaciones móviles debemos empezar realizando una búsqueda de información relevante, a partir de la selección de palabras clave.

La selección de palabras clave se enfoca en las metodologías y métodos de desarrollo de aplicaciones móviles, utilizando como principales repositorios de búsqueda a: Google Académico, ProQuest, IEEE, Repositorio de la Universidad de Fuerzas Armadas ESPE y otras publicaciones relevantes.

Las palabras clave seleccionadas para la búsqueda de información son: " metodologías desarrollo + aplicaciones móviles", "metodologías ágiles + aplicaciones móviles", "methodologies development + mobile applications" y "agile methodology + mobile application".

En base a los criterios establecidos, se seleccionaron 20 estudios de diversos autores que se presentan en el Anexo 1. Los estudios seleccionados se agrupan a partir de los siguientes parámetros:

- **Tipo**, de estudio
- **Título**, del estudio
- **Autores**, que desarrollaron el estudio
- **Año**, de publicación del estudio
- **País**, donde se desarrolló el estudio
- **N.º de metodologías**, que están incluidas en el estudio
- **Metodologías incluidas**, en el estudio
- **Descripción**, del estudio

De los estudios seleccionados, se identifican las metodologías con más coincidencias de uso a nivel teórico, ya que nivel práctico no se encontró información relevante de la aplicación de metodologías en el mercado. En la Tabla 4 se detalla la frecuencia de aparición de las metodologías de los estudios seleccionados.

**Tabla 4**  
*Metodologías más usadas en estudios seleccionados*

Metodología	Frecuencia
XP	8
Mobile-D	9
Cascada	2
Rápido RAD	2
Scrum	8
MASAM	5
HYBRID HDM HME	8
Spiral	2
Rapid7	1
Sless	4
Metodologia Y	1
MASEF	1
MDAM	1
DSDM	3
Lean	2
Kanban	1
NPD	3
ICONIX	1
MDAEM	1
M-BUSINESS	1
TDD	2
ASD	3
AUP	1
RUP	2
AGILE	4
PACIE	1
FDD	1
CRYSTAL	1
MDE	1

Mobile-D, Scrum, Hybrid y XP son las metodologías con más apariciones en los estudios seleccionados, por tal motivo se describirá más adelante sus principales características, así como ventajas y desventajas.

### **3.18 Metodología para el desarrollo de aplicaciones para dispositivos móviles**

La ingeniería de software móvil es una disciplina que brinda metodologías y técnicas para desarrollar un proyecto. Establece un conjunto de etapas básicas tales como análisis de requerimientos, especificación, arquitectura, codificación, pruebas, documentación y mantenimiento. (Sarango, 2017)

También presenta ciertas particularidades que lo diferencian del software convencional, tales como la simplicidad de ciertas aplicaciones, constante innovación debido a la amplia competencia del sector, corta duración de los desarrollos, o cambios continuos en el hardware.

Para que una metodología logre ser aplicable a los proyectos de desarrollo de aplicaciones móviles, debe tener una serie de características básicas que se detallan a continuación.

### **3.19 Características de una metodología para el desarrollo de aplicaciones móviles**

Diversas investigaciones y autores concuerdan un conjunto de características básicas que debe tener una metodología para el desarrollo de aplicaciones. (Villagrán, 2013).

Estas características se describen a continuación:

#### **Agilidad**

Los métodos ágiles mejoran la flexibilidad y la productividad de desarrollo, proporcionando los medios para adaptarse a los cambios en requerimientos y entornos, sumado al aprendizaje de experiencias de desarrollo.

#### **Conciencia del mercado**

El mercado actual para dispositivos móviles está dirigido a productos de software o aplicaciones, por tal motivo el proceso de desarrollo debe estar orientado en el modelo de negocio y al mercado potencial.

#### **Soporte para la línea de productos de software**

Las empresas desarrolladoras de software se enfocan en desarrollar una familia de productos software móvil, debido a la rapidez de los avances tecnológicos móvil, donde el ciclo de vida de las aplicaciones móviles es cada vez más corto y cuyo beneficio es reducir los costos de desarrollo.

#### **Desarrollo basado en arquitectura**

El proceso de desarrollo debe estar centrado en una plataforma común, con una arquitectura general y reconfigurable.

### **Soporte para reusabilidad**

La reutilización de componentes funcionalmente similares permite reducir costos y tiempos de desarrollo.

### **Inclusión de sesiones de revisión y aprendizaje**

El éxito del producto está en la habilidad de abstraer los conocimientos obtenidos durante las sesiones de revisión durante todo el proceso y posterior a la liberación.

### **Temprana especificación de la arquitectura física**

La definición de la arquitectura de los dispositivos móviles debe ser considerada en etapas tempranas del diseño, debido a que el gran número de riesgos técnicos proviene de las limitaciones en los dispositivos móviles y sus diferencias en la implementación de funciones básicas.

### **Valoración de la experiencia de usuario**

El diseño centrado en el usuario y usabilidad se consideran prioritarios, ya que permiten una interacción adecuada entre el dispositivo móvil y el usuario.

### **Soporte para el mantenimiento de aplicaciones móviles**

El desarrollo de aplicaciones móviles debe incorporar procesos que permitan afrontar los cambios tecnológicos, manteniéndolas actualizadas para que tengan un tiempo de vida útil en el mercado.

### **Integración de un equipo multidisciplinario**

Un equipo multidisciplinario de desarrolladores, diseñadores UX, administradores de proyecto y arquitectos de software trabajan en conjunto en cada fase del ciclo de desarrollo.

Una vez definido el conjunto de características básicas que debe tener una metodología para el desarrollo de aplicaciones, el siguiente paso es conocer las metodologías de desarrollo disponibles en el mercado. En la siguiente sección se realiza un análisis los principales enfoques tradicionales y ágiles.

#### **3.19.1 Métodos tradicionales**

Los métodos tradicionales nacen en contraposición a lo que representaba un desarrollo ad hoc (solución específica), a principios de la década de los 60. Se enfocan en los controles del proceso, donde se establecen actividades rigurosas, los artefactos a producir, y las herramientas y notaciones que se utilizaran.

Con el paso del tiempo demostraron ser efectivas y necesarias en un gran número de proyectos, pero también presentaron problemas en otros. Para corregir sus deficiencias



se podrían incluir en el proceso de desarrollo más actividades, más artefactos o más restricciones.

Sin embargo, el resultado final sería un proceso de desarrollo más complejo que podría limitar la habilidad del equipo para llevar a cabo el proyecto. A continuación, se describe el ciclo de desarrollo genérico de un proyecto de software.

### **3.19.1.1 Ciclo de vida**

Es el conjunto de etapas planificadas y controladas por donde pasa un producto software mientras dura su proceso de producción. Las distintas etapas del ciclo de vida se ejecutan secuencialmente, generando una documentación que servirá para las siguientes etapas. A este proceso se lo conoce como modelo tradicional o cascada, que es la base fundamental de la Ingeniería del Software.

Todos los demás modelos de ciclo de vida existentes están basados de una forma u otra en el modelo tradicional. Dependiendo de la complejidad del producto, las etapas del ciclo de vida pueden repetirse para conseguir mejores resultados. En la actualidad resulta complicado seguir un modelo secuencial, es por ello que para adaptarse al entorno los proyectos implementan más de un modelo.

El proceso de desarrollo de software o SDLC (ciclo de vida del desarrollo de software), es considerado una subcategoría del ciclo de vida de desarrollo de sistemas. Los modelos se encuentran estandarizados en base a la norma ISO/IEC 12207, que provee un número

de actividades para el desarrollo y mantenimiento de software. En la Figura 19 podemos visualizar el ciclo de desarrollo genérico de software.



**Figura 19.** Ciclo de Desarrollo Genérico

Fuente: (Fred, 2014)

En el siguiente apartado se describen las fases del proceso básico del ciclo de vida de desarrollo de software.

### **Fases de planificación y análisis**

La fase de planificación incluye el análisis de los requerimientos del cliente. El cliente puede tener una idea general del tipo de producto que va a necesitar, luego a través de demostraciones prácticas de funcionamiento de la aplicación se va realizando ajustes de esta idea. En esta fase también se fija el alcance del proyecto y se genera documentación de manera clara y concisa.

### **Fases de implementación, pruebas y documentación del código**

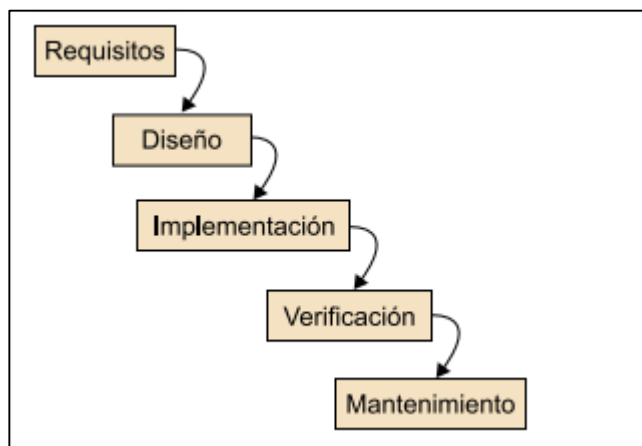
La fase de implementación consiste en codificar el producto por parte de los programadores. La fase de pruebas del software son una parte fundamental, porque permiten detectar y corregir errores en fases tempranas. La fase de documentación del diseño del software se crea durante toda la etapa de desarrollo para mantener y mejorar la aplicación en el futuro.

### **Fases de despliegue y mantenimiento del software**

La fase de despliegue permite instalar, personalizar, testear e iniciar un periodo para evaluar el rendimiento. Las fases de mantenimiento, soporte y mejora de productos ocurren cuando el software ha sido expuesto a pruebas y validado de manera exhaustiva para estar disponible en el mercado. Además, permiten corregir defectos que han ido surgiendo a partir de los requisitos definidos del producto.

#### **3.19.1.2 Modelo Waterfall (en cascada)**

Es un modelo tradicional, aplicable cuando los requisitos están completamente cerrados y no van a cambiar. Tiene énfasis en la planificación, los tiempos, las fechas límite y el presupuesto fijo. Las fases se realizan secuencialmente y sin retroalimentación. En la Figura 20 se muestra un ejemplo de fases del modelo waterfall.



**Figura 20.** Ejemplo de fases del modelo waterfall

Fuente: (Vique, 2012)

Este modelo es aconsejable en proyectos para móviles muy controlados y poco previsibles, donde no existe la incertidumbre ni influencia en los cambios de la industria. (Rodríguez, 2011)

### 3.19.2 Métodos Ágiles

Los métodos ágiles son aquellos que permiten adaptar la forma de trabajo al contexto y naturaleza de un proyecto, basándose en la flexibilidad y la rapidez, y teniendo en cuenta las exigencias del mercado y los clientes. Los métodos ágiles nacen en contraposición a lo que representaban los métodos tradicionales.

Todos los sistemas de trabajo y procesos ágiles se basan en los cuatro valores del Manifiesto ágil:

**Individuos e Interacciones sobre procesos y herramientas.** El principal factor para lograr el éxito son los recursos humanos, por tal motivo, tienen un gran valor en el desarrollo del proyecto. Un equipo capacitado y motivado, es garantía de una mayor productividad.

**Software funcionando sobre documentación extensiva.** Es importante documentar, sin embargo, se debe realizar esta actividad siempre y cuando sea necesario.

**Colaboración con el cliente sobre negociación contractual.** Es importante fomentar la participación del cliente, ya que sus aportes minimizaran los riesgos del desarrollo del proyecto.

**Respuesta ante el cambio sobre seguir un plan.** El equipo de desarrollo debe responder al cambio, para generar valor con el cliente. Durante las revisiones y retrospectivas los planes pueden cambiar en función de los comentarios del cliente.

Según (Ganesh, Hamsini, & Smitha, 2016), al estar usando procesos ágiles para la gestión de proyectos, se perciben ciertos beneficios. El desarrollo ágil se enfoca en el diseño simple, tiene la capacidad de lanzar un producto en un corto período de tiempo, se adapta a los cambios para utilizarlos como ventaja, posee una amplia cooperación en equipo, fomenta la programación en parejas y pone énfasis en realizar pruebas durante el desarrollo.

Las prácticas ágiles simplifican el desarrollo de aplicaciones móviles de modo que resultan ser adaptables incluso posterior a su lanzamiento. En la Tabla 5 se describen las principales metodologías ágiles para el desarrollo de proyectos.

**Tabla 5**  
*Métodos ágiles*

MÉTODO	ACRÓNIMO	AUTOR (ES)	AÑO
<b>Scrum</b>	Scrum	Sutherland	1994
<b>Microsoft solutions framework</b>	MSF	Microsoft	1994
<b>Rapid development</b>	RAD	McConnell	1996
<b>Método de Desarrollo de Sistemas Dinámicos - Dynamic solutions delivery model</b>	DSDM	Stapleton	1997
<b>Crystal methods</b>	CM	Cockburn	1998
<b>Agile RUP</b>	dX (AUP)	Booch, Martin, Newkirk	1998
<b>eXtreme Programming</b>	XP	Beck	1999
<b>Adaptive software development</b>	ASD	Highsmith	2000
<b>Desarrollo Dirigido por Características- Feature-driven development</b>	FDD	Charette, Mary y Tom Poppendieck	2001
<b>Agile modeling</b>	AM	Ambler	2002
<b>Lean Software Development</b>	LSD	Mary y Tom Poppendieck	2003
<b>Kanban</b>	KANBAN	David J. Anderson	2010

### 3.19.3 Ventajas

Los principios y valores ágiles proporcionan beneficios tanto para los clientes como a los equipos de trabajo. Las prácticas ágiles ayudan a resolver inconvenientes comunes

del proyecto, tales como el desfasamiento del alcance, el control del presupuesto y la planificación de horarios. Las principales ventajas del desarrollo de software ágil se detallan a continuación:

**Entrega Iterativa, Incremental y Rápida.** La entrega de pequeñas versiones funcionales permite obtener comentarios frecuentes de los clientes que ayudan a controlar el riesgo potencial. Los planes de proyecto, requisitos, diseño, codificación y pruebas se producen al principio y se reorganizan de forma incremental para adaptarse a los cambios del proyecto.

**Alto Rendimiento.** Las reuniones diarias del equipo de desarrollo permiten intercambiar información, mejorar la comunicación, compartir conocimiento y asegurar un progreso continuo.

**Diseño Flexible.** Capacidad para cambiar de dirección rápidamente, manejando los cambios con facilidad frente a los requisitos inciertos.

**Adaptable al entorno cambiante.** Se puede implementar nuevos cambios por parte del usuario en cualquier etapa.

**Reduce los riesgos de desarrollo.** El riesgo de desarrollo se reduce a través de comentarios tempranos realizados por el cliente, ya que permiten a los desarrolladores

estar al tanto de cualquier problema que pudiera surgir en las últimas etapas del desarrollo.

**Software funcional.** Entregas frecuentes de pequeñas piezas de software funcional en base a las necesidades del usuario.

**Satisfacción del cliente.** Se garantiza la satisfacción del cliente con la entrega de un producto de alta calidad.

**Calidad alta.** La ejecución de pruebas continuas para obtener comentarios del cliente, permiten identificar errores que conducen a la reutilización y mejora de la calidad del código.

**Poca documentación.** Se documenta solo lo necesario. La lista de características del producto, la duración de cada iteración, el desarrollo del producto y la duración de entrega.

**Detección de fallos.** Al realizarse las pruebas en cada iteración, los errores y defectos se identifican y se corrigen inmediatamente para que no constituyan un riesgo potencial.

**Buenas Prácticas.** Las fallas tempranas en el proyecto ahorran tiempo y dinero.



### 3.19.4 Desventajas

Los métodos ágiles tienen ciertas desventajas que se detallan a continuación:

**No funcionan en proyectos grandes.** Son considerados más adecuados para proyectos pequeños con equipos reducidos, debido a que las iteraciones tienen un alto costo al construir características específicas de un producto

**Interacción con el cliente.** Se necesita un gran compromiso del cliente para participar durante todo el ciclo de vida del proyecto.

**Requisitos insuficientes y pocos claros.** En la etapa inicial los requisitos del usuario son insuficientes y poco claros, pero se aclaran y especifican durante la fase de desarrollo. Si el cliente no está claro acerca de los requisitos, las características y el resultado, el proyecto puede salirse fácilmente de la realidad.

**Requisitos cambiantes.** Al cambiar los requisitos a lo largo del ciclo de vida del desarrollo del producto se crea el riesgo de tener un proyecto sin fin con menos previsibilidad. Sin una visión suficiente y clara, se hace difícil controlar el costo y alcance del proyecto.

**Entrega Frecuente.** Las entregas frecuentes, seguidas de las pruebas de aceptación del usuario consumen mucho tiempo y pueden ser mentalmente agotadoras, por lo tanto, es imprescindible determinar un ritmo sostenible para el equipo.

**Falta de documentación.** Debido a los ajustados plazos de los proyectos, no es fácil mantener la documentación detallada de los cambios de diseño solicitados por los usuarios. La documentación limitada provoca dificultad para los nuevos desarrolladores.

**Cultura y Equipos Dispersos.** Utilizar herramientas de videoconferencia para unir equipos remotos impacta en la cultura de la organización.

**Recursos experimentados.** Los programadores con mayor experiencia pueden tomar decisiones durante el proceso de desarrollo frente a los desarrolladores principiantes.

**Pensamiento tradicional del desarrollo.** Las organizaciones que usan métodos tradicionales les cuesta mucho adoptar métodos ágiles.

**Falta de familiaridad con ágil.** Al iniciar en el mundo ágil, se realizan muchas suposiciones que pueden cambiar el rumbo del proyecto.

Una vez revisadas las características, ventajas y desventajas del modelo ágil, el siguiente paso es describir las metodologías ágiles más usadas en el mercado: Extreme programming (XP), Kanban, Lean y Scrum

### 3.2.1 XP

La Programación Extrema (XP) es una disciplina ligera de desarrollo de software que se centra en las prácticas de ingeniería. Posee un conjunto de prácticas para mejorar la calidad del software y responder rápidamente a los requerimientos cambiantes del cliente. Es una metodología centrada en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y fomentando un buen clima de trabajo.

El método XP propone un ciclo de desarrollo de seis fases, que se muestra a continuación:

1. Exploración
2. Planificación de la entrega (release)
3. Iteraciones
4. Producción
5. Mantenimiento
6. Muerte del proyecto.

El conjunto de prácticas que propone el método XP según (José H. Canós, 2012), es el siguiente:

**El juego de la planificación.** Permite una comunicación fluida entre desarrolladores y cliente, a la hora de estimar el esfuerzo requerido para el desarrollo de las historias de usuario. El cliente también puede negociar el tiempo de entregas y ámbito de iteración.

**Entregas pequeñas.** Se generan pequeñas versiones operativas del sistema en menos de 3 meses, no necesariamente con toda la funcionalidad requerida. Estas versiones se consideran como resultados de valor para el sistema.

**Metáfora.** El desarrollo del sistema es afrontado por un conjunto de metáforas creadas a partir del equipo de desarrollo y el cliente. Las metáforas son historias que detallan como debería funcionar el sistema

**Diseño simple.** Se usa la simplicidad al momento de diseñar la solución para que sea implementada en el momento más adecuado del proyecto.

**Pruebas.** El desarrollo del código está coordinado con las pruebas unitarias determinadas por el cliente y ejecutadas durante cada modificación del sistema.

**Refactorización.** Es un procedimiento que mejora la estructura interna del código sin necesidad de alterar el comportamiento externo. Su propósito es mejorar la legibilidad, duplicidad, simplicidad y flexibilidad del código.

**Programación en parejas.** La producción de código en parejas tiene como ventaja una menor tasa de errores, mejor diseño y mayor satisfacción de los desarrolladores.

**Propiedad colectiva del código.** El código puede ser cambiado por cualquier desarrollador en cualquier momento.

**Integración continua.** El sistema se forma a través de la integración y construcción de varias piezas de código.

**40 horas por semana.** No se trabaja horas extras durante dos semanas seguidas porque desmotiva al equipo, el trabajo es de 40 horas máximo por semana.

**Cliente in-situ.** El cliente debe estar disponible y presente para el equipo durante todo el proyecto. La comunicación permite aportar mayor valor de negocio, y además los desarrolladores resuelven dudas de manera inmediata.

**Estándares de programación.** La comunicación entre desarrolladores es por medio del código, deben seguir ciertos estándares de programación para mantener la legibilidad del código.

### 3.2.2 Kanban

Es un sistema visual de gestión de procesos que indica, “qué, cuándo y cuánto” producir. Es una técnica para gestionar la creación de productos, a través de la entrega

continua sin sobrecargar al equipo de desarrollo. Utiliza los conceptos de "signboard" y el estado de flujo de trabajo "TBD, WIP, Done" para tener una amplia visión del proyecto y mejorar la comunicación. La metodología Kanban se basa en cuatro principios:

**Reducción del desperdicio.** Kanban se basa en hacer lo justo y necesario, reduciendo todo aquello que sea secundario.

**Calidad garantizada.** Kanban no prioriza la rapidez, sino la calidad de las tareas realizadas. No hay margen de error, todo debe salir bien a la primera.

**Flexibilidad.** Kanban tiene la capacidad de dar respuesta a tareas imprevistas, priorizando aquellas tareas entrantes según las necesidades del momento.

**Mejora continua.** Kanban no es simplemente un método de gestión, sino es considerado como un sistema de mejora de los proyectos.

### 3.2.3 Lean

LSD (Lean software development) es una metodología iterativa centrada en la reducción de residuos y la optimización de procesos, para lograr la máxima ganancia posible. Lean tiene un récord en la industria manufacturera y ha ganado reconocimiento en la industria de desarrollo de software en los últimos años. Su filosofía es "Piensa en grande, actúa en pequeño, equivócate rápido; aprende con rapidez". Está basado en siete principios:

**Eliminar los desperdicios.** Todo lo que no añade valor al cliente se considera un desperdicio: código y funcionalidades innecesarias, retraso en el proceso de desarrollo de software, requisito poco claro, burocracia y comunicación interna lenta.

**Ampliar el aprendizaje.** Obtener retroalimentación mediante reuniones cortas con los clientes ayuda a mejorar la fase actual de desarrollo y permite ajustar esfuerzos para introducir mejoras. El desarrollo es un proceso de aprendizaje continuo, y retos de los equipos de trabajo.

**Decidir lo más tarde posible.** El desarrollo está asociado con incertidumbre, los mejores resultados se alcanzan al retrasar las decisiones para validar que se basen en hechos y no en suposiciones inciertas.

**Entregar tan rápido como sea posible.** Cuanto antes se entregue el producto más pronto se pueden recibir comentarios de mejora. En la era de la evolución tecnológica, no es el más grande quien sobrevive, sino quien es más rápido.

**Capacitar el equipo.** Las personas necesitan algo más que una lista de tareas, necesitan motivación y un objetivo alcanzable con la garantía de que el equipo puede elegir sus propios compromisos.

**Crear integridad.** Integridad significa que los componentes separados del sistema funcionan bien juntos, como en un todo, logrando equilibrio entre la flexibilidad, sostenibilidad, eficiencia y capacidad de respuesta.

**Ver todo el conjunto.** Los sistemas de software hoy en día no son simplemente la suma de sus partes, sino también el producto de sus interacciones.

De la filosofía Lean se desprende la metodología LeanUX que se centra en los proyectos de Experiencia de Usuario.

### **3.2.3.1 LeanUX**

Lean UX es un proceso para el diseño de productos y servicios, permite refinar de manera progresiva, rápida y eficiente la oferta de valor para los usuarios. Lean UX está centrado en un conjunto de principios que abarcan desde el diseño inicial hasta la gestión. Implica un cambio en la cultura organizacional, y fomenta los equipos colaborativos, multifuncionales y adaptados a la realidad actual.

Lean UX aumenta la frecuencia de contacto con clientes reales, reduciendo el despilfarro y validando lo antes posible las ideas del cliente para evitar caer en suposiciones incorrectas. Lean UX plantea una serie de principios orientados al desarrollo de experiencia del usuario:



**Equipos multifuncionales.** Equipo conformado por personas de diversas disciplinas, que trabajan de manera colaborativa y continua

**Equipos Pequeños, dedicados, coubicados.** Al limitar el tamaño de los equipos se facilita la comunicación, concentración y camaradería.

**Progreso igual a resultados, no a entregas de documentación.** El progreso del proyecto se mide según los resultados de negocio, mas no por la acumulación de documentación.

**Equipos centrados en los problemas.** El equipo está enfocado en resolver problemas a través de sus propias ideas y no se limita a implementar nuevas funcionalidades de desarrollo.

**Eliminación del despilfarro.** Se considera un despilfarro todo aquello que no ayude a conseguir resultados, debiendo eliminarse del proceso de trabajo.

**Lotes pequeños.** El equipo es eficiente al crear el diseño necesario para evidenciar resultados. Evitando crear un gran stock de ideas sin probar ni implementar.

**Descubrimiento continuo.** Los clientes reales del producto participan frecuentemente en el proceso de desarrollo para validar nuevas ideas.

**GOOB (Getting out of the building).** Mientras antes escuchemos a los usuarios, antes sabremos si las ideas deben pasar a desarrollo. Las respuestas a los problemas están en el mercado, con los clientes reales, no en la oficina del equipo.

**Entendimiento común.** El equipo trabaja de forma conjunta para lograr conocimiento colectivo y una comprensión profunda del producto.

**Antimodelos: estrellas, gurús y ninjas.** Las estrellas, gurús, ninjas y expertos dividen al grupo y reducen la colaboración.

**Exteriorización del trabajo.** Los equipos deben mostrar su trabajo de forma abierta, para facilitar el flujo de información. El uso de pizarrones, tableros, notas adhesivas y demás anotaciones visibles permite mostrar el trabajo a todos los interesados.

**Hacer en lugar de analizar.** El feedback más importante surge de la reacción de los usuarios, no del análisis interno.

**Aprendizaje en lugar de crecimiento.** Los equipos necesitan experimentar con ideas que no funcionarán. Las grandes ideas siempre proceden de altos riesgos.

**Permiso para equivocarse.** Los equipos son libres de equivocarse, ya que estarán más dispuestos a experimentar y usar su creatividad en la generación de ideas.

**Escapar de negocios basados en entregables.** El progreso de un proyecto depende de los resultados que se consiguen sobre las soluciones, mas no de su documentación.

Lean UX también describe siete fases para desarrollar un producto según (Artigas, 2015)

- 1. Inicio de definición del producto.** En esta fase se realiza brainstorming (lluvia de ideas) con todos los stakeholders. Como resultado se obtiene un plan de lanzamiento del proyecto, concepto simples y prototipos de lo que se piensa construir.
- 2. Investigación.** En esta fase se ajustan las suposiciones, basándose en el tiempo, recursos, complejidad del producto y nivel de conocimiento existente. Para el proceso de investigación utiliza el análisis de mercado, análisis competitivo, y encuestas de conducta de cliente. Cuando el producto está terminado, se revisa heurísticas, analíticas, contenido, contexto del producto y tests con usuarios.
- 3. Análisis.** En esta fase se realiza la planificación del proyecto.
- 4. Diseño.** En esta fase se crean escenarios, mapas conceptuales, prototipos, diagramas de flujo de tareas en base a las especificaciones de diseño.

5. **Implementación.** En esta fase se ensambla el código y los recursos de diseño para crear un producto en base al diseño realizado.
6. **Lanzamiento del producto real.** En esta fase el feedback de los usuarios resulta importante para la mejora del producto. El producto deber ser constantemente monitorizado para asegurar un éxito permanente.
7. **Mejora de producto continua.** A través del uso de analíticas y paneles de rendimiento del producto se asegura una mejora continua del producto.

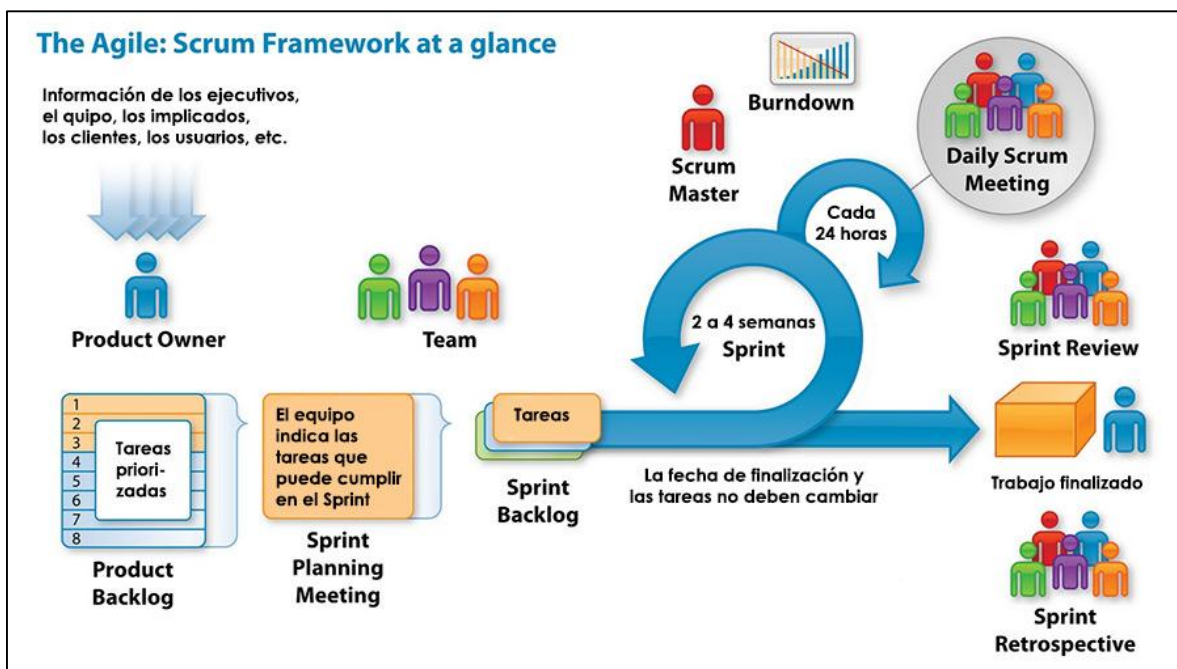
### 3.2.4 Scrum

Es un marco de trabajo para la gestión de proyectos que se centra en situaciones donde es difícil planificar el futuro. Dirigido a proyectos donde los requisitos cambian constantemente.

Scrum se encuentra organizado de la siguiente forma. El software es desarrollado por un equipo auto-organizado y autónomo. Trabaja con períodos cortos de tiempo o iteraciones llamados “sprints” (1-4 semanas). Las características que tendrá el sistema se registran en un backlog. El dueño del producto decide qué elementos del backlog se deben desarrollar en el sprint. El equipo está formado de cinco integrantes. Los miembros del equipo coordinan su trabajo en una reunión diaria “stand-up” y el Scrum Master es el encargado de resolver los impedimentos para que el equipo trabaje continuamente.

Scrum se define como útil y eficaz para el desarrollo de aplicaciones móviles, permite mantener flexibilidad en un contexto cambiante, a través de reglas claras de juego para todos los participantes.

En la Figura 21 podemos observar el proceso de Scrum:



**Figura 21.** Scrum

Fuente: (Scrum.org, s.f)

Scrum define un conjunto de roles, reuniones y artefactos. Los roles se describen a continuación:

**Product Owner.** Es la persona responsable de transmitir al equipo de desarrollo la visión del producto. Se encarga de administrar el Product Backlog, escribir las historias de usuario y ser quien da la cara al cliente. Es responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo para cada sprint. Asegura que el Product Backlog sea visible para todos. Es quien define, válida y prioriza el conjunto de requerimientos.

**Equipo de desarrollo.** Es responsable de entregar el incremento del producto al final de cada sprint. Se caracteriza por ser auto organizado, y multidisciplinario. Sus tareas incluyen: análisis, diseño, desarrollo, prueba, comunicación técnica y documentación.

**Scrum Master.** Es responsable de asegurar que Scrum es comprendido por todo el equipo. Se encarga de eliminar cualquier impedimento que pueda afectar la meta de sprint.

Las reuniones de Scrum se describen a continuación:

**Planificación de sprint.** Es la reunión donde se selecciona del Product Backlog el conjunto de requerimientos que serán desarrollados en cada sprint. El equipo determina la cantidad de historias que puede comprometerse a terminar en cada sprint.

**Diaria (Daily Scrum).** Tiene una duración corta de 15-20 minutos. Cada miembro del equipo responde las siguientes preguntas: ¿Qué hice ayer para ayudar al equipo a

terminar el sprint?, ¿Qué voy a hacer hoy para ayudar al equipo a terminar el sprint?, ¿Qué obstáculos se interponen en el camino del equipo?

**Demo y Revisión de Sprint.** Durante esta reunión, el equipo scrum y las partes interesadas colaboran sobre lo que se ha completado y que parte permanece pendiente en sprint. Se realiza una demostración del trabajo realizado al Product Owner y a los usuarios involucrados. El resultado de la reunión define los elementos que debemos añadir en el próximo sprint del Backlog del producto.

**Retrospectiva de Sprint.** Es celebrada después de la reunión de revisión de Sprint y antes de la próxima reunión de planificación de Sprint. En esta reunión todos los miembros del equipo realizan una valoración del trabajo realizado en el último sprint, identificando puntos de mejora de cara a los siguientes a realizar. Se debe responder a las siguientes preguntas: ¿Qué debemos empezar a hacer?  
¿Qué debemos dejar de hacer?, ¿Qué debemos seguir haciendo?

Los artefactos de Scrum se describen a continuación:

**Sprint.** Iteración de duración fijada, donde el equipo trabaja para convertir las historias de usuario en una nueva versión del software.

**Product Backlog.** Es una lista conjunta de requerimientos (historias de usuario) descrita en un lenguaje no técnico y priorizados por valor de negocio o retorno de inversión. Es dinámico, con descripciones genéricas y ajustado constantemente a lo largo del desarrollo. Permite identificar lo que la aplicación necesita para ser competitiva y útil. Se ordena a menudo por valor, riesgo, prioridad y necesidad, los elementos de alta prioridad son de desarrollo inmediato sobre elementos de menor prioridad.

**Sprint Backlog.** Es una lista de requerimientos seleccionados extraídos del Product Backlog. Permite hacer visible todo lo que se necesita para cumplir con la meta de Sprint. Una vez realizada la estimación de las tareas del sprint no es posible agregar historias de usuario adicionales

**Ejecución del Sprint.** Cada sprint se ejecuta en un periodo de entre 1 y 4 semanas. Una vez iniciada la ejecución del sprint, no puede ser modificado, y en caso de requerir nuevos cambios se define de otro sprint backlog una vez concluido el anterior.

**Entregable.** Una vez concluida la ejecución del sprint, se dispondrá de un entregable funcional, conocido como incremento del producto.

**Evolución del proyecto (Burn down).** Es un cuadro que refleja el estado del proyecto. Muestra los requerimientos que se encuentran pendientes de ser abordados (en el product backlog), los requerimientos que se están desarrollando (sprint backlog) y los requerimientos cuyo desarrollo se han completado en su totalidad.



### 3.3 Metodologías en el desarrollo de aplicaciones para dispositivos móviles

Las metodologías diseñadas para aplicaciones móviles son creadas en base a métodos ágiles. Las prácticas ágiles se pueden implementar en un proyecto móvil, incluso con la combinación de enfoques no ágiles para alcanzar mayor precisión.

En la Tabla 6 se describen las metodologías móviles más usadas, tomadas en base a los estudios seleccionados en la etapa “Panorama actual de las Metodologías de desarrollo”. Cabe indicar que en la tabla no se incluyen los métodos Xp, Scrum y Lean debido a que fueron descritas en la sección de “Métodos Ágiles”.

**Tabla 6**

*Metodologías para el desarrollo de aplicaciones móviles*

MÉTODO	AUTORES	AÑO	ÁGIL	NO ÁGIL
<b>Mobile D</b>	Abrahamsson et al.	2004	XP, Crystal	RUP
<b>MASAM</b>	Jeong et al.	2008	XP, TDD, Pares, Refactor	RUP, SPEM
<b>Hybrid</b>	Rahimian et al.	2008	ASD, NPD	NPD
<b>SLeSS</b>	Cunha et al.	2011	Scrum	Lean Six Sigma

En el siguiente apartado se detallan las principales características de las metodologías de desarrollo móviles seleccionadas. Cabe indicar que la información recopilada es muy limitada y en la mayoría de los casos es incompleta y sin casos prácticos de uso.

### 3.3.1 Mobile-D

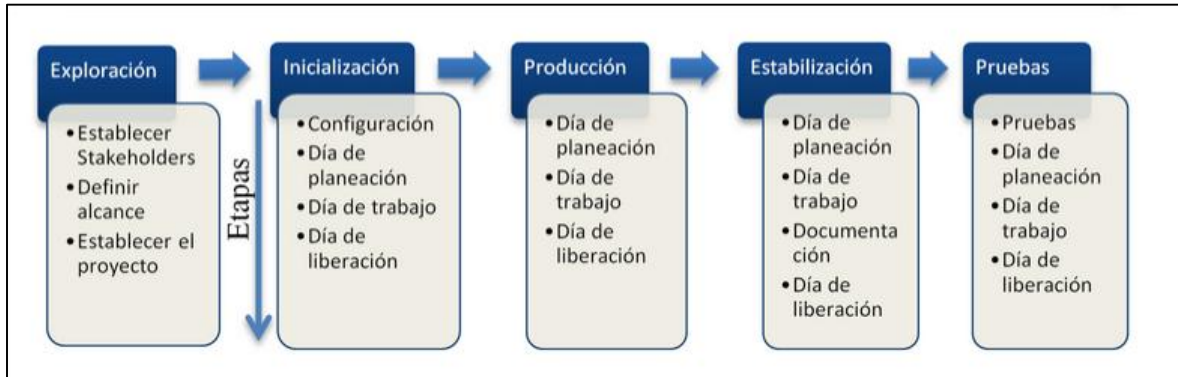
El Mobile-D es un modelo creado en Finlandia por Abrahamsson et al. (2004) Se caracteriza por ser uno de los pioneros en el uso de prácticas ágiles sobre el desarrollo de aplicaciones móviles. (Abrahamsson, y otros, 2004)

Mobile-D es una combinación de varias técnicas consolidadas: Programación Extrema (XP), del cual se han reutilizado las buenas prácticas de desarrollo; Metodologías Crystal, del cual se ha abordado la escalabilidad de los métodos y Proceso Racional Unificado (RUP), del cual se reúsa el ciclo de vida.

