



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS
EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA**

José Eliezer Pop Estrada

Asesorado por el Ing. Moisés Eduardo Velásquez Oliva

Guatemala, noviembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS
EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

José Eliezer Pop Estrada

ASESORADO POR EL ING. MOISÉS EDUARDO VELÁSQUEZ OLIVA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Ángel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Óscar Humberto Galicia Núñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. William Samuel Guevara
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS
EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha junio de 2016.


José Eliezer Pop Estrada

Guatemala, 22 de Junio de 2017

Ingeniero Carlos Azurdía
Tutor de trabajos de graduación
Escuela de Ciencias y Sistemas
Facultad de Ingeniería

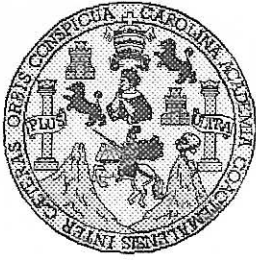
Respetable Ingeniero Azurdía:

Por este medio le informo, que como asesor del trabajo de graduación del estudiante universitario de la carrera de Ingeniería en Ciencias y Sistemas, José Eliezer Pop Estrada, carné 201114501, he revisado el protocolo y los capítulos de aporte del trabajo de graduación titulado: "Enfoque sistemático y práctico del desarrollo de aplicaciones híbridas en móviles, aplicadas a la pequeña y mediana empresa en Guatemala", y a mi criterio el mismo está completo según los alcances de su investigación y por lo tanto lo doy por aprobado.

Agradeciendo su atención a la presente, y sin otro particular me suscribo de usted, atentamente,



Ing. Moisés Eduardo Velásquez Oliva
Asesor de trabajo de graduación
Colegiado: 11,704



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 23 de Agosto de 2017


Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JOSÉ ELIEZER POP ESTRADA** con carné 201114501 y CUI 2113 25775 0101, titulado **"ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



ESCUELA DE INGENIERIA EN CIENCIAS Y SISTEMAS


UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA EN
CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA**, realizado por el estudiante **JOSÉ ELIEZER POP ESTRADA**, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"


Ing. *Martín Antonio Pérez Tur*
Director

Escuela de Ingeniería en Ciencias y Sistemas



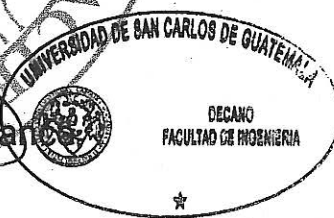
Guatemala, 03 de noviembre de 2017



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ENFOQUE SISTEMÁTICO Y PRÁCTICO DEL DESARROLLO DE APLICACIONES HÍBRIDAS EN MÓVILES, APLICADAS A LA PEQUEÑA Y MEDIANA EMPRESA EN GUATEMALA**, presentado por el estudiante universitario: **José Eliezer Pop Estrada**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Pedro Antonio Aguilar Polanco
Decano



Guatemala, noviembre de 2017

ACTO QUE DEDICO A:

Dios

Por guardarme y protegerme en el arduo camino, ser mi fortaleza y esperanza, en quien siempre confiaré, la pieza más importante en mi vida.

Mi madre

Alicia Estrada Martin, por ser al mayor ejemplo de perseverancia y esfuerzo, por sacrificar tanto y nunca dejarme solo, mis palabras no son lo suficiente para expresar lo agradecido que estoy.

Mi abuela

Irene Martin, por su confianza y amor incondicional.

Mis tíos

Toribio Estrada Martin, María Telma y María Adela Estrada, por todo el apoyo brindado en los momentos difíciles, además de los consejos que hicieron de mí una mejor persona.

Mis amigos

Francisco Luis Eduardo Illescas Rodriguez, Daniel Alejandro Rosal Tobar, Carlos Humberto Armas Ruiz, Adolfo Alejandro Sipaque Bran, Estrellita Guadalupe Armas Monroy, Alejandra Roxana Tavico Estrada, Allan Marcelo Noguera Torres y Jorge Eduardo Urías Sagastume, por todas las palabras de aliento, cada momento que me brindaron de sus vidas y el amor de hermandad que me llenó en los momentos más difíciles.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser la casa de estudios que me brindó el conocimiento y la educación suficiente para valerme como un profesional.
Facultad de Ingeniería	Por todas las experiencias vividas y por moldear mi vida con cada reto para permitirme alcanzar la meta.
Mi asesor	Ing. Moisés Eduardo Velásquez Oliva, por impartirme de forma grata el conocimiento y tiempo que han hecho posible este momento.
Mis compañeros de estudio	Marlyn Ivannia González, Francisco Antonio Noj Avendaño, María Alejandra Álvarez Vásquez, Christopher Antonio Pérez Tzoc, Luis Ricardo Hernández, Julio Estuardo Revolorio Arévalo, Rodolfo Santiago Funes Muñoz, Robinson Jonathan Fuentes, Luis Fernando Solares Arcón y Jesús Alberto Guzmán Polanco, por ser los indiscutibles aliados en esta batalla, el apoyo en los más grandes obstáculos y con quienes anhelo seguir caminando.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. APLICACIONES MÓVILES.....	1
1.1. Antecedentes y trascendencia de las aplicaciones móviles	2
1.1.1. Las <i>Apps</i> orientadas al comercio.....	4
1.1.2. Las <i>Apps</i> en mejora de procesos	5
1.2. Categorías.....	5
1.2.1. Por el entorno de ejecución	6
1.2.1.1. Aplicaciones nativas	6
1.2.1.1.1. Ventajas para el desarrollador.....	8
1.2.1.1.2. Ventajas para el usuario.....	8
1.2.1.1.3. Desventajas para el desarrollador.....	9
1.2.1.1.4. Desventajas para el usuario.....	9
1.2.1.2. <i>Web</i> móvil.....	10
1.2.1.2.1. Ventajas para el desarrollador.....	11

1.2.1.2.2.	Ventajas para el usuario	12
1.2.1.2.3.	Desventajas para el desarrollador	12
1.2.1.2.4.	Desventajas para el usuario	12
1.2.1.3.	Aplicaciones híbridas.....	13
1.2.1.3.1.	Ventajas para el desarrollador	14
1.2.1.3.2.	Ventajas para el usuario	14
1.2.1.3.3.	Desventajas para el desarrollador	15
1.2.1.3.4.	Desventajas para el usuario	15
1.2.1.4.	Comparativa	15
1.2.2.	Por sus funcionalidades	17
1.2.2.1.	Comunicaciones.....	18
1.2.2.2.	Multimedia	18
1.2.2.3.	Entretenimiento	19
1.2.2.4.	Productividad.....	19
1.2.2.5.	Utilidades.....	20
1.3.	<i>Framework</i>	21
1.3.1.	Arquitectura	22
1.3.1.1.	MVC (Modelo-Vista-Controlador)	23
1.3.1.2.	Cliente-Servidor.....	23
2.	MERCADEO DE LAS APLICACIONES MÓVILES	25
2.1.	Mercadeo	25

2.1.1.	<i>Marketing</i> de servicio.....	26
2.1.2.	<i>Marketing</i> industrial.....	26
2.1.3.	<i>Marketing</i> comercial.....	26
2.1.4.	<i>Marketing</i> social.....	27
2.1.5.	<i>Marketing</i> digital/ <i>online</i>	27
2.1.6.	<i>Marketing</i> sectorial.....	27
2.1.7.	<i>Marketing</i> corporativo	28
2.2.	<i>Marketing</i> orientado a las aplicaciones móviles.....	28
2.3.	Beneficios de las aplicaciones móviles en el mercado	31
2.3.1.	Objetivo de una aplicación móvil en el mercado.....	32
2.3.1.1.	Aplicaciones como creadoras de imagen.....	32
2.3.1.2.	Aplicaciones para la creación de lealtad empresa-cliente.....	33
2.3.1.3.	Aplicaciones como herramientas de gestión	34
2.3.1.4.	Aplicaciones como medio de venta (<i>Mobile Commerce</i>)	34
2.3.1.5.	Aplicaciones como creadoras de ingreso.....	35
2.4.	La aplicación.....	36
2.4.1.	La idea.....	37
2.4.2.	Los objetivos.....	37
2.4.3.	El ambiente.....	37
2.4.4.	El sistema	41
2.4.5.	La funcionalidad.....	41
2.4.6.	El modelo de negocios y recursos	41
2.5.	Técnicas para el <i>marketing</i> de aplicaciones móviles.....	42
2.5.1.	El inicio	43

2.5.2.	La captura	44
2.5.3.	La preferencia	46
3.	METODOLOGÍA PARA EL DESARROLLO DE <i>SOFTWARE</i> MÓVIL HÍBRIDO	47
3.1.	Patrones de diseño	47
3.1.1.	Patrones de diseño <i>Android</i>	48
3.1.1.1.	Manejo de datos.....	48
3.1.1.2.	Obtención de la información de entrada	48
3.1.1.3.	Navegación	49
3.1.1.4.	Notificación.....	49
3.1.1.5.	Personalización	49
3.1.1.6.	Interacción de pantalla	49
3.1.1.7.	Social.....	50
3.1.2.	Patrones de diseño IOS	50
3.1.2.1.	Visualización	50
3.1.2.1.1.	Deferencia.....	50
3.1.2.1.2.	Claridad.....	51
3.1.2.1.3.	Profundidad.....	51
3.1.2.2.	Anatomía	51
3.1.2.2.1.	Barras	51
3.1.2.2.2.	Contenido de la vista.....	52
3.1.2.2.3.	Control.....	52
3.1.2.2.4.	Vistas temporales.....	53
3.1.3.	Patrones de diseño en un ámbito general.....	53
3.1.3.1.	Simplicidad	54
3.1.3.2.	Consistencia.....	54
3.1.3.3.	Navegación	55

3.2.	<i>Scrum</i>	55
3.2.1.	¿Por qué utilizar <i>Scrum</i> en el desarrollo de <i>Apps</i> híbridadas?	57
3.2.2.	Iniciando <i>Scrum</i>	58
3.2.2.1.	Roles	59
	3.2.2.1.1. Propietario del producto	59
	3.2.2.1.2. El equipo	60
	3.2.2.1.3. <i>Scrum master</i>	61
3.2.2.2.	Artefactos.....	61
	3.2.2.2.1. Pila del producto.....	61
	3.2.2.2.2. Pila del <i>sprint</i>	63
	3.2.2.2.3. Incremento.....	64
	3.2.2.2.4. <i>Sprint</i>	64
3.2.2.3.	Eventos.....	65
	3.2.2.3.1. Planificación del <i>sprint</i> ..	65
	3.2.2.3.2. <i>Daily Scrum Meeting</i>	66
	3.2.2.3.3. La revisión del <i>sprint</i>	66
	3.2.2.3.4. Retrospectiva.....	67
3.2.3.	Integración de XP en <i>Scrum</i>	68
	3.2.3.1. Programación en parejas.....	69
	3.2.3.2. Desarrollo guiado por pruebas.....	70
	3.2.3.3. Diseño incremental	70
	3.2.3.4. Integración continua	71
4.	COMPARACIÓN ENTRE <i>IONIC</i> Y <i>PHONEGAP</i> PARA EL DESARROLLO DE APLICACIONES MÓVILES HÍBRIDAS.....	73
4.1.	<i>Phonegap</i>	74
4.1.1.	Descripción	74

4.1.2.	Análisis	76
4.1.2.1.	<i>Phonegap build environment</i>	77
4.1.2.2.	<i>Phonegap developer app</i>	78
4.1.2.3.	<i>Local build environment</i>	78
4.1.3.	Estructuración	80
4.1.4.	Beneficios.....	83
4.2.	<i>Ionic</i>	84
4.2.1.	Descripción.....	84
4.2.2.	Análisis	86
4.2.3.	Estructura	91
4.2.4.	Beneficios.....	92
4.3.	Comparación.....	93
CONCLUSIONES.....		97
RECOMENDACIONES		99
BIBLIOGRAFÍA.....		101

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Escala de ventajas	17
2.	Información general de “ <i>We Are Social</i> ”, 2015.....	39
3.	Información general “Google”, 2015.....	40
4.	Esquema de comunicación del <i>Phonegap</i> CLI con sus distintos ambientes de construcción.....	79
5.	Estructuración de un proyecto en <i>Phonegap</i>	82
6.	Estructuración de un proyecto en <i>Ionic</i>	91
7.	Representación de estructura <i>Cordova-Ionic/Phonegap</i>	94

TABLAS

I.	Características de los sistemas operativos en los dispositivos móviles	7
II.	Comparativa entre métodos para la creación de <i>Apps</i> móviles	16
III.	Distribución de componentes en diversos sistemas operativos	76
IV.	Distribución de sistemas operativos para el desarrollo de aplicaciones móviles en sistemas operativos móviles.....	87
V.	Tabla comparativa <i>Ionic</i> vs <i>Phonegap</i>	94

LISTA DE SÍMBOLOS

Símbolo	Significado
%	Porcentaje

GLOSARIO

- App*** Abreviación del término en inglés “*application*”, el cual define un programa informático diseñado para un propósito en particular.
- Api*** Abreviatura del inglés *Application Programming Interface*, conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como un medio de comunicación.
- Marketing*** Es el concepto proveniente del idioma inglés para referirse a la mercadotecnia o mercadeo.
- Open Source*** Es el *software* cuyo código fuente está libre de los derechos de autor, es publicado como parte de un dominio público.

RESUMEN

La tecnología expone a grandes rasgos el avance del día a día del hombre, la transformación y evolución de las sociedades. Perceptibles cambios revelan el potencial de la herramienta que llama a su vez el interés de distintos campos, entre los que resalta el entorno económico. La tecnología traza nuevas rutas que permiten expandir las posibilidades de un resultado favorable.

Guatemala cuenta con un potencial sobre cada uno de los puntos dedicados al flujo económico nacional. Determinar claves para su flote es esencial en el futuro de una nación próspera. De eso se desprende el propósito general de la investigación: hacer notar que existen fórmulas capaces de hacer crecer a la mediana y pequeña empresa, donde crear la oportunidad depende de estar informado de lo que actualmente crea resultados. Actualmente son las aplicaciones móviles, herramientas favorables si son implementadas de una manera adecuada.

Abarcando un objetivo en concreto, la investigación se concentra no solo en las utilidades de esta tecnología, sino en aprovechar el máximo de su entorno. Así es como toman énfasis las aplicaciones móviles híbridas, mediante un desarrollo que despliega un conjunto de posibilidades. Percatarse de estas ventajas depende de comprender la conformación del material, entender el mecanismo de ventas, construirlo y, finalmente, saber cuáles son los utensilios necesarios para un producto eficaz. Cada uno de estos aspectos da contenido a los capítulos del presente trabajo.

OBJETIVOS

General

Proponer a la pequeña y mediana empresa guatemalteca el panorama para el desarrollo de tecnologías móviles con ayuda de aplicaciones híbridas.

Específicos

1. Mostrar el conocimiento básico de las aplicaciones móviles a las empresas.
2. Brindar las herramientas y técnicas sobre el mercadeo desarrollado en el ámbito de aplicaciones móviles.
3. Dar a conocer la metodología de desarrollo de *software* adaptable a las aplicaciones móviles híbridas.
4. Analizar la comparación entre las distintas herramientas propuestas para el desarrollo de aplicaciones híbridas.

INTRODUCCIÓN

Las pequeñas y medianas empresas guatemaltecas son parte importante del desarrollo del país, por lo cual tener los medios suficientes para darse a conocer aporta una retribución importante al territorio guatemalteco. Además de avanzar en las tecnologías actuales, la investigación se dirige a estos entornos, brindando las herramientas y técnicas con las cuales estas empresas puedan iniciar en el difícil proceso del cambio, obteniendo mejores retribuciones.

Las empresas buscan arduamente introducirse a los límites del éxito, desarrollan métodos con ánimos de resultados positivos, y toman herramientas para valerse en la complicada lucha de competencias. Las herramientas son punto clave para la empresa, pues crean el arma eficaz para tomar mercado.

Las aplicaciones móviles son un movimiento que desde hace tiempo han captado interés entre la sociedad, hasta el punto de convertirse en un bien preciado que al día de hoy puede ser solo en un ámbito de entretenimiento o posiblemente alcanzar utilidades que resuelvan problemáticas diarias, la imaginación es la limitante para el propósito de esta gran creación tecnológica.

Se busca que el interesado pueda entender los sistemas tecnológicos de este ambiente de una forma general, con la ayuda de sus conceptos, para luego pasar a un esquema económico y de mercadeo que definirá la base para su distribución. Al mostrar los beneficios que conllevan estos medios se pasa a un marco de metodologías de desarrollo que contribuyen al conocimiento del lector para la elaboración de dicha tecnología. Finalmente, es presentada la comparación de las herramientas Phonegap y Ionic, que son utilizadas en la

última generación para la creación de aplicaciones móviles híbridas, mostrando tanto sus beneficios como sus desventajas, para que el interesado pueda elegir la opción más cómoda.

1. APLICACIONES MÓVILES

El pilar que define las aplicaciones móviles nace como fruto del ambiente tecnológico, es por ello que para determinar una definición es necesario decir de qué consta la tecnología. Esta, según las etimologías en griego, parte de dos palabras esenciales, “tekhne” y “logos”, la primera significa “técnica” y la segunda “estudio”, por lo tanto, de forma breve puede traducirse como el estudio que, por medio de técnicas, solventa problemas.

Determinar en sencillas palabras lo que significa tecnología no encierra totalmente su concepto. Para llegar a esta etapa es necesario conocer el surgimiento como tal. La tecnología se produce desde que el hombre pisó la tierra, los primeros habitantes determinaron formas de sobrevivir ante los desafíos que el ecosistema establece, de esta manera crearon herramientas que les permitieron contrarrestar dichas limitaciones. Con el pasar de muchos años, en el siglo XVII sucede el primer acontecimiento que da revuelo y concepto a la tecnología, La Revolución Tecnológica, en la que se razona que la tecnología no solo parte de conocimientos básicos, sino que necesita ser comprobada por determinadas leyes que permitan su veracidad, básicamente, necesita de medios científicos como prueba.

Dado este proceso icónico de la historia, distintos personajes permitieron determinar puntos esenciales sobre este concepto. Resaltan que el conocimiento y la ciencia son fundamentos de la tecnología y que las técnicas parten del conocimiento, de la separación de este, y de la ciencia se hace recalcar que aun cuando es un conocimiento ya comprobado, la importancia de

los sucesos cotidianos o acontecimientos individuales del hombre hacen que la tecnología pueda crear invenciones efectivas.

Entonces, partiendo de ello, la tecnología se define como “las prácticas de un estudio científico y conocimiento personal que permiten contrarrestar distintos sucesos del diario vivir humano, esto produce la invención de distintos instrumentos que permiten este fin”. En esa definición es necesario resaltar que las aplicaciones móviles son tecnología, por lo cual hacen ver ya desde un inicio que son la solución a distintas incógnitas de varios entornos.

El término de aplicación móvil se define específicamente de la siguiente forma: es *software* que ofrece funcionalidades tanto necesarias como extras, que agregan valor y utilidad a los dispositivos móviles, refiriéndose a dispositivo móvil como un sistema de carácter electrónico que una persona puede movilizar de forma sencilla (*tablets, smartphones, smartwatch*, entre otros).

Dada esta fase, es necesario resaltar los acontecimientos progresivos que formaron el cimiento de lo que hoy se presenta como un instrumento efectivo, así como los antecedentes y trascendencia que darán el punto inicial sobre esta investigación, para continuar con las siguientes fases.

1.1. Antecedentes y trascendencia de las aplicaciones móviles

A finales del siglo XIX, los primeros resultados que se tomaron en cuenta en los dispositivos fueron los causados por el teléfono. En ese momento cumplían con el único fin de establecer una mejor comunicación a distancia entre las personas, el movimiento fue creciendo tanto que dejó a un lado su objetivo y dio paso a la generación de las primeras aplicaciones móviles. En el transcurso de los 80's al 2000, la tecnología avanzó rápidamente, con algunos

momentos de caída que posiblemente dieron muerte a muchas invenciones, pero la telefonía supo salir a flote al demostrar la capacidad de un producto innovador que permitía elevar las expectativas y no tocar fondo. Tales características, como la integración de funciones de mensajería, agenda, notas e incluso juegos, aportaron un valor a menor escala pero perceptible, ya que en cierta etapa se transformó en algo necesario, en el caso de la mensajería.

Las aplicaciones móviles nacen con la necesidad de generar mercado, específicamente el mercado de dispositivos móviles a los cuales adhieren valor, pues las funcionalidades elevan en gran parte la experiencia del usuario con el dispositivo, siendo el punto esencial del consumo. Básicamente, las aplicaciones móviles pueden definir la compra de un dispositivo, así como su evolución. Estas características son las que provocan que el flujo de crecimiento del dispositivo y la aplicación sean mutuos.

La nueva era de dispositivos con cualidades de *hardware* aceptables se produjeron a finales de los 2000 y permitieron que la nueva generación aportara ideas en la creación de nuevas herramientas que abarcaban un mayor rango de utilidad. Aunque en estos años la población mundial no enfocaba su total interés en este producto, un movimiento a mediados del 2007 hizo ver de una forma totalmente nueva la telefonía móvil. *Apple*, con el lanzamiento de su reconocido producto iPhone, permitió que las aplicaciones móviles jugaran un papel más allá de solo ser un componente esencial del teléfono. La disponibilidad de no solo realizar llamadas telefónicas, sino de tener la capacidad de tomar fotografías, interactuar con los correos electrónicos, inclusive poder hacer videollamadas, promovieron la iniciativa de generar producto para este mercado. Varias y nuevas empresas iniciaron en el ámbito, provocando que el mercado creciera en opciones, con ello los desarrolladores obtuvieron la posibilidad de construir aplicaciones con libertad y objetivos distintos, algunas

entidades dieron facilidades en la construcción, por otro lado, algunas impusieron limitaciones pero no dañaron la línea de innovación. Más dispositivos empezaron a integrarse al ambiente, los reproductores mp3, las *tablets* y hoy en día los famosos *smartwatch*, construyeron un sistema donde las aplicaciones pueden desarrollarse de formas variadas.

Las aplicaciones toman objetivos dependiendo del entorno en el que se ejecuten, su prioridad es hacer más eficiente la obtención de los beneficios y ganancias de los sistemas para los cuales son creadas. El entorno móvil entrega distintas estrategias y atributos, que brindan a la entidad nuevas posibilidades de un mercado único al sistema. Los objetivos pueden estar dados de la siguiente manera:

1.1.1. Las Apps orientadas al comercio

Las aplicaciones principalmente se han generado por las corporaciones que pretenden tener mayores ganancias en sus productos móviles. Uno de los objetivos clave para una empresa que inicia se derivan de otorgar la atención del usuario hacia el producto y construir la base que hace destacar su calidad. El primer producto es esencial para generar los pilares que entregarán una preferencia al usuario, el objetivo es la formación de la figura que representará a la entidad. Los valores obtenidos por parte de una aplicación no solamente nacen de la utilidad principal sino de la expansión de esas funcionalidades, cobrando por esas atribuciones y, en dados casos, incorporando al usuario en procesos de contratos. Esto permite que las *Apps* se embarquen a métodos de mercado con grandes posibilidades de venta.

1.1.2. Las Apps en mejora de procesos

Los objetivos de conseguir beneficios por este medio son bastante claros. Sacarle provecho en la empresa a las posibilidades que permiten estos sistemas no solamente ayuda al perfeccionamiento de la misma, también fomenta globalmente la producción de importantes y mejores soluciones. Este enfoque pretende agilizar, facilitar y mejorar procesos que inicialmente han tenido un costo afectable y que de otra manera no se han logrado estructurar correctamente.

La tecnología generalmente dirige la atención del usuario hacia las áreas del entretenimiento, las Apps móviles han crecido de esta forma, a pesar de ser limitadas hacia estos parámetros no han provocado que el mayor potencial en los distintos entornos sea aplicado de modo correcto. Los casos en donde mejoran los procesos de manufactura, gestión, administración y mercadeo son prueba de que las Apps tienen mucho que entregar. Tener en la palma de la mano indicadores que faciliten la toma de decisiones, controlar procesos en línea, evaluar reportes de los trabajadores y un sinfín de recursos, brinda al empresario el medio ideal que aporta el mejoramiento a sus sistemas y que una aplicación puede producir.

1.2. Categorías

Las aplicaciones móviles pueden categorizarse conforme a sus funcionalidades, que logran en conjunto construir un objetivo que aporta valor al usuario y, finalmente, por el modo de construcción en un entorno específico, que permitirá que su ejecución pueda sustentarse por todas las cualidades del móvil o ser limitadas por las mismas.

1.2.1. Por el entorno de ejecución

Se denominan aplicaciones móviles por el entorno de ejecución aquellas que operan o funcionan en entornos detallados, es decir, que su capacidad depende del ambiente en la cual se establezcan, por ejemplo, en ciertos sistemas operativos, navegadores u otros productos de *software*.

1.2.1.1. Aplicaciones nativas

La mayor parte de entidades que se dedican a la elaboración de plataformas (sistemas operativos) para los dispositivos móviles buscan expandir su mercado de la mejor forma, frecuentemente tratan de aumentar la demanda de su *software* por medio de las Apps e introducen herramientas con el propósito de que la comunidad desarrolladora se integre y se atreva a crear *software*. Así es como nacen las aplicaciones nativas, que son las puertas que abre la industria hacia un desarrollo continuo y creciente de *software*.

Comúnmente, la definición de este tipo de aplicaciones genera conflictos entre distintas partes de la comunidad, por no ser del todo clara. Gran cantidad de desarrolladores entienden que el ser nativo solo constituye el lenguaje que es designado por el sistema operativo, dándole cabida al funcionamiento de la aplicación, y no es del todo cierto, pues dicho tipo de Apps son creadas como el propietario lo designa, utilizando no únicamente el lenguaje sino además un entorno y una estructura preestablecida, con el fin de optimizar la calidad y el rendimiento de la aplicación que finalmente tratan de mitigar la mayor parte de fallos.

De acuerdo con lo establecido, las aplicaciones nativas se definen como el tipo de aplicaciones que son elaboradas con un conjunto de normas y utilidades establecidas por cada fabricante, quienes aseguran la maximización del rendimiento y funcionamiento de la aplicación dentro de sus plataformas. Además, cabe mencionar que las aplicaciones se instalan internamente en el móvil con todos los archivos y componentes necesarios, perdurando hasta el momento de su desinstalación, cosa que otros medios no realizan. Actualmente existen 3 sistemas operativos que son los más conocidos en el mercado, razón por la cual cuentan con mayor distribución, pero es necesario mencionar que no son los únicos. Cada uno de ellos tiene su propia distribución en el desarrollo.

Tabla I. **Características de los sistemas operativos en los dispositivos móviles**

Fabricante	Sistema Operativo	Lenguaje de Programación	Herramientas	Medio de Distribución
IOS	Apple	Swift, Objective-C	Xcode	Apple Store
Google	Android	Java	Android SDK	Google Play
Microsoft	Windows Phone	Visual Basic .NET y C#	Visual Studio	Windows Phone Marketplace

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

Las metas del entorno para las aplicaciones nativas se concentran en entregar la facilidad de un espacio de construcción y brindar el mejor desempeño en los sistemas operativos. Eventualmente los fabricantes otorgan un sinnúmero de herramientas que contribuyen a la elaboración de las aplicaciones

y adhieren soporte que sustenta la misma entidad como parte de los beneficios; el generar la accesibilidad a grandes rasgos provoca una mayor expectativa en el éxito y utilidad de la aplicación.

A continuación se muestra una lista de ventajas y desventajas que competen al desarrollador, que juega un papel importante en la empresa y para el usuario, quien es la persona que utiliza la aplicación.

1.2.1.1.1. Ventajas para el desarrollador

- Le otorga un marco de trabajo.
- Distribución por medio de las tiendas de aplicaciones.
- Kits de herramientas para la construcción y prueba.
- Acceso a las API's nativas (alto y bajo nivel), las cuales permiten acceder a todos los elementos del dispositivo.
- Permite la creación de aplicaciones bastante completas.

1.2.1.1.2. Ventajas para el usuario

- Acceso rápido a las *Apps*.
- Variedad de aplicaciones de este tipo.
- Seguridad de la información, al no ser accedida por otro medio (vía Internet) frecuentemente los datos se mantienen disponibles solo para el usuario.
- Capacidad de interactuar con otras aplicaciones.

1.2.1.1.3. Desventajas para el desarrollador

- Costo de elaboración/curva de aprendizaje, por ser un lenguaje y entornos totalmente diferentes a los demás, el tiempo en adquirir el conocimiento aumenta.
- Incapacidad de adaptación, pues las aplicaciones nativas no pueden expandirse a todos los sistemas, si sucede debe construirse la aplicación nuevamente para el funcionamiento en los demás.
- Dificultad de mantenimiento.
- Frente a otros medios para el desarrollo este es el más complejo.
- Incompatibilidad entre versiones de sistemas operativos, pues en ciertos casos las versiones anteriores de las aplicaciones no son funcionales en los sistemas más recientes.
- Limitación generada por el entorno.
- Construir un sistema bastante complejo en sus procesos puede tomar demasiado tiempo.
- Los tiempos de aceptación en las tiendas para ser publicados tienden a ser altos.
- Costos adicionales por servicios *online* por parte de los fabricantes (Google, Apple, etc).

1.2.1.1.4. Desventajas para el usuario

- Necesidad de actualizaciones.
- El valor de adquisición por aplicación aumenta debido a la costosa elaboración.
- Las aplicaciones de este tipo tienden a ser muy pesadas, por lo cual la descarga se hace más tediosa.