El propósito del método es establecer un arquetipo funcional y ágil para el desarrollo de software. Pone énfasis en la cohesión del equipo, la armonía y la confianza en los demás. Trabajo con pequeños grupos muy unidos y comprometidos de hasta diez desarrolladores bajo un mismo lugar. El ciclo de desarrollo de un producto tiene diez semanas de duración.

Para Mobile-D la comunicación es crucial dentro del equipo y verticalmente en toda la organización. La interacción continua con los clientes permite obtener información valiosa sobre el mercado objetivo y sus necesidades, ya que por historia los clientes no siempre están dispuestos a cooperar con conocimientos relevantes.

El ciclo de desarrollo de Mobile-D se compone de cinco fases que se muestran en la Figura 22.



**Figura 22.** Ciclo de desarrollo de Mobile-D

Fuente: (Cruz, 2014)

Cada fase tiene etapas, tareas y prácticas asociadas, que incluyen tres días para el desarrollo, dividido en un día de planificación, un día de trabajo y un día para la liberación del producto. Existe un día adicional llamado integración que se agrega cuando varios equipos están trabajando en forma paralela sobre diferentes partes de un producto.

El enfoque Mobile-D se basa en un conjunto de prácticas. La mayoría de las prácticas son extraídos del enfoque ágil para ser adaptados al desarrollo de aplicaciones móviles. Destacando, la línea de arquitectura que proporciona soluciones de confianza y calidad; y el desarrollo orientado a pruebas automatizadas que permite mejorar el diseño en base a defectos encontrados que posteriormente se validan con pruebas de aceptación del cliente.

Las prácticas que usa Mobile-D son: integración continua, fase y ubicación, móvil test-driven development, programación en pares, mejora del proceso de software ágil, métricas, cliente fuera de lugar, línea de arquitectura y el enfoque centrado en el usuario.

El ciclo de desarrollo que propone Mobile-D es el siguiente:

**Exploración.** En esta fase se elabora el plan y se establece las características del proyecto con una alta participación del cliente. Este compuesto por tres etapas donde se establecen actores, se define el alcance y se establece las funcionalidades requeridas el proyecto.

**Inicialización.** En esta fase se realizan las tareas de: identificación de recursos a ser utilizados, configuración del proyecto, planificación de capacitaciones técnicas, definición de arquitectura, patrones, y estándares de desarrollo. Está compuesto por cuatro etapas: ejecución del proyecto, planificación inicial, día de prueba y día de salida.

**Producción.** Esta fase se lleva a cabo toda la construcción de código. Trabaja con un ciclo iterativo de tres días (planificar, trabajar y liberar) para obtener los incrementos del producto. Para el desarrollo de productos establece el uso de repositorios de código, el uso de la técnica de pruebas unitarias TDD (Test Driven Development), y la ejecución de pruebas de aceptación.

**Estabilización.** Se realizan las acciones de integración para obtener una única versión. Las pruebas de integración se realizan sobre un producto integrado y son ejecutadas para validar el correcto funcionamiento. En esta fase también se integra y estabiliza los desarrollos de los equipos separados geográficamente.

**Pruebas.** En esta última fase se obtiene un producto terminado, estabilizado, integrado y completamente funcional. Se realizan pruebas en base a los requerimientos del cliente y se ajustan defectos encontrados.

Una vez finalizadas todas las fases se obtiene una aplicación lista para ser publicada y entregada al cliente.

En base a la investigación bibliográfica realizada, se concluye que la metodología Mobile-D es un muy utilizada en proyectos de investigación a nivel académico, pero no se encontró evidencia de casos prácticos de uso en empresas públicas o privadas en la creación de aplicaciones móviles.

### **3.3.2 Masam**

Masam (mobile application software development based on agile method), es una metodología ágil propuesta por Jeong et al. (2008). Está basado en la programación extrema (XP), el proceso unificado ágil RUP, y el metamodelo de la ingeniería de proceso del software (SPEM). Utiliza el desarrollo controlado por pruebas, programación de pares y refactorización. (Corral, Sillitti, & Succi, 2015)

Está fuertemente vinculado al enfoque de Mobile D, pero con la inclusión de una herramienta de gestión de proyectos y otras pequeñas variaciones. Permite la creación de prototipos de interfaz de usuario, gestiona una base de conocimiento, usa TDD para pruebas de rendimiento y usa el modelo Model-Driven Architecture (MDA) para generar código en varias plataformas.

Masam define un conjunto de roles, tareas y productos de trabajo. Los roles son un conjunto de habilidades, competencias y responsabilidades que posee una persona. La tarea es una unidad de trabajo asignable asignada a un rol específico. A cada rol se le asigna ciertas tareas, que pueden afectar a uno o varios productos de trabajo, y el producto de trabajo es un término general para las entradas y salidas de las tareas. Siendo tres tipos de productos de trabajo. (Corral, Sillitti, & Succi, 2015)

MASAM propone un ciclo de desarrollo en cuatro fases:

**Fase de Preparación del Desarrollo.** En esta fase se trata de comprender la idea del cliente con su producto. El objetivo es tener una visión del producto para posteriormente prepararse para el desarrollo. Define el resumen y la primera noción del producto, así como roles y responsabilidades

**Fase de Personificación.** En esta fase se representa la idea del cliente en prototipos de Interfaz de Usuario. Se centra en la comprensión de las necesidades del usuario y la definición de la arquitectura del producto de software.

**Fase de Desarrollo del Producto.** En esta fase se construye el producto en forma de incrementos hasta desarrollar el producto final. La implementación del producto de software se lleva a cabo a través de las técnicas TDD, programación de pares, refactorización e integración continua.

**Fase de Comercialización.** En esta fase se programa el lanzamiento y venta del producto, con la participación del usuario se ejecutan pruebas de aceptación, pruebas de usuario y pruebas de lanzamiento al mercado. En la **Tabla 7** podemos visualizar las fases, actividades y tareas de MASAM.

**Tabla 7**  
*Fases de MASAM*

FASE	ACTIVIDAD	TAREA
<b>Preparación</b>	Agarrar el producto	Resumen del producto
		Pre-planeación
	Compartir concepto de producto	Definición del usuario
		Análisis inicial del producto
	Preparación del proyecto	Coordinación del proceso de desarrollo
Coordinación de recursos del proyecto		
		Pre estudio
<b>Personificación</b>	El usuario necesita entender	Taller de Story card
		Diseño de interfaz de usuario

	Arquitectos	Análisis de requisitos no funcionales
		Definición de la arquitectura
		Gestión de patrones
<b>Desarrollo</b>	Implementación y preparación	Configuración del entorno
		La planificación del desarrollo
	Ciclo de liberación	Planificación de la liberación
		Ciclo de iteración
		Lanzamiento
<b>Comercialización</b>	Prueba del sistema	Examen de ingreso
		Prueba del usuario
	Venta de productos	Prueba de lanzamiento
		Lanzamiento de productos

Fuente: (Corral, Sillitti, & Succi, 2015)

En base a la investigación bibliográfica realizada, se concluye que la metodología Masam es un enfoque que se presenta únicamente a nivel teórico debido a que no se encontró evidencia de casos académicos ni casos prácticos de uso para la creación de aplicaciones móviles.

### 3.3.3 Hdm

Hdm (hybrid methodology design). El método híbrido es una metodología con un enfoque diferente al ágil, basado en el riesgo y propuesta por Rahimian et al. (2007). Usa el modelo iterativo incremental para lograr una rápida entrega de productos, y mejorar la capacidad de gestión de riesgo. (Corral, Sillitti, & Succi, 2015)

Tiene como base el ciclo de desarrollo de software genérico con la inclusión de prácticas ágiles y los principios de introducción de nuevos productos. Trabaja con



iteraciones que van desde la idea hasta el lanzamiento del producto, donde se priorizan requisitos. Se basa en la combinación entre metodologías Ágiles, Desarrollo de Software Adaptativo (ASD) y Desarrollo de Producto (NPD).

El método HDM utiliza los principios de agilidad, conciencia del mercado, software de línea de productos de soporte, arquitectura basada en el desarrollo, reutilización, inclusión de revisión, sesiones de aprendizaje y especificación temprana de arquitectura física. En la Tabla 8 se muestran las fases de ciclo de desarrollo Hybrid Methodology Design.

**Tabla 8**  
*Fases de Hybrid Methodology Design*

FASE	DESCRIPCIÓN
<b>Generación de idea</b>	
<b>Iniciación del proyecto</b>	Análisis preliminar
<b>Análisis</b>	Análisis detallado Creación del Prototipo Funcional
<b>Diseño</b>	Diseño arquitectónico Diseño detallado
<b>Implementación (Motor de desarrollo)</b>	Planificación del ciclo adaptable Ingeniería de Componentes Concurrentes Actualizaciones de la biblioteca de componentes
<b>Prueba</b>	Revisión de calidad Pruebas de mercado
<b>Comercialización</b>	

Fuente: (Corral, Sillitti, & Succi, 2015)

Las prácticas que propone HDM se basan en el enfoque ágil, iterativo-incremental, entre ellas tenemos definir prioridades en los requisitos, constante participación del cliente, desarrollo basado en pruebas (TDD), calidad garantizada del producto final, programadores con alto nivel de experticia, comunicación altamente efectiva, revisión minuciosa y refinamiento de todo el proceso con aplicación de sesiones de aprendizaje, definición del nivel de abstracción para la próxima iteración, revisión y perfeccionamiento de los requisitos priorizándolos para la siguiente iteración, selección de los enfoques de diseño a utilizar en la iteración actual, aplicación de los enfoques de diseño seleccionados y aplicación de un proceso de adaptación. El proceso de desarrollo de aplicaciones móviles híbrido propone cuatro iteraciones.

**Primera Iteración.** Realizar el análisis teniendo en cuenta consideraciones de mercado. El diseño es validado con arquitectura para mitigar los posibles riesgos de desarrollo. TDD es utilizado para la implementación y pruebas. Agrega una fase de comercialización.

**Segunda Iteración.** Incluye actividades de desarrollo de nuevos productos, integración de incrementos del producto y pruebas de comportamiento de mercado. Se lanza la fase de comercialización.

**Tercera Iteración.** Se integran a la metodología las ideas de Desarrollo Adaptativo de Software (ASD).

**Cuarta Iteración.** Validar la calidad del producto e inclusión de elementos de manejo de prototipos para mitigar riesgos probables relacionados con la tecnología.

HDM es una metodología considerada de alto nivel, debido a que no se ha proporcionado tareas específicas para las etapas identificadas.

En base a la investigación bibliográfica realizada, y similar al modelo Masam se concluye que la metodología HDM es un enfoque que se presenta únicamente a nivel teórico debido a que no se encontró evidencia de casos académicos ni casos prácticos de uso para la creación de aplicaciones móviles.

#### **3.3.4 SLeSS**

SLeSS (Scrum and Lean Six Sigma -LSS). Propuesto por Cunha et al. (2011), es un enfoque de integración de Scrum y Lean Six Sigma que se centra en la gestión de proyectos y mejora de procesos, es utilizado en proyectos de desarrollo de aplicaciones para teléfonos móviles. (Corral, Sillitti, & Succi, 2015)

Por una parte, Scrum es un marco de trabajo ágil para la gestión de proyectos y el desarrollo de software, que adopta un enfoque empírico más que prescriptivo, y puede ser utilizado en complejos Proyectos. Mientras que Lean Six Sigma (LSS) es una metodología para definir y mejorar los productos, procesos y servicios con un enfoque en

la reducción de defectos y fallas, en la variación y eliminación de residuos, priorizando, de manera planeada y objetiva, el logro de calidad y resultados financieros.

El método SLess utiliza dos tipos de product backlog, customization product backlog (para personalizar proyectos de desarrollo) y LSS product backlog (para mejoras de proceso) para mejorar el proceso de desarrollo y el resultado de los proyectos. El modelo permite alcanzar los objetivos de alto rendimiento y calidad del proyecto en desarrollo.

Permite adaptarse a los cambios de requisitos en las etapas posteriores del proyecto y con menor impacto general que un enfoque tradicional. Ayuda a cumplir con los plazos, reduce las horas extraordinarias y ofrece versiones más rápidas y acorta el ciclo de desarrollo. Aumenta la productividad, mejora la calidad del proceso y ayuda en la reducción de costes. Mejora progresivamente el proceso de desarrollo, el proceso de gestión y el resultado de proyectos con menos defectos y fallas.

SLeSS se implementa como un marco de calidad con un enfoque incremental una vez que Scrum esté establecido en la organización. Es importante que el Scrum Master y el Product Owner comprendan adecuadamente las técnicas LSS, los procesos de desarrollo y las técnicas de gestión. (Flora, Chande, & Wang, 2014)

El enfoque SLeSS presenta cinco fases según (Corral, Sillitti, & Succi, 2015) que detallan en la Tabla 9.

**Tabla 9**  
*Fases del Enfoque SLeSS*

FASE	ELEMENTOS DEL BACKLOG
<b>1 Definición</b>	Contrato de Proyecto
	Análisis inicial
<b>2 Medición</b>	Diagrama SIPOC (Proveedor, Entradas, Procesos, Productos, Clientes)
	Mapa de proceso
	Diagrama de causa y efecto
	Matriz de causas y efectos
	Matriz de Esfuerzo de Impacto
	Capacidad inicial
	Sistemas de medición e inspección
<b>3 Análisis</b>	FTA (análisis de fallos)
	Análisis de los insumos críticos de los procesos
<b>4 Mejora</b>	Plan de ACCION
	SIPOC (Proveedor, Entradas, Procesos, Productos, Clientes)
	Mapa de proceso
	Capacidad final de los procesos
<b>5 Control</b>	Plan de control

En base a la investigación bibliográfica realizada, y similar al modelo Masam y HDM se concluye que la metodología SLeSS es un enfoque que se presenta únicamente a nivel teórico debido a que no se encontró evidencia de casos académicos ni casos prácticos de uso para la creación de aplicaciones móviles.

### **3.4 Propuestas para el desarrollo de aplicaciones para dispositivos móviles**

Luego de analizar las metodologías consideradas formales del mercado, a continuación, se detallan otras propuestas metodológicas consideradas no formales, debido a que son planteadas en proyectos de tesis a nivel académico:

- Un proceso integrado de desarrollo de productos de software en móviles
- Modelo de procedimientos “M-business” para el desarrollo de la interfaz de usuario en móviles
- Modelo de procesos MP–móvil para el desarrollo de aplicaciones para dispositivos móviles
- El modelo AGILEMAD
- El modelo MDAM

#### **3.4.1 Un proceso integrado de desarrollo de productos de software en móviles**

Este método se encuentra basado en el proceso de desarrollo de nuevos productos (NPD) para adaptar un producto de desarrollo holístico enfocado a servicios y aplicaciones móviles.

Este enfoque considera el uso de herramientas comunes para la estrategia de análisis y desarrollo del producto bajo un entorno competitivo más dinámico. Las características de este proceso son:

- Se basa en el Proceso de Desarrollo de Nuevos Productos en el mercado (NPD).
- Pone énfasis en la experiencia de usuario.
- Trabaja con equipos multidisciplinarios

Las cinco fases del ciclo de desarrollo según (Zeidler, Kittl, & Petrovic, 2007) se detallan a continuación:

**Generación de Idea.** El objetivo es evaluar y monitorear las tendencias tecnológicas.

**Desarrollo del Modelo y Caso de Negocio.** El objetivo es definir el valor proporcionado a los clientes y el valor agregado a los socios por el modelo de negocio. El valor es materializado a través de las utilidades generadas y satisfacción proporcionada al cliente.

**Aspectos Legales.** El objetivo es evaluar los aspectos legales que conlleva el desarrollo de servicios y software en dispositivos móviles. Aspectos como nombres, logos, marcas, contratos, políticas, convenios, protección y privacidad de datos.

**Investigación de Mercado y Diseño de Experiencia de Usuario.** El objetivo es realizar el diseño y evaluación de la Interfaz de Usuario.

**Implementación y Pruebas.** El objetivo es incluir tareas de construcción, implementación y pruebas del software. El objetivo es realizar el diseño y evaluación de la Interfaz de Usuario.

### **3.4.2 Modelo de procedimientos “M-business”**

Este modelo se enfoca en el desarrollo de la Interfaz de Usuario en Móviles apoyándose en el Diseño Centrado en el Usuario, se basa en un modelo de procedimientos que permite brindar una guía de diseño para el desarrollo de interfaces de usuario de aplicaciones para el comercio electrónico móvil conocidas como m-business. (Glissmann, Smolnik, Schierholz, Kolbe, & Brenner, 2005). El ciclo de desarrollo de este modelo está enfocado en cuatro practicas:

**Identificar Necesidades.** El objetivo es identificar las necesidades de una nueva interfaz de usuario móvil analizando aplicaciones existentes.

**Establecer Requerimientos.** El objetivo es obtener el contexto de uso, datos requeridos, requerimientos funcionales y requerimientos de usabilidad.

**Diseño Conceptual.** El objetivo es establecer actividades regulares de diseño como soluciones potenciales.

**Diseño Físico.** El objetivo es definir la apariencia y flujo que tendrá la aplicación con base en el diseño conceptual.

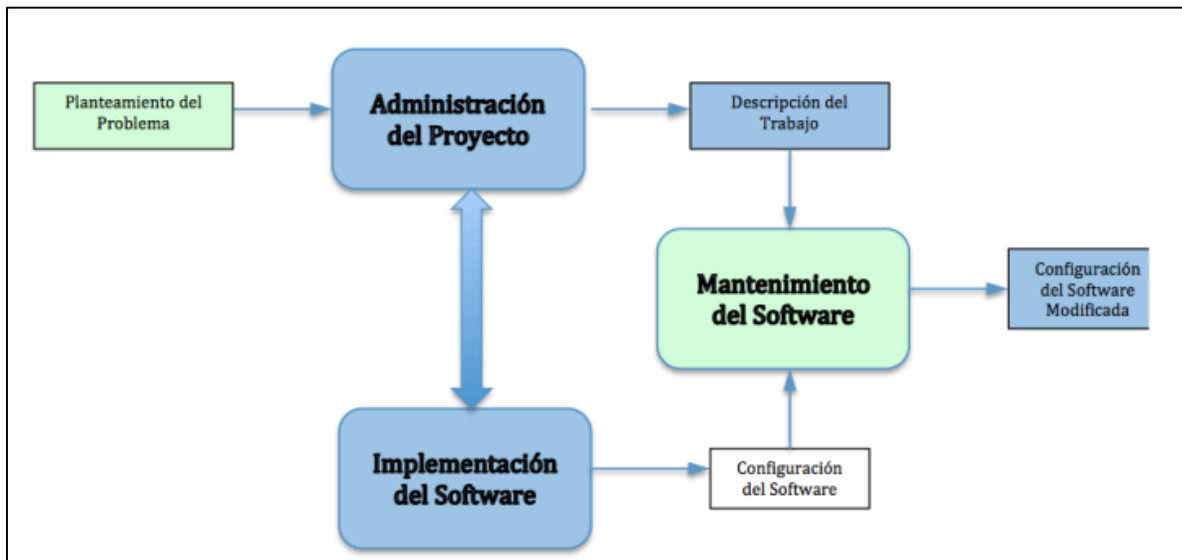


**Implementación.** El objetivo es construir e implementar la Interfaz de Usuario en los dispositivos móviles seleccionados.

### 3.4.3 Modelo de procesos MP–móvil

El modelo propone tres procesos que se describen en los siguientes apartados.

En la **Figura 23** se muestran los tres procesos y su relación:



**Figura 23.** Procesos integrados y su relación en el modelo MP-Móvil

Fuente: (Zeidler, Kittl, & Petrovic, 2007)

**Administración del Proyecto.** El propósito es establecer y realizar las tareas de un proyecto de implementación de software, que permitan cumplir con los objetivos del proyecto con calidad, tiempo y costos esperados. Se compone de las siguientes actividades:

- Inicio del Proyecto
- Planeación del Proyecto
- Ejecución del Plan del Proyecto
- Evaluación y Control del Plan del Proyecto
- Cierre del Proyecto

**Implementación del Software.** El propósito es realizar las actividades de análisis, diseño, construcción, integración y pruebas para productos de software, nuevos o modificados, de acuerdo con los requerimientos especificados. Se compone de las siguientes actividades:

- Inicio de la Implementación del Software
- Análisis de Requerimientos del Software
- Arquitectura del Software y Diseño Detallado
- Construcción del Software
- Integración del Software y Pruebas
- Entrega del Producto

**Mantenimiento del Software.** El propósito es realizar la modificación del software, de acuerdo con las peticiones de mantenimiento de clientes y usuarios finales. Se compone de las siguientes actividades:

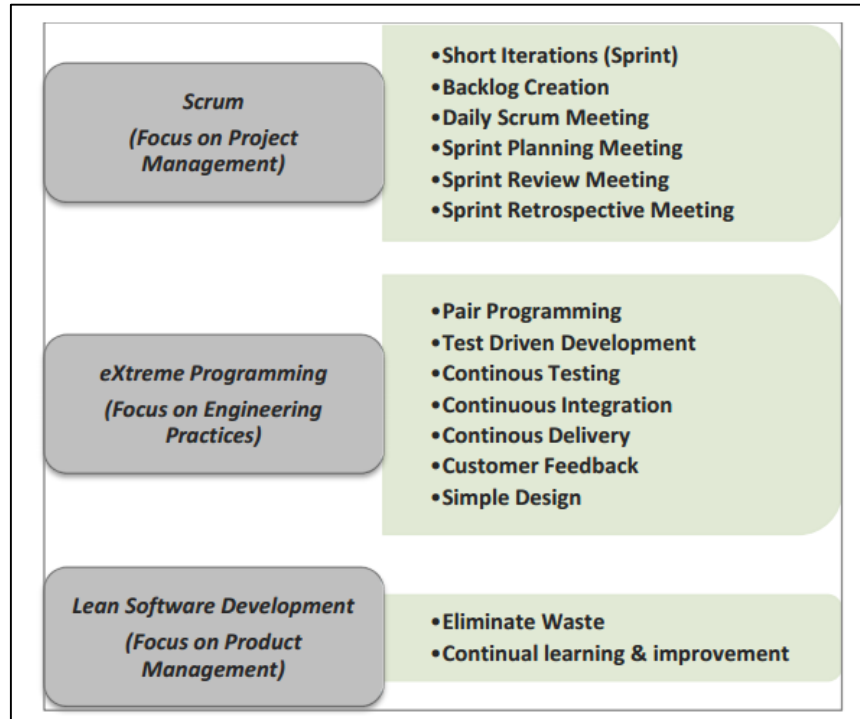
- Planificación del Proceso
- Atención de la Petición
- Mantenimiento del Software
- Reemplazo del Software
- Finalización del Mantenimiento

#### **3.4.4 El modelo AGILEMAD**

Este modelo utiliza prácticas y principios ágiles específicos para el desarrollo exitoso de aplicaciones móviles y está dirigido a la comunidad de desarrollo móvil y los investigadores.

Considera el enfoque ágil como más adecuado para el desarrollo de aplicaciones móviles debido a que la mayoría de los proyectos móviles son a corto plazo, requieren flexibilidad y poseen frecuentes solicitudes de cambio.

El modelo es el resultado de la combinación de los métodos Scrum, XP y Lean Agile. En la Figura 24 se muestran las prácticas ágiles utilizadas para el desarrollo de aplicaciones móviles.



**Figura 24.** Prácticas ágiles para el desarrollo de apps

Fuente: (Flora, Chande, & Wang, 2014)

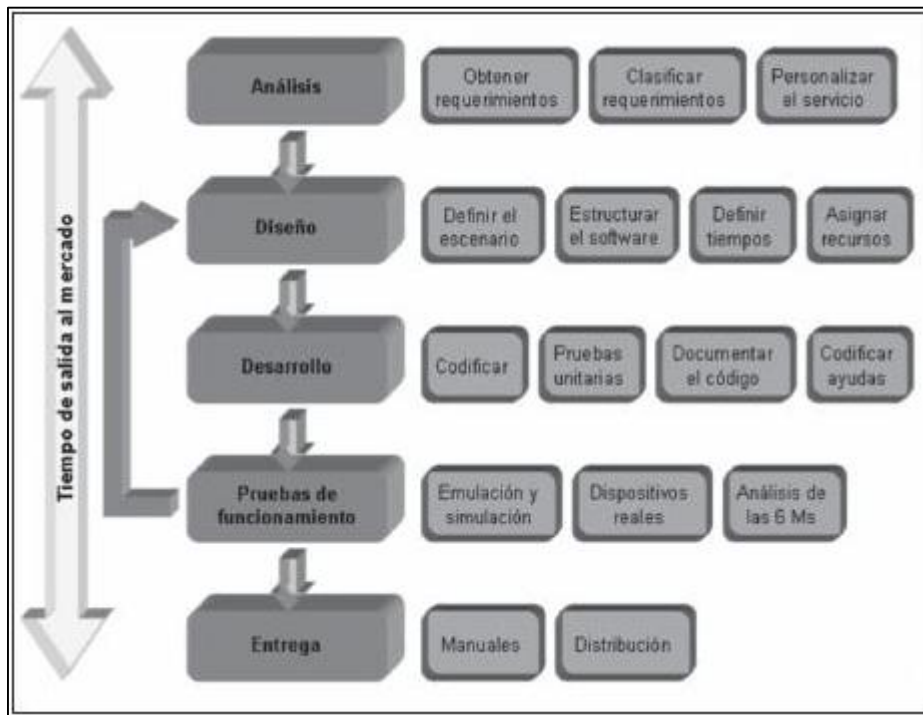
### 3.4.5 El modelo MDAM

Esta metodología propuesta se encuentra fundamentada en experiencias de investigaciones previas de desarrollo de aplicaciones móviles, la ingeniería de software educativo con modelado orientado por objetos (ISE-OO), la evaluación del potencial de éxito para servicios de tercera generación (6 M) y los valores de las metodologías ágiles.

El enfoque de micromundos interactivos y orientación por objetos se extraen de la ISE-OO, los conceptos inmersos en manifiesto ágil se heredan de las metodologías ágiles, los principios para que las aplicaciones móviles cumplan con las necesidades de los usuarios y generen ingresos se heredan de las 6 M's.

La metodología consta de cinco fases: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega. El método propuesto, fue utilizado para desarrollar una aplicación en el área de la salud, llamada m-Health en un tiempo aproximado de cuatro meses, cumpliendo el “Time to Market” definido, y con una evaluación exitosa del usuario final. (Villagrán, 2013)

En la **Figura 25** se muestra las fases de la metodología MDAM.



**Figura 25.** Fases de la metodología MDAM

Fuente: (Gasca, Camargo, & Medina, 2014)

**Análisis.** En esta fase se definen las características del entorno de la aplicación y analizan los requerimientos del cliente, en tres pasos:

- Obtener requerimientos
- Clasificar requerimientos
- Personalizar el servicio.

**Diseño.** Mediante esquemas o diagramas se plasma el pensamiento de la solución, debiendo considerar la mejor alternativa al integrar aspectos funcionales técnicos, económicos y sociales. Las cuatro actividades por realizar son:

- Definir el escenario
- Estructurar el software
- Definir tiempos
- Asignar recursos.

**Desarrollo.** En esta fase se implementa el diseño para crear un producto. Se realizan las actividades:

- Codificar
- Pruebas unitarias
- Documentar código
- Codificar ayudas

**Pruebas.** En esta fase se verifica que aplicación funcione correctamente con varias condiciones y escenarios. Las tareas por realizar son:

- Emulación y simulación
- Dispositivos reales
- Análisis de las 6 M's

**Entrega.** En esta etapa se entrega:

- Documentación
- Ejecutable
- código fuente
- manual del sistema.

### **3.5 Métodos Híbridos**

Los métodos híbridos se consideran como una mezcla de prácticas y artefactos que no necesariamente provienen de una misma metodología, ni son una variación de una metodología ágil o tradicional. Aparecen debido a las debilidades presentados en los enfoques ágiles y tradicionales, con la finalidad de crear un método robusto, pero al mismo tiempo flexible, que combine las bondades de dos o más metodologías.

Los métodos híbridos pretenden retomar las ventajas de los métodos, convirtiéndose en una combinación de mejores prácticas. La nueva tendencia en ingeniería de software

es diseñar metodologías híbridas, propuesta que es atribuida a Ivar Jacobson, uno de los tres creadores de UML (Unified Modeling Language, Lenguaje Unificado de Modelado)

### **3.6 Comparación de los métodos de desarrollo para móviles**

La Tabla 10 resume las principales características de todos los métodos encontrados en la investigación realizada, incluyendo aquellos métodos que no cuentan con información completa, ni evidencia de uso en casos prácticos, sino que son mencionados únicamente a nivel teórico.



**Tabla 10**  
Comparativa de los métodos de desarrollo para aplicaciones

METODOLOGÍAS DE ESTUDIO	PRÁCTICAS	CARACTERÍSTICAS	TAMAÑO DE PROYECTO
<b>Modelo en cascada</b>	Método tradicional	Secuencial Fechas límites Presupuestos fijos Proyectos muy controlados y poco previsibles	Medianos y Grandes
<b>Desarrollo rápido de aplicaciones</b>	Se combina con cualquier método ágil	Prototipos funcionales Patrones diseño UI Plazos de entrega son muy cortos	Pequeños
<b>Scrum</b>	Scrum	Se usa cuando es difícil planificar el futuro Manifiesto ágil	Pequeños
<b>Mobile-D</b>	Programación Extrema Metodología Cristal Proceso Racional Unificado	Pequeño equipo de 10 personas Prototipos	Pequeños y Medianos
<b>MASAM</b>	Programación Extrema Proceso unificado ágil, Proceso Racional Unificado Desarrollo impulsado por Pruebas Programación de pares Refactorización	Basado en el método Mobile-D	Pequeños
<b>HYBRID HDM</b>	Desarrollo de Software Adaptativo Desarrollo de nuevo producto	Se enfoca en la gestión del riesgo Conciencia del mercado Reutilización	Pequeños
<b>Espiral</b>	Se puede combinar con modelo en cascada	Basado en el modelo en espiral más una nueva fase. Participación del usuario Análisis de riesgo	Grandes
<b>Rapid7</b>	Todo el equipo Haga lo más simple que funcione	Producción rápida de Documentación	Pequeños, medianos y grandes
<b>SleSS</b>	Scrum Lean Six Sigma	Se usa en proyectos complejos	Pequeños
<b>Metodología en Y</b>	Enfoque evolutivo	Alinea a sus equipos a lo largo de características en lugar de plataformas,	Pequeños, medianos y grandes

Continúa



<b>Un proceso integrado de desarrollo de productos de software en móviles</b>	Proceso de desarrollo de nuevos productos	Énfasis en la experiencia de usuario.	Pequeños
<b>Modelo de procedimientos 'M – business' para el desarrollo de la interfaz de usuario en móviles</b>	Cascada	Se enfoca en el desarrollo de la Interfaz de Usuario Diseño Centrado en el Usuario	Medianos
<b>Modelo de procesos MP–móvil para el desarrollo de aplicaciones para dispositivos móviles</b>	Procesos integrados	Cumplir con los objetivos del proyecto con calidad, tiempo y costos esperados.	Pequeños y Medianos
<b>Modelo AGILEMAD</b>	Scrum Programación Extrema Lean agile	Dirigido a la comunidad de desarrollo móvil y los investigadores.	Pequeños
<b>Modelo MDAM</b>	6M, Evaluación del potencial de éxito para servicios de tercera generación	Basada en valores de las metodologías ágiles.	Pequeños
<b>MASEF</b>	Programación Extrema	Ligero Técnicas de gestión de control y riesgos.	Pequeños y Medianos
<b>DSDM</b>	Método de desarrollo del sistema dinámico	Trabaja con tiempo y recursos no fijos y con requisitos del usuario cambiantes.	Medianos y Grandes

### **3.7 Desarrollo de software ágil vs. Desarrollo de software tradicional**

El modelo en cascada es una práctica de diseño secuencial, con un enfoque de ingeniería tradicional, donde el proceso fluye hacia abajo (como una cascada) a través de las fases de concepción, planificación, análisis de requerimientos, diseño, pruebas, implementación y mantenimiento.

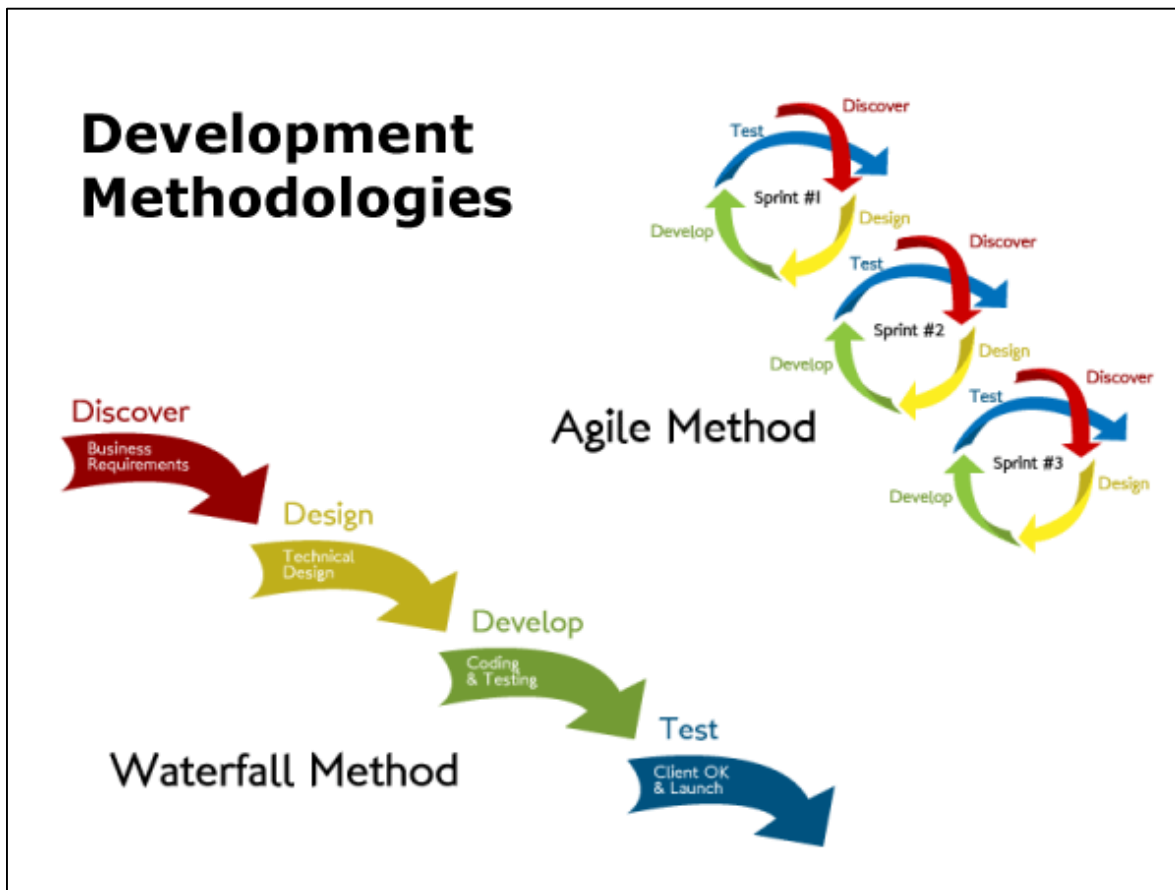
El modelo en cascada pone principal énfasis en la planificación, horarios, presupuestos y el despliegue de un sistema completo a la vez. Este método mantiene un control estricto durante toda la vida del proyecto a través de una extensa documentación y revisiones formales, seguidas de la aprobación por parte del cliente al finalizar cada fase. Normalmente no permite el movimiento de etapa a menos que esté completamente terminada y verificada.

Por tales motivos el enfoque ágil se presenta como una solución a los inconvenientes del modelo en cascada. En lugar de utilizar el diseño secuencial o enfoque intensivo, la otra opción es utilizar un método de desarrollo de software ágil con un enfoque iterativo/incremental de una manera altamente colaborativa, para construir software de alta calidad con un presupuesto eficaz, y con un control de cronograma que permita a los proyectos adoptar rápidamente cambios sobre los requisitos del cliente.

El método ágil se enfoca en entregar piezas funcionales pequeñas lo más rápido posible, acumulando funcionalidades adicionales durante el ciclo de vida del proyecto. Un enfoque ágil ayuda en gran parte a minimizar y mitigar el riesgo, permite que el

proyecto adopte cambios rápidamente sin congelar los requisitos como lo hace actualmente el modelo en cascada. Un proyecto ágil se lleva a cabo con iteraciones de una a cuatro semanas.

Las empresas están optando por elegir el proceso de desarrollo de software ágil frente a la cascada, debido a que está altamente orientado a resultados. Las características clave de la metodología ágil para ganar la aceptación en el mercado son, la entrega de un enfoque frecuente, incremental e iterativo, menos defectos, pruebas e integración continua, enfoque colaborativo y el máximo retorno de la inversión (ROI). En la Figura 26 se muestra el flujo de las dos metodologías de desarrollo de software.