- Al instalarse algunas aplicaciones dejan residuos que ocupan espacio en la memoria.
- Su rendimiento depende de las capacidades del *hardware*.

1.2.1.2. Web móvil

La tecnología *web* crece a pasos agrandados, la calidad en el diseño y desempeño que actualmente entregan los sitios no se ve truncada por la maquetación tradicional que existía en un inicio, su avance ha permitido que el límite sea establecido por la imaginación del desarrollador, generando trabajos con una mejor estructuración y superando las expectativas.

Todo dispositivo móvil al menos cuenta con un navegador, ya que la red mundial (Internet) no deja cabo suelto al tratar de abarcar todos los sistemas posibles, es así como este se transforman en parte de los elementos nativos del móvil. Conociendo este punto clave nace la idea de buscar formas que logren un desarrollo de aplicaciones móviles sencillo y que con herramientas y métodos prácticos se consiga un producto final satisfactorio. El desarrollo *web* ha sido la respuesta a esta problemática.

El *web* móvil se basa en el concepto de reutilizar los parámetros ya existentes de entornos *web* y transformarlo en sistemas con calidad de aplicación móvil o, generalmente, tomar la apariencia de una. El progreso ha permitido que las páginas *web* puedan adaptarse a la capacidad de un móvil, reduciendo su estructura y cumpliendo las mismas expectativas que un sitio habitual. Este resultado ha convencido a la industria. Las empresas toman provecho de esta metodología, ya que cuentan con servicios *web* y buscan expandir su ecosistema.

Los actuales navegadores cuentan con la capacidad de utilizar las propiedades del móvil de una forma aceptable, esto constituye la existencia del *web* móvil, un sistema que es adaptado a los dispositivos en distintos aspectos, como sus servicios y contenidos, además de brindar capacidades extras. Es necesario mencionar que no es copia total de una *web* existente, ya que se diferencia en varios aspectos. Actualmente la comunidad desarrolladora ha creado herramientas que facilitan la interacción del sistema *web* con el móvil, aunque está reducida a cierto tipo de acciones y no tiene total libertad de API's como los dispositivos nativos.

Las páginas *web* dirigidas a móviles toman el mismo camino de interacción que una *web* común, están centradas en un punto, por lo cual es necesario la conectividad. El usuario interactúa por medio de ella con la ayuda del navegador, aunque generalmente no es notado, ya que es posible generar accesos directos a estas *webs*, dando la vista de una aplicación normal donde finalmente el navegador hace el medio de comunicación con el móvil. Al igual que los elementos nativos, un *web* móvil tiene ventajas y desventajas que permiten un mejor análisis al elegir una opción. A continuación se describen estos aspectos.

1.2.1.2.1. Ventajas para el desarrollador

- Una de las principales ventajas es la compatibilidad, ya que está realizado con el lenguaje para *web* (HTML, CSS3, JavaScript) que es totalmente funcional en los navegadores, permitiendo crear las aplicaciones para distintos fabricantes sin necesidad de realizar mayores cambios.
- Reducción de costos.
- De rápido aprendizaje.

- Mayor capacidad de acceso con menores tiempos de carga.
- Sin dificultad de distribución, ya que son ambientes *web*, y en la mayoría de casos no es necesario subirlo a una plataforma de descarga.
- Cantidad de herramientas para el desarrollo, WebKits.
- El cambio de un sistema operativo no repercute grandemente en el funcionamiento de la aplicación.
- Actualizaciones de versiones dependiendo del navegador, por lo cual no es constante.

1.2.1.2.2. Ventajas para el usuario

- Ya que utiliza un entorno bastante ligero, el consumo de recursos es menor.
- Utilidades básicas.
- El cambio en *hardware* y *software* no afecta con frecuencia a la aplicación.

1.2.1.2.3. Desventajas para el desarrollador

- Es necesaria la conexión de Internet para asegurar una actualización.
- Su funcionamiento se limita a las pocas interacciones que puede tener con el móvil.
- Menos vistosos, se reduce la calidad de diseño, si acaso el objetivo sea llamar la atención del cliente con efectos como en la *web*.

1.2.1.2.4. Desventajas para el usuario

- Conexión continua a la red.

- La carga de elementos es muy pesada, limita la eficiencia.
- Solo para usuarios que no necesitan de aplicaciones tan complejas.

1.2.1.3. Aplicaciones híbridas

La complejidad que ha conllevado el camino hacia la creación de aplicaciones móviles sencillas y eficientes ha transcurrido en una larga búsqueda de recursos y elementos para llegar a cumplirlo. Las aplicaciones nativas ofrecen la utilidad de sacarle provecho a los elementos del móvil a diferencia de la *web* móvil, que se concentra en una reducción de costos. Es difícil tener ambas partes aplicadas en un solo método, así da comienzo la iniciativa de las aplicaciones híbridas que cubre los aspectos positivos de ambos sectores, la habilidad de poder interactuar con el móvil utilizando sus recursos de la mejor forma y con una facilidad de desarrollo, resaltando el aspecto más importante: la neutralidad de los ambientes donde son construidos.

El enfoque híbrido no es algo totalmente reciente, primeramente no fue aceptado como realmente se lo merecía, ya que la mayoría API's para su funcionamiento se desplazaba en un pequeño conjunto, pero avanzando el tiempo la comunidad que vio en este entorno sus ventajas puso en marcha el trabajo de elaborar API's que dieran la capacidad de igualarse a un nativo. Las aplicaciones híbridas utilizan marcos de trabajo como las *web* móviles, son desarrolladas de la misma manera, con la diferencia que son instaladas como una aplicación nativa y otorgan el mismo beneficio. Como es de notar este tipo de aplicaciones cuenta con ventajas y desventajas, las cuales a continuación se mencionan.

1.2.1.3.1. Ventajas para el desarrollador

- Fácil aprendizaje.
- Alta compatibilidad entre versiones de sistemas, es fácil migrar de un sistema a otro con un híbrido.
- Las capacidades de manipulación de los componentes de un dispositivo como un nativo.
- Bajo costo de implementación.
- Herramientas de desarrollo, creciendo eventualmente.
- Teorías de estructuración para un mejor manejo de componentes, implementadas siempre por el marco de trabajo.
- No es necesaria la conexión a la red, ya que está localmente almacenada.
- Opción de almacenarse en un espacio en la red o manejarlo fuera de línea.
- Buen rendimiento.
- Los cambios son menores al querer elegir cambiar de sistema del fabricante (multiplataforma).

1.2.1.3.2. Ventajas para el usuario

- Interfaces sencillas, pero con toda la utilidad que cualquier otra aplicación desarrollada, de otra manera, entregaría.
- Cargas rápidas.
- Economización de recursos.
- Aun cuando se instale en el dispositivo, ocupa poco espacio por ser componentes *web*.

1.2.1.3.3. Desventajas para el desarrollador

- Dependiendo de la aplicación es posible que se utilice la conectividad a la red, por el concepto de actualizaciones.
- Si se desea generar un nuevo API's es necesario crearlo en el lenguaje nativo del sistema.
- Por no ser tan conocido, es necesario que los usuarios comprueben su utilidad.

1.2.1.3.4. Desventajas para el usuario

- En caso de que la aplicación no acceda a Internet, es necesario actualizarla por medio de los paquetes en el medio de distribución (*Marketplace*).

1.2.1.4. Comparativa

Como se pudo notar con anterioridad, los ambientes de elaboración para una aplicación móvil varían en distintas facetas. Existen buenas opciones a elegir dependiendo del tipo de proyecto, seleccionar el correcto es decisión de la entidad.

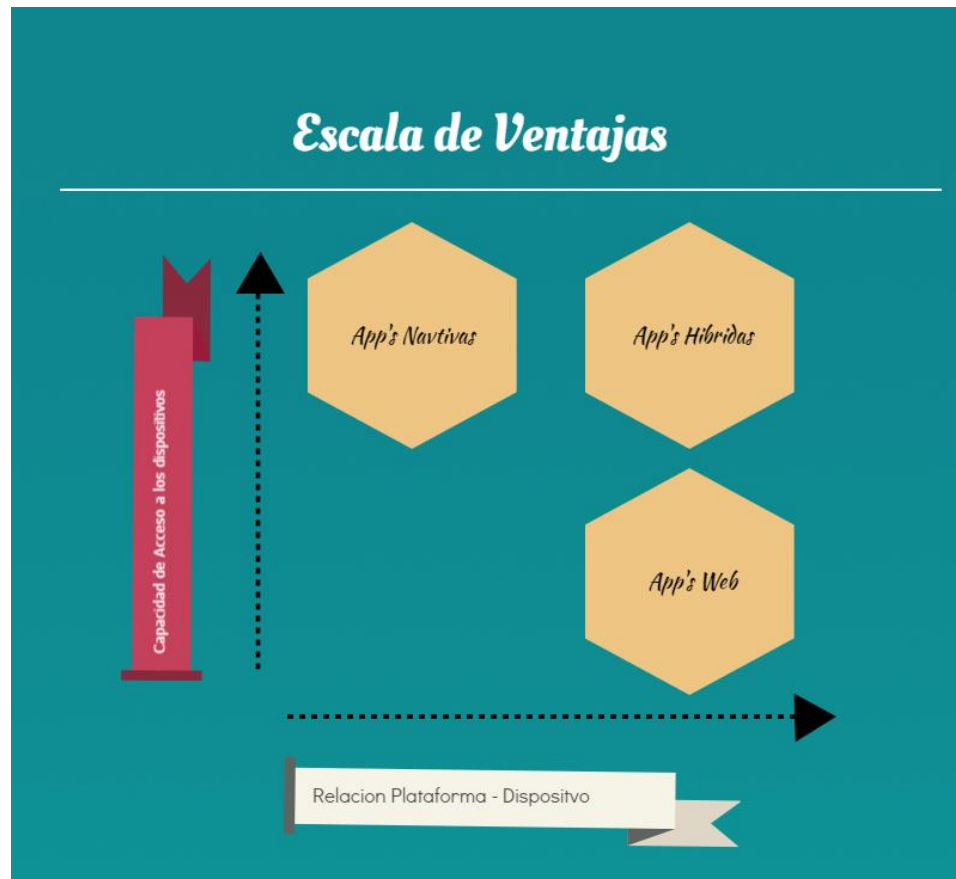
Tabla II. **Comparativa entre métodos para la creación de Apps móviles**

Característica	Aplicación Nativa	Aplicación Híbrida	Aplicación Web
Lenguaje de desarrollo	Solo nativo	Nativo y <i>web</i> o solo nativo	Solo <i>web</i>
Portabilidad y optimización de código	bajo	alto	alto
Características de acceso específicas del dispositivo	alto	medio	bajo
Uso de conocimiento existente	bajo	alto	alto
Gráficos avanzados	alto	medio	medio
Flexibilidad de actualizaciones	bajo	mediano	alto
Experiencia de instalación	alto	alto	medio

Fuente: IBM. *El desarrollo de aplicaciones móviles nativas, web o híbridas. Consulta: 25 de abril de 2016. p. 8*

El diagrama siguiente permite identificar las posiciones de cada entorno entre sus adversarios, cada uno aporta valor entre sus capacidades de acceso al dispositivo y lo atractivo del desarrollo, el tiempo, la facilidad y los inconvenientes, que son importantes para elegir una opción.

Figura 1. Escala de ventajas



Fuente: FERNÁNDEZ, Alfonso. Aplicaciones web vs aplicaciones nativas vs aplicaciones híbridas. <http://blogthinkbig.com/aplicaciones-web-nativas-hbridas/>. Consulta: 01 de marzo de 2016.

1.2.2. Por sus funcionalidades

Una aplicación catalogada por sus funcionalidades muestra idealmente esas aplicaciones que forman parte de sistemas más grandes, como las mensajerías y las redes sociales que permiten asentarse en los ambientes de la comunicación. En el mercado actual pueden encontrarse una gran cantidad de

aplicaciones y todas juegan un papel importante, desde las de entretenimiento hasta la de negocios.

1.2.2.1. Comunicaciones

Como su nombre lo indica, son aplicaciones dedicadas a la socialización entre usuarios, entre estas se encuentran:

- Mensajería instantánea
- Redes sociales
- Exploradores
- Mensajería por voz
- Videollamadas
- Clientes de correo electrónico
- Salas de conversación

1.2.2.2. Multimedia

Se refiere a la mayoría de aplicaciones que presentan la combinación variada de distintos medios, como la música, videos, etc.

- Reproductores de video
- Reproductores de audio
- Visor de imágenes
- *Streaming*
- Cámara y video (refiriéndose a la aplicación que realiza el trabajo)
- Simuladores
- Editor de fotos
- Editor de videos

- Editor de audio
- Grabadora

1.2.2.3. Entretenimiento

Toda aplicación con el objetivo de entretener o incentivar al usuario a pasar un momento agradable, están generalmente conformadas de actividades que llaman la atención del cliente.

- Juegos
 - Deportes
 - Cartas
 - De rol
 - *Puzzle*
 - Acción
 - Aventura
- Lectores de libros digitales
- TV
- Fotografías
- Música
- Radio
- Redes sociales
- Guías de entrenamiento (tutoriales)

1.2.2.4. Productividad

Las aplicaciones dirigidas a la productividad engloban los elementos que ayudan a agilizar procesos o tareas para el usuario:

- Calculadoras
- Hojas de cálculo
- Calendario
- Banca electrónica
- Organizadores
- Notas
- Recordatorios
- Almacenamiento en la nube (*App de Google Drive, Dropbox, etc*)
- Diarios en línea
- Conversores de documentos
- Editores de texto
- Traductores
- GPS
- Mapas

1.2.2.5. Utilidades

Herramientas que sirven al usuario en procesos cotidianos del hogar o generalmente en funciones esenciales del dispositivo.

- Reloj
- Gestor de contactos
- Llamadas
- Administrador de archivos
- Captura de pantalla

Generalmente un tipo de funcionalidad puede entrar en distintos campos. Clasificar una aplicación es un paso primordial en el proceso de elaboración, ya

que permite determinar los objetivos para los que será construida, que los usuarios tengan mejor elección de producto y no permitir confusiones con el *software* malintencionado; lo importante es mostrarle al usuario final la calidad.

1.3. **Framework**

Conociendo en el sentido básico lo que conforman las aplicaciones, es momento de adentrarse un poco más a los términos que permiten la construcción de una. Las *Apps* no solo se basan en sus APIS, o en el hecho de ser híbridas, *web* o nativas, parte importante son todas las características que se desglosan antes de poder iniciar con la elaboración, el *framework* es una de ellas. En simples palabras, un *framework*, como su traducción lo dice, es un marco de trabajo, es el pilar que funciona para determinar la construcción de algo, de forma más concreta, y se conceptualiza de la siguiente manera: un marco de trabajo es la conformación de elementos de utilidad y procesamiento para resolución de problemas específicos, establece tanto prácticas como conceptos y criterios, con el fin de que un problema sea resuelto de la mejor manera y con un patrón que permita un buen resultado.

En el ámbito de desarrollo de *software*, un *framework* cumple el mismo papel, pero con variaciones que permiten su extensión a distintos objetivos. La comunicación de un medio a otro es uno de ellos. Generalmente es el puente de acceso hacia otro elemento, como híbridos y *web*. Los *frameworks* incrustan herramientas que le permiten al desarrollador simplificar variedad de procesos que anteriormente llevaban un esfuerzo y tiempo bastante elevados, la comunidad hace que los *frameworks* de este tipo crezcan a la conveniencia de problemas comunes, gracias a que estos son generalmente de desarrollo libre.

Un *framework* de desarrollo agrega soporte a la construcción de *software*, mantiene la línea de producción en el caso de que el equipo de desarrollo sea de gran tamaño, incluso algunos de ellos recomiendan otros *frameworks* para una mejor elaboración de aplicaciones.

1.3.1. Arquitectura

Un marco de trabajo se rige por patrones y es necesaria una arquitectura predefinida para su funcionamiento. Una arquitectura va más allá de nombrar con qué lenguaje se va construir el *software*, se centraliza en cómo generar una estructura y un funcionamiento entre todos los elementos que jugarán un papel importante. Estos pueden ser base de datos, servicios *web*, otras aplicaciones, etc. Una arquitectura interviene en la totalidad de procesos del sistema con la intención de tener en cuenta todos los puntos de los cuales partirá un resultado óptimo. La arquitectura recopila metodologías, técnicas, composiciones y todo lo relacionado con ensamblar un ambiente al alcance del sistema.

Las arquitecturas se van mejorando a través del tiempo, con cada proyecto de *software* hecho se retroalimenta, se pone a prueba la capacidad de responder a eventos inesperados, donde se recolecta todo factor resultante y fallos que son considerados para reestructurar nuevamente la arquitectura. En el sistema general del desarrollo de *software* existen gran variedad de arquitecturas, cada una ajustable a cualquier tipo de proyecto. En el de aplicaciones móviles se reducen las arquitecturas existentes, ya que este tipo de *software* es un caso especial y hay que tomar en cuenta que las capacidades de un computador no son las mismas que un dispositivo móvil, ya que son fabricados de forma más liviana. Algunas de las arquitecturas que se pueden implementar a las aplicaciones móviles son:

1.3.1.1. MVC (Modelo-Vista-Controlador)

La arquitectura MVC se concentra en separar la interfaz gráfica mostrada al usuario (vista), para luego dirigirse al accionador de los eventos (controlador), que realiza las conversiones o traducciones necesarias para informarle al módulo pertinente que ejecutará todas las peticiones necesarias para el procesamiento de la información (modelo), en este mismo se encuentra toda la data almacenada y principalmente la lógica de negocio. Regularmente las acciones que ejecutan en los tres componentes son cíclicas, el usuario realiza un proceso que el controlador regula y el modelo desarrolla, para finalmente enviar de nuevo el modelo al controlador y el controlador mostrárselo a la vista.

1.3.1.2. Cliente-Servidor

La arquitectura cliente-servidor es un modelo en el cual el “cliente”, que son los módulos de interacción con el usuario, realiza peticiones para las cuales necesitan cierto procedimiento de transformación. Este servicio es procesado y resuelto por la parte llamada “servidor”, que contiene tanto la información como la ejecuciones necesarias para regresar la información al cliente. En dados casos este sistema no cuenta solo con un servidor, habitualmente el peso es distribuido por varios de estos.

Esta es la arquitectura más sencilla de estructurar, generalmente es utilizada cuando se necesita demanda de respuestas rápidas, ya que los cambios de información se realizan conceptualmente entre dos entes. La arquitectura puede desglosarse en las siguientes:

- Cliente liviano: su principal objetivo es ejecutar el transporte de entradas y salidas del cliente, para que el proceso de servicios solamente el servidor lo concrete.
- Cliente inteligente: un cliente inteligente es aquel cuyo procesamiento no necesita del todo al servidor, el servidor funciona como fuente de información centralizada, pero generalmente, al tener un acceso a la base de datos, se realiza localmente, ya que en el cliente se instala la base de datos y las actualizaciones se realizan al servidor general. En caso de no existir conexión al servidor, el cliente puede funcionar sin ningún problema.
- Cliente robusto: el cliente robusto, al contrario del liviano, realiza el máximo de procedimientos y operaciones permitidos por el cliente y envía al servidor los datos necesarios para una respuesta rápida.

2. MERCADEO DE LAS APLICACIONES MÓVILES

2.1. Mercadeo

El mercado es el ecosistema donde se reúnen distintos puntos de intercambio para generar beneficios en las entidades participantes, sistema por el cual todo negocio se sumerge, desde el momento en que nace con un capital hasta el instante donde posiblemente muera por la bancarrota. Los ideales en el proceso se desglosan en obtener mejores oportunidades, mayor escalabilidad con las relaciones compra/venta (proveedores y clientes) y potenciar el desarrollo individual. No todo es fácil en este entorno, ya que se puede encontrar todo tipo de obstáculos y barreras en la que incluso los sistemas externos intervienen, como por ejemplo los gobiernos.

El mercadeo nace de la necesidad de crear herramientas para la venta de bienes por parte de las empresas, es el medio por el cual los negocios indagan dentro del comportamiento de los clientes para generar conductas de compra, se determina como una filosofía que define procesos de los cuales se arman estructuras estratégicas en el ámbito financiero y busca entender el mercado objetivo, es decir al cliente, entender su entorno, necesidades y comportamientos, con el fin de presentar de manera convincente un producto con mayor porcentaje de compra. El mercadeo es el empuje de un mercado próspero.

El mercadeo abarca gran variedad de áreas de implementación, la segmentación se amplía a causa de los distintos sectores y servicios en todo sentido, cualquier producto con utilidad puede promocionarse y venderse. Antes

de saber qué mercado es el indicado para el producto, primeramente se debe conocer qué se está vendiendo. Es entonces que se avanzará a los siguientes pasos de un buen mercadeo. Esta investigación se concentra directamente en el *software* para móviles, pero las técnicas que se presentarán funcionan generalmente para cualquier producto, siempre y cuando cumpla con los requerimientos necesarios para su implementación. A continuación se habla de los tipos de mercadeo más utilizados.

2.1.1. *Marketing de servicio*

Este *marketing* se concentra en sectores como la educación, el ocio, la seguridad, entre otros. Los servicios son bienes que pueden devengarse, así como todo producto, y ser calificados por el cliente (una persona individual o regularmente otro negocio que solicita el servicio). El comprador finalmente establece una ponderación de calidad que afectará al producto en un futuro.

2.1.2. *Marketing industrial*

El *marketing* industrial se centra en el sector que está antes del consumidor final, es decir las empresas que consumen materia prima de otros negocios, en que verifican su calidad y valor en el mercado para tomar las mejores decisiones en elección. Es irremediable en este *marketing* concentrarse en los costos/precios y eficiencia, ya que la competencia tiende a ser amplia.

2.1.3. *Marketing comercial*

El *marketing* comercial centraliza una estrecha relación entre los bienes y el cliente, siendo estos las empresas o la sociedad en general, dejando a un

lado los sistemas gubernamentales. Primordialmente genera una importante identidad con la marca y, a diferencia de otras estrategias de *marketing*, se familiariza con el producto vendido, procurando posicionarlo en el mercado.

2.1.4. *Marketing social*

Este tipo de *marketing* se concentra en las necesidades de la sociedad, su fin es el mejoramiento del ambiente social, por eso realiza cambios constantes en las actividades diarias, la producción de beneficios y la aceptación social en el bienestar del individuo.

2.1.5. *Marketing digital/online*

Este *marketing* tiene como herramienta principal los elementos provenientes del Internet, se concentra en proporcionar publicidad por medios *online*, además de realizar transacciones provocadas por el mercado en el mismo medio. Hoy en día es un instrumento efectivo para iniciar en el ámbito del mercadeo. La proporción de usuarios que se encuentra navegando por la *web* es cada vez más grande, y aprovechar esta clase de comportamiento permite crear mejores estrategias y resultados, ejemplo de ello es aumentar las capacidades de mostrar un nuevo producto, generar portafolio de clientes o ampliar los territorios abarcados.

2.1.6. *Marketing sectorial*

El mercadeo sectorial se concentra en un determinado grupo de empresas que dirigen sus elementos estratégicos con una metodología semejante, siendo homogéneos, es decir que cuentan con un mismo precio, producto, público u otras cualidades. Es fundamental conocer a qué división del mercado se

pertenece y especializarse en el tema, ya que de ello depende la elaboración de una estrategia exitosa. La competencia sectorial acostumbra tendencias agresivas, es por lo tanto indispensable contar con las bases suficientes para sobrevivir a este ambiente.

2.1.7. Marketing corporativo

Es conocido también como *marketing* interno, se especializa en publicitar normas y estrategias dentro de la misma organización, su fin determina el fortalecimiento de lazos entre los empleados y el mejoramiento estructural de procesos. Implementa una filosofía de trabajo óptima que atribuye mejoramientos y adaptación a los cambios. Este mercado, además del mejoramiento de la mano de obra, establece un *estatus* frente a los medios y cataloga a la empresa como una entidad confiable.

2.2. Marketing orientado a las aplicaciones móviles

Cada día se suman nuevas clasificaciones de *marketing* gracias al crecimiento de los campos sociales y los productos como tal (ineludible recalcar su íntima atribución en las metodologías). Los bienes no definen directamente el público comprador, debido a que en muchos casos lo que realmente se pensaba como propósito termina siendo transformado por los consumidores, circunstancia que sucede comúnmente con el *software*. Las metodologías se renuevan con la actitud del cliente, lo esencial para satisfacer tan inesperados cambios es prepararse a que sucederán. Aunque la mención de mercadeo en esta investigación sea corta, conforma el conocimiento básico que la pequeña y mediana empresa guatemalteca pueden aplicar en los sistemas de *software* móvil; la intención es sacar a relucir este factor.

En el *marketing* orientado a las aplicaciones móviles el producto es el *software* que brindará la utilidad según el negociante proponga o, por otro lado, el *software* es el medio por el cual otro producto se promocionará. Independientemente del caso, el resultado es el crecimiento del negocio. Cabe recordar que a diferencia de las grandes empresas que poseen mercado en este entorno, las pequeñas y medianas que deciden iniciar abarcan poco, es por ello que este material se dirige a estas para crear posibilidades de un amplio mejoramiento en ventas y procesos. Para exhibir los puntos fuertes de la inversión en todo sentido de las aplicaciones móviles se hace útil mencionar los siguientes acontecimientos.

En el 2015 se hace constatar que existen aproximadamente 3.649 billones de usuarios con dispositivos móviles, lo cual muestra que la cantidad de esta tecnología supera las cantidades de cualquier otra. Solamente establecer que una aplicación cobre 1 centavo (de cualquier moneda) diario, estando en la mitad de los dispositivos, expone las grandes ganancias que producen dichos recursos. Las aplicaciones actuales cobran más de \$1 por servicios extras, y en una proporción mayoritaria los usuarios pagan por obtenerlos.¹ Si bien estos datos aún no evidencian el beneficio de las *Apps*, es indispensable citar que los usuarios pasan en promedio 177 minutos diarios en sus dispositivos. La oportunidad que se le brinda a la empresa de ser reconocida en nuevos servicios a través de aplicaciones móviles (como banca móvil, compras, etc.) aumenta debido a que la sociedad está siendo consumida por este movimiento.

La intención no es el aumento evidente de ingresos sino el optimizar aspectos como el rendimiento de los trabajadores en la producción, la reducción de gastos de recursos innecesarios por actividades que se tienen que

¹ <http://wearesocial.com/uk/special-reports/digital-social-mobile-worldwide-2015>. Consulta: 22 de enero de 2016.

ejecutar o tareas que necesiten un tratamiento (lo que indirectamente producirá una mejor producción y mayores ganancias). Siendo así, una aplicación móvil introduce el punto clave en el cambio.

Los usuarios de móviles se multiplican diariamente, por lo cual no es de extrañar que los empleados cuenten con alguno y lo revisen a cada momento aun cuando están laborando. El 40 % de las personas mira su dispositivo más de 50 veces al día. Beneficioso es aprovechar las circunstancias, tomando como ejemplo el caso de alguna actividad de monitoreo de procesos por medio de un App. El empleado, aun cuando no necesariamente revise lo que está sucediendo en su labor, directamente estará apegado a su dispositivo (según el comportamiento mencionado anteriormente), lo que implícitamente permitirá que observe el estado del proceso.

El mercado de aplicaciones móviles es inmenso, respaldado con el crecimiento de ventas en los dispositivos, la mayor parte de interacción con los móviles se invierte en el uso de aplicaciones, la pelea contra miles de competidores es inevitable, es frecuente el parecido a otros productos, y predominar ante estas adversidades proporcionará el éxito inevitable de la aplicación. La importancia de estas claves es decisiva para el desarrollo del *marketing*.²

La mención de los tipos de *marketing* realizada con anterioridad tiene como fin mostrar las posibilidades para una aplicación móvil, que puede estar dirigida a un sector en particular, para uso de la misma corporación, para la sociedad, o bien, como un servicio (el propósito con que comúnmente es creada). Es este el fundamento que implica el argumento de implementar la

² <https://think.storage.googleapis.com/docs/mobile-app-marketing-insights.pdf>. Consulta: 22 de enero de 2016.

publicidad habitual en estos entornos. Una *App* es un producto más que precisa publicidad para tomar trayecto en el mercado.

Antes de finalizar este tema, es importante mencionar que en concreto la investigación no prioriza solamente indagar en los sistemas móviles, sino también en la creación de aplicaciones directamente híbridas. Ser híbrido expande campos de ejecución en el mercado, rentabilidad que principalmente abarcará el producto del mismo modo que el mercadeo. Las *Apps* móviles destacan por el valor que no todas las aplicaciones actuales tienen, el provecho de producción que se reduce económicamente en costos de reconstrucción que muy probablemente exigirán distintos sistemas de alojamiento, la disminución de tiempos en desarrollo, el aprovechamiento de la variedad disponible en las cualidades del móvil y, entre otras razones más, el avance a pasos agrandados de una nueva arma de crecimiento en la empresa. En los siguientes temas se resalta con mayor exactitud el beneficio que las aplicaciones brindan, además de cómo una aplicación puede desglosarse en nuevas ideas para el consumo del producto y cómo el mercado concede que estas mismas se promocionen sin realizar mayor trabajo del demandado.

2.3. Beneficios de las aplicaciones móviles en el mercado

Las aplicaciones móviles desde sus principios no obtuvieron el valor que realmente merecían, a pesar del infortunio fue posible explotarlas. Definitivamente la adquisición sobre este producto sobrepasa increíblemente a otras tecnologías, además de no detenerse es inimaginable lo que en la actualidad una aplicación puede brindar, incluso llega a los puntos de la realidad virtual que aún se desarrolla. Lo que antes se creía como ciencia ficción hoy se está logrando con la ayuda de un teléfono y su *software*.