**Figura 26.** Metodologías de Desarrollo

Fuente: (Picone, 2013)

En base a la investigación realizada se determina que la información acerca de metodologías para el desarrollo de aplicaciones móviles es muy limitada y sin casos de uso prácticos en su mayoría. De todos los métodos identificados, Mobile-D es el más usado a nivel de proyectos de investigación, ya que tiene más apariciones en la búsqueda bibliográfica. También podemos concluir que ningún método propuesto se puede considerar incompleto o poco útil para el desarrollo móvil debido a que no se encontró evidencia de casos de éxito o fallidos. La realidad es que ningún método puede funcionar de manera sola, necesita complementarse con otro modelo para afrontar de manera

completa los proyectos. Finalmente, a partir del análisis de la información de los métodos tradicionales y ágiles se concluye que lo ágil encaja de forma en los proyectos de desarrollo para móviles por su flexibilidad y adaptabilidad, por tal motivo se selecciona Scrum como principal aporte de la guía metodológica propuesta. A continuación se describen los criterios de selección de las metodologías y buenas prácticas seleccionadas para la elaboración de la guía.

### **3.8 Criterios para la selección de buenas prácticas**

A través de la revisión de literatura sobre las metodologías y métodos propuestos de desarrollo de aplicaciones móviles, se establece al *enfoque ágil* como principal aporte de buenas prácticas para la guía metodológica propuesta. El proceso de selección del *enfoque ágil* se realizó en base a los criterios que se describen a continuación:

#### **Existen más probabilidades de tener éxito**

- El entorno de desarrollo de aplicaciones para dispositivos móviles es rápido, incierto y susceptible de cambios rápidos.
- Los costes de entrada del desarrollo móvil son bajos y las demandas de tales proyectos son dinámicas y altamente impredecibles.
- Un equipo de desarrollo de aplicaciones móviles es exitoso cuando responde con rapidez las sugerencias de los usuarios.
- Un marco de gestión de proyectos ágil permite iteraciones rápidas basadas en datos del mundo real.

**Es adaptable a la naturaleza del entorno móvil**

- El desarrollo ágil se adapta a la naturaleza evolutiva de la movilidad
- Una aplicación móvil puede ajustarse fácilmente a los requisitos actuales del negocio.
- El desarrollo ágil hace una aplicación móvil más estable y menos problemática.
- La alta volatilidad del entorno obliga a equipo de desarrollo adaptarse a los nuevos cambios de plataforma, entornos de desarrollo y terminales. El ritmo cambiando requiere una alta respuesta de cambio frente a seguir un plan establecido.

**Brinda una alta participación del cliente y el equipo**

- El enfoque ágil se centra en la participación del cliente, la planificación adaptable, la evaluación continua y la gestión de riesgos.
- Permite que los proyectos se pueden completar de manera eficaz y eficiente.
- La comunicación constante con los clientes de la aplicación móvil permite asegurar que el producto final es exactamente lo que se espera y se necesita
- La metodología ágil proporciona a los clientes una retroalimentación continua, donde ven hitos cada 2-3 semanas, sin necesidad de esperar hasta el final del proyecto.
- El equipo de desarrollo se centra en diseñar una interfaz de usuario óptima, y garantiza una buena experiencia de usuario mediante la prueba de la aplicación con los usuarios antes de la liberación.
- Los equipos de desarrollo suelen estar integrados por pocas personas e incluso gestionadas por un solo desarrollador. Al ser pocos los integrantes la comunicación

entre los miembros es más fluida, y las interacciones en el proceso y herramientas mucho más controlados.

### **Permite lanzar una aplicación rápidamente**

- Las prácticas ágiles ayudan a insertar al mercado más rápidamente una aplicación móvil
- Luego de ser lanzada la aplicación, el enfoque ágil permite desarrollar en forma incremental la experiencia del usuario basándose en las opiniones de los usuarios

### **Aumenta la vida útil de una aplicación**

- La infraestructura y el hardware de las aplicaciones móviles evolucionan muy rápidamente, dándole una vida útil de aproximadamente 12 meses.
- El uso de ágil facilita un enfoque iterativo e incremental que encaja perfectamente con los requisitos de evolución rápida de los proyectos móviles.

### **Es complemento a otras metodologías**

- El enfoque ágil es uno de los enfoques y metodologías de desarrollo de software más populares, al acoplarse a otras técnicas permite identificar el gran aumento de amenazas y las vulnerabilidades en un sistema de información
- Las metodologías ágiles, XP, Scrum y LeanUx, son métodos apropiados para el desarrollo de aplicaciones móviles, ya que tienen el potencial de ayudar a los desarrolladores móviles a ofrecer una mejor calidad de las aplicaciones móviles en



poco tiempo. Combinados con las metodologías Mobile-D, MASAM y las técnicas Design Thinking y DevSecOps permiten abordar más ampliamente todo el ecosistema de desarrollo móvil.

En base a estos criterios se consideran los métodos Scrum, Programación Extrema, LeanUX, Mobile-D y MASAM como principales aportes metodológicos para el desarrollo de la guía. Los criterios de selección de cada método se describen a continuación.

Scrum al ser el marco de trabajo ágil más utilizado para el desarrollo móvil, utiliza una estrategia "flexible" donde el cliente puede participar con más transparencia en el progreso actual del proyecto. El cliente puede realizar ajustes sobre la marcha del proyecto, característica que no es permitida en un enfoque tradicional o secuencial.

La metodología LeanUX nos permitirá crear valor para los clientes con menos recursos, proporcionará un producto con valor perfecto, a través de la eliminación del desperdicio. Lean nos permitirá construir e iterar el producto rápidamente, teniendo retroalimentación en cada paso hasta llegar al producto perfecto en lugar de construir todo el producto de una sola vez y luego lanzarlo.

La Programación Extrema nos permitirá afrontar los requerimientos imprecisos o rápidamente cambiantes., a través de sus prácticas de refactorización, integración continua, diseño sencillo, estándares de codificación, las historias y los flujos de trabajo.

El método Mobile-D permitirá fomentar una comunicación vertical dentro del equipo y toda la organización; además de la interacción continua y la comunicación con el cliente para obtener información valiosa sobre el mercado objetivo y sus necesidades.

El método MASAM permitirá promover el desarrollo controlado por pruebas, programación de pares y refactorización. Al estar fuertemente vinculado al enfoque de Mobile D, presenta unas pequeñas variaciones como una herramienta de gestión de proyectos.

## CAPÍTULO IV

### GUÍA METODOLÓGICA PARA EL DESARROLLO DE APLICACIONES MÓVILES TRANSACCIONALES

#### 4.1 Introducción

La presente guía metodológica es una herramienta fundamentada en las mejores prácticas de desarrollo, ofrece a las empresas la posibilidad de gestionar los proyectos de desarrollo de aplicaciones móviles transacciones a través de un conjunto de pasos bien definidos.

La guía es una herramienta flexible y escalable que identifica los principales pasos a seguir en el desarrollo de aplicaciones móviles. Una “guía metodológica” se define como el documento que describe el conjunto de pasos a seguir para completar una actividad.

La guía está creada en base a mejores prácticas de desarrollo, metodologías de desarrollo para móviles y marcos de trabajo ágiles. Ofrece la posibilidad de descubrir las actividades que se requieren para crear, poner a prueba y desplegar una aplicación móvil en el mercado.

Según (Withers, 2000) una mejor práctica es una acción técnica o de gestión adoptada por el equipo, que ha demostrado una mejora sobre uno o más aspectos como: productividad, uso eficiente de recursos, coste, calendario, calidad o satisfacción del usuario. El uso de las buenas prácticas propuestas dependerá exclusivamente del tipo de proyecto y del tipo de aplicación a desarrollar.

## 4.2 Visión General

El mundo de la telefonía móvil se encuentra en constante evolución, nuevas herramientas, nuevas soluciones, nuevos métodos salen a la luz cada día: realidad aumentada, GPS combinado, acelerómetro. Nuevas aplicaciones son lanzadas al mercado, pero sólo unas pocas sobreviven y se mantienen. Lanzar una aplicación móvil al mercado es tan complejo como lanzar un nuevo negocio, debido a muchos factores que involucran al ecosistema móvil.

Para el desarrollo de aplicaciones resulta crucial poner en marcha una metodología de requisitos y una estrategia de diseño que permita construirlas para que sean eficientes, funcionales y seguras. La calidad de su desarrollo, las metodologías flexibles, los requisitos, el entorno de los diferentes sistemas operativos, son algunos aspectos que se debe tomar en cuenta. La problemática gira en torno al elevado número de dispositivos, diferentes sistemas operativos con múltiples versiones y un mercado heterogéneo.

Surge la necesidad de crear una guía en base a conocimientos obtenidos de metodologías y procesos existentes en el mercado, donde los métodos ágiles son una parte clave para el desarrollo de las aplicaciones móviles. La guía constituye la intersección que existe entre prácticas ágiles de gestión de proyectos y desarrollo de aplicaciones móviles. Permitirá que las empresas TI puedan reaccionar con mayor rapidez a alta demanda de aplicaciones del mercado. Para las empresas las aplicaciones

móviles ya no son una opción, sino constituyen un requisito para mantener a sus clientes a la vanguardia del mercado.

La creación de una aplicación móvil se realiza en ocho fases, ver Tabla 11. Cada fase tiene un conjunto de pasos bien definidos que guiarán al equipo de trabajo en el desarrollo. En términos generales cada fase ofrece lo siguiente.

**Generación de la idea.** Levantamiento de información necesaria para el proyecto.

**Gobierno.** Planificación, elicitación y clasificación de requerimientos.

**Diseño.** Estructura de la solución y definición del escenario tecnológico;

**Prototipado.** Personificación de las ideas del cliente en prototipos de interfaz de usuario.

**Construcción.** A partir del diseño se desarrolla una aplicación con sus sistemas frontend y backend.

**Integración.** Integración de incrementos de software e integración con sistemas internos y externos.

**Verificación.** Prueba y simulación del producto, ajustando detalles, para su posterior instalación en equipos reales y poder así evaluar su rendimiento.

**Lanzamiento y monitoreo.** Despliegue y monitoreo de la aplicación.

**Tabla 11**  
*Fases de la Guía Metodológica*

	<b>Fase I</b>	<b>Fase II</b>	<b>Fase III</b>	<b>Fase IV</b>	<b>Fase V</b>	<b>Fase VI</b>	<b>Fase VII</b>	<b>Fase VIII</b>
	<b>Generación de la idea</b>	<b>Gobierno</b>	<b>Diseño</b>	<b>Prototipado</b>	<b>Construcción</b>	<b>Integración</b>	<b>Verificación</b>	<b>Lanzamiento y Monitoreo</b>
1	Comprender necesidades e ideas del cliente y usuarios	Realizar la planeación inicial del proyecto	Definir la arquitectura funcional y técnica de la aplicación	Establecer soluciones potenciales	Configurar el entorno de desarrollo para el proyecto	Integrar los incrementos de producto	Realizar pruebas rigurosas y de validación	Gestionar la liberación de la solución móvil
2	Establecer los stakeholders y potenciales usuarios	Seleccionar los miembros del equipo de trabajo	Definir la arquitectura de seguridad de la aplicación	Diseñar los prototipos de Interfaz de Usuario (UI)	Implementar el código de manera efectiva	Integrar con sistemas internos o externos	Verificar que el producto tiene las funcionalidades requeridas por el cliente	Realizar el proceso de evaluación con usuarios
3	Identificar tendencias tecnológicas y Plataformas de desarrollo	Definir y validar los requerimientos	Revisión y refinamiento de diseño	Evaluar los Prototipos	Aplicar prácticas de seguridad en la codificación	Realizar pruebas de integración sobre el producto consolidado	Ejecutar diferentes tipos de pruebas	Ejecutar una estrategia de marketing
4	Generar las Épicas (Historias de Usuario generales)	Definir el tipo de solución a desarrollar	Definir la Experiencia de Usuario (UX)	Solicitar retroalimentación/feed back	Gestionar los requisitos de Plataforma Móvil	Utilizar un enfoque DevOps - DevSecOps	Realizar la corrección de defectos	Lanzar la aplicación al mercado
5	Establecer compromisos con el cliente	Definir el entorno de la solución	Definir los estándares de usabilidad y accesibilidad		Gestionar las actividades de Scrum	Utilizar el modelo de Integración Continua	Realizar Pruebas de Seguridad	Realizar el monitoreo de la aplicación móvil en el mercado
6		Establecer el enfoque ágil como marco de referencia para el desarrollo					Automatizar las Pruebas	
7		Planificar un ciclo de vida de desarrollo seguro (SDL)						
8		Realizar un análisis de impacto y riesgos						

La guía metodológica se encuentra estructurada de la siguiente manera: cada fase empieza con un nombre y una breve descripción, a continuación, se lista un conjunto de pasos, cada uno incluye el objetivo, descripción con un sinnúmero de recomendaciones y prácticas generadoras de valor, al finalizar cada paso se detallan artefactos, técnicas y entregables generados.

La estructura de guía metodológica está basada en la “Guía del emprendedor para negocios TI” propuesta según (Ordoñez Ortiz & Valle Caicedo, 2014). Su estructura está compuesta por 4 etapas y 31 pasos, cada paso propone un objetivo, descripción, artefactos, técnicas y entregables que guiarán en la construcción de una empresa.

En el siguiente apartado se describen los principios, objetivos, características, alcance, público objetivo, controversia y mejores prácticas de desarrollo de la presente guía propuesta.

### **4.3 Principios de la guía**

Están basados en los principios de la metodología LeanUX (Leal, 2017), y se centran en las necesidades de usuarios, ubicación, objetivos y métodos de trabajo en equipo.

**GOOB (Getting out of the building) Salir de Edificio.** Las respuestas sobre los usuarios están en el mercado, no en una oficina, mientras más escuchemos a los usuarios, mejor sabremos si sus ideas implican un desarrollo.

**Equipos multifuncionales, dedicados y autoorganizados.** Los equipos altamente colaborativos con tareas particulares mejoran su comunicación, y dan mejores resultados que un gran equipo dedicado a todo a la vez.

**Incremento de visibilidad.** Entrega regular y temprana de artefactos de alta calidad para que el cliente pueda ver el progreso del proyecto.

**Entrega de valor.** Entrega constante de software con mayor valor de negocio para el cliente.

**Aborda el riesgo temprano.** Concentrarse en las historias más arriesgadas desde el principio para reducir el riesgo total del proyecto.

**Promueve la adaptabilidad.** Capacidad para realizar cambios en el proyecto, presentar nuevas funcionalidades incluso en las últimas etapas, lo cual es muy importante para satisfacer las demandas del mercado.

**Seguridad y tendencias tecnológicas.** Permite realizar cambios, agregar nuevas funciones, controles de seguridad y seguir evolucionando con las últimas tendencias.

**Comunicación Efectiva.** La comunicación es uno de los pilares fundamentales de la guía. Fomenta la interacción y comunicación continua entre los desarrolladores, testers y el cliente en todas las etapas del proyecto.



**Reglas favorables.** Se promueve condiciones de trabajo favorables tanto para el equipo como para el cliente.

**Optimización de Recursos.** Permite descomponer un gran número de requisitos en trozos manejables y, por lo tanto, permite la máxima utilización de los recursos. Se promueve los ciclos cortos de trabajo y el principio Keep it simple (Mantenlo simple)

**Ritmo Constante.** Los desarrolladores, clientes y los interesados pueden controlar y mantener un ritmo constante de trabajo con una rápida retroalimentación.

**Descubrimiento y mejora Continua.** Comprometer a los clientes con el proceso de diseño y desarrollo, mediante actividades, validando nuevas ideas y reduciendo trasposos de información. El equipo debe aceptar la realidad de los cambios para lograr una mejora permanente.

**Pruebas.** Las pruebas se pueden realizar en cada etapa y simultáneamente para que los proyectos pueden ser lanzados de forma más temprana.

**Centrado en los usuarios.** Cliente constantemente involucrado en el proceso de desarrollo y alta participación del cliente en la creación de prototipos de interfaz de usuario.

**Centrado en los problemas.** Los equipos deben resolver el problema asignado y no limitarse a implementar un grupo de funciones. Tienen confianza, libertad para probar ideas y capacidad para reaccionar tan rápido como sea posible.

**Progreso es igual a resultados.** El progreso se mide en base a resultados y reportes de progreso que permiten validar si las funciones desarrolladas son eficaces.

**Eliminación de desperdicio.** Se debe eliminar todo lo que no ayude a conseguir resultados y no agregue valor al proceso de trabajo. Se promueve la agilidad y eficacia para obtener mejores resultados. Entre algunos tipos de desperdicio tenemos, talento no utilizado, exceso de inventario, sobreproducción de código, retrabajo, reprocesos, esperas o movimientos innecesarios de los miembros del equipo de trabajo.

**Lotes pequeños.** El equipo debe crear los diseños necesarios para avanzar, evitando crear un gran stock de ideas sin implementar, ya que el trabajo de gran magnitud es ineficiente y no permite ver resultado de esfuerzos realizados.

**Entendimiento común.** La comprensión profunda del espacio, el producto y los usuarios se construye a medida que se trabaja en equipo.

#### **4.4 Objetivos de la guía**

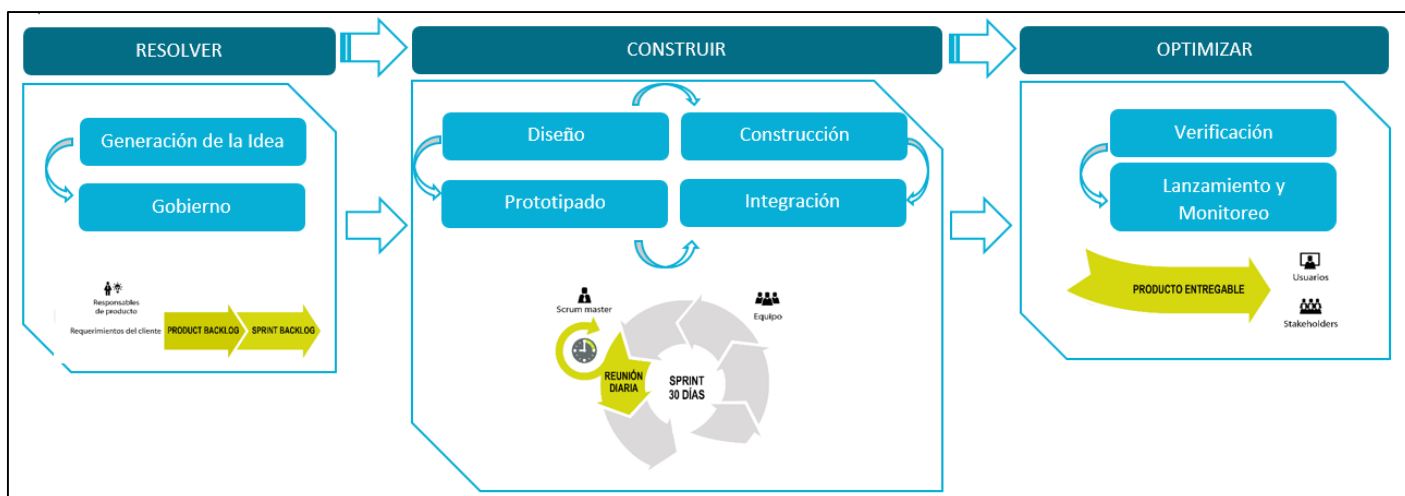
Los objetivos de la guía metodológica se detallan a continuación:

- Proporcionar al equipo de trabajo, los pasos metodológicos para la creación de la aplicación móvil.
- Contribuir con una solución integral desde la concepción hasta la liberación de la aplicación móvil.
- Fomentar el ciclo de desarrollo seguro y la seguridad de datos, para desarrollar aplicaciones móviles robustas.

#### 4.5 Características

La guía metodológica incluye buenas prácticas de gestión, planificación, y desarrollo de proyectos móviles, sobre un conjunto de fases y pasos bien detallados. Sus principales características son:

- Basada en las mejores prácticas de desarrollo de aplicaciones móviles.
- Centrada en el desarrollo ágil de software, por consiguiente, es continua, iterativa e incremental. ver **Figura 27**.
- Orientada a mejorar el proceso de desarrollo de aplicaciones, brindando flexibilidad y escalabilidad.
- Diseño centrado en el usuario, a través de participación del cliente y la creación de prototipos.
- Enfocada en actividades de mejora la experiencia de usuario
- Encaminada a crear aplicaciones seguras.
- Fomenta la integración de las áreas seguridad, desarrollo y operaciones



**Figura 27.** Ciclo iterativo de desarrollo.

#### 4.6 Alcance

El alcance de la guía abarca el desarrollo de aplicaciones móviles de tipo transaccional, abarcando aquellos procesos que facilitan tareas, atienden una necesidad, optimizan procesos internos o permiten acceder a servicios del ámbito financiero, comercial (m-commerce), empresarial (m-business), académico (m-learning), o salud (m-health), No contempla el desarrollo de aplicaciones para el entretenimiento u otros que no realicen procesos transaccionales.

#### 4.7 Público Objetivo

Esta guía es recomendada para los gerentes y líderes de proyecto que se encuentran en etapa de transición de desarrollo convencional de software (escritorio, web) a desarrollo de aplicaciones móviles, así como a start-ups que necesitan un marco de

referencia para empezar con sus estrategias de desarrollo y lanzamiento de aplicaciones móviles. También está dirigida a profesionales y estudiantes.

## **4.8 Controversia**

La presente guía no es una metodología rígida, una fórmula única o una receta para los proyectos. Se constituye como un marco de referencia de buenas prácticas adaptables a las necesidades y cuyo aprendizaje obtenido permitirá mejorar la gestión de desarrollo de aplicaciones móviles.

### **4.8.1 Mejores prácticas de desarrollo**

Bajo un enfoque de selección empírico, se identificó un conjunto de mejores prácticas que podrían considerarse como una solución integral frente al desarrollo móvil. Estas mejores prácticas impactan directamente en todas las fases de la guía propuesta, y su adopción proporcionará información valiosa para el desarrollo de las aplicaciones. Las mejores prácticas propuestas permiten que los proyectos de desarrollo de aplicaciones móviles transaccionales se alineen al entorno tecnológico móvil actual para alcanzar mejores resultados. Las buenas prácticas son obtenidas a partir de los métodos de desarrollo seleccionados para el desarrollo de software. móviles En la Figura 28 se observa los métodos utilizados en la presente guía. Mientras que en la Figura 29 se observa las principales prácticas seleccionadas y usadas en la presente guía.

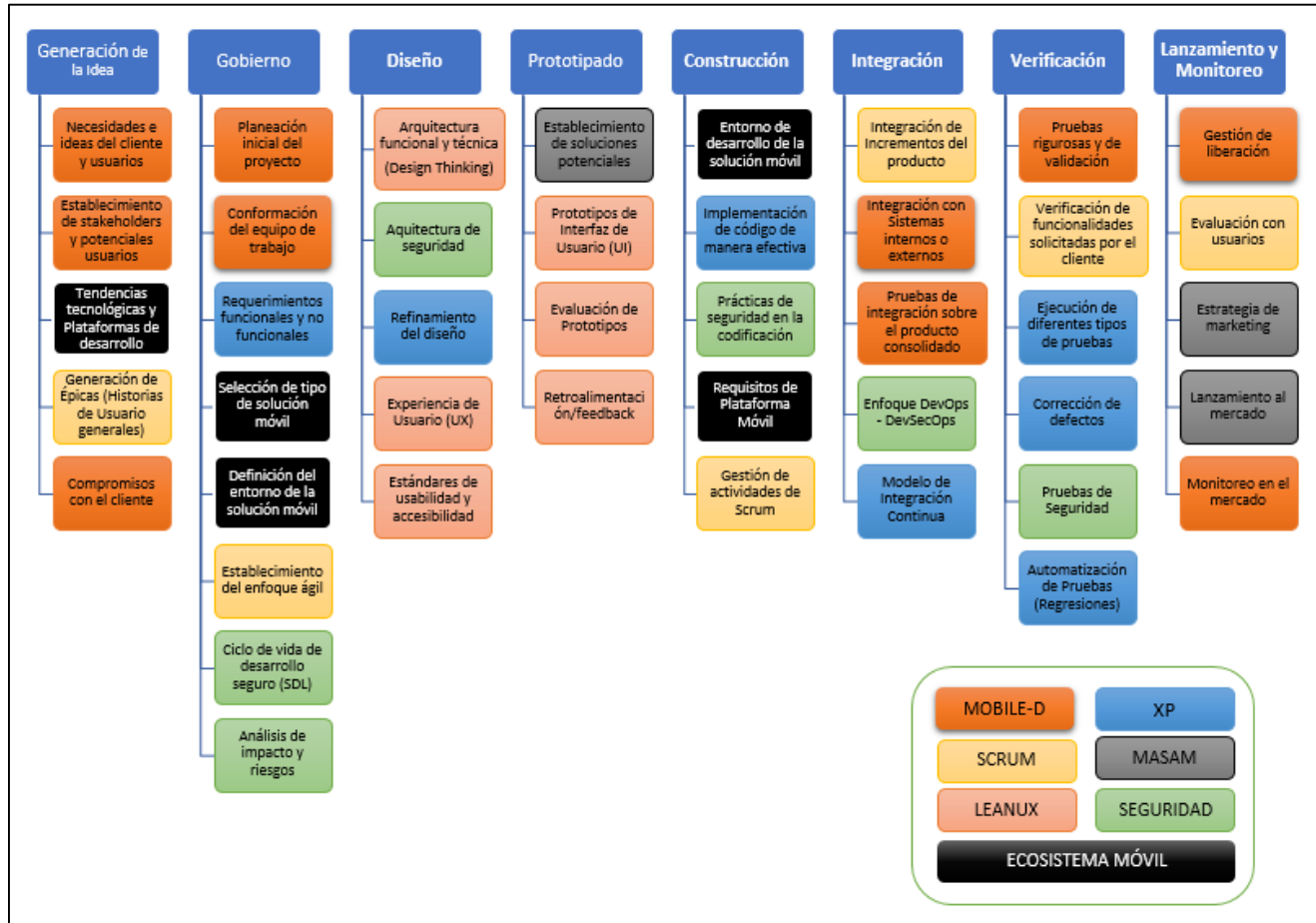
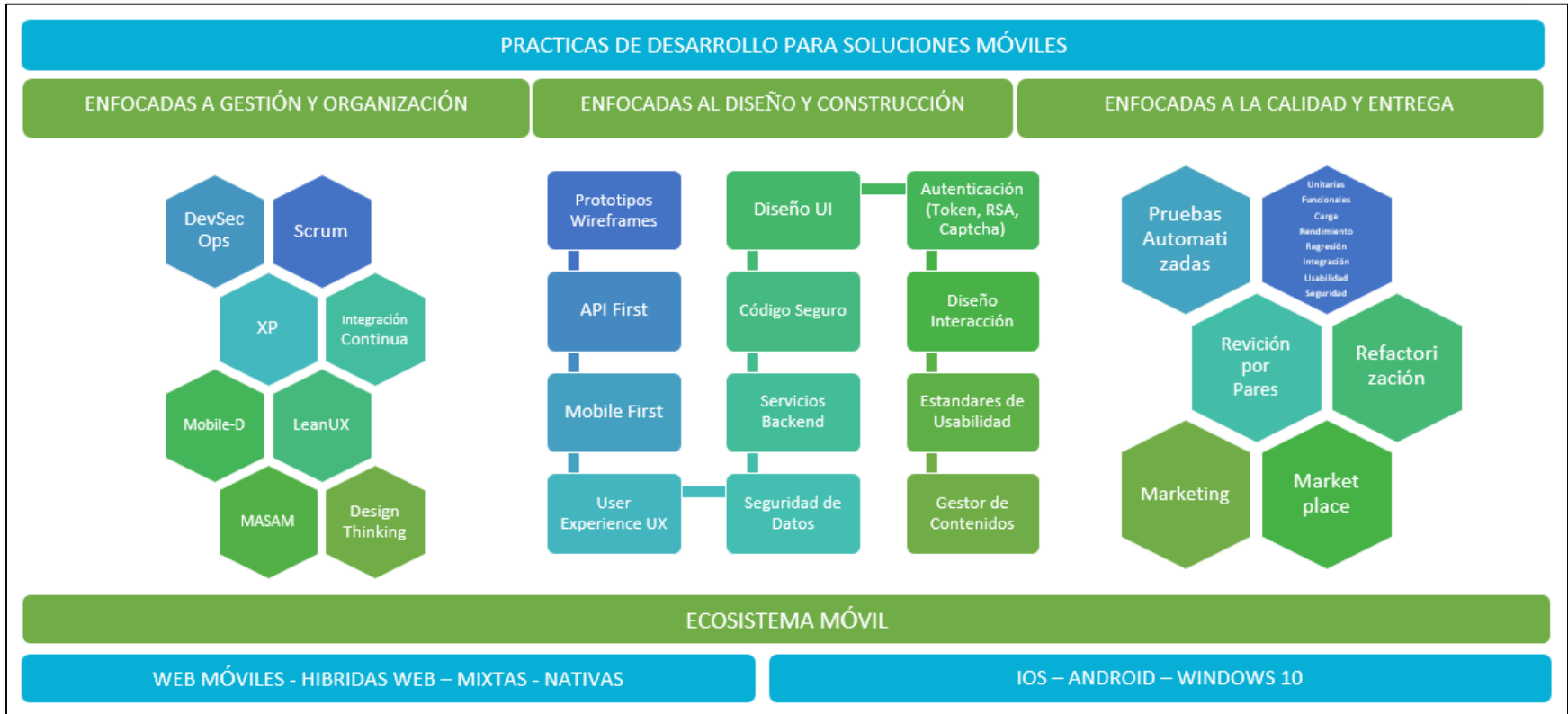


Figura 28. Métodos usados en la guía metodológica



**Figura 29.** Prácticas para el desarrollo de soluciones móviles transaccionales

## **4.9 Descripción de los pasos de la guía**

A continuación, se detallan las fases y pasos de la guía propuesta.

### **4.12.1 Fase I. Generación de la Idea**

Se identifica las necesidades del cliente a través de una recopilación de ideas para posteriormente contrastarlas con la realidad tecnológica actual. Se busca obtener una visión y una propuesta de valor que obtendrá del producto, a través de un análisis preliminar del entorno móvil. Resulta primordial realizar una revisión bibliográfica referente a las tendencias móviles, ya que con la información obtenida se podrá orientar adecuadamente el proyecto. Esta fase está compuesta por cinco pasos que se describen a continuación.

#### **Paso 1. Comprender necesidades e ideas del cliente y usuarios.**

##### **Objetivo:**

Identificar necesidades y requerimientos del mercado en base a las tendencias tecnológicas, y contrastarlas con las ideas del cliente.

##### **Descripción:**

En esta actividad el equipo obtiene la mayor cantidad de información posible y necesaria para el proyecto a realizar, tanto acerca del cliente como de los usuarios. A través de una primera sesión de lluvia de ideas, se valida la idea inicial de la aplicación.



La fuente de información primordial son las personas y lo que experimentan en relación con el producto, servicio o la operación del negocio. Para ello resulta necesario experimentar de primera mano lo que viven dentro de su propio entorno para poder entender de mejor manera sus necesidades, problemas y deseos.

En la definición del problema se determina cuáles son las áreas de oportunidad, especificando cuáles son los problemas o desafíos que se resolverán y permitirán encontrar soluciones innovadoras.

### **Artefactos y Técnicas:**

- Técnicas de investigación (Entrevistas y Encuestas)
- Tormenta de ideas (Brainstorming).
- Exploración del mercado.

### **Entregables:**

- Entrevistas y encuestas realizadas a los clientes.
- Documento obtenido al aplicar la técnica de “Tormenta de ideas”.
- Documento obtenido al aplicar de realizar la técnica de “Exploración del mercado”.

### **Ejemplo:**

**Formato de Tormenta de ideas.** El formato varía en función del negocio del cliente, pero principalmente contiene una lista con todas las ideas de negocio que se le ocurra al cliente, luego se identifica y prioriza las ideas que mejor respondan a las necesidades de los usuarios.

## **Paso 2. Establecer los stakeholders y potenciales usuarios.**

### **Objetivo:**

Definir los grupos de interés relevantes, teniendo claro que los stakeholders no son solo los clientes potenciales, sino también forman parte los propios empleados, los medios de comunicación y la competencia.

### **Descripción:**

Se definen los grupos de interés relevantes. Para que se garantice la aceptación del servicio se deben analizar aspectos de la cotidianidad del cliente como costumbres, preferencias o particularidades del usuario. Una de las primeras cosas a considerar para crear una aplicación móvil son los hábitos y el comportamiento de los usuarios objetivo. Una vez comprendido cómo interactúan con el negocio y los productos o servicios que va a ofrecer la aplicación, se crea una aplicación que permite resolver necesidades de manera eficiente.

Se establecen los medios y estrategias de comunicación entre los clientes, stakeholders y los miembros del equipo. Investigar, observar el entorno, a las personas que están alrededor y las dificultades que encuentran los usuarios en su vida cotidiana. Esto permitirá dar forma a las ideas o descubrir otras nuevas y conocer si el proyecto es viable.

### **Artefactos y Técnicas:**

- Técnicas de investigación (Entrevistas y Encuestas)
- Herramientas que proporcionen información sobre los tipos de usuarios móviles visitan la web. (Google Analytics).

**Entregables:**

- Lista de usuarios potenciales
- Entrevistas y Encuestas.

**Paso 3. Identificar tendencias tecnológicas y Plataformas de desarrollo****Objetivo:**

Obtener una visión y una propuesta de valor del producto, a través de un análisis preliminar del entorno móvil. Resulta primordial realizar una revisión bibliográfica referente a las tendencias móviles, ya que con la información obtenida se podrá orientar adecuadamente el proyecto.

**Descripción:**

Se utilizan diversas herramientas para identificar tendencias tecnológicas. La obtención de información del mundo real se realiza a través de investigación en internet, consultoras como Gartner, o a través de redes de apoyo.

Una vez analizado el panorama tecnológico se asesora al cliente antes de abordar el proyecto, sobre todo si el cliente se enfrenta por primera vez a un desarrollo móvil. Se

contextualiza al cliente sobre las tecnologías líderes del mercado y las implicaciones que existen al desarrollar sobre los diferentes tipos de tecnologías móviles.

**Artefactos y Técnicas:**

- Herramientas de búsqueda de información
- Técnicas de investigación (Entrevistas y Encuestas)
- Redes de contactos
- Herramienta de gestión Benchmarking

**Entregables:**

- Informe sobre plataformas de desarrollo.
- Informe sobre tendencias tecnológicas
- Entrevistas y Encuestas

**Paso 4. Generar las Épicas (Historias de Usuario generales)****Objetivo:**

Las Épicas son historias de usuario generales con todos los interesados en el proyecto.

**Descripción:**

Se genera las Épicas o historias de usuario indefinidas que serán divididas en historias más pequeñas en las siguientes etapas. Una Épica puede ser vista como una historia de usuario de alta complejidad (petición de negocio de alto nivel y complejidad, no clara)

La creación de épicas se realiza a través de la técnica “lluvia de ideas” con las personas involucradas en el proyecto, de tal manera que se obtenga la primera versión de las Historias de usuario.

**Artefactos y Técnicas:**

- Herramienta de Gestión de Proyectos (Jira, Confluence, etc)
- Tormenta de ideas (Brainstorming).

**Entregables:**

- Lista de requisitos generales o épicas agrupadas por funcionalidad y complejidad

**Paso 5. Establecer compromisos con el cliente****Objetivo:**

Negociar términos, condiciones y firma del contrato.

**Descripción:**

Se realiza un conjunto de dinámicas (Agile Inception) para orientar a todas las personas involucradas, reducir las incertidumbres, y explicar los riesgos más evidentes. El resultado esperado determina si el proyecto es viable o de alto riesgo.

El cliente se involucra en todas las fases del desarrollo, garantizando, por un lado, que el producto es lo que el cliente ha pedido y, por otro, satisfaciendo las necesidades de los usuarios potenciales.

Se negocian los términos, condiciones y firma del contrato, tomando en cuenta que esta parte puede o no ejecutarse en esta fase, toda dependerá del cliente y las características del proyecto.

**Artefactos y Técnicas:**

- Técnica Agile Inception.

**Entregables:**

- Contrato firmado

**4.12.2 Fase II. Gobierno**

El Gobierno está estrechamente relacionado con el análisis de solución móvil a desarrollar. El análisis es la clave para un adecuada gestión y construcción del proyecto móvil. Se ajustan las suposiciones de la fase anterior, se define el alcance, presupuesto, cronograma, se identifican las funcionalidades, se especifican las historias de usuario, se identifican riesgos y los problemas críticos. Esta fase está compuesta por ocho pasos que se describen a continuación.

**Paso 1. Realizar la planeación inicial del proyecto****Objetivo:**

Definir las estrategias y actividades para la gestión del proyecto.

**Descripción:**

Se coordina el proceso de desarrollo y se gestiona los recursos del proyecto. No es recomendable apretar las estimaciones para acortar el proyecto, ya que puede conducir al fracaso. Se fijan indicadores (KPI), métricas y objetivos para asegurar que los requisitos, diseño, código, pruebas y otras tareas vayan coordinadas y alineadas. Se elabora un plan de mejora de procesos que fomenta temas de visibilidad, escalabilidad, capacidad, lecciones aprendidas, reuniones, documentación y comunicación.

Se utiliza herramientas ágiles de Gestión de Proyectos como JIRA (issues y project tracking), para planificar, supervisar y gestionar el desarrollo de aplicaciones. Se utilizan herramientas colaborativas como Confluence para crear, organizar, hablar con el equipo y gestionar el conocimiento. Estas herramientas nos permiten tener una Base de conocimiento a medida que van creciendo los requerimientos y necesidades del proyecto.

Se define el plan estratégico, alcance, cronograma, y presupuesto en base a las siguientes consideraciones.

**Plan estratégico.** Se establece un plan a través de selección de software, análisis de procesos de negocio, investigación de tendencias competitivas, políticas de Gobierno, estrategias de Marketing, compromiso digital, estrategia de gestión de contenido (estático o dinámico), arquitectura, hojas de ruta, y planes de personalización. Se definen estrategias ganadoras, construyéndolas a partir de talleres colaborativos y sesiones de observación de usuarios, con el fin de construir un plan ejecutable adaptado al negocio de la empresa.

**Alcance.** Se establece un alcance claro delimitando los casos de uso, esquemas y simulaciones, para ello se debe dividir y agrupar características en fases y establecer reglas de negocio. El alcance debe estar alineado con un tiempo fijo y presupuesto fijo.

**Cronograma.** Se realiza el cronograma estableciendo hitos en una línea de tiempo, para las entregas iterativas, integración con dependencias externas y fases finales de lanzamiento. Considerando tiempos para la versión completa del producto, el candidato de liberación y el candidato de lanzamiento.

**Presupuesto.** Se establece el presupuesto en función de los recursos estimados, el tiempo necesario para la ejecución del proyecto y las suposiciones en términos de dependencias externas e internas. Se realiza presupuestos y estimaciones razonables para obtener un retorno de la inversión y para evitar que se prolonguen plazos por estimaciones poco realista. Se utilizan técnicas de estimación y creación de presupuesto, que incluye el monitoreo de la aplicación.

**Marco ágil.** Se establece al enfoque ágil como el marco de gestión del proyecto. Un enfoque incremental y ágil es la mejor práctica para el desarrollo de aplicaciones para móviles, debido a la naturaleza del software móvil y los desafíos únicos presentados. A continuación, algunas consideraciones del porque se usa el enfoque ágil.

El enfoque ágil permite que los proyectos de desarrollo de aplicaciones móviles se puedan completar de manera eficaz y eficiente. Se centra en la participación del cliente, la planificación flexible, la evaluación constante y la gestión del riesgo. La comunicación regular con el cliente permitirá asegurar que el producto final sea exactamente lo que se



espera y necesita. Las prácticas de desarrollo ágil simplificarán el desarrollo de aplicaciones móviles de modo que las aplicaciones móviles resultantes sean adaptables después del lanzamiento.

El flujo de enfoque ágil se basa en sprints, con entrega de una pieza funcional después de cada iteración y reuniones stand-up diarias para revisar el progreso. Cada sprint tiene un alcance definido de características y casos de uso. El cliente puede cambiar cualquier entregable en el ámbito de la próxima iteración, siempre y cuando no afecte los recursos disponibles para los futuros sprints o los márgenes generales del proyecto. Los cambios pueden incluir ajustes de la interfaz de usuario, funcionalidad de reglas de negocio, una nueva función que reemplaza una característica definida como parte del ámbito o un cambio en una API de backend.

Al final de la etapa se debe documentar y tener claro el flujo ágil seleccionado en un lugar visible para todo el equipo de trabajo.

### **Artefactos y Técnicas:**

- Reuniones, entrevistas, observación.
- Herramienta de Gestión de proyectos (JIRA)
- Herramientas de Gestión del conocimiento (Confluence)
- Talleres colaborativos
- Técnicas de estimación y creación de presupuesto

### **Entregables:**

- Plan estratégico del proyecto (alcance, presupuesto, tiempo)

- Documento con lineamientos de desarrollo ágil

## **Paso 2. Gestionar las actividades de Scrum**

### **Objetivo:**

Establecer el flujo de actividades de Scrum.

### **Descripción:**

Se realiza un ciclo iterativo de desarrollo a través de sprints, considerando las pruebas como parte del proceso de desarrollo. Para cada iteración se realiza el siguiente flujo.

Se estima la velocidad y capacidad del equipo, se analizan los requerimientos, priorizan tareas, se planea la iteración en función a las historias de usuario, se define y revisa las pruebas de aceptación, se fijan las metas del sprint, se crea el sprint backlog, se ejecutan las tareas de desarrollo, se ejecutan las pruebas, integrar la iteración a la rama principal, se realizan las reuniones de control (planning, daily meeting, release planning, retrospective) y se proporciona retroalimentación tomando en cuenta la anterior iteración.

Durante todo el proceso de construcción se toma en cuenta el flujo de la aplicación, wireframes y/o prototipos UI, diseños elaborados por arquitectura, arquitectura de seguridad, integración con API y arquitectura de microservicios, gestión del contenido, desarrollo basado en pruebas unitarias, pruebas de funcionamiento y la creación del producto enfocado a servicios e integración continua.

- Reuniones, artefactos, roles de Scrum

**Entregables:**

- Demo del producto

**Paso 3. Seleccionar los miembros del equipo de trabajo****Objetivo:**

Seleccionar el talento humano con las destrezas y experiencia necesaria.

**Descripción:**

Se selecciona un equipo de trabajo multidisciplinario, estableciendo ubicación y nivel de habilidades de desarrollo para asignar el trabajo apropiado a la persona indicada. Los equipos seleccionados tienen sesiones de capacitación y coaching dependiendo de sus funciones. Es recomendable que los equipos utilicen herramientas de gestión del tiempo para optimizar sus tareas.

**Artefactos y Técnicas:**

- Herramientas de Gestión del talento humano (ERP)
- Capacitación y coaching
- Herramientas de Gestión del tiempo (Timesheet)

**Entregables:**

- Lista de Miembros del equipo de trabajo.

## **Paso 4. Definir y validar los requerimientos**

### **Objetivo:**

Definir los requerimientos básicos e implícitos tanto funcionales como no funcionales.

### **Descripción:**

Definir y validar los requerimientos del proyecto, tomando en cuenta ciertas consideraciones.

**Definición de Requerimientos.** El éxito de cualquier sistema de software radica en el correcto levantamiento de requerimientos, para poder darles seguimiento durante cada fase de desarrollo. Se trabaja cerca del cliente no solo para escuchar los requerimientos, sino para comprender porque existen y plantear opciones que permitan ajustarlos a la realidad. Se analizan los requerimientos en base al alcance de la aplicación, para seleccionar los dispositivos móviles de ejecución que mejor se adapten a las necesidades.

Se identifican los requisitos en base a rendimiento, sistema, arquitectura y diseño a prueba de fallos, Los requerimientos son priorizados utilizando la regla de 80/20, es decir, el 80% del tiempo deberá ser empleado en el desarrollo del 20% de las características con mayor prioridad para el cliente. Los requerimientos deberán incluir pruebas de aceptación las cuales son escritas y verificadas por el cliente.

El Product Backlog y las historias de usuario son creados en base a los criterios de aceptación (datos, usabilidad, contexto de uso, funcionales) y los requerimientos. Se reúnen todas las partes involucradas como clientes, responsables de empresa y líderes, para documentar los requisitos de forma clara y efectiva, alineándolos a los objetivos empresariales y evitando la improvisación. También se considera explícitamente la seguridad durante el procesamiento de captura de requisitos de software.

En la definición de los requisitos funcionales y no funcionales, se toma en cuenta ciertas consideraciones:

Los requerimientos funcionales describen lo que debe hacer la aplicación, como funciones, entradas, salidas, o excepciones. Se definen claramente las tareas que va a realizar la aplicación mediante escenarios de casos de uso. Los requerimientos funcionales se plasman en diagramas de casos de uso para comprender el funcionamiento deseado de la aplicación móvil y determinar sus características.

Los requerimientos no funcionales se crean partir de:

- Rendimiento del sistema, desempeño de la aplicación, portabilidad, fiabilidad, tiempo de respuesta, disponibilidad, manejo de los recursos, consumo de energía, seguridad, consumo de datos, tamaño del almacenamiento, etc.
- Interfaces, dispositivos de E/S, usabilidad, interoperabilidad, interacción, etc.
- Proceso de desarrollo, estándares, herramientas, plazo de entrega, tiempo de salida al mercado, costo, etc.

**Validación de requerimientos.** El proceso de validación de requerimientos se realiza tomando en cuenta los siguientes parámetros:

- Validez, tener puntos de vista distintos de diferentes usuarios.
- Consistencia, no debe haber contradicciones entre requerimientos.
- Completitud, en un desarrollo iterativo deben estar disponibles los requisitos de la iteración en curso.
- Realismo: se pueden implementar con la tecnología actual.
- Verificabilidad: comprobar el cumplimiento de cada requisito.

**Artefactos y Técnicas:**

- Herramienta de gestión de proyectos (JIRA)
- Sesiones de observación de usuarios

**Entregables:**

- Lista priorizada de requerimientos funcionales y no funcionales

**Paso 5. Definir el tipo de solución a desarrollar**

**Objetivo:**

Seleccionar el tipo de tecnología de desarrollo, híbrida(multiplataforma) o nativa tomando en cuenta sus ventajas y desventajas.

**Descripción:**

Se selecciona el tipo de aplicación móvil híbrido o nativo a través de ciertas consideraciones.

Un enfoque de desarrollo híbrido afecta en tiempo el lanzamiento de la aplicación sobre las tiendas de aplicaciones oficiales. En Android, las aplicaciones nativas generalmente tardan 24 horas en ser aprobadas, mientras las aplicaciones híbridas pueden tomar entre 3 días hasta una semana. A pesar de que Apple y Google publican una lista de requisitos sobre la aplicación para ser publicada, la mayoría de las aplicaciones híbridas, quedan estancadas en la aprobación de la publicación debido a los rechazos por cambios y posteriormente por ser vueltos a presentar.

La inclusión de nuevas librerías como por ejemplo WebGL (API multiplataforma destinada a crear gráficos 3D en un navegador web), puede ocasionar problemas de rendimiento y seguridad para los sitios web móviles y las aplicaciones híbridas.

Las plataformas de aplicaciones híbridas son menos seguras contra la ingeniería inversa y la modificación de aplicaciones, y pueden ser tan vulnerables como los sitios web. Se debe tener cuidado al escribir el código, y poner énfasis en el almacenamiento de datos dentro de la aplicación. Como solución existen complementos que pueden ayudar a proporcionar protección a la aplicación.

Existen soluciones como Checkmarx (herramienta de análisis de código estático) que permiten escanear aplicaciones construidas con Apache Cordova para detectar las principales amenazas móviles de OWASP, y detectar vulnerabilidades en el código. Las

aplicaciones híbridas diseñadas con Ionic, Titanium, o Cordova, se acercan a la apariencia de una aplicación nativa real, aunque con pequeña diferencia en la UI / UX.

Las aplicaciones híbridas son de tipo “responsive”, puedan adaptarse de manera fluida a las dimensiones de cualquier dispositivo, integran notificaciones web, tienen un diseño con tolerancia a fallos y que están optimizadas para SEO (Posicionamiento en buscadores)

En base a estas consideraciones entre nativo e híbrido el equipo decide cuál se adapta al tipo de proyecto que pretende desarrollar en base a los requisitos del cliente.

#### **Artefactos y Técnicas:**

- Documentación técnica referente a tipos de aplicaciones móviles.

#### **Entregables:**

- Tipo de tecnología seleccionada.

### **Paso 6. Definir el entorno de la solución**

#### **Objetivo:**

Definir características del entorno de la solución móvil.

#### **Descripción:**

Se seleccionan los diferentes dispositivos móviles a utilizar para el desarrollo y ejecución de la aplicación móvil. Los dispositivos incluyen teléfonos, tablets, tv, laptop,



etc. La rápida expansión de teléfonos inteligentes y su fragmentación ha hecho que se constituya un gran reto la selección de los dispositivos sobre los que se va a desarrollar. Para una correcta elección primero se debe analizar el tráfico existente y luego evaluar a una audiencia en términos de preferencia de dispositivo. Considerar si el desarrollo de la aplicación involucra la integración con sistemas backend y si hay una infraestructura preexistente. Si es así considerar la capacidad de sistemas existentes para manejar transacciones y tráfico. Comprender qué tipos de funcionalidad del dispositivo se requieren. Decidir en qué modalidad trabajará la aplicación móvil dependiendo del tipo de operaciones que va a realizar, la modalidad puede ser online, offline o ambas. Para el sector empresarial la mejor opción siempre será una modalidad combinada.

**Artefactos y Técnicas:**

- Datos sobre dispositivos más usados en el mercado

**Entregables:**

- Lista de dispositivos seleccionados

**Paso 7. Planificar un ciclo de vida de desarrollo seguro (SDL)****Objetivo:**

Construir un plan de seguridad para asegurar ciclo de vida de desarrollo de aplicaciones y así reducir el riesgo frente a ataques potenciales.

**Descripción:**

Se construye un plan de seguridad compatible con la cultura y tecnología del proyecto. El plan se construye teniendo en mente las siguientes consideraciones.

El plan incluye pequeños ciclos incrementales que entregan ganancias al aseguramiento de software. Se crea una hoja de ruta para agregar elementos de seguridad al ciclo de vida de desarrollo y reducir el número de errores de seguridad. El software seguro es diseñado con fallos en mente, construido y probado para ser seguro, para que se ejecute correctamente ante ataques.

Los lineamientos referentes a las actividades de seguridad deben ser específicos, y los pasos en la construcción y medición del programa de aseguramiento deben estar bien definidos, ser medibles y simples. El equipo comprende la importancia de la seguridad, la capacidad fuera de línea y la interoperabilidad. Se define políticas para asegurar la privacidad de los desarrolladores y el secreto de la información. Se formulan requisitos de seguridad especificando el comportamiento esperado del software con respecto a la seguridad y aplicando un modelo de madurez, ver Anexo 2, para el aseguramiento de software (SAMM) que permita implementar una estrategia de seguridad en base a las necesidades de la organización.

**Artefactos y Técnicas:**

- Modelo de Madurez para el aseguramiento del software

**Entregables:**

- Marco de Seguridad

## **Paso 8. Realizar un análisis de impacto y riesgos**

### **Objetivo:**

Definir una estrategia móvil de menor impacto, que permita tener el control y seguimiento del proyecto.

### **Descripción:**

Se realiza una evaluación de amenazas centradas en la identificación y comprensión de riesgos a nivel de proyecto, basándose en la funcionalidad de la solución móvil y las características del entorno de ejecución. Se establece mini-hitos de seguimiento, control para mejorar la gestión de riesgos y mitigar incidencias de manera más controlada. Se realiza un análisis cualitativo para priorizar, determinar el impacto, probabilidad de ocurrencia y probabilidad de detección del riesgo.

Los miembros del equipo fijan hitos para cumplir plazos y reducir los retrasos que pueden surgir por tareas dependientes.

La definición de la estrategia móvil con menor impacto se realiza respondiendo las siguientes interrogantes:

- ¿Qué tan claros son los requisitos del proyecto?
- ¿Qué tan disperso está el equipo?
- ¿Qué tan grande es el proyecto?
- ¿Quiénes son sus usuarios finales?

**Artefactos y Técnicas:**

- Indicadores para el análisis de impacto y riesgos

**Entregables:**

- Matriz de riesgos

**4.12.3 Fase III Diseño**

Permite diseñar la solución móvil. La solución se plasma mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. Se puede realizar en paralelo con la fase de Prototipado para obtener mejores resultados. Esta fase está compuesta por cinco pasos que se describen a continuación.

**Paso 1. Generar la arquitectura funcional y técnica de la aplicación****Objetivo:**

Definir la arquitectura del sistema para obtener una visión del producto.

**Descripción:**

Se define la estructura de la aplicación, detallando un diseño de alto nivel con dos propósitos: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo.

El arquitecto de aplicaciones garantiza la elección apropiada de la arquitectura, tomando en cuenta requerimientos, limitaciones y restricciones del ecosistema móvil. Se determinan los componentes que van a interactuar con la aplicación, tales como el cliente, servidor y base de datos, también se definen lineamientos tanto para el desarrollo del Backend (servicios web) como para el desarrollo del Frontend (patrón MVC)

El arquitecto aplica toda la experiencia y conocimiento para identificar las amenazas y anti-patrones, para crear un diseño eficiente acorde al tiempo de desarrollo, que simplifique las operaciones de la aplicación y que permita tener un tiempo de respuesta aceptable.

Para el diseño de la arquitectura, se toma en cuenta las siguientes consideraciones propias del ecosistema móvil:

**Acceso a datos:** Seleccionar los servidores de acceso a datos que permitirán realizar búsquedas, actualizar información o mostrar resultados en tiempo real.

**Geoposicionamiento:** Tomar en cuenta el manejo de la información en base a la localización del usuario.

**Pasarela de pago:** Considerar la integración de la aplicación con el pago de las tiendas de apps u otros medios de pago

**Registro de usuarios:** Validar la cantidad y que datos se deben recoger en el registro de usuarios.

**Envío de notificaciones push:** Incluir la posibilidad del envío de mensajes personalizados a los usuarios a través de la aplicación. Identificando el tipo de notificaciones y si es necesario la creación de una interfaz de gestión de notificaciones.

**Integración con otros sistemas:** Considerar la integración de la aplicación con sistemas existentes (gestor de contenidos, bases de datos, gestión de usuarios, envío de notificaciones push, etc.)

**Plataformas:** Tomar en cuenta el número de plataformas donde va a funcionar la aplicación. Como mínimo debe estar disponible para Android y sistemas iOS. En función del país y/o el tipo de usuarios, puede crearse una versión para Windows, Blackberry o incluso para Firefox OS.

**Aplicaciones nativas/híbridas:** Considerar que la aplicación nativa ofrece una mayor calidad y rendimiento con un costo mayor, mientras que las aplicaciones híbridas ofrecen menor rendimiento, pero con un costo inferior.

**Gestión de Capacidad:** Considerar el uso de servidores dedicados que brinden escalabilidad, debido a que el flujo de datos y tráfico pueden aumentar en el futuro. Se puede elegir entre un VPS (Servidor Privado Virtual) o un Servidor Cloud.

**Escalabilidad:** La escalabilidad permitirá que la aplicación pueda soportar miles, millones o miles de millones de conexiones concurrentes, reaccionado ante picos de carga de trabajo. Permitirá mantener la calidad del servicio y su rendimiento, a través de una adecuada gestión de los servidores y recursos.

**Artefactos y Técnicas:**

- Desarrollo de diagramas UML

**Entregables:**

- Diagrama UML de arquitectura de la aplicación (Cliente, Servidor, base de datos)
- Diagrama de Despliegue (UML)

**Paso 2. Definir la arquitectura de seguridad de la aplicación****Objetivo:**

Definir lineamientos de seguridad de la información y mecanismos de autenticación de la aplicación.

**Descripción:**

El arquitecto de seguridad enfoca sus esfuerzos en crear lineamientos de seguridad de información, siendo un proceso de mejora continua que abarca desde la fase de requerimientos hasta la fase de lanzamiento. Para que el riesgo de seguridad de datos sea reducido se debe tomar medidas proactivas con diseño y construcción seguro, e incluir servicios seguros conocidos y con componentes reutilizables. Se desarrolla un modelado ágil de riesgos y puntos de contacto en actividades relativas a la seguridad del desarrollo. Se considera la protección de toda la información, utilizando repositorios privados, software de monitoreo de información, sistemas que encripten la información almacenada y uso de canales de comunicación seguros a través de SSL.

La Autenticación de Usuarios, se enfoca en diseñar aplicaciones seguras con identificación única (single sign-in) e incluye diversos mecanismos de autenticación para diferentes aplicaciones y sitios. Se utiliza técnicas para codificar o cifrar datos sensibles para protegerlos de intrusos y ladrones. Se diseñan políticas de autenticación de usuario para definir qué información es sensible, quién puede tener acceso a esta información y bajo en qué circunstancias. Incluir la autenticación de usuario móvil de dos factores a través de un token de seguridad generado por un dispositivo de hardware dedicado, una aplicación móvil Google Authenticator o un mensaje SMS. Se publican políticas de privacidad para asegurar la información personal de los usuarios.

**Artefactos y Técnicas:**

- Reuniones centrales y de seguimiento
- Matriz de riesgos

**Entregables:**

- Diagrama UML de mecanismos de autenticación
- Lineamientos y políticas de seguridad
- Diagrama UML de arquitectura de seguridad

**Paso 3. Revisión y refinamiento de diseño de arquitectura****Objetivo:**



Evaluar el diseño de arquitectura de la aplicación en busca de problemas de todo tipo, poniendo énfasis en la seguridad para detectar problemas que eviten grandes costos potenciales.

**Descripción:**

Se realiza una inspección detallada de diseños, a nivel de datos y de la aplicación, que conduzcan a una evaluación de diseño y revisión de fallos antes de iniciar la fase de codificación. Se asegura que existan lineamientos de mitigación de riesgos conocidos y mecanismos de seguridad.

Se optimiza el diseño de la aplicación para convertirlo en un diseño único. Las aplicaciones del mercado actual deben poseer un diseño que las convierten en el centro de popularidad y eficacia; esto se consigue a través de la inclusión de controles de interfaz intuitivos y transiciones elegantes. Cuando dos aplicaciones realizan una tarea o función similar, el diferenciador único descansa en el diseño visual y en la experiencia del usuario. Al finalizar este paso se refina las Historias de Usuario.

**Artefactos y Técnicas:**

- Reuniones de seguimiento

**Entregables:**

- Historias de usuario refinadas
- Diseño refinado

## **Paso 4. Definir la Experiencia de Usuario (UX)**

### **Objetivo:**

Diseñar una estrategia de experiencia del usuario

### **Descripción:**

Se diseña la experiencia del usuario final para múltiples dispositivos, teniendo en mente el desarrollo multiplataforma. Para ello se elige estándares web abiertos para el desarrollo de aplicaciones, debido a que el desarrollo nativo a pesar de tener ventajas requiere un esfuerzo más grande para lanzar una aplicación multiplataforma, y el costo en muchos casos no es asumible para la mayoría de las empresas. Se asegura que, en todos los dispositivos, los usuarios pueden completar su tarea con facilidad y eficiencia, y la experiencia se sintonice a las expectativas desde ese dispositivo.

Se considera un diseño completamente intuitivo, que permita crear experiencias digitales únicas y funcionales adaptadas a las necesidades del usuario. Se tiene en mente un concepto creativo y de optimización de procesos de negocio. Se considera el diseño de la arquitectura únicamente para los dispositivos móviles seleccionados. Se crea el Mapa de ruta de usuario (RoadMap).

Los dispositivos móviles no sólo se usan para ver el contenido, sino también para interactuar con ese contenido, se diseña un producto que resuelve necesidades y se ajusta a las capacidades, expectativas y motivaciones del usuario. Los diseñadores consideran ergonomía, gestos, transiciones y, por último, los patrones de interacción específicos para móviles.

Se realiza un diseño de interacción para comprender cómo todas las partes interactúan entre sí, definiendo escenarios para determinar todas las funciones que tendrá la aplicación. El diseño visual de la aplicación se produce en base a los requerimientos capturados, integrando las mejores prácticas para el diseño de interfaces de usuario de la mano de UX. El diseño considera la portabilidad de la aplicación, debiendo ejecutarse en distintas plataformas con la misma funcionalidad. El diseño también permite integrarse en los sistemas TI existentes de las empresas.

Se usan diferentes técnicas de investigación para conocer a fondo a los usuarios finales, ya sea investigación cualitativa o cuantitativa. Al diseñar aplicaciones de tipo web se considera el uso de la filosofía de experiencia de usuario Mobile First, que construye comenzando siempre por los dispositivos con pantallas más pequeñas e las adapta posteriormente a las más grandes. Finalmente, el diseño permite obtener una solución moderna y elegante a la vez.

**Artefactos y Técnicas:**

- Técnicas de investigación (encuesta, entrevista, observación)
- Técnica Mobile First

**Entregables:**

- Estrategias de experiencia del usuario
- Diseños UX
- Diseño de interacción

## **Paso 5. Definir los estándares de usabilidad y accesibilidad**

### **Objetivo:**

Definir los principios de usabilidad para que la aplicación tenga una interacción alta con usuarios.

### **Descripción:**

Se establece la usabilidad y accesibilidad que tendrán al interactuar con el usuario, debido a que la mayoría de los dispositivos modernos móviles emplean pantallas táctiles. Los principios se crean tomando en cuenta los siguientes criterios.

**Acceso multidispositivo.** Capacidad de la aplicación para funcionar y visualizarse sobre las diferentes plataformas móviles.

**Recursos de accesibilidad.** Capacidad de la aplicación para utilizar los servicios del sistema operativo que facilitan la accesibilidad.

**Navegación y orientación.** Valoración de estructura de interfaz de aplicación nativa para brindar comodidad a la navegación a través de un dispositivo móvil.

**Formularios.** Asociación de etiquetas con controles de formularios, como ayudas complementarias al usuario o posibles errores que se pueda cometer

**Imágenes.** Existencia de textos alternativos como el grado de corrección sobre las imágenes

**Uso del sonido.** Los contenidos que utilicen sonidos como métodos de transmisión de información deben proporcionar una alternativa visible.

**Estructura.** Evaluación de dos subcriterios, la correcta estructura de los contenidos textuales presentes mediante encabezados, y la asociación de elementos homogéneos mediante listas. Evaluación de la estructura semántica de documentos y contenidos textuales largos, como manuales de ayuda, documentos mostrados a través de la interfaz o información acerca de la empresa o aplicación. El incumplimiento de estos subcriterios provocará barreras de acceso sobre todo a usuarios ciegos que utilicen ayudas técnicas (lectores de pantalla).

**Color.** Evaluación de tres posibles barreras: el contraste de imágenes, el contraste del texto con el fondo y el uso semántico del color. Los dos primeros subcriterios afectan a las personas con deficiencia visual, y el último a las personas con ceguera del color o que no diferencian los colores, y a quienes tengan alguna discapacidad intelectual.

**Tablas de datos.** Evaluación de marcado de tablas de datos. Un marcado incorrecto impide que los lectores de pantalla utilizados por las personas ciegas interpreten la información y puedan asociar los títulos de las columnas con el contenido.

**Ubicación del foco.** El foco de entrada debe quedar reflejado en la pantalla de forma inequívoca, de tal forma que asegure que hay un modo de operación en la que el indicador de foco del control direccional puede ser localizado visualmente. Las personas con limitaciones de atención o las limitaciones de memoria a corto plazo, o limitaciones en los procesos ejecutivos se benefician por ser capaces de descubrir dónde está situado el foco.

**Multimedia.** Valoración de diferentes aspectos de presentaciones multimedia presentes como recursos. Las barreras pueden provocar dificultades de acceso para personas con diversas discapacidades: intelectual, visual o auditiva.

**Organización de la interfaz.** Hay que asegurar que el usuario evite desplazamientos (uso de scrolling) y pueda percibir las opciones de navegación y el contenido relevante en el primer vistazo inicial de la aplicación. Es importante que el usuario tenga una idea de la aplicación con el primer uso. Al ser las pantallas de los dispositivos limitadas en tamaño, el incumplimiento obliga al usuario a realizar un desplazamiento de la interfaz para navegar, con la dificultad que esto implica en el dispositivo móvil a algunos perfiles.

**Artefactos y Técnicas:**

- Dispositivos móviles de prueba.

**Entregables:**

- Estándares de usabilidad y accesibilidad

#### **4.12.4 Fase IV. Prototipado**

Permite crear el diseño UI (interfaz de usuario) de la aplicación y tangibilizarla bien sea con un prototipo físico o digital. Los resultados de las investigaciones y las ideas se colocan frente a los usuarios, para obtener comentarios, refinarlos y repetir el proceso. Esta fase está compuesta por cuatro pasos que se describen a continuación.

## **Paso1. Establecer soluciones potenciales**

### **Objetivo:**

Establecer los lineamientos para el diseño de interfaz de usuario de la aplicación.

### **Descripción:**

Se define los lineamientos para manejar diferentes diseños de aplicaciones evitando malgastar tiempo generando código, y evitando invertir demasiado esfuerzo y dinero. Antes de comenzar a diagramar se determina cómo los usuarios van a interactuar con la aplicación. La interfaz de usuario determinará la sensación de la aplicación y permitirán mostrarle al cliente como se verá exactamente. Se toman en cuenta los pequeños detalles como decidir cómo se verán los botones en la página principal de la aplicación ya que pueden marcar una gran diferencia. Para el tipo de desarrollo híbrido se debe considerar un diseño responsive es decir se debe diagramar las diferentes pantallas en al menos tres vistas: PC, tablet y smathphone. Para el diseño UI, se toma en cuenta las siguientes prácticas generadoras de valor:

**Design Thinking.** Esta técnica permite generar ideas innovadoras al centrarse en entender y dar solución a las necesidades reales de los usuarios. El enfoque utiliza la sensibilidad del diseñador y su método de resolución de problemas para satisfacer las necesidades de las personas de una forma que sea tecnológicamente factible y comercialmente viable.

**API FIRST.** Esta técnica permite desde un principio diseñar una API (interfaces de desarrollo de aplicaciones) que dará soporte a todos los posibles clientes que necesiten

consumir el negocio base del sistema. Se debe realizar una construcción basada en APIs, diseño de bus de servicios (ESB), consumo de servicios de terceros, desarrollo de servicios REST API, desarrollo de servicios Web SOAP y Diseño de arquitecturas SOA. Las APIs de almacenamiento en la nube permitirán exponer productos, servicios o funciones de la solución móvil.

**Artefactos y Técnicas:**

- Reuniones, entrevistas
- Herramienta Gestión de Contenidos (Confluence)
- Design Thinking.
- API FIRST

**Entregables:**

- Lineamientos de interfaz de usuario

**Paso 2. Diseñar los prototipos de Interfaz de Usuario (UI)****Objetivo:**

Crear los prototipos UI en base a los lineamientos de las soluciones potenciales.

**Descripción:**



El equipo de Experiencia de Usuario (UX) se encarga de plasmar las soluciones potenciales en diseños UI a través de herramientas de creación de prototipos. El prototipo es útil para validar funcionalidades concretas, observar el aspecto de distintos diseños y realizar tests de usabilidad. Una buena práctica es crear varios tipos de prototipos como sketch, wireframe, mockup y prototipos de un diseño según se requiera para tener una idea cómo será el producto final.

**Artefactos y Técnicas:**

- Lápiz y papel (Paper Prototyping)
- Power Point o herramientas similares
- Herramientas UX (UXPin, Proto.io, Balsamic, etc)

**Entregables:**

- Prototipos construidos.

**Paso 3. Evaluar los Prototipos****Objetivo:**

Presentar los prototipos a los usuarios para obtener retroalimentación en base a su evaluación.

**Descripción:**

El usuario interactúa con el prototipo, brindando retroalimentación que permita mejorar y perfeccionar los diseños. Se realizan pruebas sobre una amplia gama de usuarios para

obtener información más relevante y útil. La evaluación se considera como un proceso repetitivo que permite perfeccionar el modelo y lograr satisfacer al usuario. Los primeros prototipos permiten realizar propuestas de interacción entre los clientes y el equipo.

**Artefactos y Técnicas:**

- Prototipos UI

**Entregables:**

- Prototipos rediseñados

**Paso 4. Solicitar feedback****Objetivo:**

Generar escenarios de prototipado para obtener comentarios, refinar y repetir el proceso.

**Descripción:**

Se establece un conjunto de reglas generales para solicitar comentarios a los usuarios. Se compara varias versiones que logren obtener retroalimentación crítica y más honesta. Se validan los prototipos con las personas adecuadas para obtener información relevante y útil. Se realizan las preguntas correctas, para estar seguro sobre lo que realmente se está probando y para revelar puntos clave que el equipo no puede encontrar. Ser neutral al presentar las ideas, siendo objetivo y flexible al resaltar los puntos negativos y positivos

de la solución. En resumen, se debe encontrar diversas maneras de mejorar a través del aporte de ideas del cliente.

**Artefactos y Técnicas:**

- Prototipos UI
- Reuniones, entrevistas

**Entregables:**

- Retroalimentación de Usuarios, comentarios positivos y negativos
- Lista de características refinadas del producto (Product Backlog)

**4.12.5 Fase V. Construcción**

Una vez diseñada y evaluada la interfaz de usuario se construye el software. En esta fase el código y los recursos de diseño son ensamblados para crear un producto que siga las especificaciones de diseño de la solución móvil. Un equipo de desarrollo multidisciplinario será encargado de construir el producto gradualmente con la ayuda de Scrum; y basándose en el análisis y el diseño para convertir la aplicación en realidad. Esta fase está compuesta por cinco pasos que se describen a continuación.

**Paso 1. Configurar el entorno de desarrollo para el proyecto.****Objetivo:**

Seleccionar las herramientas para el desarrollo de la aplicación.

**Descripción:**

La selección de ambientes de desarrollo y herramientas de gestión del producto son factores cruciales en desarrollo de la aplicación. Se invierte en herramientas que aumentan la productividad y eficiencia del equipo de desarrollo como equipos con hardware moderno, software, plataformas de desarrollo, y herramientas actualizadas.

Para el desarrollo de código se seleccionan los tipos de dispositivos móviles con sus periféricos, lenguajes de desarrollo, herramientas de calidad de código, herramientas de control de versiones (Git), repositorios de artefactos y entornos de desarrollo integrado (IDE). Se usan herramientas de despliegue de software para construir, enviar y ejecutar una aplicación, como Docker, Vagrant o Ansible, ya que son una alternativa de virtualización, requieren de recursos mínimos para la automatización de tareas y realizan despliegues de forma simple.

Se realiza el análisis de la infraestructura necesaria a utilizar como bases de datos, hosting, alojamiento, servidores, capacidad o hardening, con la capacidad para crecer y escalar tan rápido como las aplicaciones móviles transaccionales. Se fomenta un plan de capacitación en el uso de herramientas y conceptos a manejar, poniendo énfasis en la seguridad en aplicaciones móviles. Al final de la etapa el equipo de trabajo se realiza la estimación de tiempos de las Historias de Usuario.

**Artefactos y Técnicas:**

- Herramientas de control de calidad de código (SONAR)

- Repositorios de código (Bitbucket, GIT)
- Repositorio de artefactos (Nexus)

**Entregables:**

- Entorno de desarrollo configurado

**Paso 2. Implementar el código de manera efectiva****Objetivo:**

Codificar la aplicación utilizando lineamientos de calidad

**Descripción:**

Se realiza la gestión efectiva y documentación del código fuente para que estén versionados, y permitan mantener la trazabilidad y reversión controlada del código. La optimización y reutilización del código puede ahorrar mucho esfuerzo y reducir costes. Se codifican módulos pequeños probados unitariamente para que se integran continuamente (Integración, entrega, despliegue continuo).

Se cuenta con equipos suficientemente potentes y capaces a nivel de hardware y de software, para no tener problemas de capacidad a la hora de codificar. Se incorpora feedback de los clientes en el proceso de desarrollo para acelerar la respuesta a posibles errores. Para tener un código de calidad se siguen las siguientes recomendaciones.

**Documentar el código.** Redactar una breve documentación luego de codificar y testear una funcionalidad realizada.

**Utilizar repositorios de fuentes.** Almacenar en una misma ubicación todos los elementos que se utilizarán durante la construcción. Esta estrategia permite acelerar el desarrollo de múltiples cambios en forma simultánea.

**Utilizar repositorios de ejecutables:** Permiten recuperar los artefactos que se van a instalar. Generados por equipos de desarrollo o por motores de construcción

**Codificación en Pares:** Utilizar técnicas de codificación en pares y refactorización para obtener mejores resultados. Planificar revisiones de código en pares para encontrar errores en la aplicación.

**Código Limpio:** Una de las estrategias más importantes en el desarrollo de aplicaciones consiste en tener el código más simple posible, que permita reducir la complejidad, evitar posibles errores, y facilitar su mantenimiento.

**Desarrollo guiado por pruebas (TDD):** El desarrollo de una funcionalidad implica la creación de una prueba que permita verificar su correcto funcionamiento.

**Desarrollo en paralelo:** Brinda la posibilidad de aislar una versión con todas sus fuentes, permitiendo realizar modificaciones a esta sin alterar otras versiones que puedan estar siendo desarrolladas en ese momento.

**Coordinación con UX:** El equipo UX entrega al programador los archivos, documentación e indicaciones necesarias, así como el soporte durante todo el proceso de desarrollo. La comunicación entre diseñador y programador debe ser constante ya que surgen dudas, y errores que hay que corregir.

**Elaborar interfaz de usuario (UI):** Codificar la interfaz de usuario mediante las especificaciones definidas hasta el momento.

**Calidad del código:** Para evitar que el software se oxide se recomienda seguir los principios SOLID y usar patrones de diseño. Utilizar herramientas de evaluación de calidad de código fuente como SONAR, para detectar bugs, code smells y vulnerabilidades de seguridad.

**Artefactos y Técnicas:**

- Herramientas de código de calidad, limpio y seguro (SONAR)

**Entregables:**

- Incrementos del producto
- Wireframes y/o prototipos de UI.
- Lineamientos de Arquitectura

**Paso 3. Aplicar prácticas de seguridad en la codificación**

**Objetivo:**

Codificar teniendo en mente los alineamientos de seguridad establecidos en la etapa de diseño.

**Descripción:**

Se aplica el desarrollo de código seguro, enfocándose en las principales vulnerabilidades de la red y en base a lineamientos de seguridad. Durante el desarrollo

se hacen pruebas de seguridad más completas, precisas y eficientes en forma automatizada. Se realizan revisiones de código integral para identificar vulnerabilidades básicas, problemas de seguridad de alto riesgo o descubrir riesgos específicos de la aplicación y a nivel del lenguaje. Se evita liberar una aplicación insegura o vulnerable, ya es menos productiva porque se tendrá que desarrollar dos veces.

Para que una aplicación web sea más segura, a la hora de codificar se toma en cuenta los 10 riesgos más críticos de seguridad web, según la Organización OWASP (Open Web Application Security Project) Proyecto abierto de seguridad de aplicaciones web, ver Anexo 3. Mientras que para que una aplicación móvil se más segura se toma en cuenta el top 10 de los riesgos más críticos de seguridad móvil, según OWASP Mobile Security, ver Anexo 4. El objetivo de OWASP es clasificar los riesgos de seguridad móviles y proporcionar controles en el desarrollo para reducir su impacto y probabilidad de explotación.