La pregunta que se plantea para sacar el máximo provecho a todo lo antes mencionado radica en cómo se beneficia la empresa al querer implementar una aplicación móvil. Se han mencionado las ventajas que tienen tanto para su equipo desarrollador como para el usuario final, dependiendo del enfoque, pero hablando del mercado se especifican 5 objetivos que se resaltan en una aplicación móvil. Cada uno favorece sin duda al negocio y entran en los dos desgloses generales, como comercio y mejoramiento de procesos, lo que a continuación se mencionará se destaca en los casos que han tenido éxito en el mercado, así como lo que usualmente se desarrolla en el sistema.³

2.3.1. Objetivo de una aplicación móvil en el mercado

2.3.1.1. Aplicaciones como creadoras de imagen

Las aplicaciones otorgan un valor único al usuario, provocan un interés sobre cada uno de ellos dándoles razón para adquirirlas. Antes de crear notoriedad sobre X producto, es preciso promocionarlo y luego desatar futuras utilidades atractivas al consumidor, por esto se mantienen retribuciones constantes donde una aplicación móvil puede encajar. La razón de una aplicación en esta perspectiva es aumentar la interacción del cliente con dicho producto e indirectamente promocionarlo.

Generalmente los productos mostrados tienden a ser nuevos y como método de publicidad utilizan las vías donde la sociedad concurre frecuentemente. Ejemplo son las compañías fabricantes de autos, las cuales crean aplicaciones que permiten al usuario interactuar con el producto (generalmente nuevo), con lo cual brindan servicios que resaltan. La aplicación puede ser de larga usabilidad (otorgando servicios perdurables que no

³ Libro blanco de Apps. Guía de Apps móviles. p. 8.

necesariamente dependen del producto nuevo, aunque aporte ventajas a este), o bien de uso momentáneo (muestran el producto por medios temporales, como concursos, juegos, etc.). La siguiente lista es de algunas aplicaciones que llenan las especificaciones de este objetivo:

- *Nissan Connect*
- *Nissan Innovation*
- *Audi Magazine*
- *Toyota Entune*
- Catálogo Digital Claro

2.3.1.2. Aplicaciones para la creación de lealtad empresa-cliente

Una aplicación puede ser el inicio de una cadena de compras, provocando un lazo entre la compañía y el usuario. La lealtad del cliente hacia el producto se vuelve más estrecha, lo cual es de importancia para la empresa. Según el principio 80-20 de Pareto, que se aplica a las sociedades, el 20 % de un sector causa el 80 % de algún evento y el 20 % restante de esos eventos es causado por el 80 % del sector. Si es orientado a las empresas el 20 % de los clientes genera un 80 % de las ganancias y el 80 % de clientes restantes genera un 20 % de ganancias; ese 20 % generalmente está formado por clientes que toman parte del portafolio de fieles a la empresa.

Ejemplo de las tácticas habituales sobre este tema son empresas donde brindan un servicio *web* que el cliente sigue constantemente, por lo cual se integra otro medio (en este caso móvil) para que el usuario pueda estar atento a cualquier evento que la marca produzca. Algunas de las aplicaciones que entran en esta rama son las siguientes:

- *ESPN App*
- *Android News* (exclusivo para android)
- *Cerveza Gallo App*
- *Vans App*
- *Bershka App*

2.3.1.3. Aplicaciones como herramientas de gestión

Ya está por demás mencionar que esta porción del tema ha sido recalcada desde el inicio de la investigación, por lo cual se resume conceptualmente como toda aplicación que aporte beneficios de optimización o solución hacia procesos de las empresas, otorgándoles la calidad y eficiencia que necesitan. Ejemplos de aplicaciones de este tipo son:

- *SAP Business One*
- *Oracle BI Mobile*
- *Zoho CRM*
- *Microsoft Office Mobile*
- *MSN Dinero: Bolsa y Noticias*
- *SAP Fiori Client*
- *GoToMeeting*

2.3.1.4. Aplicaciones como medio de venta (*Mobile Commerce*)

El comercio en móviles es uno de los otros medios que las empresas utilizan como herramienta para generar ingresos, habitualmente algunos comercios no cuentan con el recurso físico para la venta (locales, tiendas, etc.)

o no se centralizan en ello, de manera que utilizan otras vías, como páginas *web*, telefonía u otras. Hoy en día, con el crecimiento de la población, los sistemas no sustentan la demanda, dando paso a la búsqueda de otra alternativa en la tecnología móvil.

La creación de aplicaciones que facultan al usuario con la habilidad de realizar compras de forma rápida y práctica desde su móvil son elementos base que constituyen una aplicación de este tipo. El *mobile commerce* crea un comportamiento de compra constante en los usuarios, el concepto que influye al usuario con la disponibilidad de obtener sus productos en unos cuantos segundos y cualquier momento hace que el consumo sea altamente frecuente. Pueden verse en este tipo las aplicaciones como:

- McDonald's GT
- Amazon Compras
- *Ebay Mobile*
- Pollo Campero GT

2.3.1.5. Aplicaciones como creadoras de ingreso

Los objetivos de las aplicaciones anteriores indirectamente generan ingresos a la empresa. En el caso de esta división se crea un ingreso directo, el producto es la aplicación, lo esencial es que el negocio se beneficie económicamente por cada usuario que utilice la aplicación. Los procedimientos para estas aplicaciones se edifican estratégicamente, un ejemplo de ello es la difusión de aplicaciones gratuitas, en las que habitualmente se genera un espacio publicitario en la aplicación y demás empresas publican sacándole provecho. Otro método es generar cargos extras por utilidades únicas de la aplicación. Esto constantemente es aplicado a los juegos, donde la moneda

dentro del juego debe ser comprada con dinero real, finalmente es posible que la aplicación se venda a un precio específico. Por mencionar algunas de estas aplicaciones se presentan las siguientes:

- *Deezer App*
- *Spotify App*
- *TeamSpeaker*
- *Netflix App*
- *Crunchyroll App*
- Mi Presupuesto

2.4. La aplicación

Antes de continuar con una estrategia de *marketing*, es de suma importancia tener en claro qué se va a construir, tiene que pasar un proceso de planeamiento donde cada etapa definirá con claridad el producto final. Esta labor no se puede omitir, una aplicación móvil no sirve como contribuyente al negocio sino que cumple una finalidad. Muchas personas hacen aplicaciones como *hobbies* y aunque existen casos en que se obtuvieron grandes cantidades de dinero (como con el juego *Flappy Bird*) las empresas no deben confiarse y dejar una aplicación a su suerte. Una entidad prudente debe planificar sus proyectos de manera que presenten ventaja competitiva. Desarrollar en su totalidad las ideas que definirán el *software* es un proceso arduo que necesita recaudar todas las metas propuestas y analizar cómo el producto lo elaborará, de este modo los siguientes temas tomarán parte en la labor de guiar en este proceso básico.

2.4.1. La idea

Para concebir una buena idea se pueden establecer métodos comunes como el planteamiento de soluciones a problemáticas diarias, atacando distintos aspectos cotidianos de la sociedad o de la empresa misma, observando los alrededores y todo elemento que pueda ser parte de la solución, pues abarcando la mayor cantidad de piezas se abrirá paso a una nueva idea generadora de iniciativas o innovación. El conocimiento es parte de la herramienta principal de desarrollo. Conocer el mercado, al cliente, los empleados, etc., son armas que solamente un buen emprendedor usaría en su beneficio. La mejor inversión del tiempo se implementa en el fundar una buena idea y madurarla para crear un producto confiable.

2.4.2. Los objetivos

¿Qué propósito tendrá la aplicación? Como en todo proyecto, los objetivos son el medio que define el camino que tomará la empresa para llegar a un resultado. Sin rutas claras un proyecto no puede concretar sus metas. Los objetivos son planteados con base en las retribuciones, dichos elementos son los que se debe analizar. Los objetivos de una aplicación ya se han mencionado anteriormente, solo es de constatar qué es lo que la empresa requiere actualmente, así como para quién, cómo y dónde.

2.4.3. El ambiente

Luego de plantear esa idea, se procede a explorar el campo de las aplicaciones, visualizar quiénes son las posibles competencias o las competencias directas en el mercado, tomando en cuenta también la demanda del producto, lo fiable de construir una aplicación de ese tipo, si es mejor evitar

riesgos altos o lanzarse al instante con una idea innovadora y si la finalidad es medir el terreno. Existen varios estudios que presentan el comportamiento de las aplicaciones móviles en el mercado, y es importante tomar en cuenta dichos artículos, instituciones privadas o independientes que presentan información para que toda pequeña y mediana empresa, corporaciones o cualquier individuo, puedan estudiar el panorama. Como parte de la investigación se han seleccionado algunos indicadores de importancia para analizar el ambiente (si se desea observar los documentos completos se encuentran mencionados en la sección bibliográfica).

Figura 2. Información general de “We Are Social”, 2015



Fuente: *Digital, Social and Mobile*. <http://wearesocial.com/uk/special-reports/digital-social-mobile-worldwide-2015>. Consulta: marzo de 2016.

Figura 3. Información general “Google”, 2015



Fuente: *Mobile App marketing insights: how consumers really find and use your Apps.*
<https://think.storage.googleapis.com/docs/mobile-app-marketing-insights.pdf>. Consulta: marzo de 2016.

2.4.4. El sistema

Son los sistemas operativos móviles para los cuales la aplicación estará disponible. Las aplicaciones híbridas otorgan la ventaja de poder ser adaptables a la mayoría de sistemas actuales, pero aun con esta ventaja es necesario analizar los ambientes para evitar pérdidas de los recursos en la adaptación de la aplicación. Aunque sean cambios que no requieren un esfuerzo mayor es conveniente evitarlos y utilizar los recursos de la mejor forma posible para un máximo provecho.

2.4.5. La funcionalidad

Cuando ya se tiene establecida la idea se continúa fijando las funcionalidades, es elemental enlistarlas o documentar estas acciones. Si en ciertos casos se requiere de la interacción con otros sistemas, tomar en cuenta todos los posibles contratiempos que atraería unirse a ellos y, si realmente es favorable, es necesario examinar los recursos que se exhibirán y cómo serán mostrados. También es aconsejable que se cree un bosquejo de la posible apariencia que tendrá la aplicación, centrar las ideas y elaborar el proceso común de todo usuario dentro de la aplicación. El objetivo de todo es tener más claridad del producto, además de facilitar el trabajo al personal que desarrollará el proyecto.

2.4.6. El modelo de negocios y recursos

Incluye conformar el modelo de negocio a trabajar y evaluar si el proyecto puede ser gratuito o necesariamente un servicio de cobro. En el caso de la pequeña empresa, es aconsejable que el proyecto pueda auto mantenerse, no tratar de vender a un alto precio sin que el público conozca del

producto, tomar con calma cada parte y establecer la mejor solución adaptable a la necesidad económica. Si es irremediable el cobro, realizarlo con un proceso moderado y justo u optar por obtener ingresos de terceros, con ayuda de publicidad. Es útil planificar adecuadamente los recursos, si al mismo personal de la empresa se le asignará la construcción del *software* debe tenerse en consideración el uso mayor de tiempo (en los casos de necesitar capacitaciones), lo contrario de contratar personal calificado pero, en ese caso, el recurso monetario será indispensable.

2.5. Técnicas para el *marketing* de aplicaciones móviles

Luego de pasar por todo el proceso para la elaboración del producto, que en el caso de este capítulo solo es la concertación de la idea, se procede al *marketing* de aplicaciones. El *marketing* no es excluyente, ya que todo producto que la empresa se disponga a ofrecer será aprobado dentro del proceso que conlleva todo este desarrollo. Es necesario estudiar el sector ampliamente para plantear el mejor ofrecimiento al público con base en preguntas, desde cómo demostrar que el producto es el mejor, hasta por qué los usuarios deben obtenerlo, dónde se tiene que exponer y demás dudas que serán la base para un excelente inicio de mercado. Toda la teoría se centraliza en el comportamiento humano, la empresa debe analizar a su comprador y el cumplimiento de metas, pues sin ello el sentido del producto se perdería.

Se analizará este proceso por etapas, las cuales se encadenan una tras otra. Es útil tomar en cuenta que estos son consejos que podrían seguirse o no al pie de la letra, toda empresa siempre tiene su propio estilo para realizar sus ventas. Esta sección ayudará a encontrar una fórmula rápida de iniciar en la materia.

La metodología se basa en cómo captar al cliente, pasando primeramente en saber quién es, para luego promocionar tácticamente el producto e ir avanzando lentamente hasta transformarlo en un cliente absoluto. Destaca la sencillez del sistema, pero todo lleva un proceso que debe madurar poco a poco.

2.5.1. El inicio

El inicio de todo proceso es un punto importante, pero aún cuando se pueda sacar provecho de ello no establecerá un final seguro, como buen empresario hay que marcar el inicio y mantener ese éxito en todo el camino. Hoy el sector está inundado de tantas aplicaciones que contribuyen interesantemente a la sociedad, de igual modo existen aplicaciones que son descartadas por ser innecesarias. Como buen inicio de mercadeo es de importancia analizar los errores más comunes que tienden a producirse, para luego tomar de ello la experiencia necesaria y combatir todo obstáculo de un enfrentamiento futuro.

La primera idea que todo vendedor plantea es la publicidad, lo cual no es erróneo, pero debe implementarse de un modo adecuado. Existen medios muy tradicionalistas para realizar este proceso, habitualmente costosos, que vistos para las grandes corporaciones son funcionales. Los típicos anuncios de televisión, radio, vallas, panfletos, etc., son recursos que en publicidad son marcados. Aquí se encuentra el primer error de todo principiante, pues es necesario recordar que se juega el papel de pequeña y mediana empresa, y creer que la publicidad de las grandes compañías funcionará en cualquier sector no es lo correcto. Importante es examinar la posición del negocio antes de lanzar la primera piedra, los recursos invertidos en una publicidad

excesivamente grande pueden causar una bancarrota total al negocio. Buscar medios correctos para publicitar es decisivo para no invertir innecesariamente.

Actualmente el *marketing* digital es una opción bastante fiable, todo lo que envuelve a las redes sociales y a la *web* son opciones donde fácilmente se adquiere mucho público. La cantidad de personas que navegan por dichos medios son exorbitantes. No es bueno depender ciegamente de una sola respuesta, este beneficio puede convertirse en un arma de doble filo, lo importante es sectorizar por campañas para localizar al público ideal. También lanzar publicidad sin medida es uno de los errores más convencionales, no es eficiente, el campo digital entrega suficiente material de clientes para trabajar, ya que divide las audiencias de público e identifica mejor los objetivos de una forma sencilla.

Ya que la publicidad ha permitido abarcar a los clientes supuestos, verificar que es la correcta decisión en el público. Supervisar continuamente las campañas beneficia en la captura de distintos indicadores (conducta de compras, estacionales, mensuales, anuales, lugares de más frecuencia, el número de visitas en la publicidad, etc.) que permitirán una buena decisión para la siguiente fase de *marketing*. El mercadeo digital, además de ser un método económico, es un excelente sector para la recaudación de recursos que conforman esos indicadores antes mencionados. Herramientas como *Google Analytics* y *Facebook Insights* reducen el trabajo de análisis de mercado.

2.5.2. La captura

Realizado un buen proceso de publicidad, se debe integrar al cliente a la utilidad (aplicación móvil), entonces, ¿cómo debería hacerse? El siguiente paso integrado en la publicidad se concentra en familiarizarse más con el cliente, no

forzarlo a comprar y adquirir el producto, sino hacer de ello un proceso calmado y sin prisa. El comportamiento humano es un análisis esencial. Colocarse en la posición del cliente deja ver que demasiada publicidad y presión provocan automáticamente rechazo al producto, sucediendo antes de conocerlo a fondo. Es mejor respetar el espacio personal del cliente, es un factor que debe tomarse muy en cuenta. Es difícil encontrar un cliente por eso no es bueno intimidar; una publicidad amigable es un cliente satisfecho.

Es necesario conocer en detalle al cliente, entender qué es lo que espera de la compañía y encontrar el medio para mantenerse al tanto. Estas acciones provocan un comprador seguro y la buena relación con el cliente es imprescindible. Para lograrlo es necesario mantenerse comunicado, una de las mejores formas es por medio de correo electrónico, es tan personal como un número telefónico pero menos molesto que llamadas de prestamistas.

Es determinante obtener esta información no de manera forzosa sino que retribuyendo ganancias al cliente por darlo a conocer. En el caso de las aplicaciones móviles, pueden aplicarse de manera que brinden servicios extras al usuario solo con integrar su correo a la aplicación. Existen muchas aplicaciones *free trial* que brindan este tipo de utilidades. En el caso de las aplicaciones gratuitas sería buena opción no integrar todas las herramientas al subirse a la tienda, sino entregar una versión completa hasta terminar el proceso de enlazar la cuenta de correo a la *App*. Varias son las maneras de permitir este paso, lo importante es que el cliente pueda sentirse confiado de que la aplicación es la correcta para depositar su información.⁴

⁴ IPPOLITO, Peter. *La guía definitiva de mercadeo digital que realmente vende en Latinoamérica*. p. 13.

2.5.3. La preferencia

Finalmente, ya establecido el medio de enlace, es momento de incentivar al cliente a adquirir beneficios del producto, si es que se comercia alguno, o realizar la compra definitiva de la aplicación. La construcción de la fidelidad del cliente es un factor que no se debe dejar a un lado, el avance de ello es preciso para que la empresa pueda ganar mercado. La aplicación no debe decepcionar las expectativas del cliente, entregar un producto de mala calidad simplemente atrae el mal nombre de la empresa. Todo este planteamiento de una buena aplicación ya fue considerado anteriormente, por ende se concentra en que el cliente se quede junto al producto, que forme parte del conjunto de aplicaciones diarias con las cuales interactúa y contribuya con un *marketing* automático al recomendar su uso a los miembros de su entorno. Es importante recordar que una aplicación debe ser sencilla, objetiva y clara, no excederse si el intención es darse a conocer, si se trata de un proceso complicado es mejor mostrarlo en un conjunto de partes, entregar primeramente los puntos claves para que el cliente espere con ansias las demás actualizaciones u otras aplicaciones que ayudarán a que su experiencia sea más gratificante. La garantía y la seguridad del cliente son lo que importa para dar nombre a la empresa.

Los clientes fieles no deben tomarse a la ligera. El aumento de clientela es necesario pero sin duda no se debe permitir perder a los que ya establecieron relación con el negocio. Promover eventos donde la situación le da un privilegio más grande al cliente sobre todos es un método para generar ingresos, las famosas cuentas Premium, oro, diamante, etc., son absolutamente métodos para favorecer al cliente, establecer mayores privilegios en la aplicación o expandir su uso. Son esos beneficios que al cliente le agrada obtener, pues percibirá que el negocio tiene la atención sobre él y así es como debe ser: no desestimar a los clientes, por ellos es que la empresa se mantendrá a flote.

3. METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE MÓVIL HÍBRIDO

3.1. Patrones de diseño

Los patrones de diseño se enfocan en soluciones óptimas para casos habituales en la construcción de *software*, por esta razón pueden definirse como herramientas de importancia del desarrollador. Estos patrones inicialmente fueron creados para la conocida programación orientada a objetos, pero debido a la expansión de instrumentos en el desarrollo de *software* ha tomado, de modo conveniente, utilidad en otros sistemas, como en el caso del *software* móvil, donde se define cómo debe estructurarse la interacción del *software* con el usuario, desde el estilo de presentar la información hasta dónde colocar botones o cualquier otro objeto.

El objetivo principal de los patrones es estandarizar la estructura de las aplicaciones con las cuales un cliente de algún sistema en particular puede familiarizar la modalidad del funcionamiento con distintos sistemas anteriores y obtener una mejor experiencia al utilizar aplicaciones. La división se realiza por medio del sistema operativo del dispositivo móvil, destacando la personalidad del sistema, pues de este depende la cantidad de modificaciones con las cuales contará la aplicación. La investigación se enfocará en los dos sistemas más utilizados: IOS y Android.

Cabe mencionar que aun cuando el enfoque sea dirigido a las aplicaciones híbridas, es necesario tener este conocimiento para conformar una aplicación que no es nativa a una como tal, en estilo específicamente. Con

ello se cubren las necesidades de una aplicación que puede responder a los más arduos procesos y continuar siendo híbrida con apariencia nativa.

3.1.1. Patrones de diseño *Android*

Los patrones que pueden encontrarse en los sistemas *Android* cambian dependiendo de la versión, de tal manera que tener previsto el ambiente *Android* donde se efectuará el desarrollo es primordial. De ser necesario cubrir la más amplia gama se recomienda optar por las versiones más recientes del sistema, ya que estas comprenden un mejor soporte y contenido. Para los patrones de diseños mostrados en esta sección se tomarán como referencia las recomendadas por el sitio *Android Patterns*, ya que el contenido ha sido dividido de una forma tal que pueda implementarse de manera adecuada.

3.1.1.1. Manejo de datos

Se enfoca en las aplicaciones que se encargan de mostrar gran cantidad de información al usuario, por lo cual los patrones son generados de manera que pueden establecerse posiciones adecuadas a los objetos para la mejor vista de datos: filtros, ordenamientos, navegación, búsquedas y vistas.

3.1.1.2. Obtención de la información de entrada

Se encarga de la utilización de componentes dedicados a la recaudación de información donde el usuario habitualmente interactúa. Abarca los cuadros de texto, selecciones múltiples, componentes para ingreso de fechas y hora, listas, *spinners* y *sliders*.

3.1.1.3. Navegación

Siendo un proceso importante, la navegación se concentra en permitir al usuario desplazarse desde un punto hacia otro, cumpliendo con las funciones de mostrarle lo que ofrece la aplicación y brindar de manera rápida lo que necesita. Algunos componentes que entran en la navegación son las barras de acciones, *dashboards*, listas de navegación, pestañas y más.

3.1.1.4. Notificación

Una notificación se dedica a mostrarle al usuario los eventos acontecidos en su dispositivo, sobre condiciones del móvil o bien redes sociales. Una notificación es de importancia ya que es el medio de comunicación de la aplicación con el usuario. Entre los componentes se encuentran las barras de progreso, diálogos de alerta y de *estatus*.

3.1.1.5. Personalización

Los componentes de personalización permiten al usuario establecer propiedades a la aplicación que le brinda la autorización de poder administrar. Componentes como las plantillas de *logeos*, creación de cuenta, favoritos, entre otros.

3.1.1.6. Interacción de pantalla

En las interacciones de pantalla se concentran todas las actividades que puede realizar el usuario con el dispositivo físicamente, excluyendo de ellas el *single tap* que se reserva para la acción de seleccionar. Movimientos como el

deslizar, arrastrar y el doble toque son parte de los abarcados en las interacciones.

3.1.1.7. Social

Entre las aplicaciones son muy comúnmente las de actividades sociales, es por ello que esta parte se dedica a las acciones que corresponden a la comunicación entre usuarios de la aplicación, entrando en esta categoría los *rating*, comentarios, acciones de compartir, etc.

3.1.2. Patrones de diseño IOS

IOS establece las métricas para el diseño de aplicaciones con las cuales pretenden brindar la mejor experiencia de interacción entre sus usuarios. Estas se encuentran establecidas en la página oficial iOS Human Interface Guidelines.

3.1.2.1. Visualización

IOS tiene su propia filosofía para la estructura de sus aplicaciones. Se concentran en:

3.1.2.1.1. Deferencia

Donde las interfaces permiten facilitar la interacción del usuario con la aplicación y no provocar un mayor conflicto entre estos.

3.1.2.1.2. Claridad

Donde todos los aspectos visuales, desde el texto hasta las imágenes, se componen de una forma sutil, permitiendo ir directamente de la funcionalidad al diseño.

3.1.2.1.3. Profundidad

Permite que el modelo visual de las aplicaciones proporcione una distribución jerárquica reconocible, donde los elementos no están envueltos en capas complejas y las transiciones puedan ser manejables. Denota su facilidad de comprensión.

3.1.2.2. Anatomía

Igual que otras aplicaciones, IOS establece los componentes básicos para una aplicación en sus sistemas guiados por el *framework* UIKit. Se divide en 4 categorías:

3.1.2.2.1. Barras

Son divididas entre barras de navegación y pestañas, las cuales indican al usuario su posición en la aplicación y las acciones futuras a ejecutar:

- Barras de estatus
- Barras de navegación
- Barra de herramientas
- Pestañas
- *Scope Bar*

3.1.2.2.2. Contenido de la vista

Son las actividades en las cuales el usuario tiene permitido efectuar acciones como las de desplazamiento, inserciones, eliminación y reubicación de elementos:

- Actividad (*Activity*)
- Vista del controlador de actividades
- Vista de colección
- Vista de imágenes
- Vista de mapas
- Controlador para vista de páginas
- *Popover*
- Vista de desplazamiento
- Vista de tablas
- Vista de texto

3.1.2.2.3. Control

Permite la realización de acciones en pantalla u obtención de información:

- Indicador de actividades
- Botón de “agregar contacto”
- Selector de fechas
- Botón de detalle
- Botón de información
- Etiquetas
- Indicador de actividades de red

- Control de página
- Selectores
- Vista de progreso
- Control de refrescado
- Control de segmentación
- *Slider*
- *Stepper*
- *Switch*
- Sistema de botones

3.1.2.2.4. Vistas temporales

Como una notificación, una vista temporal brinda brevemente al usuario la información de los sucesos ocurridos en la aplicación, así como de incorporar funcionalidades especiales o extras:

- Alertas
- *Popup* de acciones
- *Modal view*

3.1.3. Patrones de diseño en un ámbito general

Dejando a un lado la exclusividad de los sistemas y las tendencias que generan en sus clientes, estos mismos comprenden un concepto en común a lo que se refiere en patrones de diseño, con ello se unifica el siguiente esquema de patrones, que es dividido en distintos puntos que toman en cuenta cada aspecto principal de la aplicación. Los aspectos son propuestos por el libro *Diseñando apps para móviles*, escrito por Javier Cuello y José Vittone, el cual permite visualizar globalmente las propiedades de diseño de una aplicación.

3.1.3.1. Simplicidad

La simplicidad es la pieza que permite a una aplicación móvil ser lo que es, un producto con aspecto totalmente minimalista dada sus limitaciones espaciales, que otorga una funcionalidad específica y no de aspecto sobrecargado (tal como la mayoría de *software*). Las aplicaciones otorgan un conjunto de funciones que representa sus objetivos, es por ello que, si cumple estas acciones, el cliente estará satisfecho y dejará a un lado los aspectos llamativos que posiblemente otra aplicación pueda ofrecer. Cuando se habla de simplicidad, no quiere decir que los aspectos deben ser vacíos o aburridos, sino que existen métodos que proponen un número pequeño de colores e imágenes que dan el mejor de los resultados para el aspecto de la aplicación.

3.1.3.2. Consistencia

La consistencia permite a la aplicación generar aspectos intuitivos entre los usuarios. Radica en establecer las funcionalidades de una manera habitual, así como todo usuario comprende que el botón de bloqueo de su dispositivo realiza dicha acción, con ese mismo formato una aplicación debe estandarizar su funcionamiento, para que todo usuario pueda conocerla sin necesidad de adentrarse en profundidad en cuestiones de versionamiento del sistema.

La intuición y el accionar del usuario comprenden también los aspectos visuales que conforman la aplicación y permiten generar un conjunto de conocimientos esenciales en los usuarios. Encontrar los medios donde todos los sistemas del mismo entorno puedan familiarizarse se debe a la fluidez de cada uno.

3.1.3.3. Navegación

El diseño de un desplazamiento bien estructurado es una de las cualidades que determinan la buena elaboración de aplicaciones, disminuye la línea de aprendizaje del usuario y crea la denotada intuitividad. Causar el efecto de predicción en los controles del *software* brinda sostenibilidad y confiabilidad. Estas funciones van de la mano con la consistencia, ya que la navegación constituye parte del sistema operativo como tal. El desplazamiento entre pantallas dentro de la aplicación no debe ser complejo, así como toda la aplicación debe ser simplificada, la navegación debe permitir solamente hacer el trabajo que le corresponde.

3.2. Scrum

Entrando al entorno de desarrollo de las aplicaciones híbridas, es necesario contar con la herramienta que permita orden al llevar a cabo la construcción de la aplicación. Es por ello que conocer un marco de trabajo que genere los mejores resultados en un producto final entrega una gran ventaja al vendedor como tal. Para poder definir Scrum es necesario conocer qué son las metodologías ágiles para el desarrollo de *software*. El concepto de metodologías ágiles se basa en aspectos que comúnmente otras formas de desarrollo no cumplen o pasan por alto, es por ello que una metodología ágil se rige por los siguientes aspectos, conocidos como el Manifiesto Ágil:

- El primer valor es permitir que el cliente tenga el producto de forma temprana, además de poder visualizar su construcción de forma constante.
- Los cambios son bien recibidos en una metodología ágil, aun siendo en etapas tardías, por lo cual encontrar diferencias en alguna etapa de

construcción es permitido, ya que el producto es realizado a la medida de lo solicitado.