**Artefactos y Técnicas:**

- Informe sobre riesgos más críticos de seguridad móvil y web (OWASP)

**Entregables:**

- Código seguro

**Paso 4. Gestionar los requisitos de Plataforma Móvil****Objetivo:**



Tomar en cuenta las consideraciones de Plataforma Móvil

**Descripción:**

Se toman en cuenta ciertas consideraciones de Plataforma Móvil para llegar preparados a la fase de lanzamiento. Apple y Google poseen ciertas restricciones a nivel de seguridad, es por ello que los servicios que consume la aplicación deben estar expuestos a través de certificados seguros emitidos por entidades certificadoras reconocidas como: Secure GlobalSing, Symantec, Trustwave, Geotrust, etc., caso contrario no brindarían el soporte necesario y no permitirán subir al Marketplace. Se debe tener la última versión de sistema operativo, tomando en cuenta las actualizaciones que se liberan, en el caso de iOS existe una sola versión y para Android existen versiones beta. Se toma en cuenta los tiempos de revisión que tardan las tiendas para el lanzamiento de las aplicaciones móviles, así como la entrega de ambientes y credenciales de acceso para evitar contratiempos y retrasos.

**Artefactos y Técnicas:**

- Lineamientos de Plataforma Móvil

**Entregables:**

- Credenciales de acceso para cada plataforma móvil

**4.12.6 Fase VI. Integración**

Permite integrar la solución móvil con subsistemas o aplicaciones internas o externas. También se integran los incrementos de productos para obtener un producto consolidado. Maneja el concepto de interoperabilidad, ya que las empresas no solo implementan sistemas, sino también los conectan para aprovechar al máximo, la lógica de negocios, funcionalidad, comunicaciones, facilitar la reusabilidad, la integridad de la información y el consumo de servicios. Esta fase está compuesta por cinco pasos que se describen a continuación.

### **Paso 1. Integrar los incrementos de producto**

#### **Objetivo:**

Integrar el incremento de producto de cada iteración con el entregable principal

#### **Descripción:**

Los incrementos de producto son ensamblados para crear un producto consolidado que sigue las especificaciones de diseño. Si se encuentra inconsistencias con los requerimientos del cliente se lleva a cabo las tareas de refactorización. Al finalizar, se realiza pruebas de regresión para asegurar la calidad de la solución móvil.

#### **Artefactos y Técnicas:**

- Pruebas unitarias, integración y regresión
- Refactorización

#### **Entregables:**

- Artefacto final

## **Paso 2. Integrar con sistemas internos o externos**

### **Objetivo:**

Integrar la aplicación con subsistemas backend encargados de exponer servicios del lado del servidor de aplicaciones.

### **Descripción:**

Se integra la solución móvil con aplicaciones empresariales externas o internas tales como gestor de contenidos, bases de datos, gestión de usuarios o envío de notificaciones push. Se asegura que lo que pase en la aplicación móvil impacte al backend y lo que pase en el backend se refleje en la aplicación móvil. Se usa APIs para integrarse a otras aplicaciones de terceros consumiendo los servicios expuestos. Se cuenta con acuerdos de nivel de servicio (SLA) para integrarse con diferentes servicios gratuitos y no gratuitos como Google Maps o Twitter.

### **Artefactos y Técnicas:**

- Acuerdos de nivel de servicio (SLA)
- APIs de integración de servicios

### **Entregables:**

- Fuentes del Producto integrado

### **Paso 3. Realizar pruebas de integración sobre el producto consolidado**

#### **Objetivo:**

Realizar pruebas de integración para verificar la conexión de dependencias entre sistemas y las aplicaciones

#### **Descripción:**

Se realizan pruebas de integración para conectar las dependencias entre sistemas y aplicaciones, utilizando como opción las tecnologías de integración de la nube. Se definen y revisan las pruebas de aceptación de integración. Se repite el proceso de pruebas y se realizan pruebas de regresión hasta que se hayan agregado todos los módulos y el sistema se prueba como un todo. Se considera no lanzar aplicaciones a producción sin pruebas formales de integración, ya que puede conducir a problemas mayores. Es recomendable hacer pruebas de integración con pruebas de carga de forma temprana

#### **Artefactos y Técnicas:**

- Escenarios de prueba

#### **Entregables:**

- Actas de Pruebas

## **Paso 4. Planificar un flujo de trabajo colaborativo a través del uso del enfoque DevSecOps**

### **Objetivo:**

Planificar el uso del enfoque DevOps para promover una cultura centrada en la comunicación, colaboración e integración entre líneas de negocio, desarrolladores de software y operaciones TI.

### **Descripción:**

Se promueve una cultura centrada en la comunicación, colaboración e integración entre líneas de negocio, desarrolladores de software y operaciones TI. Tanto desarrolladores, testers y especialistas en operaciones trabajan en colaboración mediante las herramientas compartidas de computación en la nube para reducir los riesgos de seguridad en el software. La colaboración implica compartir información que puede poner en amenaza a la organización, por tal motivo se deben definir políticas de control de acceso entre los participantes del proyecto.

Se utiliza el enfoque DevSecOps para agregar profesionales de seguridad a los equipos DevOps. Se utiliza tecnología habilitadora para el despliegue de software como Chef o Docker. Impulsar las buenas prácticas de entrega continua, automatización, reutilización y agilidad para agregar valor a todos los departamentos TI. Finalmente, se garantiza que la seguridad no sea una idea tardía de un departamento aislado, sino que se integre en todas las etapas del proyecto.

**Artefactos y Técnicas:**

- Reuniones

**Entregables:**

- Herramientas DevSecOps

**Paso 5. Utilizar el modelo de Integración Continua****Objetivo:**

Automatizar las tareas de despliegue de aplicaciones

**Descripción:**

Se automatiza lo máximo posible todas las acciones necesarias para implantar una nueva versión de la aplicación y todas las tareas necesarias para validar esa nueva versión. Para detectar errores de forma temprana se compila, realiza análisis de código estático y se ejecuta pruebas automáticas de forma periódica. Se implanta en producción con el menor tiempo posible las nuevas funcionalidades garantizado siempre las funcionalidades anteriores.

El despliegue automatizado permite reducir los tiempos de cada ciclo de desarrollo, es decir, reduce el tiempo para que cada uno de los requisitos solicitados por los usuarios finales llegue a sus manos.

**Artefactos y Técnicas:**

- Herramientas de Integración Continua como Jenkins, Badoo o Electric Flow

**Entregables:**

- Artefactos desplegados automáticamente

**4.12.7 Fase VII. Verificación**

Permite realizar pruebas de calidad sobre la solución móvil. Una vez terminado el desarrollo de cada iteración se pasa a la verificación, para obtener una versión estable del producto en base a los requisitos definidos. Se enfoca en los procesos y actividades relacionadas a verificar y probar artefactos producidos en el desarrollo de la solución móvil, incluyendo revisiones y actividades de evaluación. Al llevarse las pruebas durante el proceso de desarrollo, los desarrolladores y testers pueden detectar fácilmente problemas. Los desarrolladores pueden comunicarse fácilmente con las partes interesadas para mostrar su progreso. Esta técnica permitirá asegurar que el proyecto se mantenga en el presupuesto y se logre liberar a tiempo. Esta fase está compuesta por seis pasos que se describen a continuación.

**Paso 1. Realizar pruebas rigurosas y de validación****Objetivo:**

Planificar escenarios para las pruebas de control de calidad

**Descripción:**

Se realiza una planificación de rutas y se establece un conjunto de casos pruebas para validar las funcionalidades desarrolladas. Se realiza un proceso de prueba exhaustivo sobre varios sistemas operativos y a través de una amplia variedad de dispositivos. Las pruebas son parte integral del desarrollo de software y se ejecutan en todo el proceso de codificación del software. Se documenta cada prueba, se informa de manera efectiva los errores, el rastreo de los errores y la corrección de los mismos.

El equipo omite técnicas tradicionales y adopta un enfoque innovador, incluyendo pruebas de público con usuarios para garantizar el éxito de la aplicación en el mundo real. Se cuenta con un equipo combinado de desarrolladores y testers para cumplir mejor con los objetivos de negocio, generando software de calidad en los plazos establecidos del proyecto.

**Artefactos y Técnicas:**

- Reuniones
- Historias de usuario

**Entregables:**

- Casos de prueba
- Planes de rutas

**Paso 2. Verificar que el producto tiene las funcionalidades requeridas por el cliente****Objetivo:**



Verificar que la aplicación cumpla con los requerimientos del cliente y documentar todo el proceso de desarrollo.

**Descripción:**

Se documenta todo el proceso de desarrollo, desde el plan del proyecto, requisitos, especificaciones, diseños, planes de pruebas, informes de las pruebas, informes de estado y la documentación para los usuarios. Se revisa minuciosamente todos los entregables, el código y de la documentación. La clave de las pruebas es la colaboración y cooperación entre los desarrolladores de aplicaciones, testers, el equipo de diseño y los usuarios finales.

Si el software no cumple con los requerimientos definidos, o si existen fallas en tiempo de ejecución, o si el cliente solicita cambios de última hora, debemos regresar a la fase de diseño para solucionar los inconvenientes presentados.

**Artefactos y Técnicas:**

- Reuniones
- Historias de usuario
- Casos de prueba
- requisitos y especificaciones

**Entregables:**

- Requerimientos refinados

### **Paso 3. Ejecutar diferentes tipos de pruebas**

#### **Objetivo:**

Ejecutar las pruebas de calidad en base a los casos establecidos

#### **Descripción:**

Se ejecutan diferentes tipos de pruebas sobre la aplicación, tales como, pruebas unitarias, pruebas de integración, pruebas de funcionalidad, pruebas del sistema y pruebas de rendimiento. Los tipos de pruebas para dispositivos móviles están dirigidos hacia la funcionalidad, compatibilidad, interoperabilidad, localización y la usabilidad de la aplicación. Se utilizan testers y expertos en rendimiento para que se centren en casos más complejos y determinen escenarios de prueba para cubrir áreas fuera del código, como infraestructura o pruebas de terceros. El tester busca situaciones límite para exponer las limitaciones de la implementación del componente, ya sea tratando éste como una caja negra "pruebas de caja negra" o fijándonos en su estructura interna "pruebas de caja blanca".

Para validar la calidad de la aplicación se ejecutan las siguientes pruebas.

**Pruebas unitarias.** Permiten ejecutar inspecciones automatizadas para validar el correcto funcionamiento.

**Pruebas de usabilidad:** Permiten utilizar métodos heurísticos o empíricos para determinar el nivel aceptable de usabilidad. Los métodos heurísticos implican la participación de expertos en usabilidad y los métodos empíricos constan de técnicas e instrumentos que requieren de la participación de usuarios en base a atributos de

usabilidad tales como facilidad de aprendizaje, eficiencia, manejo de errores, presentación visual apropiada y satisfacción.

**Pruebas de accesibilidad.** Permiten determinar la flexibilidad del entorno en cuanto a la presentación, métodos de control, formas de acceso, ayudas y disponibilidad de contenidos.

**Pruebas de rendimiento.** Permiten validar y verificar los atributos de calidad de la aplicación como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento permiten observar el comportamiento de la aplicación bajo una carga de usuarios concurrentes.

**Pruebas de carga.** Permiten validar la carga transaccional de los servicios que se exponen a la aplicación.

**Pruebas de regresión.** Ejecutar pruebas de regresión en el caso de cambios de versión de artefactos o APIs, para aumentar la calidad total.

**Revisión por pares:** Tener el trabajo revisado por otro experto funcional aumenta la credibilidad de los examinadores y la credibilidad de las pruebas, La evaluación de la información permitirá eliminar distorsiones, prejuicios y supuestos.

#### **Artefactos y Técnicas:**

- Casos de prueba

#### **Entregables:**

- Informes de las pruebas, informes de estado y la documentación para los usuarios

#### **Paso 4. Realizar la corrección de defectos**

##### **Objetivo:**

Ajustar la aplicación móvil en base a los defectos detectados

##### **Descripción:**

Se analizan defectos y realizan pruebas de aceptación. Se definen métricas claves para asegurar la calidad y detectar fallos. Al encontrar defectos se regresa a la fase de construcción para solucionarlos, luego si las pruebas son exitosas se empieza una etapa de pruebas con dispositivos reales. Se realizan pruebas a lo largo del proceso en lugar del final permite descubrir defectos en la codificación u otros errores. Durante las pruebas, se pueden encontrar errores que deben ser corregidos o pueden existir mejoras de interfaz de usuario a realizar por el equipo de desarrollado.

##### **Artefactos y Técnicas:**

- Lineamientos UX

##### **Entregables:**

- Aplicación móvil certificada funcionalmente

#### **Paso 5. Realizar Pruebas de Seguridad**

##### **Objetivo:**

Definir el proceso de pruebas de seguridad basándose en la implementación y los requisitos del software

**Descripción:**

Se construye casos de prueba desde los requisitos de seguridad conocidos para validar el código. Se realizan pruebas de intrusión y pruebas de alto nivel en cada publicación del software basándose en la funcionalidad de la aplicación para detectar vulnerabilidades. Se enfocan los esfuerzos en encontrar problemas de seguridad en ambientes de pre y producción para hacer visible las malas configuraciones operacionales o errores en la lógica de negocio que son difíciles de detectar en ambientes de desarrollo. Se automatizan pruebas de seguridad y verificación de correcciones sobre vulnerabilidades detectadas.

**Artefactos y Técnicas:**

- Casos/Rutas de prueba

**Entregables:**

- Actas de pruebas
- Aplicación móvil certificada por seguridad

## **Paso 6. Automatizar las pruebas**

### **Objetivo:**

Realizar pruebas automatizadas para que se ejecuten en dispositivos propios de usuarios

### **Descripción:**

Se usan herramientas automatizadas y procesos contrastados en las fases más tempranas, para identificar y resolver errores con el menor coste posible. Las pruebas se realizan con escenarios simulados y dispositivos móviles emulados, explorando todas las funcionalidades e, ingresando datos correctos o erróneos que permitan medir la robustez de la aplicación. Se realiza pruebas sobre equipos reales para medir el desempeño y el rendimiento de la aplicación. Existen servicios en la nube que proporcionan un elevado número de dispositivos reales que permiten realizar pruebas automatizadas.

### **Artefactos y Técnicas:**

- Servicios en la nube

### **Entregables:**

- Herramientas automatizadas de pruebas

## **4.12.8 Fase VIII. Lanzamiento y Monitoreo**

Permite el lanzamiento y monitoreo de la solución móvil. En esta fase la aplicación es finalmente puesta a disposición de los usuarios, ya sea en forma web o a través de las

tiendas de aplicaciones (“markets”): App Store, Google Play, Window Phone o Firefox Marketplace. Es crucial que no exista prisa para liberar el nuevo producto en el mercado ya que puede ser el principal conductor de inyección de vulnerabilidades de seguridad sobre la aplicación. Esta fase está compuesta por cinco pasos que se describen a continuación

### **Paso 1. Gestionar la liberación de la solución móvil**

#### **Objetivo:**

Coordinar con el cliente el lanzamiento de la aplicación

#### **Descripción:**

Se definen procesos y actividades para gestionar la liberación del producto. Se incluye el envío de productos a usuarios finales, instalación de productos en ambientes internos o externos y el detalle de las operaciones normales a realizar en un ambiente de producción. Se despliega la aplicación de manera eficaz a través de un buen plan de despliegue y una lista a modo de checklist que asegure el proceso.

Se exige evidencias de pruebas de seguridad específicas sobre la aplicación para asegurarse que los lineamientos de seguridad están implementados antes de la publicación. Se determina si el producto es lo suficientemente maduro como para ser liberado en producción, a través de versiones alpha o beta testing.

#### **Artefactos y Técnicas:**

- Reuniones
- Cronograma

**Entregables:**

- Lista de chequeo de despliegue

**Paso 2. Realizar el proceso de evaluación con usuarios****Objetivo:**

Realizar pruebas con un grupo de personas para obtener comentarios

**Descripción:**

Se obtienen analíticas, estadísticas y comentarios de usuarios, para evaluar el comportamiento y desempeño de la aplicación en un ambiente de producción. Generalmente se usa aplicaciones betas antes del lanzamiento. Se integra la documentación que pueda requerir el usuario final, así como se define las estrategias para su publicación y distribución entre el público en general. Al finalizar la etapa se corrige errores, realiza mejoras y se actualiza la aplicación.

**Artefactos y Técnicas:**

- Herramientas de información estadística

**Entregables:**

- Planes de prueba



- Resultado de evaluación de usuarios (comportamientos, sugerencias y comentarios)

### **Paso 3. Ejecutar la estrategia de marketing**

#### **Objetivo:**

Poner en marcha las estrategias de marketing luego del lanzamiento de la aplicación

#### **Descripción:**

Se define el canal de comercialización de la aplicación. Por lo general las empresas destinan el presupuesto en el desarrollo de aplicaciones y no mantienen presupuesto suficiente para la parte de comercialización. Una de las etapas finales es tener una estrategia de marketing. Esta estrategia requiere que se ejecute una planificación y la investigación adicional. Es clave para crear conciencia y tener éxito en aplicación. Se cuenta con un equipo que pueda desarrollar una estrategia de marketing para crear conciencia y lanzar una aplicación.

Existen en el mercado, innumerables canales de comercialización disponibles, algunos de los más populares son: medios de comunicación social, difusión por correo electrónico, compra de medios, redes de afiliados, banner syndication o publicidad de pago por clic.

Se toman en cuenta ciertas estrategias avanzadas de Marketing:

Se realiza el lanzamiento temprano para conocer a los usuarios a través de sus comentarios, pero eso no significa que la aplicación contenga errores, sino se trata de

liberar una versión básica. Para garantizar que los usuarios descarguen la aplicación se utiliza un "Landing Page" que describa de forma breve lo mejor de la aplicación.

Se comercializa la aplicación móvil incluyendo vídeos e imágenes, siendo creativos con el contenido y gráficos para hacer que la marca sea reconocible. Se mejora la búsqueda de la aplicación móvil en los marketplace, colocando las palabras clave correctas, títulos específicos y descripciones bien escritas para que los usuarios puedan encontrarla sin esfuerzo. Finalmente se mantiene una estrategia de contenido lista antes de que se publique la aplicación móvil.

**Artefactos y Técnicas:**

- Estrategia de marketing

**Entregables:**

- Canales de comercialización

**Paso 4. Lanzar la aplicación al mercado****Objetivo:**

Poner la aplicación a disposición de los usuarios en el mercado

**Descripción:**

La aplicación se pone a disposición de los usuarios, por medio de la web o a través de las tiendas de aplicaciones("marketplace") tales como App Store, Google Play, Window

Phone o Firefox Marketplace. Para que una aplicación tenga éxito, necesita exposición en los principales Marketplace del mercado tanto en Google Play como en la App Store, la mejor manera de conseguir exposición es siguiendo las siguientes recomendaciones.

Se presenta o muestra la aplicación como una "Opción del editor". Se distribuyen versiones beta, se recopilan informes de fallos significativos en vivo, se obtiene retroalimentación de usuarios reales y se analiza la cobertura de las pruebas. Se llena la información solicitada en cada apartado de la ficha de los Marketplace ya que en base a su contenido el usuario decidirá si descarga la aplicación.

La aplicación al ser enviada a las tiendas de aplicaciones se involucra en un proceso que puede tardar hasta una semana y, a menudo, requiere realizar cambios sobre la aplicación. Finalmente, cada mercado de aplicaciones es completamente diferente, algunos permiten versiones beta, tienen un promedio de espera que es entre 2-6 semanas de ida y vuelta, o se espera a presentar varias versiones antes de un gran impulso de marketing.

#### **Artefactos y Técnicas:**

- Marketplace

#### **Entregables:**

- Aplicación desplegada en los Marketplace

### **Paso 5. Realizar el monitoreo de la aplicación móvil en el mercado**

#### **Objetivo:**

Monitorear la aplicación para detectar errores o mejoras.

**Descripción:**

Se define un proceso de soporte y mantenimiento que permita informar y reportar errores al equipo de desarrollo. Se analizan patrones de uso, se monitorea y se mantienen las aplicaciones en una evolución continua con valor para el usuario.

Se realiza un monitoreo continuo debido a que las aplicaciones móviles están en constante actualización, con un promedio de una por mes. Los motivos son diversos tales como, actualización de sistema operativo, inclusión de notificaciones push, nuevas librerías, nuevas funcionalidades o por el feedback de los usuarios.

Se realiza una implementación de servicios de Google Analytics para obtener analíticas que permitan optimizar el producto. Se realiza seguimiento de la experiencia del usuario. Toda la información debe ser monitorizada y entregada en informes para asegurar un éxito continuo. Las herramientas de tracking sirven para hacer seguimiento de campañas en diferentes servidores o soportes publicitarios. Se conoce la fuente de tráfico para tomar decisiones de marketing en función de la rentabilidad de las distintas campañas. Un equipo de desarrollo ágil maduro continuará actualizando y mejorando la aplicación bien después de su lanzamiento. Finalmente, los datos de feedback, reportes de errores y otras analíticas generadas permiten mejorar la aplicación.

**Artefactos y Técnicas:**

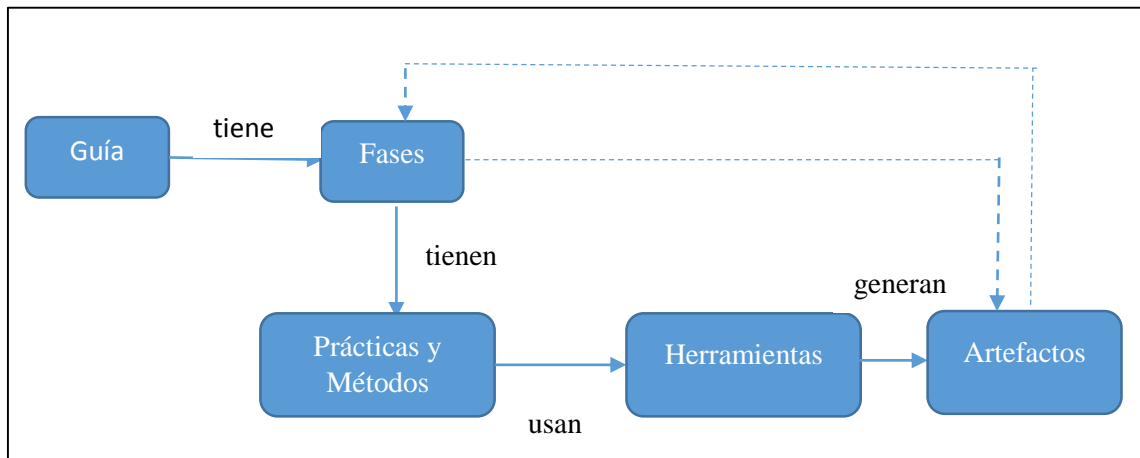
- Herramientas de tracking (Google Analytics, redes sociales)

**Entregables:**

- Reporte de fallos y errores
- Reportes de herramientas de tracking

#### 4.10 Artefactos

Los artefactos son creados o producidos con un propósito específico en cada fase de desarrollo. En la Figura 30 se visualiza el proceso para obtener un artefacto.



**Figura 30.** Obtención de un artefacto

Los pasos de la guía metodológica descritos permiten guiarnos en el desarrollo de una aplicación móvil, pero para obtener mejores resultados se sugiere ejecutar cada paso en el orden establecido.

A continuación, para complementar la guía se detallan los principales roles que intervienen en el proceso de desarrollo de aplicaciones móviles.

#### 4.11 Roles

El equipo de desarrollo está formado por las siguientes personas. Un desarrollador backend encargado de crear los servicios que serán consumidos por la aplicación, desarrollador frontend encargado de crear las pantallas que se comunicaran con los servicios, desarrollador UX encargado de definir la experiencia de usuario, un tester

encargado de verificar la calidad del software, un arquitecto encargado de definir la estructura de la aplicación, un arquitecto de seguridad encargado de definir los lineamientos de seguridad, gerente de proyecto encargado de gestionar las actividades y establecer la comunicación entre el equipo de desarrollo y los demás equipos, además se encarga de asignar la prioridad en las tareas, así como sus responsables. Un Scrum master encargado de gestionar los impedimentos, así como los artefactos y reuniones, un product owner encargado de definir los lineamientos funcionales del negocio con el cliente e interesados. A continuación, se describen los roles antes mencionados:

#### **4.14.1 Diseñador de Experiencia de Usuario (UX)**

Es el encargado de crear los wireframes y/o prototipos con el uso de una amplia gama de herramientas y de diferentes niveles de fidelidad en función de preguntas formuladas, público destinatario y cumpliendo con los requisitos planteados y una interacción definida. Un profesional UX tiene una comprensión profunda de la psicología cognitiva, la interacción persona-ordenador y diseño de investigación. Tienen experiencia en el manejo de entrevistas de usuarios finales, evaluaciones de usabilidad y los estudios etnográficos. Los entregables se pasan al equipo de Desarrollo front-end.

#### **4.14.2 Desarrollador Frontend**

Es la persona que se encarga de construir las interfaces de usuario, dándole comportamientos e interacción, también se conoce como Diseñar Web.

#### **4.14.3 Desarrollador Backend**

Es el encargado de crear los servicios para el correcto funcionamiento de la aplicación, gestionar el envío y recepción de datos, darle robustez, entre otras tareas.

#### **4.14.4 Equipo de Desarrollo (Team)**

El equipo de desarrollo está formado regularmente por 5 personas, uno de ellos es el líder encargado de establecer la comunicación entre el equipo de desarrollo y los demás equipos, se encarga de asignar la prioridad en las tareas, así como sus responsables. El equipo es el encargado de ejecutar las historias de usuario que se comprometieron en el inicio de cada sprint.

#### **4.14.5 Arquitecto**

Es un programador experimentado capaz de visualizar el software como un todo, se encarga en base a los requerimientos del cliente de armar la arquitectura del software.

#### **4.14.6 Scrum master**

Es la persona que guía al equipo para que cumpla con las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y el desperdicio de tiempo.

#### **4.14.7 Analista en Diseño de Soluciones/ Dueño del Producto**



Es el responsable de comprender las necesidades del cliente, mantiene un contacto permanente y se apropia de los requerimientos solicitados. Se encarga de transformar las características funcionales y no funcionales del futuro software en requerimientos entendibles, y representa al equipo como intermediario o mediador ante posibles conflictos. Formaliza los requerimientos en historias y las prioriza de forma regular.

#### **4.14.8 Tester**

Es el encargado de realizar las pruebas necesarias para las características del software, así como documentar el proceso y hacer un seguimiento de errores y correcciones generadas.

#### **4.14.9 Gerente de Proyecto/Responsable de proyecto**

Es el encargado de gestionar el proyecto: el alcance, la calidad, el tiempo, los costos, los riesgos y la satisfacción del cliente (interno y externo).

#### **4.14.10 Arquitecto de Seguridad**

Es el responsable del diseño, desarrollo y adecuación de controles de seguridad informática, típicamente controles de software, como control de acceso, funciones criptográficas, filtros, bitácoras de seguridad de aplicativos, etc.

#### **4.14.11 Interesados/Stakeholders**

Son aquellos que se ven afectados materialmente por el resultado del proyecto.

#### **4.14.12 Cliente**

Son aquellos individuos u organizaciones que están activamente involucrados en el proyecto, o cuyos intereses pueden verse afectados, positiva o negativamente, como resultado de la ejecución y término del proyecto.

#### **4.14.13 Usuario**

Son aquellos que utilizarán directamente el producto del proyecto.

### **4.12 Encuesta**

Las encuestas permitirán validar la guía metodológica para conocer el grado de aceptación. Permitirán conocer si los encuestados en este caso Líderes y Gerentes de proyecto están satisfechos o insatisfechos con el proceso de desarrollo de soluciones móviles propuesto.

La encuesta se realizó a través de la herramienta en línea [www.surveymonkey.com](http://www.surveymonkey.com), se encuentra formada por 22 preguntas, su formato se encuentra en el Anexo 5, y su publicación se encuentra en el siguiente enlace:

- ***Encuesta De Valoración De Una Guía Metodológica Para El Desarrollo De Apps Transaccionales:***

<https://www.surveymonkey.com/survey/d/l3S0S3A1J9A5N9M4H>

La encuesta fue realizada a 10 personas el mes de abril del 2018, entre los encuestados están líderes y gerentes de proyecto de seis empresas de servicios y desarrollo de software de la ciudad de Quito, dos empresas de sector público y cuatro empresas del sector privado, quienes tuvieron la apertura de colaborar en la realización de la misma.

La encuesta contiene dos videos explicativos sobre el proceso de desarrollo de la guía metodológica. Los videos fueron creados con la herramienta en línea PowToon, y se encuentran publicados en los siguientes enlaces:

- ***Guía Metodológica para el desarrollo de apps:***

[https://www.youtube.com/watch?v=K4c0\\_3KFOvM&rel=0](https://www.youtube.com/watch?v=K4c0_3KFOvM&rel=0)

- ***Ciclo de Desarrollo de la Guía Metodológica para el desarrollo de apps:***

<https://www.youtube.com/watch?v=2gZI55kV9mM&rel=0>

#### **4.15.1 Resultados**

Los resultados obtenidos se tabularon y se adjuntan en el Anexo 6. En base a los resultados, la guía metodológica es altamente recomendada, cumple con las expectativas esperadas, ya que las fases del ciclo de desarrollo propuesto contienen todas las prácticas necesarias para abordar un proyecto móvil en su totalidad. También se pudo identificar puntos de mejora que pueden abordarse en futuras investigaciones, para hacer que la guía metodológica sea aún más robusta y mejore el proceso del desarrollo de soluciones móviles.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 4.15 Conclusiones

- La guía metodológica proporciona al gerente de proyecto un conjunto de pasos metodológicos creados a partir de las mejores prácticas de desarrollo que lo guiarán en la gestión y desarrollo de una aplicación móvil transaccional.
- La guía metodológica se constituye como una herramienta integral de desarrollo de aplicaciones móviles debido a que establece un conjunto de ocho fases de ciclo de desarrollo: Generación de la idea, Gobierno, Diseño, Prototipado, Construcción, Integración, Verificación, Lanzamiento y Monitoreo.
- La guía metodológica fomenta la seguridad de la información en una aplicación, a través de la inclusión de prácticas que hacen énfasis en el código seguro, mecanismos de autenticación y seguridad de datos, haciéndola más robusta frente a vulnerabilidades y problemas de alto riesgo.
- La guía metodológica permite gestionar los proyectos de desarrollo de aplicaciones móviles a través de prácticas ágiles que se adaptan a las particularidades, restricciones del ecosistema móvil, y las necesidades específicas de cada proyecto.
- La guía metodológica combina las mejores prácticas de los métodos XP, Scrum, LeanUX, Mobile-D, MASAM, DevSecOps y Design thinking para construir un

enfoque robusto que eleva las probabilidades de tener éxito en el desarrollo de aplicaciones móviles.

- Los métodos de desarrollo de aplicaciones móviles más populares del mercado Mobile-D y MASAM describen el 'que hacer', mientras que la guía metodológica al adoptar estos métodos en conjunto con otras técnicas como Scrum y LeanUX permiten describir el 'como hacerlo' través de una serie de pasos detallados.
- La guía metodológica cumple con las expectativas esperadas de la mayoría de los gerentes de proyecto encuestados, ya que un 80% manifestó que probablemente si la recomendaría para el desarrollo de proyectos móviles, mientras que un 30% probablemente no la recomendaría.
- La guía metodológica es considerada muy complicada para una menor proporción de la población encuestada, siendo el 30%, mientras que para el 70% es considerada de complejidad media-fácil.
- La guía metodológica no es una herramienta rígida, una receta o fórmula única, por tal motivo en base a los resultados se detectaron puntos de mejora que podrán hacerla más robusta en futuros trabajos como la inclusión de plantillas de documentos, datos técnicos o las actividades específicas de cada rol.

#### **4.16 Recomendaciones**

- Se recomienda a las empresas implementar la guía metodológica para el desarrollo de aplicaciones móviles, siguiendo los pasos detallados, manteniendo la forma de trabajo ágil, y la obtención temprana de retroalimentación del usuario.

Esta forma de trabajo permitirá abordar todas las características del ecosistema móvil, y permita obtener un proceso optimizado de desarrollo que conducirá a resultados exitosos.

- Se recomienda al equipo de trabajo e interesados mantener una comunicación efectiva que permita identificar nuevos puntos de mejora de la guía para complementarla y hacerla aún más robusta.
- Se recomienda incentivar a estudiantes el desarrollo de proyectos de aplicaciones móviles, utilizando como base la guía metodológica para que puedan identificar los pasos a seguir en la gestión de un proyecto móvil.
- Se recomienda el uso de la guía a gerentes y líderes de proyecto que se encuentran en etapa de transición de desarrollo convencional (web, escritorio) a desarrollo de aplicaciones móviles. A empresas start-up que necesitan de un marco de referencia para empezar con sus estrategias de desarrollo y despliegue de aplicaciones móviles; y a aquellas personas que a través del uso de buenas prácticas de desarrollo sientan el deseo de agregar valor a sus proyectos.
- Para crear aplicaciones seguras se recomienda mantener el proceso de seguridad en cada fase del ciclo de desarrollo, además mantener políticas de seguridad de datos, el uso de repositorios privados, el uso de código seguro, el uso de sistemas de encriptación, mecanismos de autenticación, canales de comunicación seguros y seguir los lineamientos de riesgos más críticos de seguridad según la Organización OWASP.

- En base a la retroalimentación de las encuestas realizadas se recomienda para futuras investigaciones, incluir datos técnicos como el análisis de la infraestructura necesaria a utilizar (bases de datos, hosting, alojamiento, servidores, capacidad, hardening, etc), así como modelos de los entregables, o el detalle de las funciones de cada rol del equipo de trabajo.

## ANEXOS

## Anexo 1. Estudios de las metodologías de desarrollo de aplicaciones móviles

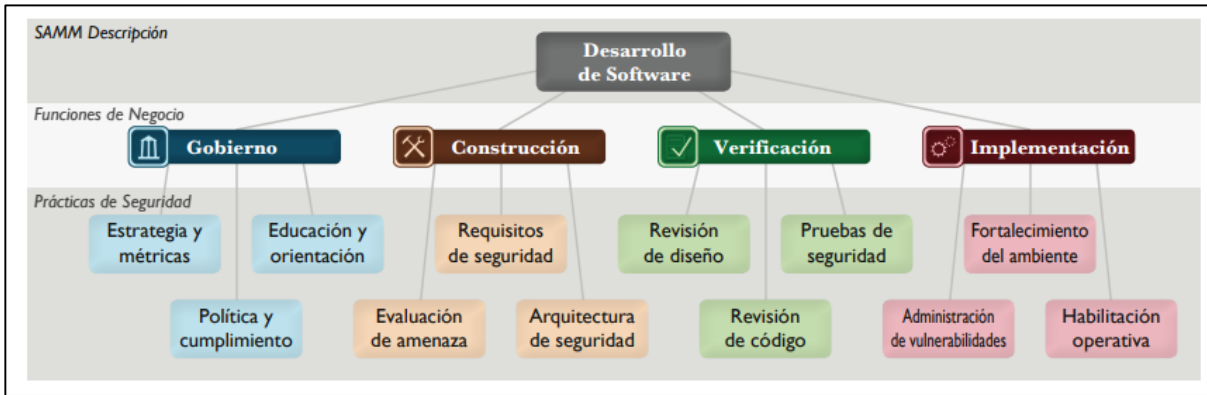
TIPO	TÍTULO	AUTORES	AÑO	PAÍS	N.º METODOS	METODOLOGÍAS INCLUIDAS	DESCRIPCIÓN	
1	Artículo	Adopting an agile approach for the development of mobile applications	Flora, Chande & Wang	2014	India	8	Mobile-D HME-HYBRID MASAM SLess Scrum XP Lean Kanban NPD	<ul style="list-style-type: none"> <li>• Análisis de las principales características de las metodologías de desarrollo de aplicaciones móviles.</li> <li>• Evalúa como adoptar varios enfoques ágiles para mejorar el desarrollo de aplicaciones móviles y mejorar los procesos dentro de una organización.</li> </ul>
2	Artículo	An integrated product development process for mobile software	Zeidler, Kittl, & Petrovic	2007	Canada	1		<ul style="list-style-type: none"> <li>• Describe un proceso de desarrollo de productos de cinco pasos para software y servicios móviles que abarcan cuestiones organizativas, comerciales y técnicas e informes sobre las experiencias prácticas realizadas en un proyecto de la vida real.</li> </ul>
3	Tesis	Propuesta de una metodología para el desarrollo de aplicaciones móviles en el campo educativo (Mobile Learning) para la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.	Sarango, Stalin	2017	Ecuador	4	MDAEM Mobile-D XP ICONIX	<ul style="list-style-type: none"> <li>• Recopilación de información sobre las metodologías para el desarrollo de aplicaciones móviles educativas</li> <li>• Análisis de la situación actual en cuanto al uso de metodologías para el desarrollo de aplicaciones móviles</li> </ul>
4	Tesis	Marco de trabajo para control de calidad en el desarrollo de aplicaciones móviles bajo metodologías ágiles	Puglla, Gabriela	2017	Ecuador	1	Mobile-D	<ul style="list-style-type: none"> <li>• Establece la interacción entre las fases de desarrollo del enfoque ágil y las fases del proceso de calidad definidos, para facilitar y agilizar la concepción, idea, diseño, construcción y validación del software a través del uso de artefactos, métricas de calidad, técnicas y la ejecución de pruebas adecuadas a este tipo de aplicaciones y obtener un producto de calidad acorde a las necesidades y exigencias del entorno y de los usuarios.</li> </ul>
5	Tesis	Procesos en el desarrollo de aplicaciones para dispositivos móviles	Rangel, Ian	2013	Mexico	6	Mobile-D MASAM HYBRID Proceso integrado Modelo M-Business Modelo MP-Movil	<ul style="list-style-type: none"> <li>• Recopila las propuestas existentes sobre procesos de desarrollo de aplicaciones para dispositivos móviles.</li> <li>• Integra un modelo de procesos para el desarrollo de aplicaciones para dispositivos móviles, basado en la revisión bibliográfica de propuestas actuales y experiencia de profesionales</li> </ul>
6	Artículo	Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual	Amaya, Yohn	2013	Colombia	6	XP Scrum TDD Mobile-D HMD Spiral	<ul style="list-style-type: none"> <li>• Describe el uso de los métodos ágiles en el desarrollo de aplicaciones para dispositivos móviles</li> </ul>



7	Artículo	Guía metodológica ágil, para el desarrollo de aplicaciones móviles "AEGIS-MD"	Amaya, Yohn	2013	Colombia	3	XP Scrum TDD	<ul style="list-style-type: none"> <li>Desarrolla una guía metodológica basada en características de las metodologías XP, SCRUM y TDD con la finalidad de proporcionar elementos conceptuales, pasos estructurados, roles y herramientas a ser aplicadas en el desarrollo de aplicaciones para dispositivos móviles, con el fin de aumentar la productividad en el proyecto y asegurar la calidad del producto entregado</li> </ul>
8	Artículo	Metodología para el desarrollo de aplicaciones móviles	Gasca, Camargo & Medina	2013	Colombia	1	MDAM	<ul style="list-style-type: none"> <li>Conceptualización de las tecnologías y las metodologías ágiles para el desarrollo de software</li> </ul>
9	Artículo	A combined agile project management approach for mobile application development	Daković, Aljaž	2000	Slovenia	2	Scrum XP	<ul style="list-style-type: none"> <li>Adaptación original e integración de dos metodologías ágiles: Scrum y Extreme Programming, creando una base adecuada y validada para la gestión de proyectos.</li> </ul>
10	Artículo	Agile Development Methodology and Testing for Mobile Applications - A Survey	Prasad, Hamsini & Smitha	2016	India	9	XP HME MASAM DSDM NPD ASD AUP RUP SLess	<ul style="list-style-type: none"> <li>Describe las principales metodologías ágiles para el desarrollo de proyectos móviles.</li> </ul>
11	Artículo	Software Process Models for Mobile Application Development: A Review	Ashishdeep, Bhatia & Varma	2015	India	8	Agile Mobile-D HME MASAM SLess Scrum XP Lean	<ul style="list-style-type: none"> <li>Describe varios modelos de procesos de software y como la combinación del enfoque ágil puede convertirse en proceso de software ideal para el desarrollo de aplicaciones móviles</li> </ul>
12	Artículo	A Proposed Hybrid Agile Framework Model For Mobile Applications Development	Ammar Khader Almasri	2016	Jordania	11	Cascada RAD Spiral RUP Shaped Model Agile XP Scrum Crystal DSDM FDD MDE	<ul style="list-style-type: none"> <li>Busca un modelo de proceso adecuado que cumpla con los requisitos de la aplicación móvil.</li> <li>Investiga los métodos de desarrollo ágiles para que el desarrollo de la aplicación móvil sea fácil y compatible con las características del dispositivo móvil.</li> </ul>
13	Tesis	Modelo para el desarrollo de aplicaciones nativas en Android basado en mejores prácticas, metodologías ágiles y elementos del área interacción humano - computadora	Veloz, Carlos	2016	México	1	MDE	<ul style="list-style-type: none"> <li>Considera las fases del ciclo de desarrollo de software para el desarrollo de aplicaciones para la plataforma Android</li> </ul>
14	Artículo	Método ágil híbrido para desarrollar software en dispositivos móviles	Leiva & Villalobos	2015	Chile	1	Agile	<ul style="list-style-type: none"> <li>Presenta un nuevo método ágil híbrido para desarrollar software en dispositivos móviles, la evaluación de cuán ágil es, si es o no adecuada para el desarrollo de aplicaciones en dispositivos móviles y su aplicación en un caso real para evaluar sus bondades</li> </ul>

15	Artículo	Agile: A Natural Fit in Mobile Application Development Process	Mahmud & Abdullah	2015	Malasia	8	ASD Mobile-D Scrum MASAM HYBRID SLess & Six Sigma MASEF DSDM	<ul style="list-style-type: none"> <li>• Presenta un proceso de desarrollo de aplicaciones móviles utilizando una metodología ágil.</li> </ul>
16	Artículo	Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con <i>Android</i> y el <i>iPhone</i>	Blanco, Camarero, Fumero, Werterski, & Rodriguez	2014	España	2	Mobile-D HYBRID	<ul style="list-style-type: none"> <li>• Describe la idoneidad de las metodologías de tipo ágiles para el desarrollo de software móvil.</li> </ul>
17	Artículo	Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach	Rahimian & Ramsin	2008	Irán	3	ASD NPD HYBRID	<ul style="list-style-type: none"> <li>• Describe una metodología híbrida para el desarrollo de software móvil</li> </ul>
18	Artículo	Metodología de desarrollo de aplicaciones para el aprendizaje móvil basadas en software libre	Otero, Martinez & Diaz	2016	México	1	PACIE	<ul style="list-style-type: none"> <li>• Presenta una metodología para el desarrollo de aplicaciones de m-learning que se ejecuten bajo el sistema operativo Android especificando los pasos a seguir para lograr su implementación de modo adecuado; finalmente se aporta una aplicación enfocada a la educación superior</li> </ul>
19	Artículo	Métodos para el desarrollo de aplicaciones móviles	Ramírez, Robert	2011	España	4	Cascada Rápido Agile Mobile-D	<ul style="list-style-type: none"> <li>• Detalla un método de desarrollo de aplicaciones móviles y expone las mejores prácticas en cada una de las fases del desarrollo.</li> </ul>
20	Libro	Agile Project Management For Mobile Application Development	Estrella & Sornabala	2017	Canada	1	Scrum	Describe el ciclo de desarrollo de aplicaciones móviles utilizando Scrum y la gestión de proyectos.

**Anexo 2. El modelo de madurez para el aseguramiento de software.**



Fuente: (Chandra, s.f)

### **Anexo 3. OWASP Top 10 Application Security Risks - 2017**

---

#### **OWASP TOP 10 APPLICATION SECURITY RISKS – 2017**

A1 – Inyección

A2 – Pérdida de Autenticación y Gestión de Sesiones

A3 – Secuencia de Comandos en Sitios Cruzados (XSS)

A4 – Ruptura del Control de Acceso

A5 – Configuración de Seguridad Incorrecta

A6 – Exposición de Datos Sensibles

A7 – Protección de ataque insuficiente

A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)

A9 – Uso de Componentes con Vulnerabilidades Conocidas

A10 – API desprotegidas

---

### **Anexo 4. OWASP Mobile Security Top 10 2016**

---

#### **OWASP MOBILE SECURITY TOP 10 2016**

M1 - Uso incorrecto de la plataforma

M2 - Almacenamiento inseguro de datos

M3 - Comunicación insegura

M4 - Autenticación insegura

M5 - Criptografía insuficiente

M6 - Autorización insegura

M7 - Calidad del código del cliente

M8 - Manipulación de códigos

M9 - Ingeniería inversa

M10 - Funcionalidad extraña


---

## Anexo 5. Formato de encuesta

### ENCUESTA DE VALORACIÓN DE UNA GUÍA METODOLÓGICA PARA EL DESARROLLO DE APPS TRANSACCIONALES

Buenos días,

estimad@, por favor dedique unos minutos de su tiempo para responder la siguiente encuesta, con el fin de conocer el grado de aceptación de una guía metodológica propuesta para el desarrollo de aplicaciones móviles transaccionales

**EMPEZAR ENCUESTA** 

**1**

✖

**Seleccione las principales metodologías de desarrollo que utiliza para el desarrollo de proyectos móviles**

- Scrum
- Mobile-D
- Masam
- Hybrid (HDM)
- Sless
- ICONIX
- XP
- Waterfall
- Agile
- Otras:

*100 caracteres restantes*

**2**

✖

**¿Las metodologías seleccionadas son adaptadas según las necesidades del proyecto o cumple con las fases establecidas?**

- Todas las fases son adaptadas
- No todas las fases son adaptadas
- Son adaptadas en un porcentaje mínimo
- Cumple con todas las fases
- Otra Respuesta:

*100 caracteres restantes*

**Coloque la frecuencia sobre los problemas que ha tenido a la hora de gestionar el desarrollo de una app móvil.**

	Nunca	Algunas Veces	Casi siempre	Siempre
Planificaciones demasiado optimistas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mala estimación de tiempos de desarrollo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gestión del riesgo insuficiente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Destinar poco tiempo al control de calidad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño inadecuado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Confiar demasiado en tecnologías-herramientas no exploradas previamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Motivación débil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Añadir más personal a un proyecto atrasado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Planificaciones demasiado optimistas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Excesivas tareas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Problemas con el equipo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Falta de involucración del usuario	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Se prepara el producto para su liberación demasiado pronto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Corrupción del alcance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Falta de automatización del control del Código Fuente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4

Antes de contestar las preguntas de los siguientes apartados, debe observar con atención el siguiente video explicativo acerca del proceso que tiene la guía metodológica para el desarrollo de apps. Finalmente para continuar debe calificar el video.

Guía Metodológica para el desarrollo de apps



★ ★ ★ ★ ★

0/5

5

La guía propone las siguientes fases. ¿Cómo valora el ciclo de desarrollo de la guía metodológica propuesta?

Generación de la idea	Levantamiento de la información necesaria para el proyecto.
Gobierno	Planificación, obtención y clasificación de los requerimientos
Diseño	Estructuración de la solución y definición del escenario tecnológico
Prototipado	Personificación de ideas del cliente en prototipos de interfaz de usuario
Construcción	Implementación del diseño en un producto de software
Integración	Integración de la solución con subsistemas
Verificación	Validación y simulación del producto ajustando detalles para luego instalarlo en equipos reales y evaluar su rendimiento
Lanzamiento y Monitoreo	Despliegue y monitoreo de la aplicación.

★ ★ ★ ★ ★

3/5



6

Antes de contestar las preguntas de los siguientes apartados, debe observar con atención el siguiente video explicativo acerca del ciclo de desarrollo que tiene la guía metodológica para el desarrollo de apps. Finalmente para continuar debe calificar el video.

Ciclo de Desarrollo de la Guía Metodológica para el desarrollo de apps

3. Evaluar los Prototipos

4. Solicitar feedback



★ ★ ★ ★ ★

0/5

7

**La guía propone los siguientes pasos. En que porcentaje usted considera que la guía puede mejorar el desarrollo de proyectos móviles**

Generación de la idea	Gobierno	Diseño	Prototipado	Construcción	Integración	Verificación	Lanzamiento y Monitoreo
Necesidades e ideas del cliente y usuarios	Planeación inicial del proyecto	Arquitectura funcional y técnica (Design Thinking)	Establecimiento de soluciones potenciales	Entorno de desarrollo de la solución móvil	Integración de Incrementos del producto	Pruebas rigurosas y de validación	Gestión de liberación
Establecimiento de stakeholders y potenciales usuarios	Conformación del equipo de trabajo	Arquitectura de seguridad	Prototipos de Interfaz de Usuario (UI)	Implementación de código de manera efectiva	Integración con Sistemas internos o externos	Verificación de funcionalidades solicitadas por el cliente	Evaluación con usuarios
Tendencias tecnológicas y Plataformas de desarrollo	Requerimientos funcionales y no funcionales	Refinamiento del diseño	Evaluación de Prototipos	Prácticas de seguridad en la codificación	Pruebas de integración sobre el producto consolidado	Ejecución de diferentes tipos de pruebas	Estrategia de marketing
Generación de Épicas (historias de Usuario generales)	Selección de tipo de solución móvil	Experiencia de Usuario (UX)	Retrosalimentación/feedback	Requisitos de Plataforma Móvil	Enfoque DevOps - DevSecOps	Corrección de defectos	Lanzamiento al mercado
Compromisos con el cliente	Definición del entorno de la solución móvil	Estándares de usabilidad y accesibilidad		Gestión de actividades de Scrum	Modelo de Integración Continua	Pruebas de Seguridad	Monitoreo en el mercado
	Establecimiento del enfoque ágil					Automatización de Pruebas (Regresiones)	
	Ciclo de vida de desarrollo seguro (SDL)						

MOBILE-D

XP

SCRUM

MASAM

LEANUX

SEGURIDAD

No puede mejorar

En un 5%

En un 10%

En un 20%

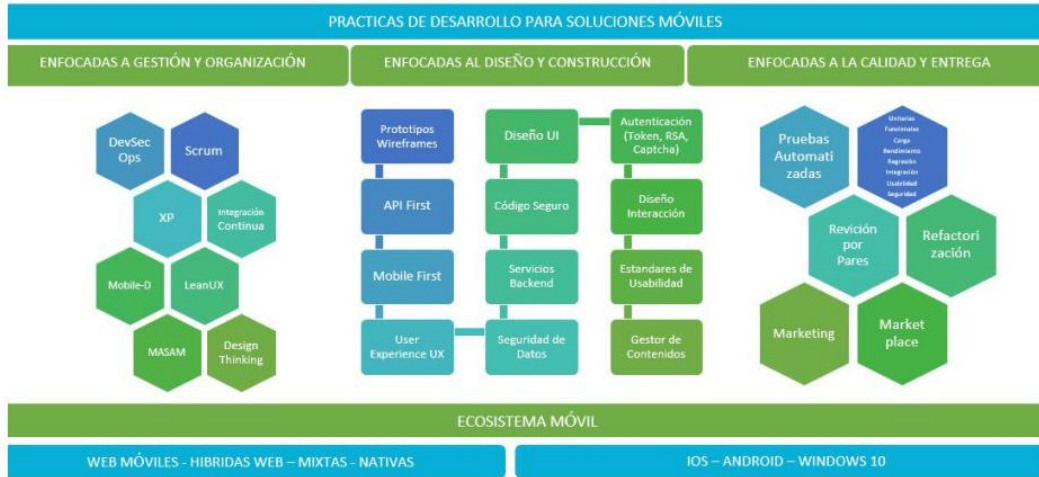
En un 50%

Una mejora completa

Otro:

Escribir tu respuesta

**Esta desuerdo con las siguientes prácticas propuestas para el desarrollo de apps móviles transaccionales**



Totalmente en desacuerdo  
 En desacuerdo  
 Ni de acuerdo, ni en desacuerdo  
 De acuerdo  
 Totalmente de acuerdo  
 Otra respuesta:

100 caracteres restantes

9

\*

**Considera importante identificar las necesidades reales del cliente en base a las tendencias del panorama tecnológico actual, como se describe en la fase "Generación de la idea"**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

10

\*

**En la fase "Gobierno", se establece el enfoque ágil como marco de referencia y se planifica un ciclo de desarrollo seguro. ¿Considera adecuado ambos enfoques?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

11

**En la fase "Diseño", se incluyen definiciones de Experiencia de Usuario, usabilidad, accesibilidad y tareas de revisión de definiciones de arquitectura. ¿Considera necesario que en estas definiciones se involucre a todo el equipo, cliente e interesados?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

100 caracteres restantes

12

**En la fase "Diseño" se propone realizar una construcción basada en APIs, que permitan exponer servicios base de la solución. Usted considera importante el uso de APIs en el desarrollo de apps móviles.?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

13

**¿Considera relevante la inclusión de una fase “Prototipado” que permita personificar las ideas del cliente?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

100 caracteres restantes

14

**¿Usted cree que se debería poner énfasis en la creación de código efectivo y seguro según como se establece la fase "Construcción"?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

15

**¿Considera necesario fomentar el uso del enfoque DevSecOps (Desarrollo, Seguridad, Operaciones) para una adecuada integración continua de la app según lo establecido en la fase "Integración"?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

16

★

**En la fase "Verificación", es importante realizar todo tipo de pruebas para validar la app. ¿Cree usted que es necesario automatizarlas para obtener mejores resultados?**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

17

★

**¿Como considera que se garantiza la calidad de un producto final?**

Pruebas de control de calidad

Uso de Prototipos

Participación del usuario

Planificando hitos de revisión

Todas las anteriores

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*

18

★

**Considera usted que el monitoreo de la app debe ser constante una vez desplegada en Producción, según lo expresado en la fase "Lanzamiento y monitoreo"**

Totalmente en desacuerdo

En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

Otra respuesta:

Escribir tu respuesta

*100 caracteres restantes*



19

¿Como considera la guía metodológica propuesta?

Muy sencilla

Fácil

De dificultad media

Dificil

Muy complicada

20

¿La guía metodológica propuesta cumple con sus expectativas? Qué tan satisfecho se encuentra?

Muy satisfecho

Satisfecho

Medianamente Satisfecho

Insatisfecho

Totalmente insatisfecho

21

¿Recomendaría a esta guía metodologica para el desarrollo de proyectos móviles?

Definitivamente sí

Probablemente sí


No lo sé

Probablemente no

Seguramente no

22

**¿Cómo se podría mejorar o que recomendaciones daría a la presente guía propuesta?**

 Escribir un párrafo

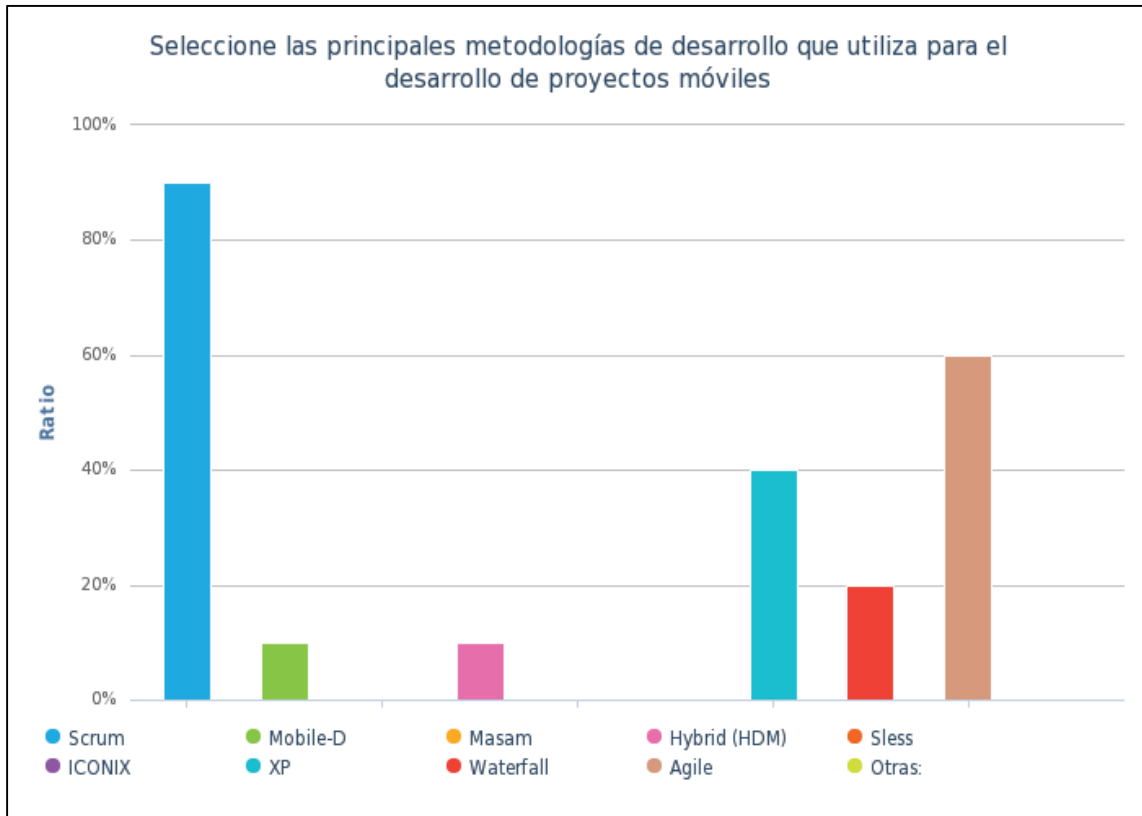
*1500 caracteres restantes*

Gracias por su colaboración.

**Anexo 6. Resultados de la encuesta**

**Pregunta 1**

**Seleccione las principales metodologías de desarrollo que utiliza para el desarrollo de proyectos móviles**



**Figura 31.** Principales metodologías de desarrollo

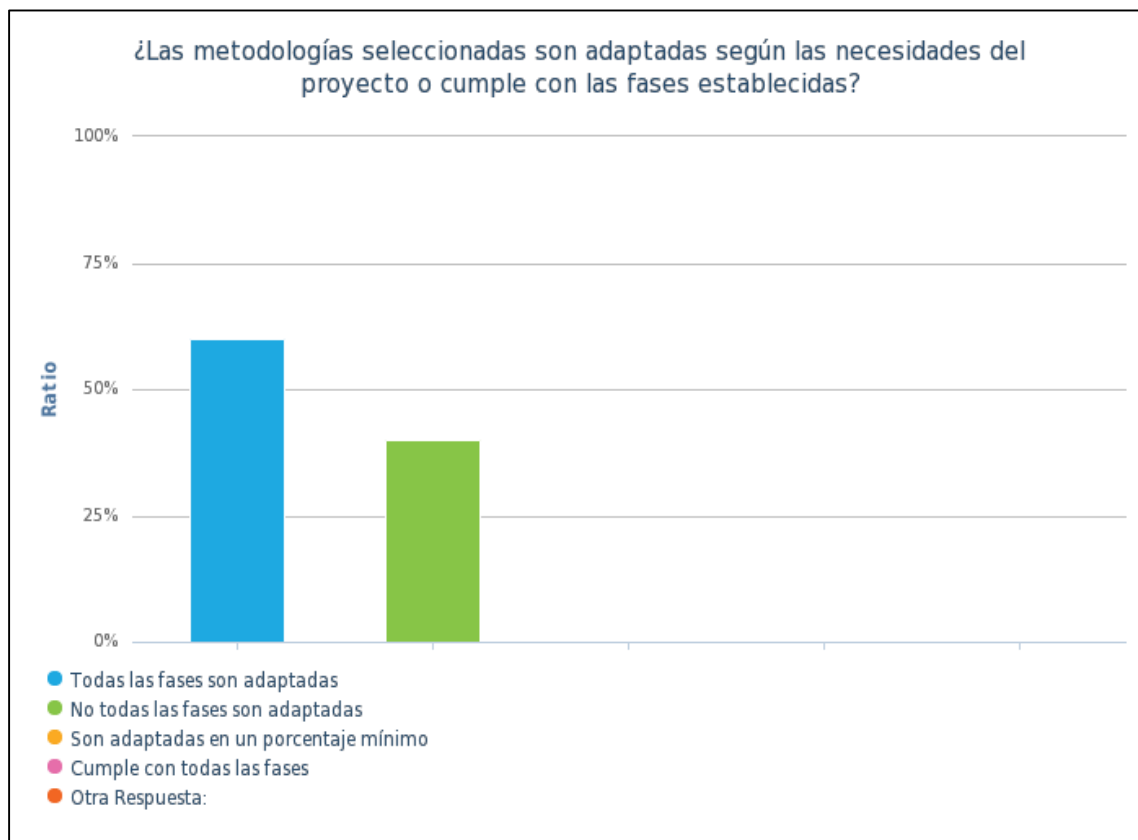
**Tabla 12**  
*Principales metodologías de desarrollo*

	RESPUESTAS	PORCENTAJE
<b>Scrum</b>	9	90 %
<b>Otras:</b>	0	0 %
<b>Mobile-D</b>	1	10 %
<b>Masam</b>	0	0 %
<b>Hybrid (HDM)</b>	1	10 %
<b>Sless</b>	0	0 %
<b>ICONIX</b>	0	0 %
<b>XP</b>	4	40 %
<b>Waterfall</b>	2	20 %
<b>Agile</b>	6	60 %

Según la **Figura 31** y la **Tabla 12**, los encuestados indican que las metodologías más usadas o de mayor importancia para el desarrollo de proyectos móviles son Scrum, Agile y XP, mientras que de menor importancia o menos usadas son Mobile-D, Hybrid y Waterfall.

## Pregunta 2

¿Las metodologías seleccionadas son adaptadas según las necesidades del proyecto o cumple con las fases establecidas?



**Figura 32.** Metodologías adaptables

**Tabla 13**

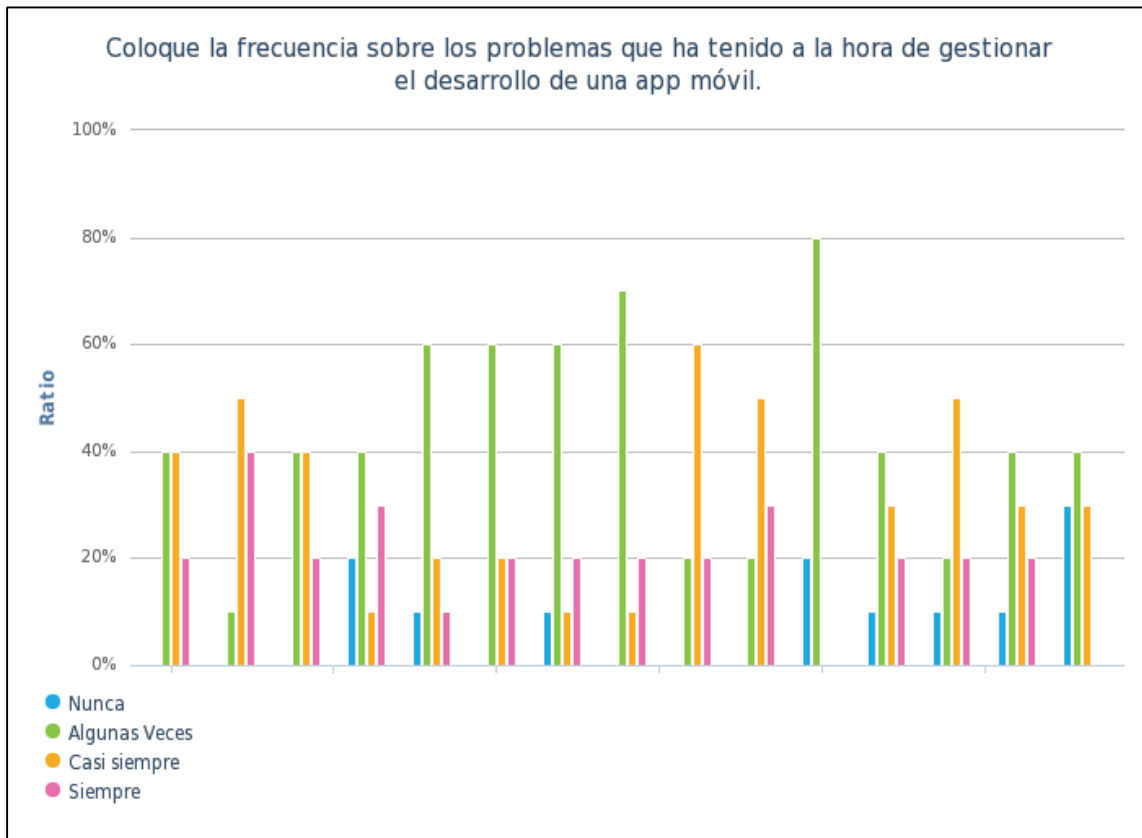
*Metodologías adaptables*

	RESPUESTAS	PORCENTAJE
<b>Todas las fases son adaptadas</b>	6	60 %
<b>No todas las fases son adaptadas</b>	4	40 %
<b>Son adaptadas en un porcentaje mínimo</b>	0	0 %
<b>Cumple con todas las fases</b>	0	0 %
<b>Otra Respuesta:</b>	0	0 %

Según la **Figura 32** y la **Tabla 13**, el 60% de la población encuestada adapta las metodologías de desarrollo en base a las necesidades del proyecto, mientras que un 40% cumple con todas las fases predefinidas.

**Pregunta 3**

**Coloque la frecuencia sobre los problemas que ha tenido a la hora de gestionar el desarrollo de una app móvil.**



**Figura 33.** Frecuencia de problemas

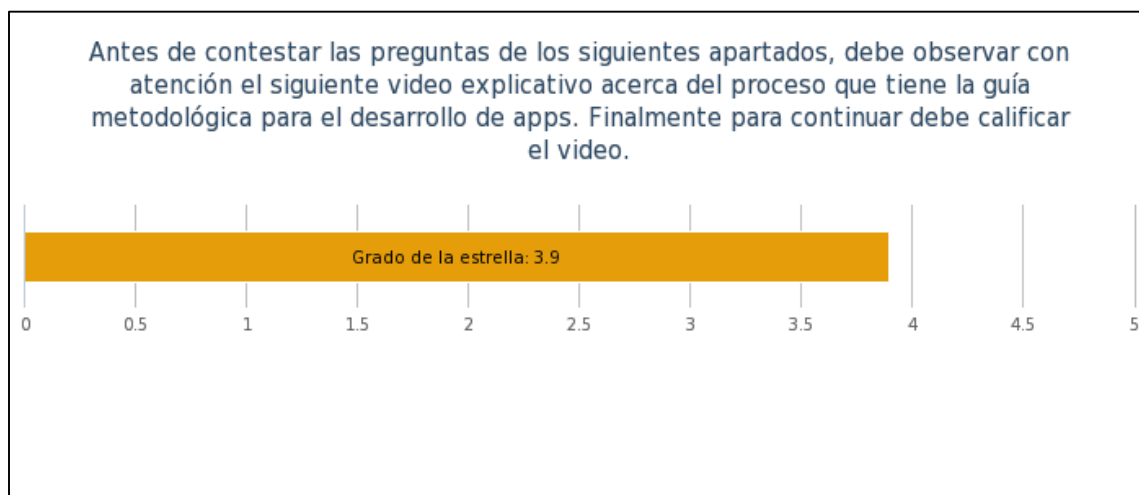
**Tabla 14**  
*Frecuencia de problemas*

	NUNCA	ALGUNAS VECES	CASI SIEMPRE	SIEMPRE
<b>Planificaciones demasiado optimistas</b>	0	4	4	2
<b>Mala estimación de tiempos de desarrollo</b>	0	1	5	4
<b>Gestión del riesgo insuficiente</b>	0	4	4	2
<b>Destinar poco tiempo al control de calidad</b>	2	4	1	3
<b>Diseño inadecuado</b>	1	6	2	1
<b>Confiar demasiado en tecnologías-herramientas no exploradas previamente</b>	0	6	2	2
<b>Motivación débil</b>	1	6	1	2
<b>Añadir más personal a un proyecto atrasado</b>	0	7	1	2
<b>Planificaciones demasiado optimistas</b>	0	2	6	2
<b>Excesivas tareas</b>	0	2	5	3
<b>Problemas con el equipo</b>	2	8	0	0
<b>Falta de involucración del usuario</b>	1	4	3	2
<b>Se prepara el producto para su liberación demasiado pronto</b>	0	4	4	2
<b>Corrupción del alcance</b>	0	1	5	4
<b>Falta de automatización del control del Código Fuente</b>	0	4	4	2

Según la **Figura 33** y la **Tabla 14**, podemos evidenciar que todos los problemas expuestos están presentes con diferentes frecuencias durante el desarrollo de un proyecto móvil, que de una u otra forma están impactando en el proceso.

#### Pregunta 4

Antes de contestar las preguntas de los siguientes apartados, debe observar con atención el siguiente video explicativo acerca del proceso que tiene la guía metodológica para el desarrollo de apps. Finalmente, para continuar debe calificar el video.



**Figura 34.** Proceso de la Guía

#### Tabla 15

*Proceso de la Guía*

	RESPUESTAS	PORCENTAJE
1/5	0	0 %
2/5	1	10 %
3/5	3	30 %
4/5	2	20 %
5/5	4	40 %

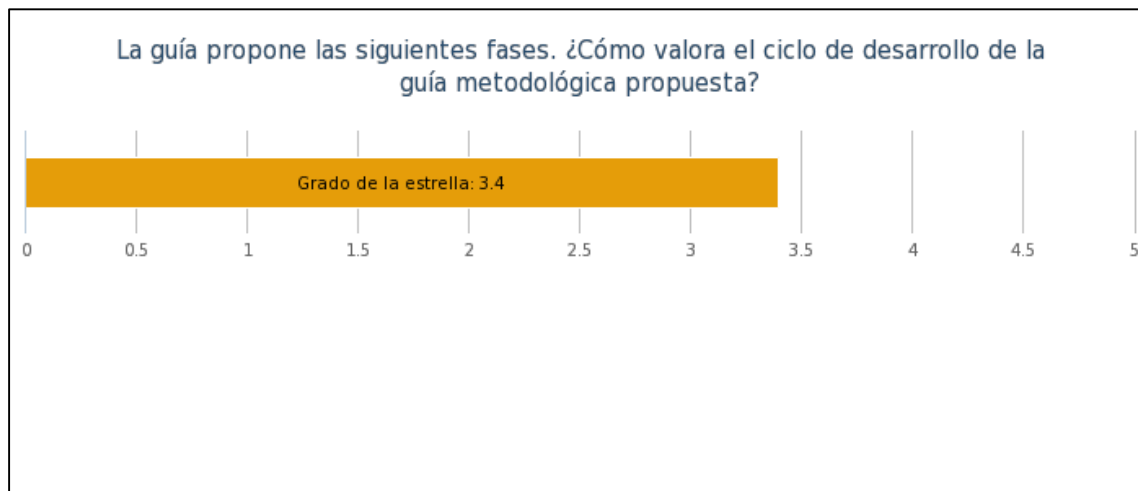
Según la **Figura 15** y la **Tabla 34**, podemos evidenciar que la información mostrada en el video explicativo acerca del proceso de la guía tiene un promedio de aprobación del 3.9/5 equivalente a Bueno. Donde el 40% lo califica con cinco estrellas que equivale a



Muy Bueno, un 20% con cuatro estrellas equivalente a Bueno, otro 30 % con tres estrellas equivalente a Regular, y mientras que un porcentaje menor de 10% con dos estrellas equivalente a Malo.

### Pregunta 5

La guía propone las siguientes fases. ¿Cómo valora el ciclo de desarrollo de la guía metodológica propuesta?



**Figura 35.** Valoración Ciclo Desarrollo

**Tabla 16**

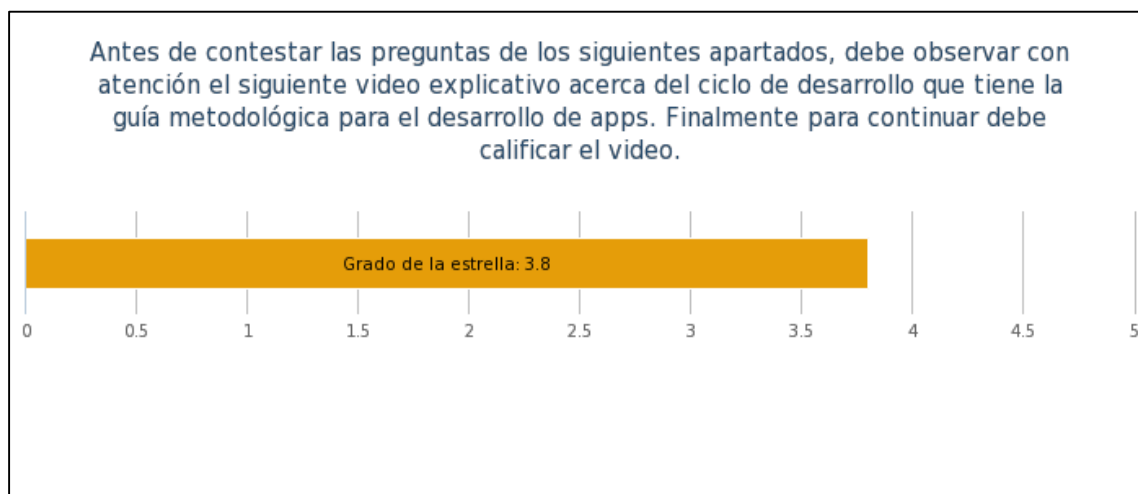
*Valoración Ciclo Desarrollo*

	RESPUESTAS	PORCENTAJE
<b>1/5</b>	1	10 %
<b>2/5</b>	0	0 %
<b>3/5</b>	5	50 %
<b>4/5</b>	2	20 %
<b>5/5</b>	2	20 %

Según la **Figura 35** y la **Tabla 16**, podemos evidenciar que las fases del ciclo desarrollo de la guía propuesta tiene un promedio de aprobación del 3.4/5 equivalente a Regular/Bueno. Donde el 50% lo califica con tres estrellas equivalente a Regular, un 20% con cinco estrellas equivalente a Muy Bueno, un 20% con cuatro estrellas equivalente a Bueno, y mientras un porcentaje menor de 10% como Malo.