- El producto es entregado constantemente, dependiendo de la exigencia del solicitante. Las metodologías ágiles permiten contemplar la estructuración del *software* en lapsos cortos.
- El trabajo de construcción depende de un equipo muy bien organizado para cumplir las etapas necesarias para la construcción del *software*.
- La relación cliente-desarrollo es bastante estrecha, ya que es de importancia que el cliente esté integrado a la generación del producto, desde un inicio hasta el final.
- Fuera de los aspectos del cliente, es necesario que el equipo de trabajo internamente esté motivado en la construcción del producto, ya que la metodología depende del entusiasmo de los trabajadores.
- El buen trabajo depende de una buena comunicación, de ello es que nace la unión de equipo.
- El objetivo es el funcionamiento del producto.
- Mantener un desarrollo con ritmo constante, por lo cual todo involucrado en el desarrollo debe comprender su responsabilidad en el.
- Finalmente, concentrarse en el mejoramiento constante del trabajo realizado con la metodología, equipos organizados por sí mismos y contribuciones propias para maximizar al límite las técnicas del desarrollo.

Estos principios estructuran los pilares fundamentales de las metodologías ágiles. Scrum nace de los principios que son transformados por su singular forma de actuar ante las problemáticas provenientes del desarrollo del *software*.

3.2.1. ¿Por qué utilizar *Scrum* en el desarrollo de *Apps* híbridas?

Las ventajas de alejarse de la complejidad de aplicaciones nativas y la sustentación de cualidades funcionales permite distinguir el punto favorable en las aplicaciones híbridas. *Scrum*, al ser una metodología bastante razonable, acoge de manera positiva y eficiente el desarrollo híbrido. *Scrum* aporta todos los beneficios de un desarrollo ágil (por lo cual el cliente podrá visualizar constantemente el proceso de construcción), además de brindar los cambios necesarios para un resultado requerido.

Generalmente puede ser acoplado a los desarrollos pequeños, donde una aplicación móvil encaja a la perfección con reuniones de planificación y revisión. *Scrum* crea el ambiente óptimo para que un *App* pueda ser desarrollado sin ningún inconveniente. Es necesario mencionar que el desarrollo en esta investigación fue dirigido a una implementación individual, pero es ventajoso que si se desea crear un *software* por medio de esta metodología es necesario elaborar un equipo de trabajo. *Scrum* fue desarrollado para que su funcionamiento óptimo sea implementado por un equipo muy bien organizado, pero no deja de lado que ciertos aspectos del mismo sean aplicados por una sola persona. Los beneficios de crear un equipo de trabajo son poder extender las responsabilidades del desarrollo, generar roles necesarios para que cada punto en el proceso de elaboración a través del método sea implementado de una forma exacta y precisa, por lo tanto, individualizar estos procesos es una ventaja que prevé acciones que controlan las tareas en aumentos inesperados del proyecto sin poner en riesgo el trabajo realizado.

Optar por la elección de *Scrum* representa constantes ventajas, siendo estas las que aportan la mayor parte en el desarrollo, además de proporcionar

el acoplamiento de *Scrum* al proyecto. Más adelante se define otra metodología que expande los beneficios al marco de trabajo, compuesto de *Scrum* y la siguiente metodología, con lo cual se sacará provecho al desarrollo de la *App* propuesta.

3.2.2. Iniciando *Scrum*

Scrum está conformado por aspectos que lo individualizan de otros marcos de trabajo, por lo cual está dividido en componentes principales que permiten que la elaboración del producto esté organizada. Los siguientes puntos que tomarán parte de la elaboración de *Scrum* han sido tomados del libro *Gestión de proyectos Scrum manager* y del libro *Scrum y XP desde las trincheras*.

- Roles
 - El equipo
 - El propietario del producto
 - El *scrum master*
- Artefactos
 - La pila del producto
 - La pila del *sprint*
 - Incrementos
 - *Sprint*
- Eventos
 - Reuniones de planificación
 - *Daily scrum meeting*
 - Revisión del *sprint*
 - Retrospectiva del *sprint*

3.2.2.1. Roles

Los roles en *Scrum* son parte de la distribución del trabajo que implica la producción del proyecto, en este caso existen dos tipos de roles generales que integran el equipo: los comprometidos y los implicados. Los comprometidos son aquellos que están dentro de la creación del producto, es decir aquellos que físicamente desarrollan el producto. Los implicados son los interesados en la existencia del producto, como puede ser el *marketing*. La división específica de estos roles se desglosa por el propietario del producto y los que componen el equipo, que juegan el papel de comprometidos.

3.2.2.1.1. Propietario del producto

El propietario del producto es el que delega la accesibilidad de las cualidades que el cliente presenta abriendo las puertas a un producto real e interacciona directamente con el cliente. Este rol depende de que solo sea un individuo el que lo represente, las razones son simples: es necesario que las decisiones sean tomadas rápidamente, evitando conflictos en el caso de tener más de un *product owner*. Las responsabilidades de este son la toma de decisiones como un supuesto cliente, además de ello se concentra en que el flujo de trabajo y que todos los elementos en él sean adecuados, mantiene una visión global del proyecto, su interacción en los puntos clave del desarrollo da el visto bueno a la finalización del producto. Es conocedor de la inversión y el plan, debe satisfacer el esfuerzo generado por el equipo.

Para poder tomar parte como *product owner*, la persona debe ser conocedor de *Scrum*, para poder impartir los aspectos que le corresponden, como lo son la administración y el desarrollo de pila del producto, tener la visión del producto, para que pueda tomar las decisiones correctas de qué se debe y

qué no hacer durante el proceso de construcción. Finalmente, y no menos importante, debe tener la debida comunicación con el equipo de trabajo, ya que forma parte principal de las interacciones de cada uno de ellos.

3.2.2.1.2. El equipo

El equipo que conformará el desarrollo de *Scrum* es importante, ya que genera el producto en bruto. La mano de obra permitirá representar los planes en estados físicos para una entrega completa del proyecto. Los equipos de desarrollo tratan de ser unidos en los aspectos de comunicación, ya que de ello depende la localización de posibles fallas en la elaboración. El tamaño se reduce de 4 a 8 miembros en casos muy extremistas.

Los equipos de *Scrum* trabajan estratégicamente, ya que de ellos depende la entrega de un buen resultado, son dirigidos por medio de las decisiones del propietario, por lo cual todas las acciones son dependientes del exterior. El equipo de trabajo debe conocer altamente la visión del propietario, ellos deben poder integrarse a un objetivo y no desviarse drásticamente del camino dispuesto. Además de ello, el equipo permite la creación de la pila del producto conjunto al propietario, por lo cual colabora con sus opiniones y mejoras. Absolutamente todos los miembros deben ser conocedores de *Scrum*, ya que forman parte de las decisiones en cada *sprint*, es por ello que todo miembro debe tener un carácter responsable e igualitario, ya que una acción fuera de las normas de *Scrum* puede dañar la integridad del resultado.

3.2.2.1.3. Scrum master

El *Scrum master* es aquel experto en el tema que asesora y organiza al equipo de tal forma que pueda cumplir con las responsabilidades asignadas. Forma parte de las revisiones y aceptación de la pila de trabajo, mantiene el control sobre las acciones en las reuniones diarias, hace gestión debida ante todo el proyecto y, principalmente, es quien permite que *Scrum* se ejecute al pie de la letra, desde los tiempos definidos en las reuniones hasta el personal que está implicados en el desarrollo.

3.2.2.2. Artefactos

Los artefactos son los componentes con los cuales los comprometidos en el proyecto trabajan para llevar el producto a un resultado final. Están conformados por elementos que interactúan en cada fase del desarrollo de *Scrum* y son empleados por cada uno de los roles de la metodología.

3.2.2.2.1. Pila del producto

Las pilas de producto son el conjunto de requerimientos que el cliente describe sobre el mismo, conforman la existencia de la pila, son parte de ella, por lo cual están constituidas de todos los elementos esenciales para la construcción y producción del proyecto. Esta lista está ordenada por medio de las prioridades brindadas a cada elemento. La pila de productos o *backlog* es el elemento inicial para elaborar un resultado final, tiende a un crecimiento conforme avanza el proyecto, cada una de estas partes deben ser concretadas para darle un punto final al mismo. Los elementos del *backlog* están compuestos por los siguientes descriptores:

- Identificador: así como todo elemento individual descrito dentro de un conjunto, es necesario saber diferenciarlo de los demás, por esta misma razón se utiliza un medio por el cual es posible identificarlo, con ello se salda todo cambio que pueda ocurrir en su nombre original. El ID puede ser propuesto por los comprometidos del proyecto, brindándole un valor incremental o bien dependiendo del caso.
- Nombre: el nombre es un descriptor que especifica la acción del elemento de la pila, debe ser concreto y diferenciado de los demás, es necesario que no sobrepase un estimado de 10 palabras y no menos de 4. Un ejemplo es “mejorar el crecimiento del sistema”.
- Prioridad: es la escala de importancia que posee el elemento de la pila. El propietario la genera conforme a sus decisiones. Permite establecer un ordenamiento desde lo más importante de la aplicación hasta lo que realmente forma parte de ella pero no derrumba el resultado si no se realiza primero.
- Estimación: el esfuerzo en la elaboración es parte importante en el desarrollo, ya que de ello depende el tiempo en el cual ese elemento será terminado en completo estado para una integración final. Poder dar una evaluación inicial para la construcción de cierto componente de la pila es primordial, estableciendo un punteo considerando todas las acciones que permitirán concluir completamente cierta parte. Tanto como el propietario y los miembros de equipo forman parte de la deducción de los tiempos de elaboración, siempre un punteo de mucho más alto valor llevará aún más tiempo que uno de menor. Debe considerarse que en el mejor de los casos una de estas partes dura 2 días, el propietario debe establecer de 6 a 8 días como tiempo adecuado.

Es importante mencionar que en la pila hay partes concretas, partes que pueden ser entregadas al cliente, con un funcionamiento y vista ya terminadas.

Entre ellas es necesario mencionar que un prototipo no forma parte de la pila de productos y menos de procesos como elaboración de espacios de trabajo o bases de datos.

3.2.2.2. Pila del *sprint*

El *sprint backlog* es la separación de cada elemento de la pila, descomponiendo estos en tareas o actividades que conformarán la construcción del proyecto. Básicamente es crear una nueva pila pero, como referencia, se utilizará una de las partes generadas en la pila global. Cada una de estas tareas es asignada a los miembros del equipo en la reunión de planificación del *sprint* y comprendida en tamaños adecuados para un monitoreo constante sobre su avance, para permitir la prevención de riesgos futuros. El *sprint backlog* se compone de condiciones necesarias para que pueda producirse correctamente. Adecuar estos reglamentos permitirá un avance constante en cada inicio de *sprint*.

- El *sprint backlog* debe estar conformado por todos los miembros del equipo, tomando en cuenta sus opiniones y deberá haber unanimidad en la decisión final.
- Toda tarea es tomada en cuenta para poder finalizar con éxito el *sprint*.
- El equipo es el único capaz de realizar los cambios durante el proceso de elaboración del *sprint*.
- Toda tarea excesiva, o que implique más tiempo de lo que normalmente un elemento de la pila tomaría, es necesario descomponerla para que no altere la secuencia de elaboración.
- Toda tarea debe ser visible físicamente ante todo el equipo de trabajo, generalmente en una pizarra o tablero ubicado en el mismo espacio de labores.

Para poder concluir con el proceso de un *sprint backlog* es necesario tener organizadas todas las acciones a realizarse. Tener el orden adecuado y de quiénes son los participantes permitirá que en todo final de *sprint* pueda obtenerse el rendimiento, además de comunicarse en el momento necesario con el que realiza ciertas acciones en cuestiones de cambios.

3.2.2.2.3. Incremento

El incremento es el componente del producto generado en el *sprint*, una parte papable y funcional lista para ser entregada al cliente. Todo *sprint* tiene como resultado un incremento, por lo cual debe ser muy bien planificado para tener confrontación con las siguientes entregas. Finalmente, un incremento debe de ser probado antes de salir del *sprint*, y en caso de que sea necesaria alguna documentación sobre el desarrollo esta también forma parte de lo necesario para generar el incremento.

3.2.2.2.4. Sprint

El *sprint* básicamente es la ejecución de todas las tareas que han sido seleccionadas para dicha acción, tiene que estar preparado el ambiente para que el *sprint* se genere, es decir debe existir el *product backlog*, y que todos los miembros estén entendidos de lo que existe en él y de las prioridades claras para una ejecución en el tiempo planeado, además de que el *product owner* debe tener control sobre las historias generadas, ya que estas plantean los posibles cambios que intervendrán en la priorización de cada parte de la pila.

3.2.2.3. Eventos

3.2.2.3.1. Planificación del *sprint*

El *sprint* tiene una reunión de planificación donde se brinda información, toda la necesaria para la ejecución, del objetivo del *sprint* y lo necesario para que finalice de la forma adecuada, tanto tareas como procesos que comprende.

Lo que produce una planificación son las metas del *sprint*, el conjunto de miembros integrados en dicho *sprint*, además del ya mencionado *sprint backlog* y, finalmente, el tiempo establecido para la entrega del *sprint* y el lugar donde se conformarán las reuniones diarias. Es importante recalcar que todos los miembros del equipo, incluyendo al *product owner*, deben asistir a las reuniones de planificación, debido a las variables del *sprint* y el alcance, estimación e importancia, ya que dos de ellas las propone el *product owner* y la otra el equipo. Es necesario que cuando se esté planteando un resultado, estas vayan siendo pulidas hacia un estado en el cual esté todo estimado concretamente, porque hay cuestiones que a través del proceso producen contratiempos, por lo cual cada miembro debe aclarar sus dudas y opiniones para llegar a un arreglo adecuado sobre lo que se producirá y lo que no en dicho *sprint*.

El trabajo de un Scrum *master* en estas reuniones se trata de mantener el estado del mismo al finalizar el *sprint*, además de ello debe tomar en cuenta todas las variables ejecutadas en el *sprint backlog* y estar en constantes reuniones con el cliente para mantenerlo al tanto sobre las modificaciones que deben implementarse para entregar un producto de calidad. También debe ayudar al equipo a mantener el objetivo claro sobre el *sprint*, resumidamente que todo lo que se está planteando en la metodología Scrum sea concretado al pie de la letra en las planificaciones.

3.2.2.3.2. Daily Scrum Meeting

Básicamente son las reuniones programadas al final del día para proporcionar la información generada en el *sprint*. Cada reunión se realiza en no más de 15 minutos y se enfoca en establecer los procesos para la siguiente jornada. Se sincronizan los avances al final del día de cada miembro, además de establecer nueva información de lo logrado en la jornada, la pila del *sprint* es actualizada en cada reunión diaria identificando los posibles riesgos o necesidades del *sprint*.

Las herramientas, como la pizarra para el avance del *sprint*, son de utilidad para visualizar físicamente las acciones realizadas en el proceso; en ellas todo miembro comparte la información designada en su tarea. Se especifica tanto lo que se ha logrado individualmente como lo que se realizará en la siguiente etapa del mismo *sprint*, es necesario que todo miembro comente todos los sucesos que pueden causar posibles problemas que afecten la finalización del proceso. Un resumen detallado de todo lo que se ha elaborado en el día.