### Pregunta 6

**Antes de contestar las preguntas de los siguientes apartados, debe observar con atención el siguiente video explicativo acerca del ciclo de desarrollo que tiene la guía metodológica para el desarrollo de apps. Finalmente, para continuar debe calificar el video.**



**Figura 36.** Ciclo Desarrollo

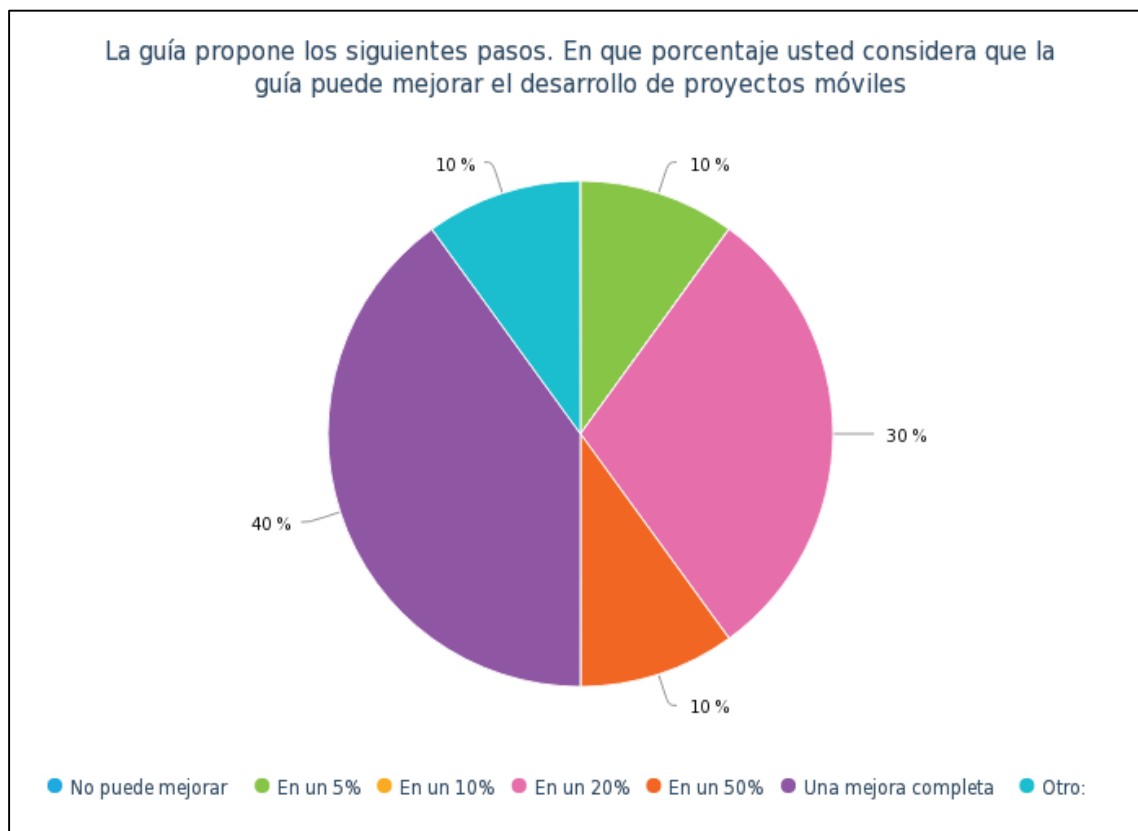
**Tabla 17**  
*Ciclo Desarrollo*

	RESPUESTAS	PORCENTAJE
<b>1/5</b>	0	0 %
<b>2/5</b>	0	0 %
<b>3/5</b>	4	40 %
<b>4/5</b>	4	40 %
<b>5/5</b>	2	20 %

Según la **Figura 36** y la **Tabla 17**, podemos evidenciar que la información acerca del ciclo de desarrollo de la guía mostrada el video explicativo, tiene un promedio de aprobación del 3.8/5 equivalente a Bueno. Donde el 40% lo califica con tres estrellas equivalente a Regular, un 40% con cuatro estrellas equivalente a Bueno un 20% con cinco estrellas equivalente a Muy Bueno, mientras que nadie lo califico como Malo.

### **Pregunta 7**

**La guía propone los siguientes pasos. En qué porcentaje usted considera que la guía puede mejorar el desarrollo de proyectos móviles**



**Figura 37.** Pasos de la Guía

**Tabla 18**  
*Pasos de la Guía*

	RESPUESTAS	PORCENTAJE
<b>No puede mejorar</b>	0	0 %
<b>En un 5%</b>	1	10 %
<b>En un 10%</b>	0	0 %
<b>En un 20%</b>	3	30 %
<b>En un 50%</b>	1	10 %
<b>Una mejora completa</b>	4	40 %
<b>Otro:</b>	1	10 %

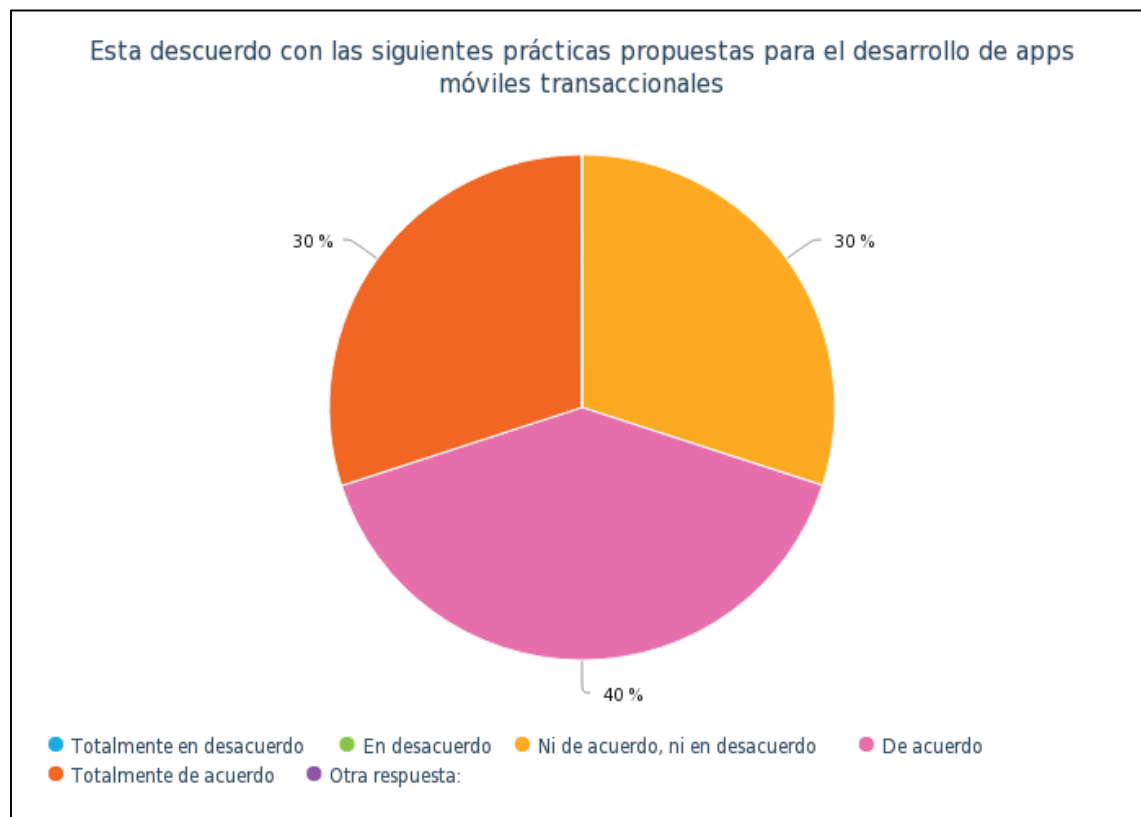
Según la **Figura 37** y la **Tabla 18**, el 40% de la población encuestada considera que la guía puede brindar una mejora completa al desarrollo de los proyectos móviles, otro 30% considera que puede mejorar “en un 20%”, el 10% puede mejorar “en un 50%”, otro 10% puede mejorar “en un 5%”, y mientras que nadie respondió que “no se puede mejorar”.

Entre otras respuestas tenemos:

- Una guía aporta, pero la mejora dependerá de que tan factible es su aplicación práctica

### Pregunta 8

**Esta descuerdo con las siguientes prácticas propuestas para el desarrollo de apps móviles transaccionales**



**Figura 38.** Prácticas propuestas

**Tabla 19**

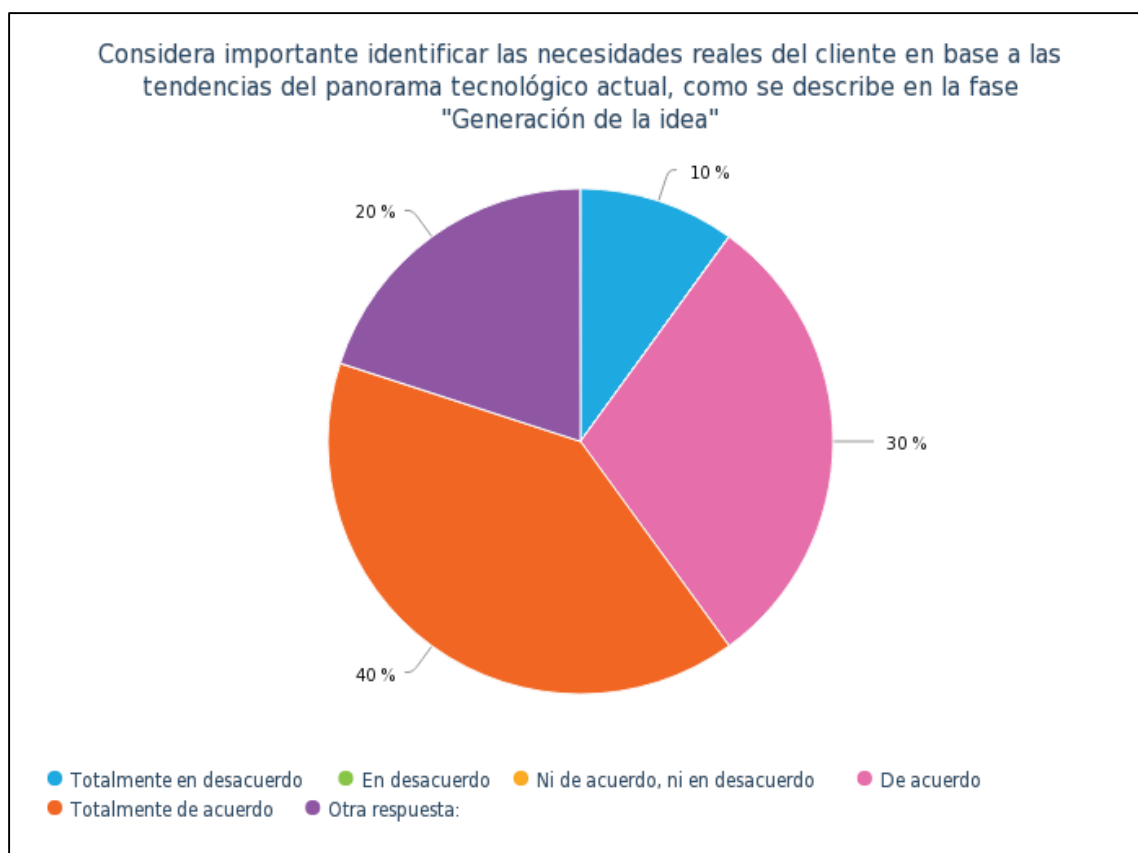
*Prácticas propuestas*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	3	30 %
<b>De acuerdo</b>	4	40 %
<b>Totalmente de acuerdo</b>	3	30 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 38** y la **Tabla 19**, el 40% de la población encuestada está de acuerdo con las prácticas propuestas para el desarrollo de soluciones móviles, un 30% está totalmente de acuerdo, otro 30% no está ni de acuerdo, ni en desacuerdo, y mientras que nadie respondió que está en desacuerdo.

### Pregunta 9

**Considera importante identificar las necesidades reales del cliente en base a las tendencias del panorama tecnológico actual, como se describe en la fase "Generación de la idea"**



**Figura 39.** Necesidades del cliente

**Tabla 20**  
*Necesidades del cliente*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	1	10 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	0	0 %
<b>De acuerdo</b>	3	30 %
<b>Totalmente de acuerdo</b>	4	40 %
<b>Otra respuesta:</b>	2	20 %

Según la **Figura 39** y la **Tabla 20**, el 40% de la población encuestada está totalmente de acuerdo en que dentro de la fase “Generación de la idea” se identifiquen las necesidades reales del cliente en base a las tendencias del panorama tecnológico actual, un 30% está también de acuerdo, y mientras en menor porcentaje el 10% está en desacuerdo.

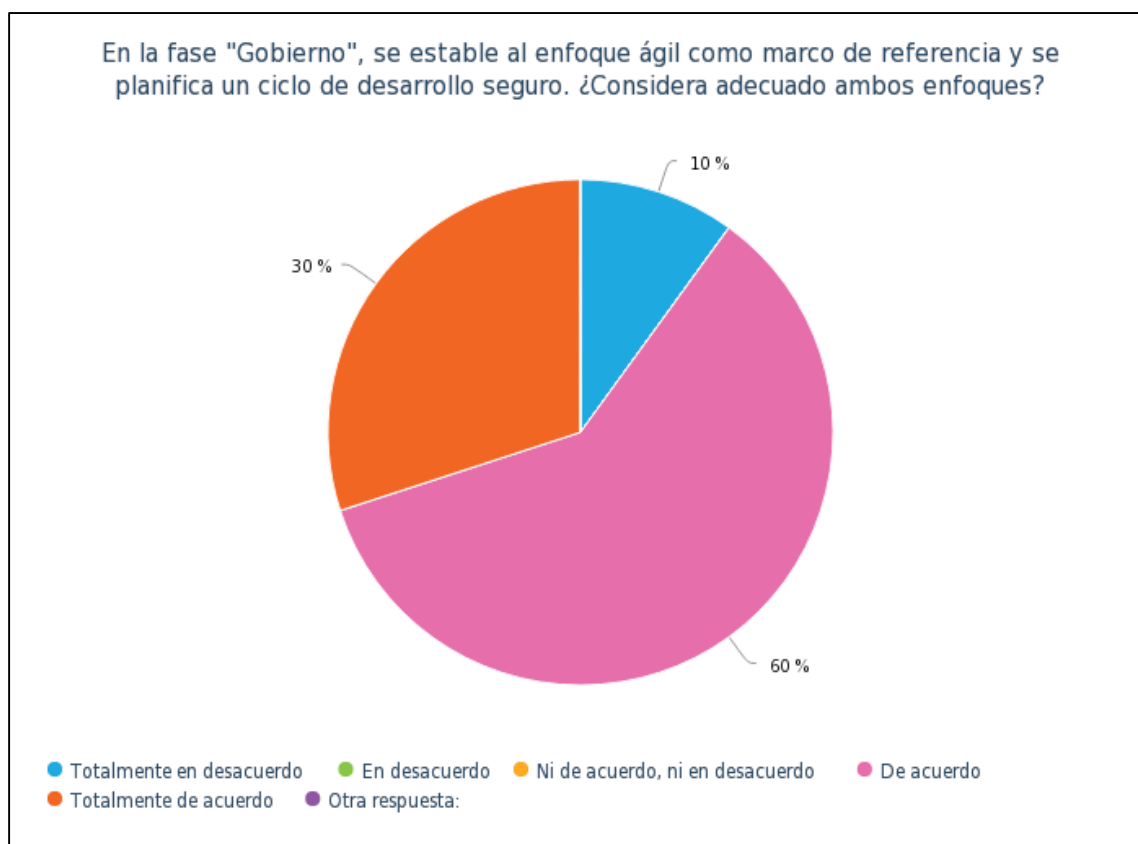
Entre otras respuestas tenemos:

- La clave siempre será entender las necesidades del cliente.
- Pocas veces el cliente sabe que quiere realmente



**Pregunta 10**

**En la fase "Gobierno", se establece al enfoque ágil como marco de referencia y se planifica un ciclo de desarrollo seguro. ¿Considera adecuado ambos enfoques?**



**Figura 40.** Enfoque ágil

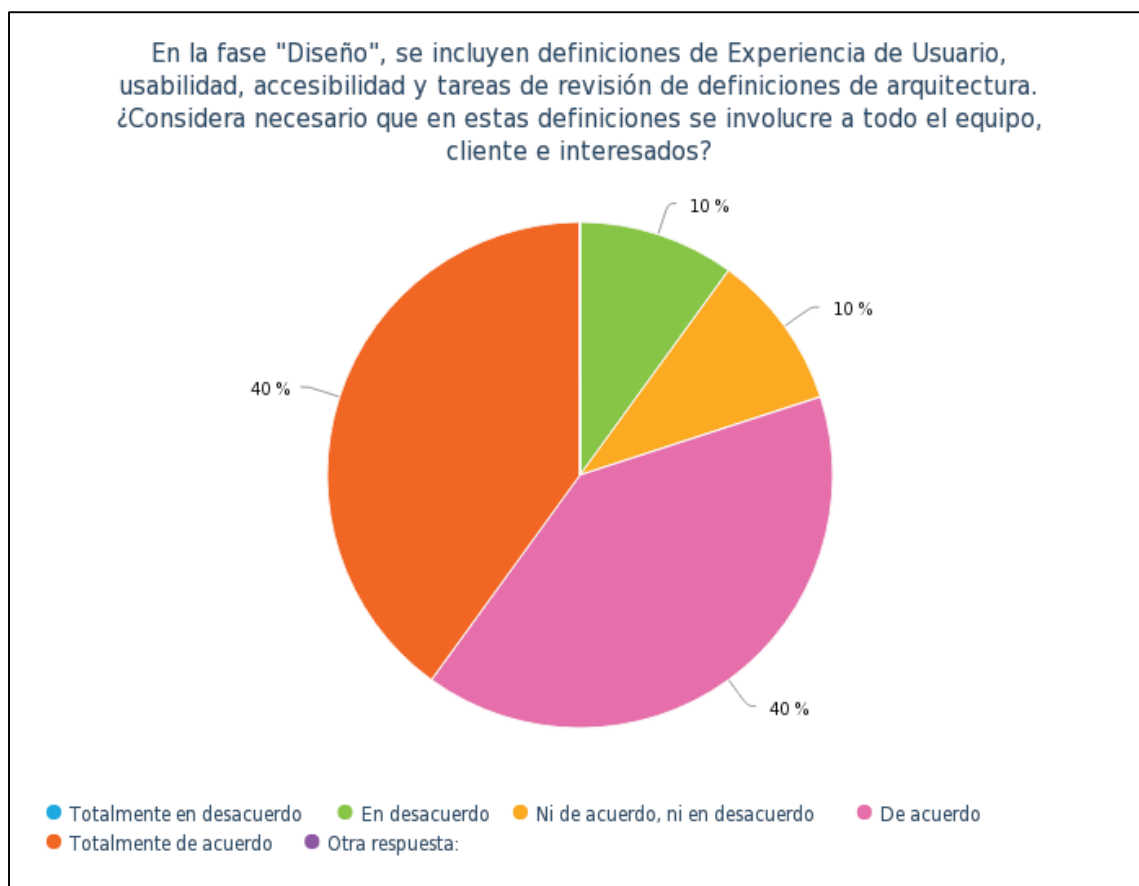
**Tabla 21**  
*Enfoque ágil*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	1	10 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	0	0 %
<b>De acuerdo</b>	6	60 %
<b>Totalmente de acuerdo</b>	3	30 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 40** y la **Tabla 21**, el 60% de la población encuestada está de acuerdo en que dentro de la fase “Gobierno” se establezca al enfoque ágil como marco de referencia y se planifique un ciclo de desarrollo seguro, un 30% está totalmente de acuerdo, mientras que en menor porcentaje el 10% está en desacuerdo.

### Pregunta 11

En la fase "Diseño", se incluyen definiciones de Experiencia de Usuario, usabilidad, accesibilidad y tareas de revisión de definiciones de arquitectura. ¿Considera necesario que en estas definiciones se involucre a todo el equipo, cliente e interesados?



**Figura 41.** Experiencia de Usuario

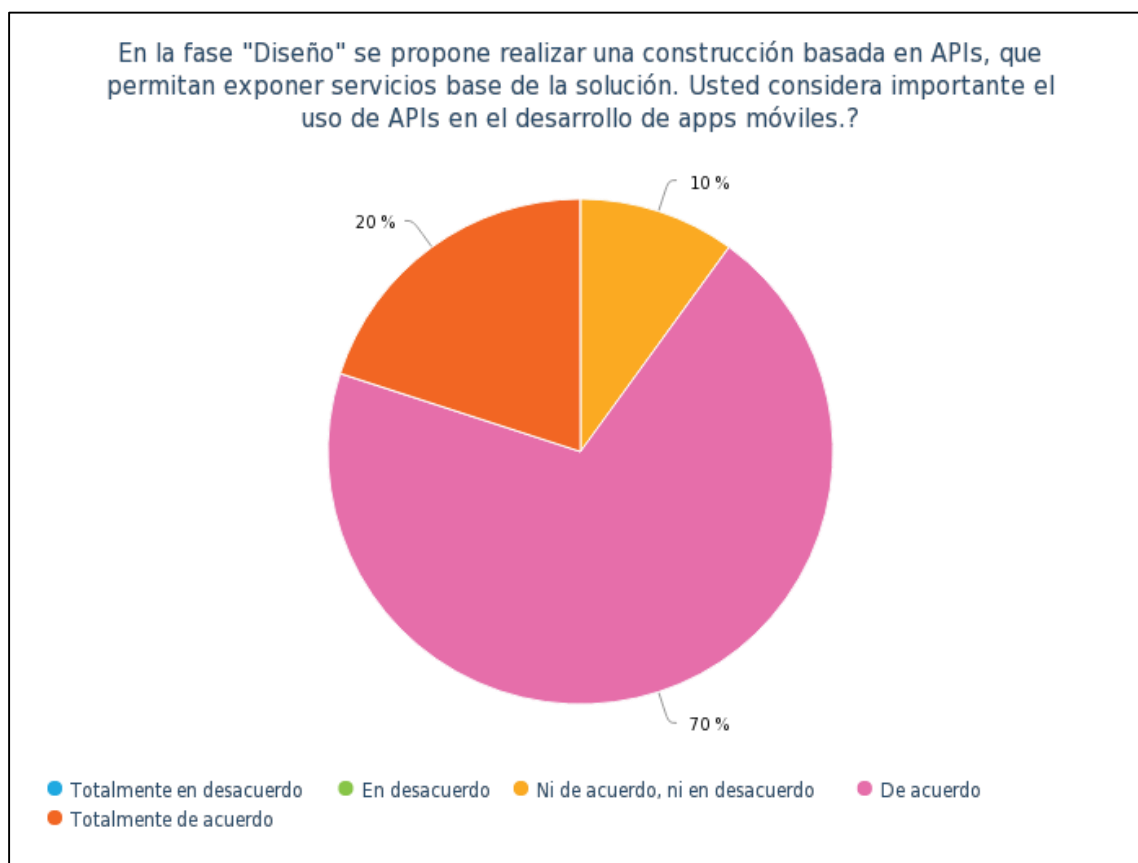
**Tabla 22**  
*Experiencia de Usuario*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	1	10 %
<b>Ni de acuerdo, ni en desacuerdo</b>	1	10 %
<b>De acuerdo</b>	4	40 %
<b>Totalmente de acuerdo</b>	4	40 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 41** y la **Tabla 22**, el 40% de la población encuestada está de acuerdo en que dentro de la fase “Diseño” se involucre a todo el equipo, cliente e interesados dentro de definiciones de experiencia de usuario, usabilidad y accesibilidad, otro 40% está totalmente de acuerdo, un 10% está ni de acuerdo, ni en desacuerdo, y mientras que en menor porcentaje otro 10% está en desacuerdo.

### Pregunta 12

En la fase "Diseño" se propone realizar una construcción basada en APIs, que permitan exponer servicios base de la solución. Usted considera importante el uso de APIs en el desarrollo de apps móviles.?



**Figura 42.** Construcción de APIs

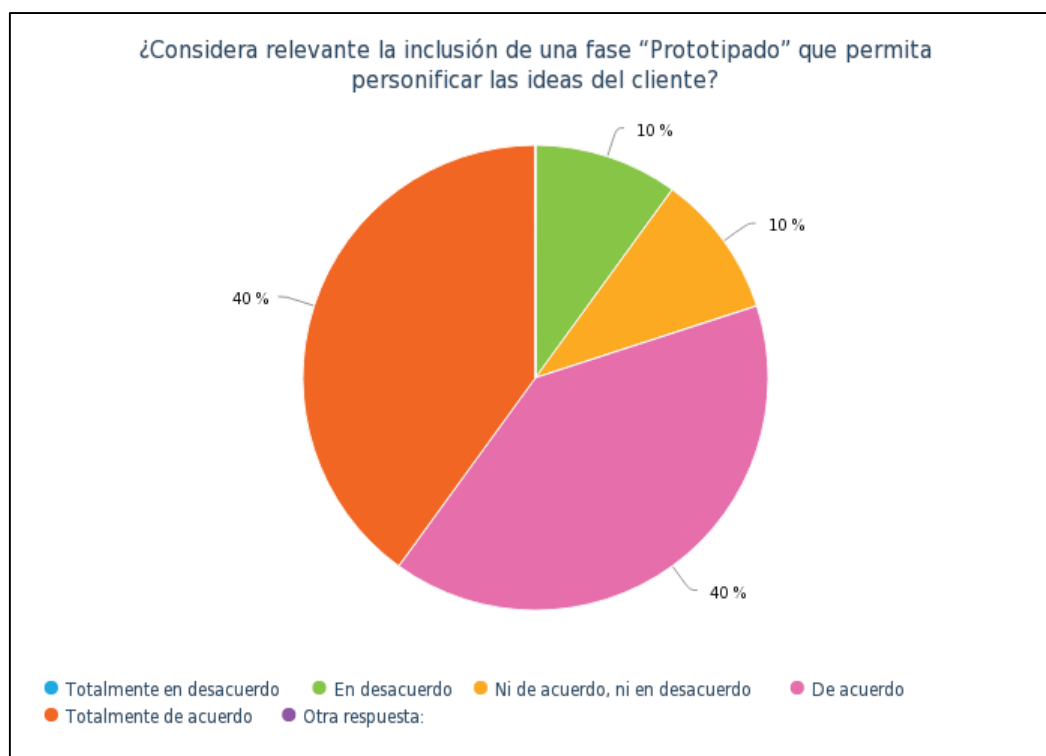
**Tabla 23**  
*Construcción de APIs*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	1	10 %
<b>De acuerdo</b>	7	70 %
<b>Totalmente de acuerdo</b>	2	20 %

Según la **Figura 42** y la **Tabla 23**, el 70% de la población encuestada está de acuerdo en que dentro de la fase “Diseño” se realice una construcción basada en APIs para exponer servicios base, un 20% está totalmente de acuerdo, un 10% está ni de acuerdo, ni en desacuerdo, mientras que ninguno de los encuestados está en desacuerdo.

### Pregunta 13

¿Considera relevante la inclusión de una fase “Prototipado” que permita personalificar las ideas del cliente?



**Figura 43.** Inclusión de Prototipado

**Tabla 24**

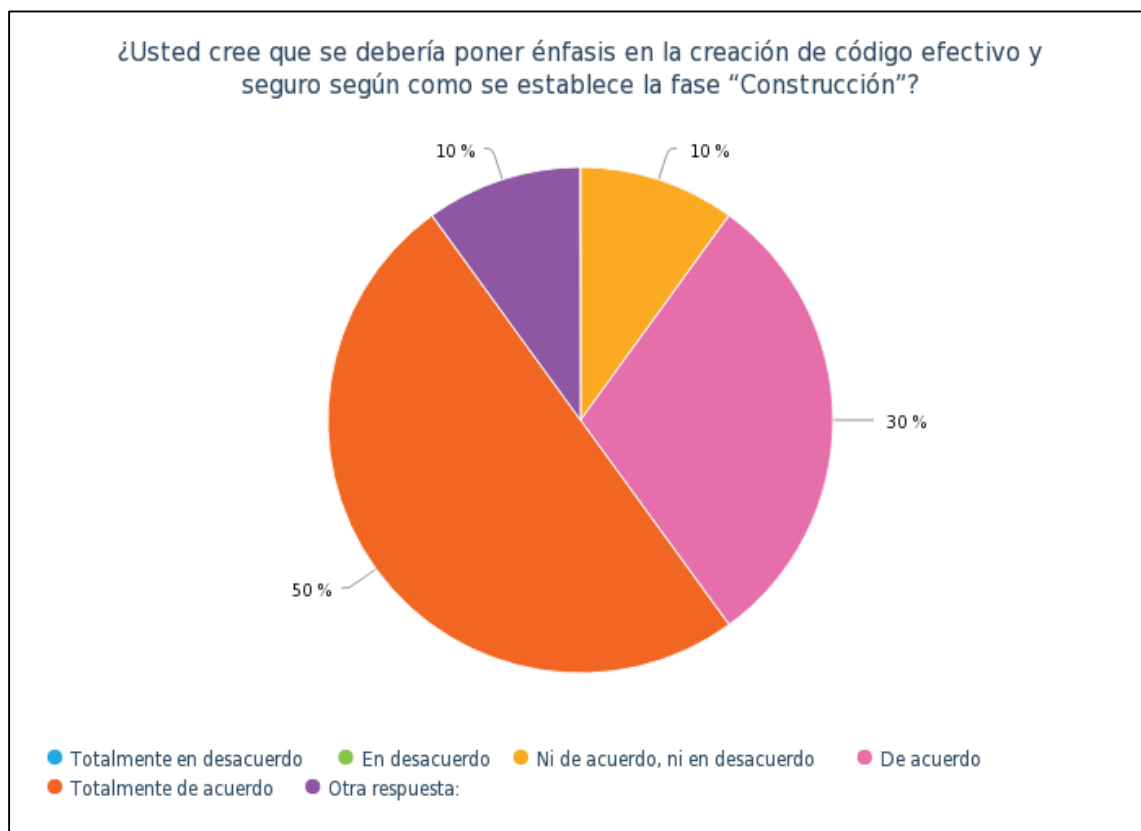
*Inclusión de Prototipado*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	1	10 %
<b>Ni de acuerdo, ni en desacuerdo</b>	1	10 %
<b>De acuerdo</b>	4	40 %
<b>Totalmente de acuerdo</b>	4	40 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 43** y la **Tabla 24**, el 40% de la población encuestada está totalmente de acuerdo en incluir una fase “Prototipado” que permita personificar las ideas del cliente, otro 40% también está de acuerdo, un 10% está ni de acuerdo, ni en desacuerdo y mientras un porcentaje menor del 10% está en desacuerdo.

#### Pregunta 14

**¿Usted cree que se debería poner énfasis en la creación de código efectivo y seguro según como se establece la fase “Construcción”?**



**Figura 44.** Creación de código



**Tabla 25**  
*Creación de código*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	1	10 %
<b>De acuerdo</b>	3	30 %
<b>Totalmente de acuerdo</b>	5	50 %
<b>Otra respuesta:</b>	1	10 %

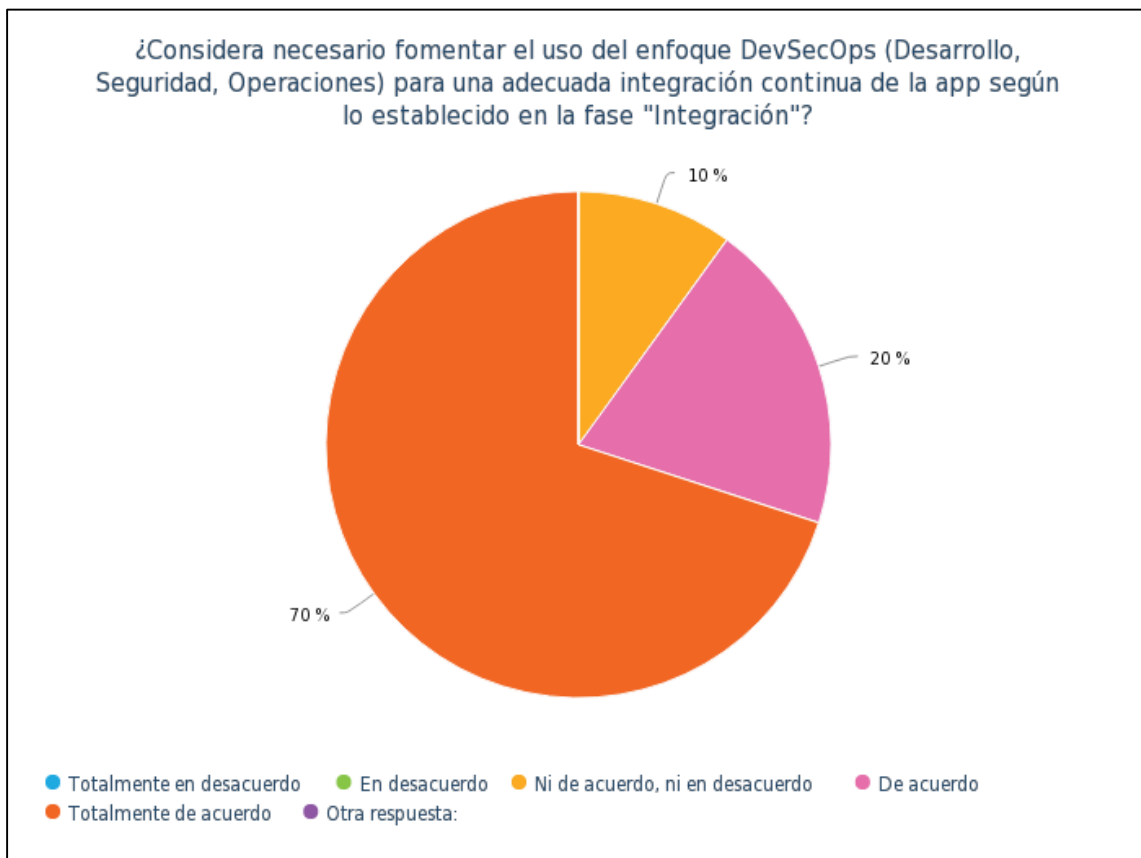
Según la **Figura 44** y la **Tabla 25**, el 50% de la población encuestada está de acuerdo en que dentro de la fase “Construcción” se debe poner énfasis en la creación de código efectivo y seguro, un 30% está de acuerdo, un 10% está ni de acuerdo, ni en desacuerdo, mientras que ninguno de los encuestados está en desacuerdo.

Entre otras respuestas tenemos:

- El término "código efectivo" puede ser subjetivo

**Pregunta 15**

**¿Considera necesario fomentar el uso del enfoque DevSecOps (Desarrollo, Seguridad, Operaciones) para una adecuada integración continua de la app según lo establecido en la fase "Integración"?**



**Figura 45.** Enfoque DevSecOps

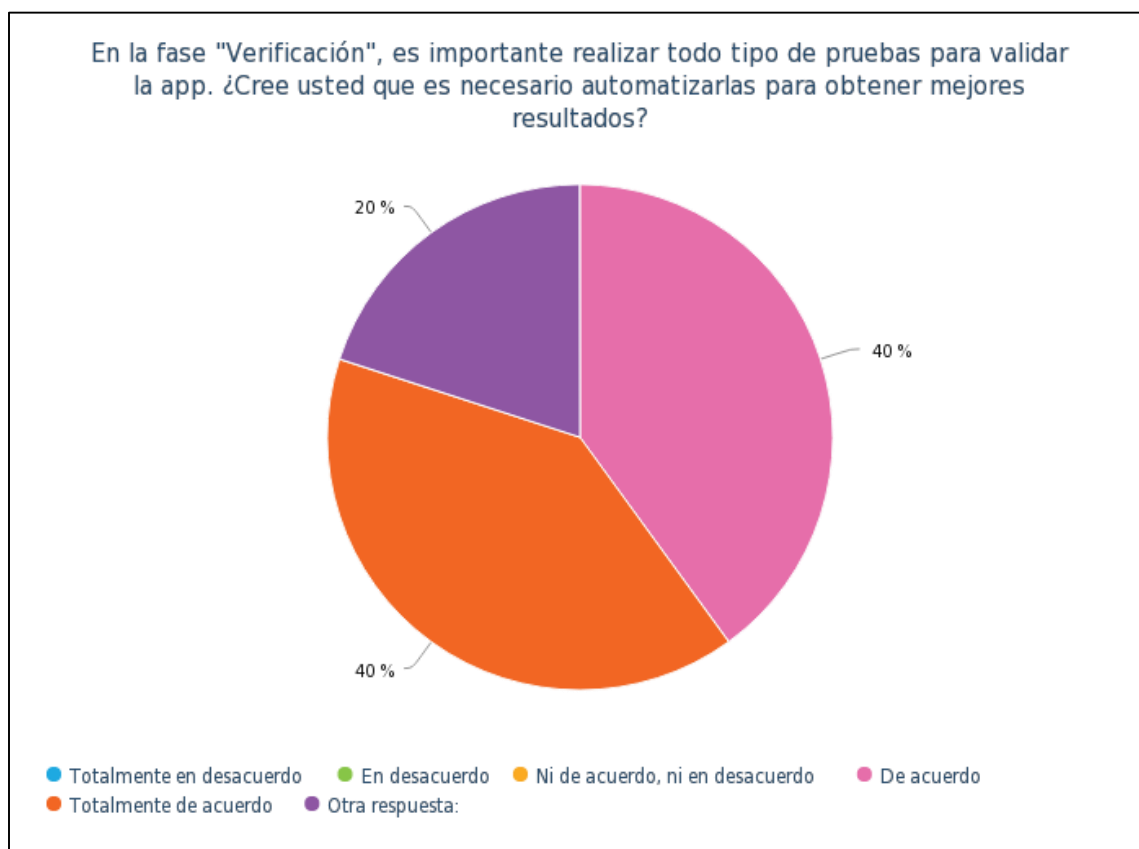
**Tabla 26**  
*Enfoque DevSecOps*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	1	10 %
<b>De acuerdo</b>	2	20 %
<b>Totalmente de acuerdo</b>	7	70 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 45** y la **Tabla 26**, el 70% de la población encuestada está totalmente de acuerdo en que dentro de la fase “Integración” se fomente el uso del enfoque DevSecOps para lograr una adecuada integración continua de la solución móvil, un 20% también está de acuerdo, un 10% está ni de acuerdo ni en desacuerdo, mientras que ninguno de los encuestados está en desacuerdo.

**Pregunta 16**

**En la fase "Verificación", es importante realizar todo tipo de pruebas para validar la app. ¿Cree usted que es necesario automatizarlas para obtener mejores resultados?**



**Figura 46.** Pruebas de calidad

**Tabla 27**  
*Pruebas de calidad*

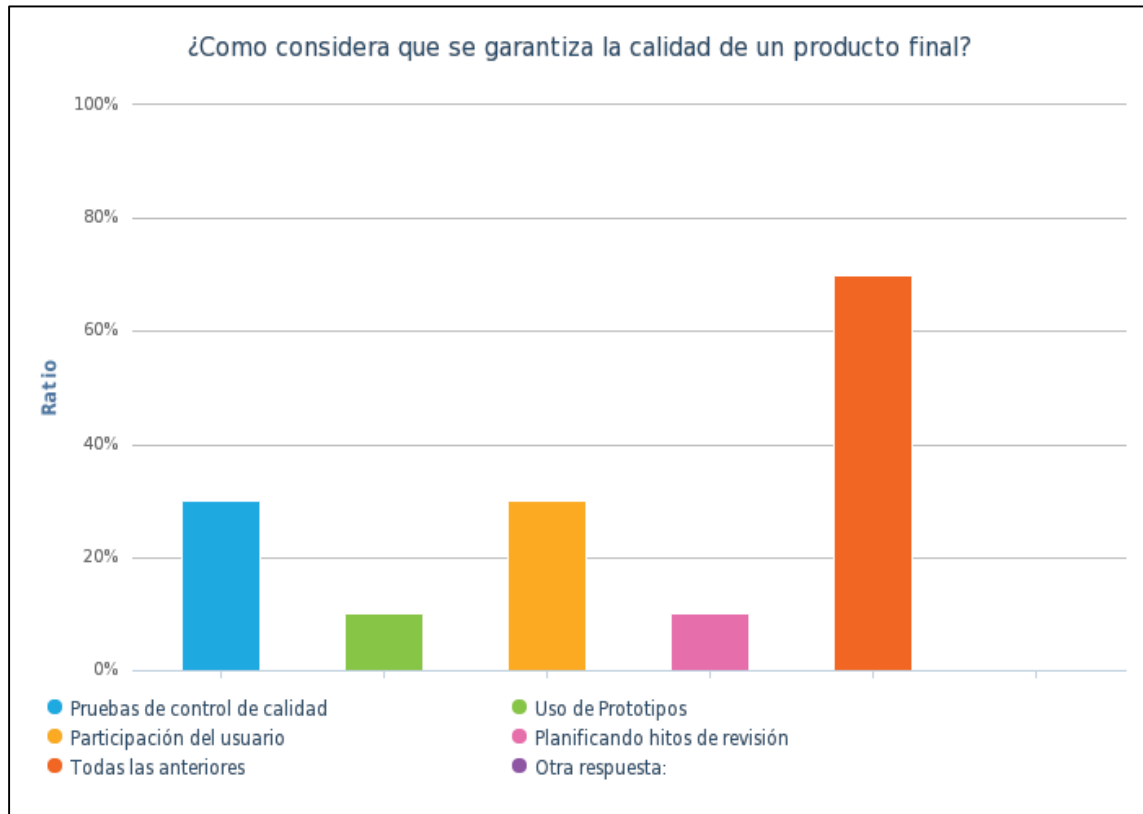
	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	0	0 %
<b>De acuerdo</b>	4	40 %
<b>Totalmente de acuerdo</b>	4	40 %
<b>Otra respuesta:</b>	2	20 %

Según la **Figura 46** y la **Tabla 27**, el 40% de la población encuestada está totalmente de acuerdo en que dentro de la fase “Verificación” se automatice las pruebas para obtener mejores resultados, otro 40% también está de acuerdo, mientras que ninguno de los encuestados está en desacuerdo. Entre otras respuestas tenemos

- En un entorno híbrido el éxito está en la elaboración de la lista de chequeo.
- Hay que ser más específicos con las pruebas que ameritan automatización, "todo tipo" es muy genérico

## Pregunta 17

¿Cómo considera que se garantiza la calidad de un producto final?



**Figura 47.** Calidad del Producto

**Tabla 28**

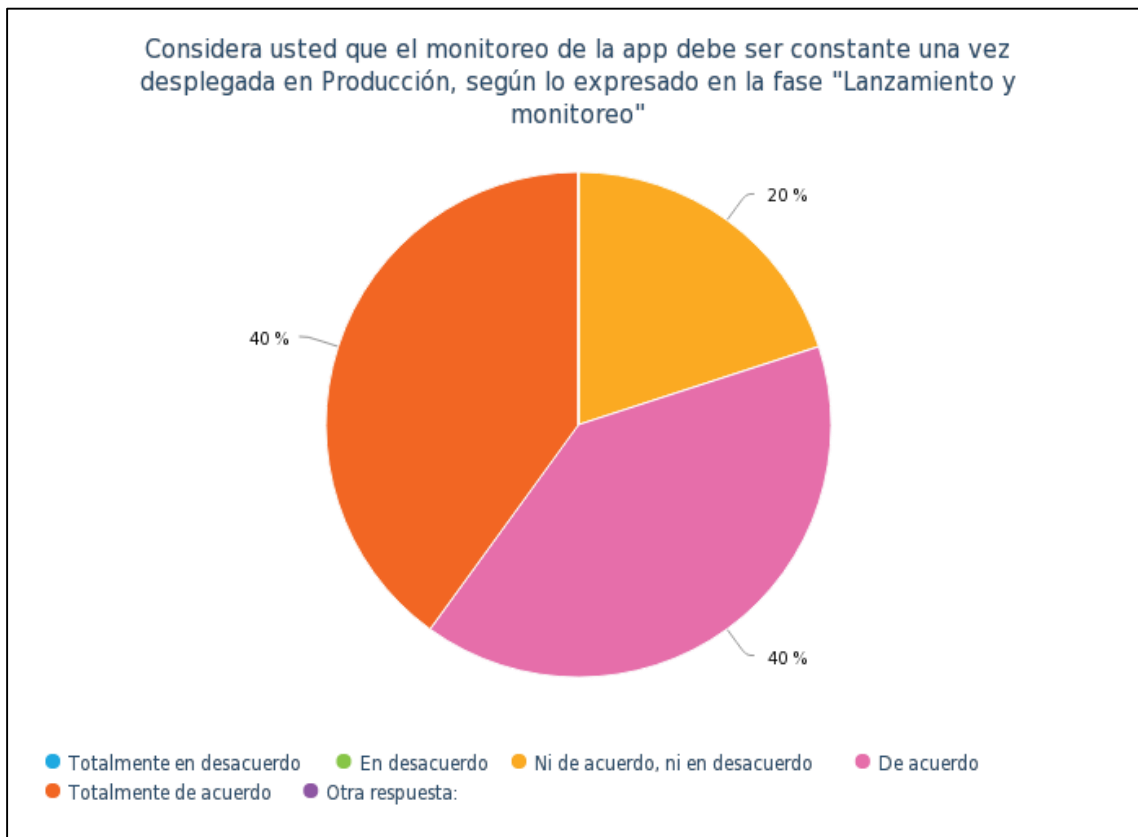
*Calidad del Producto*

	RESPUESTAS	PORCENTAJE
<b>Pruebas de control de calidad</b>	<b>3</b>	<b>30 %</b>
<b>Uso de Prototipos</b>	<b>1</b>	<b>10 %</b>
<b>Participación del usuario</b>	<b>3</b>	<b>30 %</b>
<b>Planificando hitos de revisión</b>	<b>1</b>	<b>10 %</b>
<b>Todas las anteriores</b>	<b>7</b>	<b>70 %</b>
<b>Otra respuesta:</b>	<b>0</b>	<b>0 %</b>

Según la **Figura 47** y la **Tabla 28**, el 70% de la población encuestada considera que todos los ítems enumerados son importantes para garantizar la calidad del producto final, un 30% considera que solo la “Participación del Usuario” es primordial, otro 30% considera que las “Pruebas de control” garantizan la calidad, un 10% considera primordial el “Uso de prototipos”, y mientras que otro 10% considera importante la “Planificación de hitos de revisión”.

### **Pregunta 18**

**Considera usted que el monitoreo de la app debe ser constante una vez desplegada en Producción, según lo expresado en la fase "Lanzamiento y monitoreo"**



**Figura 48.** Monitoreo de App



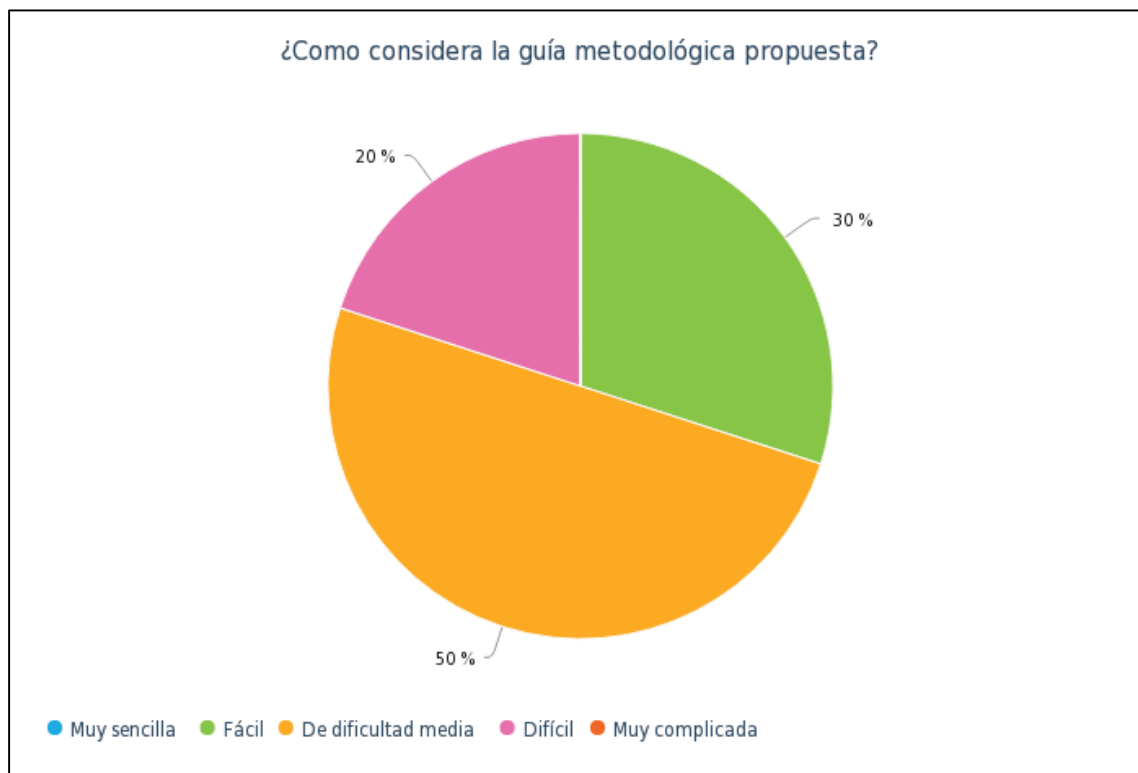
**Tabla 29**  
*Monitoreo de App*

	RESPUESTAS	PORCENTAJE
<b>Totalmente en desacuerdo</b>	0	0 %
<b>En desacuerdo</b>	0	0 %
<b>Ni de acuerdo, ni en desacuerdo</b>	2	20 %
<b>De acuerdo</b>	4	40 %
<b>Totalmente de acuerdo</b>	4	40 %
<b>Otra respuesta:</b>	0	0 %

Según la **Figura 48** y la **Tabla 29**, el 40% de la población encuestada está de acuerdo en que dentro de la fase “Lanzamiento y Monitoreo” se realice un monitoreo constante de las aplicaciones una vez desplegadas en producción, otro 40% está totalmente de acuerdo, un 20% está ni de acuerdo, ni en desacuerdo, mientras que ninguno de los encuestados está en desacuerdo.

## Pregunta 19

¿Cómo considera la guía metodológica propuesta?



**Figura 49.** Calificación de la guía

**Tabla 30**

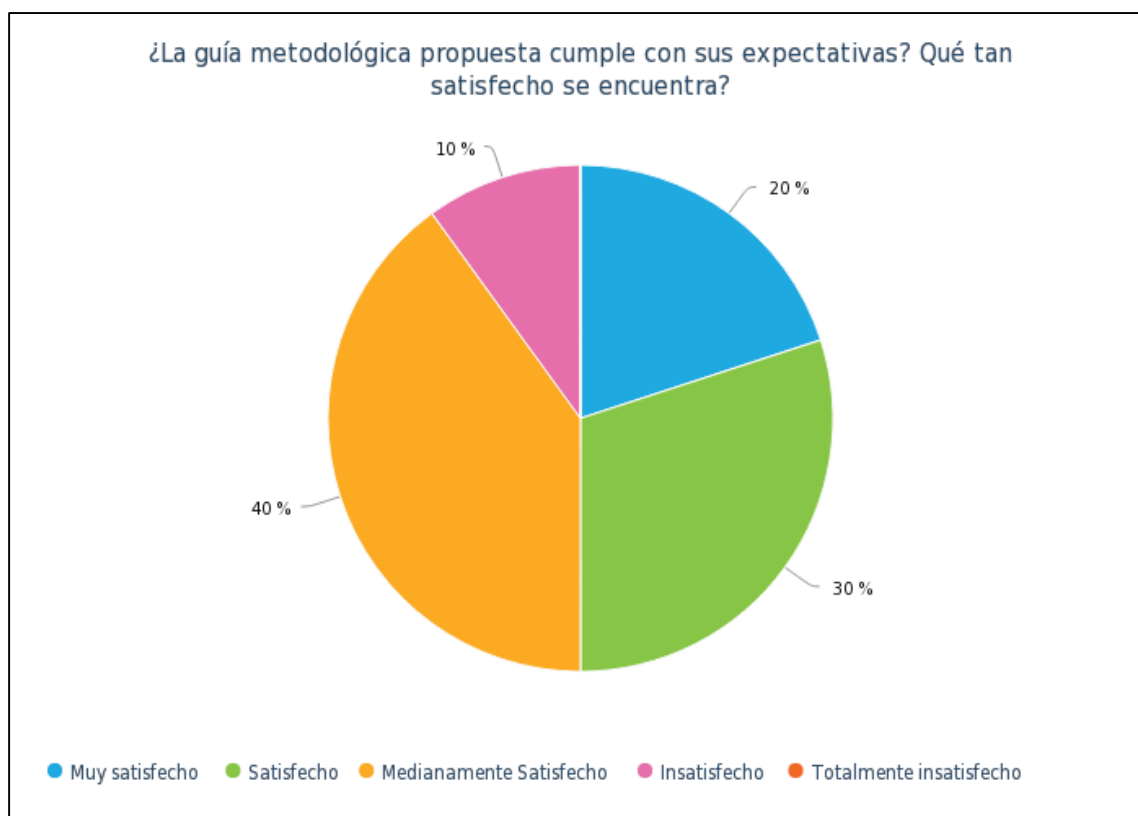
*Calificación de la guía*

	RESPUESTAS	PORCENTAJE
<b>Muy sencilla</b>	0	0 %
<b>Fácil</b>	3	30 %
<b>De dificultad media</b>	5	50 %
<b>Difícil</b>	2	20 %
<b>Muy complicada</b>	0	0 %

Según la **Figura 49** y la **Tabla 30**, el 50% de la población encuestada considera que la guía metodológica propuesta es de dificultad media, un 20% con un grado difícil, un 30% de grado fácil, y mientras que ninguno de los encuestados respondió como muy complicada.

### Pregunta 20

**¿La guía metodológica propuesta cumple con sus expectativas? ¿Qué tan satisfecho se encuentra?**



**Figura 50.** Nivel de Satisfacción de la guía

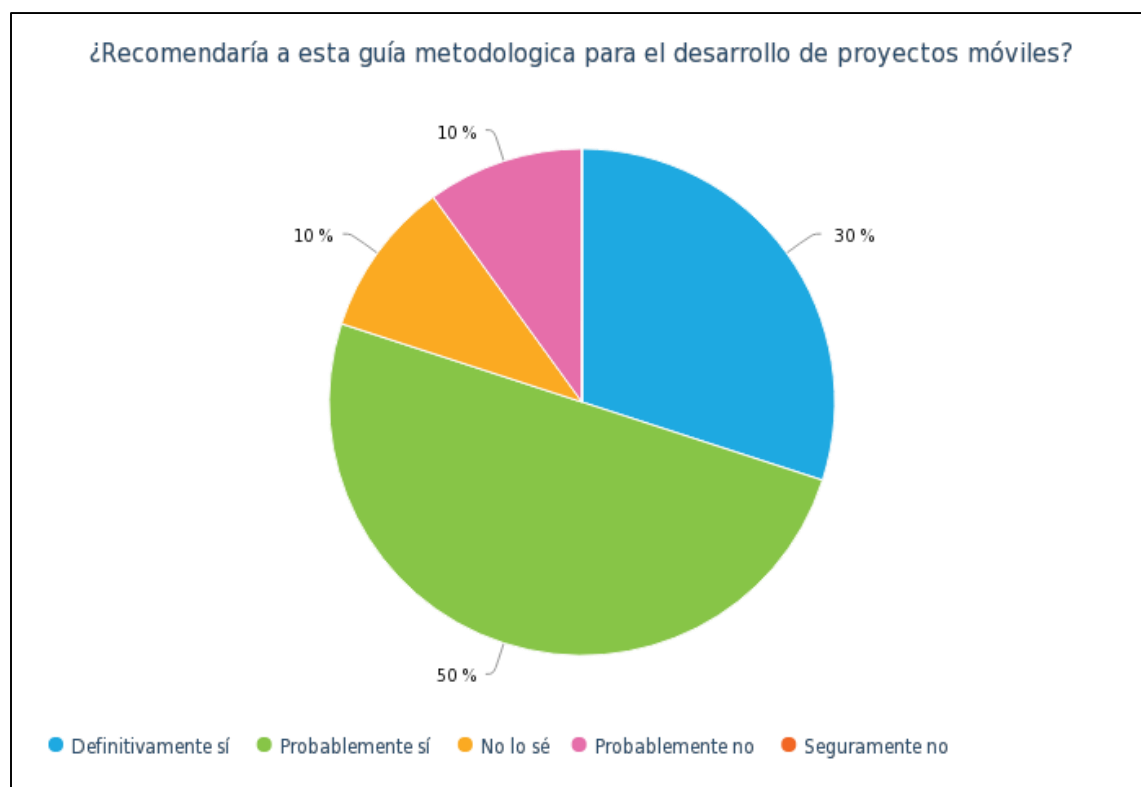
**Tabla 31***Nivel de Satisfacción de la guía*

	RESPUESTAS	PORCENTAJE
<b>Muy satisfecho</b>	2	20 %
<b>Satisfecho</b>	3	30 %
<b>Medianamente Satisfecho</b>	4	40 %
<b>Insatisfecho</b>	1	10 %
<b>Totalmente insatisfecho</b>	0	0 %

Según la **Figura 50** y la **Tabla 31**, el 40% de la población encuestada están medianamente satisfechos con la guía metodológica propuesta, un 30% están satisfechos, un 20% están muy satisfechos, un porcentaje menor del 10% están insatisfechos, es decir para una menor parte de la población la guía no cumple con las expectativas, y mientras que ninguno de los encuestados respondió con “totalmente insatisfecho”.

### Pregunta 21

¿Recomendaría a esta guía metodológica para el desarrollo de proyectos móviles?



**Figura 51.** Recomendación de la guía

### Tabla 32

Recomendación de la guía

	RESPUESTAS	PORCENTAJE
<b>Definitivamente sí</b>	3	30 %
<b>Probablemente sí</b>	5	50 %
<b>No lo sé</b>	1	10 %
<b>Probablemente no</b>	1	10 %
<b>Seguramente no</b>	0	0 %

Según la **Figura 74** y la **Tabla 33**, el 50% de la población encuestada probablemente si recomendaría la guía metodológica para el desarrollo de proyectos móviles, un 30% definitivamente si la recomendaría, un 10% no lo sabe, una menor parte con un 10% probablemente no lo haría, y mientras que ninguno de los encuestados respondió con “seguramente no”.

## **Análisis Pregunta 22**

### **¿Cómo se podría mejorar o que recomendaciones daría a la presente guía propuesta?**

La pregunta 22 es de tipo abierta, por ende, las respuestas se enlistan a continuación:

- Demostración práctica. Entregables en cada fase. Modelos de documentos que no existe en ninguna metodología
- Un resumen técnico un poco más conciso.
- Se puede crear un resumen de la metodología para que los desarrolladores la tengan presente en cada momento, así mismo agregar un poco más la definición de los roles del equipo de trabajo para que todos puedan trabajar enfocados en un mismo fin.
- Al final siempre se termina combinando modelos, ya que la flexibilidad permite dar versatilidad y valor agregado al negocio.

- La guía cubre todas las etapas del desarrollo de una aplicación móvil, que generalmente no se encuentran detalladas en ninguna metodología.

## REFERENCIAS BIBLIOGRÁFICAS

- Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jääliñoja, J., Korkala, M., . . . Salo, O. (24-28 de Octubre de 2004). Mobile-D: an agile approach for mobile application development. *In Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, 174-175. doi:10.1145/1028664.1028736
- Agilemanifesto.org. (s.f.). *Manifiesto por el Desarrollo Ágil de Software*. Recuperado el 15 de Julio de 2017, de <http://agilemanifesto.org/iso/es/manifiesto.html>
- Alaimo, D. M. (2013). *Proyectos ágiles con Scrum : flexibilidad, aprendizaje,.* Ciudad Autónoma de Buenos Aires: Kleer EBook.
- Alarcón, J. (Agosto de 2016). *Qué son las Aplicaciones Web Progresivas o "Progressive Web Apps"*. Recuperado el 1 de Julio de 2017, de <https://www.campusmvp.es/recursos/post/Que-son-las-Aplicaciones-Web-Progresivas-o-Progressive-Web-Apps.aspx>
- Amaya, Y. (23 de Julio de 2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología*, 12(2), 111-124. Recuperado el 12 de Enero de 2018
- Amrhein, D. M. (2013). *Developing native mobile apps with IBM Worklight*. Recuperado el 2 de Julio de 2017, de [https://www.ibm.com/developerworks/websphere/techjournal/1310\\_amrhein/1310\\_amrhein.html](https://www.ibm.com/developerworks/websphere/techjournal/1310_amrhein/1310_amrhein.html)
- Apple, I. (2017). *iPhone SE*. Recuperado el 12 de Junio de 2017, de <https://www.apple.com/es/iphone-se/>
- Artigas, S. G. (11 de 12 de 2015). *Lean UX, una nueva metodología para la experiencia de usuario*. Recuperado el 23 de 2 de 2018, de <http://www.torresburriel.com/weblog/2015/12/11/lean-ux/>
- Belucci, M. (2017). *5G: la tecnología móvil que va a revolucionar la vida cotidiana*. Recuperado el 12 de Junio de 2017, de [https://www.clarin.com/sociedad/bancha-ancha-movil-5g-internet-cosas-big-data\\_0\\_Hy3tV6T9e.html](https://www.clarin.com/sociedad/bancha-ancha-movil-5g-internet-cosas-big-data_0_Hy3tV6T9e.html)
- BlackBerry. (2017). Recuperado el 18 de Junio de 2017, de [http://us.blackberry.com/content/dam/bbCompany/Desktop/Global/media-gallery/Logos/BlackBerry\\_10\\_logo.png](http://us.blackberry.com/content/dam/bbCompany/Desktop/Global/media-gallery/Logos/BlackBerry_10_logo.png)



- Cantú, A. (2016). *Herramientas imprescindibles para UX Designers*. Recuperado el 2 de Agosto de 2017, de <http://blog.acantu.com/9-herramientas-ux-designers/>
- Chandra, P. (s.f). *Software Assurance Maturity Model, Una guía para integrar seguridad en el desarrollo de software*. OWASP, The Open Web Application Security Project. Recuperado el 05 de 10 de 2017, de <http://www.opensamm.org/>
- Corral, L., Sillitti, A., & Succi, G. (25 de Mayo de 2015). *Software Development Processes for Mobile Systems, Is Agile Really Taking Over the Business?* Recuperado el 12 de Agosto de 2017, de <http://www.sei.cmu.edu/community/mobs2013/upload/MOBS2013-Corral.pdf>
- Cruz, R. (2014). *Sistema Móvil De Sondeo Preventivo De Vehículos Con Soporte Obdi Para Mejorar La Vida Útil Del Automotor*. Recuperado el 15 de Julio de 2017, de Metodología: <http://pegasus.javeriana.edu.co/~PA133-05-PMovVidaAutomotor/Metodologia.html>
- Curran, P. (2016). *Hybrid versus Native Mobile App Development: Methodologies, Risks and Tools*. Recuperado el 20 de Agosto de 2017, de <https://www.checkmarx.com/2016/08/10/2016081020160809hybrid-versus-native-mobile-development>
- Fling, B. (14 de Agosto de 2009). *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps*. (O. Media, Editor) Recuperado el 5 de Junio de 2017, de [https://books.google.com.ec/books/about/Mobile\\_Design\\_and\\_Development.html?id=LyMeulBTkH0C&redir\\_esc=y](https://books.google.com.ec/books/about/Mobile_Design_and_Development.html?id=LyMeulBTkH0C&redir_esc=y)
- Flora, H. K., Chande, S. V., & Wang, X. (Mayo de 2014). Adopting an agile approach for the development of mobile applications. *International Journal of Computer Applications*, 94(17), 975 - 8887. Recuperado el 21 de Agosto de 2017, de <https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjEp9vOtoDWAhUFeCYKHbAwBO4QFgg5MAA&url=http%3A%2F%2Fresearch.ijcaonline.org%2Fvolume94%2Fnumber17%2Fpxc3896199.pdf&usg=AFQjCNGGri1EAq2KQekwwnaWYUmLCXOUdA>
- Fred. (3 de noviembreN de 2014). *El ciclo de vida del desarrollo de software*. Recuperado el 30 de Agosto de 2017, de <https://velneo.es/el-ciclo-de-vida-del-desarrollo-de-software/>
- Ganesh, P., Hamsini, R., & Smitha, G. (Septiembre de 2016). Agile Development Methodology and Testing for Mobile Applications-A Survey. *International Journal of New Technology and Research (IJNTR)*, 98-101. Recuperado el 24 de Junio de 2017, de [https://www.ijntr.org/download\\_data/IJNTR02090047.pdf](https://www.ijntr.org/download_data/IJNTR02090047.pdf)

- Gasca, M. C., Camargo, L. L., & Medina, B. (2014). Metodología para el desarrollo de aplicaciones móviles. *Revista Tecnura*, 18, 18(40), 20-35. Recuperado el 23 de Junio de 2017, de <http://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>
- Gisbert, J. (2017). *Google presenta Android Go, Android para smartphones con menos de 1GB de RAM*. Recuperado el 20 de Junio de 2017, de <https://andro4all.com/2017/05/android-go-para-smartphones-menos-1-gb-ram>
- Glissmann, S., Smolnik, S., Schierholz, R., Kolbe, L., & Brenner, W. (11-13 de Julio de 2005). Proposition of an M-Business Procedure Model for the Development of Mobile User Interfaces. (IEEE, Ed.) *International Conference on Mobile Business*. doi:10.1109/ICMB.2005.83
- Gómez, M. (2016). *TOP 5 de plataformas de desarrollo iOS y Android*. Recuperado el 3 de Julio de 2017, de <http://cice.es/noticia/top-5-plataformas-desarrollo-ios-android/>
- Google. (2014). *Android*. Recuperado el 12 de Junio de 2017, de <https://android.com>
- Guevara, A. (2017). *Universidad Nacional Autónoma de México*. Recuperado el 16 de Junio de 2017, de Dispositivos Móviles: <https://revista.seguridad.unam.mx/node/2237>
- hernan. (23). 23. 2323: 23.
- IEBS, C. (23 de 2 de 2018). *Pilares y principios de Lean UX*. Obtenido de <https://comunidad.iebschool.com/vicman/2016/04/26/pilares-y-principios-de-lean-ux/>
- Institute, I. F. (1 de 10 de 2013). *¿QUÉ ES EL DESIGN THINKING?* Recuperado el 23 de 2 de 2018, de <https://www.innovationfactoryinstitute.com/blog/que-es-el-design-thinking/>
- José H. Canós, M. C. (13 de 3 de 2012). *Métodologías Ágiles en el Desarrollo de Software*. Recuperado el 26 de 2 de 2018, de <http://roa.ult.edu.cu/handle/123456789/476>
- Kobayashi, A. (2015). *Lean UX : deseing thinking, lean startup, agile : la combinación para crear startup exitosos*. Universidad Autónoma del Perú. Recuperado el 23 de 2 de 2018, de <http://repositorio.autonoma.edu.pe/handle/AUTONOMA/317>
- Leal, B. (2017). *Principios de la metodología Lean UX*. Recuperado el 12 de Julio de 2017, de <https://www.ida.cl/blog/estrategia-digital/principios-metodologia-lean-ux/>

- Lee, C. (3 de Agosto de 2014). *Gartner Says Traditional Development Practices Will Fail for Mobile Apps*. Recuperado el 1 de Junio de 2017, de <http://www.gartner.com/newsroom/id/2823619>
- Leiva, I., & Villalobos, M. (2015). Método ágil híbrido para desarrollar software en dispositivos móviles. *Revista Chilena de Ingeniería*, 473-488. Recuperado el 2 de Junio de 2017, de <http://www.redalyc.org/articulo.oa?id=77241115016>
- Leow, A., Baker, V., Marshall, R., Revang, M., & Wong, J. (Diciembre de 2016). *Predicts 2017: Mobile Apps and Their Development*. Recuperado el 17 de Junio de 2017, de <https://www.gartner.com/doc/3525665>
- Luzardo, I. (14 de Junio de 2016). *10 características de los actuales súper celulares*. Obtenido de <http://www.enter.co/chips-bits/smartphones/10-caracteristicas-de-los-actuales-super-celulares/>
- Mahmud, D., & Abdullah, A. (2015). *Agile: A Natural Fit in Mobile Application Development Process*. Recuperado el 12 de Junio de 2017, de [https://www.academia.edu/27725041/Agile\\_A\\_Natural\\_Fit\\_in\\_Mobile\\_Application\\_Development\\_Process](https://www.academia.edu/27725041/Agile_A_Natural_Fit_in_Mobile_Application_Development_Process)
- Mechato, J., & Gastañadui, C. (s.f). *Metodologías ágiles para el desarrollo de aplicaciones móviles*. Recuperado el 14 de Julio de 2017, de <http://www.academia.edu/16347741/Metodologias-Agiles>
- Menz, S. (2016). *UX en 30 Minutos*. Recuperado el 26 de Julio de 2017, de <https://es.slideshare.net/solmesz/ux-en-30-minutos>
- Microsoft, C. (6 de Junio de 2017). *Windows 10*. Obtenido de <https://www.microsoft.com/es-es/moviles/windows10/>
- Negar, M. (2013). *Y shaped methodology driving modern mobile development techniques*. Recuperado el 14 de Julio de 2017, de <http://mobileapptechy.blogspot.com/2013/02/y-shaped-methodology-driving-modern.html>
- Neoland. (Diciembre de 2016). *UI/UX/IxD ¿Qué es el diseño de experiencia de usuario/interfaz/interacción?* Recuperado el 28 de Julio de 2017, de <https://www.neoland.es/uiuxixd-diseno-experiencia-usuariointerfazinteraccion/>
- Ordoñez Ortiz, J. F., & Valle Caicedo, L. E. (2014). *Diseño de un marco de referencia para el emprendimiento de negocios de TI basado en la filosofía "Lean Startup*. (F. d. UDLA, Ed.) Quito, Ecuador. Recuperado el 12 de 1 de 2018, de <http://dspace.udla.edu.ec/handle/33000/3059>

- Ordoñez, J. (s.f.). Diseño de un marco de referencia para el emprendimiento de negocios de TI basado en la filosofía Lean Startup. Recuperado el 12 de 1 de 2018
- Otero, A., Martínez, J., & Díaz, J. (s.f). *Revista de Transformación Educativa*. Recuperado el 2 de Julio de 2017, de Metodología de desarrollo de aplicaciones para el aprendizaje móvil basadas en software libre: <http://www.transformacion-educativa.com/attachments/article/149/RETE%20-%20tem%C3%A1tico%20Educaci%C3%B3n%20mediada%20por%20Tecnolog%C3%ADa.pdf>
- Otherwise. (2012). *Diferenciando entre un sketch, mockup, wireframe y prototipo*. Recuperado el 12 de Julio de 2017, de <http://www.otherwiseonline.net/diferencias-entre-sketch-mockup-wireframe-prototipo/>
- OWASP, F. (2016). *Mobile Top 10 2016-Top 10*. Recuperado el 13 de 10 de 2017, de [https://www.owasp.org/index.php/Mobile\\_Top\\_10\\_2016-Top\\_10](https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10)
- OWASP, F. (2017). *OWASP Top 10 Application Security Risks - 2017*. Recuperado el 13 de 10 de 2017, de [owasp: https://www.owasp.org/index.php/Top\\_10\\_2017-Top\\_10](https://www.owasp.org/index.php/Top_10_2017-Top_10)
- Pettey, C. (Octubre de 2011). *Gartner Identifies the Top 10 Strategic Technologies for 2012*. Recuperado el 1 de Junio de 2017, de <http://www.gartner.com/newsroom/id/1826214>
- Picone, K. (2013). *Waterfall vs. Agile Development: How They Differ and Why it Matters*. Recuperado el 12 de Julio de 2017, de <https://www.business-software.com/blog/waterfall-vs-agile-development-differ-matters/>
- Pizarro, N. (5 de 7 de 2017). *Metodología de diseño de soluciones*. Recuperado el 23 de 2 de 2018, de <https://www.ida.cl/blog/estrategia-digital/que-es-el-design-thinking/>
- Quer, F. (2016). *10 conceptos clave para entender la experiencia de usuario*. Recuperado el 25 de Julio de 2017, de <http://kingeclient.com/blog/10-conceptos-clave-para-entender-la-experiencia-de-usuario>
- Redwood, C. (Julio de 2016). *Hybrid Vs. Native Mobile Apps*. Recuperado el 20 de Julio de 2017, de <https://www.linkedin.com/pulse/hybrid-vs-native-mobile-apps-charlie-redwood>
- Referente, E. (02 de 12 de 2014). *Las empresas utilizan el crecimiento de las aplicaciones móviles para posicionar sus marcas*. Recuperado el 3 de Septiembre de 2017, de <http://www.elreferente.es/tecnologicos/las-empresas->

utilizan-el-crecimiento-de-las-aplicaciones-moviles-para-posicionar-sus-marcas-27961

- Rodríguez, T. (2011). *Métodos aplicables para el desarrollo de aplicaciones móviles*. Recuperado el 13 de Julio de 2017, de <https://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>
- Santa María, L. (Julio de 2015). *5 formas para evitar fracasar en el desarrollo de aplicaciones móviles*. Recuperado el 2 de Junio de 2017, de <https://www.staffdigital.pe/blog/5-formas-para-evitar-fracasar-en-el-desarrollo-de-aplicaciones-moviles/>
- Sarango, S. (2017). Propuesta de una metodología para el desarrollo de aplicaciones móviles en el campo educativo (Mobile Learning) para la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja. Loja, Ecuador. Recuperado el 10 de 10 de 2017, de <http://dspace.unl.edu.ec:9001/jspui/bitstream/123456789/19040/1/Sarango%20Palacios%2C%20Stalin%20Joe.pdf>
- Scribd. (s.f). *IEEE 1074*. Recuperado el 4 de Junio de 2017, de <https://www.scribd.com/doc/178784956/IEEE-1074>
- Scrum.org. (s.f). *Scrum*. Recuperado el 26 de Agosto de 2017, de <http://metodologiascrum.readthedocs.io/en/latest/Scrum.html>
- Symbian, F. (s.f). *Symbian*. Recuperado el 15 de Junio de 2017, de <http://licensing.symbian.org/>
- Tecnología, Á. (s.f). *Sistemas Operativos Móviles*. Recuperado el 15 de Junio de 2017, de <http://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html>
- Universo, E. (31 de Marzo de 2014). *Supermercado crea un asistente virtual y móvil*. Recuperado el 11 de Septiembre de 2017, de <http://www.eluniverso.com/noticias/2014/03/31/nota/2537041/supermercado-crea-asistente-virtual-movil>
- Villagrán, I. R. (2013). *Procesos en el desarrollo de aplicaciones para dispositivos móviles*. Posgrado en Ciencia e Ingeniería de la Computación. Doctoral dissertation, tesis de maestría, Posgrado en Ciencia e Ingeniería de la Computación, UNAM, México. Recuperado el 12 de Agosto de 2017, de <http://132.248.9.195/ptd2013/abril/300218002/300218002.pdf>
- Vique, R. R. (2012). *Métodos para el desarrollo de aplicaciones móviles*, PID\_00176755. Recuperado el 1 de Septiembre de 2017, de Universidad Oberta de Catalunya:

[https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia\\_y\\_desarrollo\\_en\\_dispositivos\\_moviles/Tecnologia\\_y\\_desarrollo\\_en\\_dispositivos\\_moviles\\_\(Modulo\\_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)

Withers, D. H. (2000). Software engineering best practices applied to the modeling process. *Simulation Conference, 2000. Proceedings. Winter. 1*, págs. 432-439. Orlando, FL, USA, USA: IEEE. doi:10.1109/WSC.2000.899749

Zeidler, C., Kittl, C., & Petrovic, O. (9-11 de Julio de 2007). An integrated product development process for mobile software. (IEEE, Ed.) *International Conference on the Management of Mobile Business, 2007. ICMB 2007*. doi:10.1109/ICMB.2007.12