3.2.2.3.3. La revisión del *sprint*

A diferencia de una reunión diaria, las revisiones de *sprint* son evaluaciones al final de dicho proceso, se concede un tiempo adecuado conforme a los *sprints* ejecutados, pues se evalúa de 1 a 4 horas, por *sprints* ejecutados de 5 a 7 días. Los objetivos en una revisión son la comprobación por parte del *product owner* de todo el proceso que conlleva el proyecto, de ello pueden sobresalir decisiones como la modificación de estimaciones sobre algunos *products backlog*, con lo que se define una trayectoria más certera para la ejecución exitosa del producto. Se define también el punto de tareas

completadas para aquellos procesos que son producidos, además de descartar los que aun para su perspectiva necesitan una mejora.

El *feedback* es un elemento esencial en las revisiones, ya que cumple el cometido de poder tomar las buenas y malas acciones para completar mejor el siguiente *sprint*. El incremento es el resultado del *sprint*, por ello deben listarse todas las tareas y resultados que comprende el incremento, esto individualmente para ver cada punto efectuado. Las reuniones deben seguir el protocolo de metodología ágil, por lo cual deben ser precisas y no perder tiempo en discusiones que pueden resolverse de una forma concreta.

El equipo demuestra el funcionamiento del incremento logrado en el *sprint*, con ello se abre las puertas a todas las sugerencias que pueden integrar un mejor resultado. El *product owner* debe ser capaz de obtener toda la información necesaria del incremento en esta reunión, para que más adelante pueda efectuar los cambios necesarios. Finalmente, el *Scrum master* da por finalizada la reunión y fecha la siguiente para continuar con el proceso.

3.2.2.3.4. Retrospectiva

La retrospectiva es muy distinta a una *daily meeting*, o bien, a una reunión de revisión, ya que en ella se comprende cómo se está haciendo el trabajo, y no realmente lo que se está produciendo, exactamente cómo se trabaja en cada proceso y cómo se desarrolla el producto.

Estas reuniones se ejecutan después de la revisión del *sprint* y antes de la planificación del siguiente, son reuniones que son necesarias para mejorar los métodos de trabajo, ya que están alejados de la metodología con la cual se produce el proyecto *Scrum*. Ayudan a organizar de una mejor forma el proceso

de ejecución y resultados, ya que las acciones provenientes se programan o se analizan. Dichos elementos de la pila son distintivos del individuo que los ejecuta. El mejoramiento individual permite que se tenga un mejor resultado contemplando la satisfacción de una entrega exitosa.

De esta misma forma nace el razonamiento de integrar buenas prácticas de programación dentro de la metodología de *Scrum*. Más allá de gestionar un proyecto, es necesario que todo desarrollador tenga el material necesario para saber cómo producir correctamente un resultado exigente como lo es el incremento, por tanto, en esta parte se define el conjunto de métodos provenientes de la metodología XP, que formará parte importante en la producción del resultado esperado.

3.2.3. Integración de XP en *Scrum*

Una combinación altamente valiosa es la implementación de XP en *Scrum*, pero, exactamente, ¿qué es lo que XP (*eXtreme Programming*) aporta? XP es un conjunto de métodos generado como parte de metodología ágil, por lo cual la pregunta es ¿por qué no implementar solo XP? Es sencillo deducirlo, pues XP no abarca completamente una gestión de proyecto con reuniones o decisiones, como lo es *Scrum*, sino básicamente se enfoca en que el desarrollo sea cometido de una forma rápida, con estándares que proporcionarán a los desarrolladores las herramientas para finalizar el trabajo.

Aun cuando XP implementa propios roles y métodos para el desarrollo de un proyecto, es necesario enfocarse en las técnicas que lo hacen efectivo, introduciendo a una ejecución altamente ágil como lo es *Scrum*. Los siguientes procesos se integrarán como labores que los miembros del equipo deben efectuar además de las ya mencionadas en *Scrum*.

3.2.3.1. Programación en parejas

La programación por parejas es una de las técnicas distintivas de XP, donde existen dos miembros colocados en un mismo sitio generando código de una forma en la cual uno puede prever situaciones erróneas que su compañero no puede ver. Existen dos roles: el “controlador”, quien es el que efectúa la codificación, y el “navegador”, quien dirige el trabajo del controlador.

Aun cuando este método es criticado por las desventajas que pueden mostrarse a futuro, el desarrollo en parejas brinda una mejor conformidad en *Scrum*, ya que permite elaborar un mejor resultado. Las ventajas de la implementación de esta técnica son:

- La generación de un código con mayor calidad.
- El recordatorio y guía del navegador hacia el controlador.
- Su aprendizaje es rápido, depende a su vez de las personas que están implementando la técnica.
- Las revisiones antes de la finalización de la jornada tienden a ser de mejor calidad.
- Permite generar el valor de disciplina entre los miembros, ya que están atados a un resultado unificado.
- En esta parte puede definirse como ventaja o desventaja, ya que, dependiendo de las personas, puede unirse de mejor manera el equipo, o provocarse un conflicto internamente. De cualquier forma es mejor identificar los problemas antes de poner en riesgo el proyecto.

3.2.3.2. Desarrollo guiado por pruebas

El desarrollo guiado por pruebas se concentra en generar el código específico del desarrollo pero también de crear un *test* que permita sobrepasar este código de buena forma. Prueba y optimización es lo que concentra este método.

Este tipo de desarrollo tiene como contratiempo que los trabajadores tardan en entenderlo y, además, implementarlo. Es necesario que se tome el tiempo para que pueda adaptarse, pero finalmente es una inversión que contribuirá a que el programador pueda asegurar que el fragmento de código hará lo que debe de buena manera. Existen herramientas que permiten que las pruebas sean realizadas más rápidamente, queda en las manos de quien lleva a cabo el proyecto poder elegir la correcta, dependiendo tanto del equipo como las pruebas que se hacen con cada una ellas. Estas pruebas se pueden implementar desde el nuevo código generado, posiblemente retrasando el inicio del mismo proyecto, pero obtener el resultado de una forma más segura otorga una inversión altamente segura.

3.2.3.3. Diseño incremental

El mejoramiento continuo de cada proceso es fundamental en la producción de los resultados, es decir el no tener que estancarse en una sola técnica sino más bien ir cada vez probando nuevas para encontrar la que encaje concretamente en el proyecto, así como adoptar los mejores métodos para optimizar la construcción entre desarrolladores. Tiende a ser una filosofía de mejoramiento con la que todo proyecto debe contar.

3.2.3.4. Integración continua

La integración continua es otro método en el cual todo desarrollador integra habitualmente el código generado de su tarea hacia un esquema de pruebas donde todo se reúne para luego ser constituido por un resultado final. Su objetivo es eliminar todo error que pueda generar un impacto al producto final, con todo esto y los demás métodos de pruebas debe alcanzarse un resultado altamente satisfactorio y disminuir etapas finales de comprobaciones generales de todo el sistema.

No queda más que agregar que todas estas implementaciones permiten que el producto, en este caso las aplicaciones móviles, permita generar una ganancia y cumplir con su objetivo rápidamente. El desarrollo ágil fue creado para salir apropiadamente de una tarea que, no teniendo una gestión adecuada, puede causar problemas o incluso la no ejecución de la misma. Tener la idea concreta para producir, así como la gestión adecuada, contribuirá a que el resultado sea el esperado.

4. COMPARACIÓN ENTRE *IONIC* Y *PHONEGAP* PARA EL DESARROLLO DE APLICACIONES MÓVILES HÍBRIDAS

Las aplicaciones móviles son un tema de importancia no solo para el crecimiento tecnológico sino para la contribución en avances importantes para la sociedad. El desarrollo no pertenece a un selecto grupo de conocedores, es necesario que esta creencia sea erradicada. El motivo de la investigación es dar a conocer de forma simple la elaboración de las *Apps*, para poder utilizarlas como arma en el vasto comercio, obtener beneficios y visualizar de forma representativa su crecimiento. Aunque una empresa contrate a un ingeniero externo para que elabore la *App*, debe tener en cuenta que las herramientas para la construcción son amplias, hablando del ámbito híbrido, con lo cual es necesario darse a la tarea de buscar la mejor opción.

Las malas decisiones en una empresa pequeña, mediana o grande, pueden repercutir en futuros daños. Es por eso que el representante del negocio debe introducirse al desarrollo de este proyecto, no de forma que afecte el flujo de construcción, sino para que pueda supervisar o darse una idea de lo que se elabora y no terminar con un resultado que incumpla el fin deseado. Para introducirse más en este capítulo es necesario mencionar nuevamente a los *frameworks* (con anterioridad se especificó este concepto) y que la gama de estas herramientas sigue creciendo cada día, con mejoras y resultados más pulidos. Las aplicaciones híbridas se desarrollan con capacidades de mejor control en los móviles y los *frameworks* juegan un papel importante en este logro. Es por ello que se mencionan 2 icónicos en este desarrollo: *Ionic* y *Phonegap*, aunque generalmente ambos siguen un particular

patrón de estructuración y tienen el mismo lenguaje, demuestran puntos importantes que hacen la elección más conveniente.

Además de las utilidades que cada *framework* ofrece, existe la posibilidad de realizar combinaciones con terceros, estas son variadas dependiendo del caso, pero finalmente refuerzan o facilitan la manipulación de los *frameworks* principales. Es necesario realizar un análisis para verificar cuál entrega una mejor optimización en sus rutinas, lo que permitirá a la aplicación ejecutarse de mejor forma; también qué contribuciones proporcionan para el manejo de componentes del móvil, las dificultades para generar ejecutables de los distintos sistemas y, finalmente, las contribuciones estéticas, que son uno de los puntos clave que una aplicación necesita como fuente llamativa hacia el usuario final.

4.1. *Phonogap*

4.1.1. Descripción

De una forma breve, *Phonogap*, también conocido como Apache Cordova, es un *framework* para la construcción de aplicaciones móviles basado en tecnología *web*. Además de ello pertenece al conjunto de herramientas para la elaboración de *Apps* híbridas.

Nació en el 2009, como ganadora de premios en el evento iPhoneDevCamp, San Francisco, donde surge como una nueva idea de desarrollo por parte de *Nitobi Software*. Por cuestiones empresariales Nitobi es adquirido por Adobe, donde nace Apache Cordova. Es necesario especificar que, en el aspecto técnico, Apache Cordova es idéntico que *Phonogap*, solo que Cordova es la base de la estructuración de *Phonogap*, mientras que *Phonogap* es una distribución de Cordova. *Phonogap* es legalmente

perteneciente a Adobe, mientras que Apache Cordova es libre en la asociación Apache. A pesar de los diversos cambios corporativos, Adobe deja en manos a Nitobi el control del *framework*, convirtiendo a la herramienta en *open-source* o código libre, donde cada parte de la comunidad desarrollada de forma ética podría incluir mejoras y soluciones para que el *framework* creciera de mejor forma, además de ser gratuito.

Phonegap se concentra en sus principales componentes, que permiten la elaboración de sus aplicaciones móviles, CC3 y HTML5, que son la revolución actual en la fabricación de los aspectos visuales en las plataformas *web*. Además de ello, existe Javascript, que permite darle la lógica necesaria al *software*, lo cual complementa la creación de un sistema altamente funcional.

Si se habla de los sistemas operativos para móviles, cada uno presenta restricciones que limitan la utilidad o construcción del producto: *Phonegap* decide ser parte de las unidades que generan una facilidad de aceptación entre las distintas variantes de sistemas. Cabe destacar que aun cuando Apple es uno de los más serios en sus políticas, ha permitido que *Phonegap* sea una de las herramientas que dan vida a la gran variedad de aplicaciones que otorgan en su tienda. Puede que existan casos en que no sean aceptados, pero no depende del *framework* sino de las faltas que comúnmente se cometen al desarrollar para dicha empresa.

Phonegap fue diseñado para crear aplicaciones de todo tipo, ya que sus distinguidas *APIS* permiten utilizar la mayor parte de funcionalidades de los dispositivos, así que si se desea utilizar a nivel empresarial es una de las mejores opciones, permitiendo desde arquitecturas de *software* ligeras o robustas hasta protocolos de seguridad que mantendrán segura la aplicación de ataques que comúnmente puedan surgir a lo largo de la vida del *software*.

Además de ello, Adobe entrega herramientas por medio de la nube para dar soporte a los usuarios de *Phonegap*, para que puedan mantenerse al tanto de las últimas novedades, aunque estos recursos extras tienen un costo monetario.

4.1.2. Análisis

Phonegap es un *framework* que cuenta con distintas versiones, las cuales mejoran en cada distribución. Actualmente acepta la integración de varios *plugins* que permiten un mejor control o acceso al *hardware* de los móviles. Aun cuando se utilicen *plugins*, *Phonegap* no es dependiente de ellos. Las *API's* de *Phonegap* hacen el trabajo difícil de poder acceder a los distintos parámetros de los dispositivos. A continuación se muestra el conjunto de componentes que *Phonegap* puede tener como utilidades en las diferentes distribuciones de los sistemas operativos:

Tabla III. **Distribución de componentes en diversos sistemas operativos**

	Iphone/Iphone 3G	Iphone 3GS y más nuevos	Android	Blackberry OS5 x	Blackberry OS6 0+	WebOS	Windows Phone 7	Symbian	Bada
Acelerómetro	si	si	si	si	si	si	si	si	si
Cámara	si	si	si	si	si	si	si	si	si
Compás	no	si	si	no	si	si	si	no	si
Contactos	si	si	si	si	si	no	si	si	si
Archivos	si	si	si	si	si	no	si	no	si
Geolocalización	si	si	si	si	si	si	si	si	si
Media	si	si	si	no	no	no	si	no	no
Redes	si	si	si	si	si	si	si	si	si
Notificaciones (Alertas)	si	si	si	si	si	si	si	si	si
Notificaciones (Audio)	si	si	si	si	si	si	si	si	si
Notificaciones (Vibración)	si	si	si	si	si	si	si	si	si
Almacenamiento	si	si	si	si	si	si	si	si	no

Fuente: *What is Phonegap Cordova*. <http://stacktips.com/tutorials/phonegap/what-is-phonegap>.

Consulta: 19 de mayo de 2016.

Actualmente existen 1392 *plugins* que han sido elaborados para *Phonegap*, tales como los que permiten tener acceso al estado de la batería, cámara, transferencia de archivos (que es importante al realizar transacciones de información), barras de estado, etc. Principalmente, *Phonegap* trata de que ningún parámetro quede fuera del desarrollo de una *App*, es por ello que entrega gran variedad de utilidades que sin duda presentan las ganancias de una aplicación nativa aun cuando esta no lo sea.

Ya teniendo una vista breve de lo que *Phonegap* puede entregar como un producto final, es necesario trasladarse a la mano del desarrollo como parte fundamental para la creación de aplicaciones. Es necesario que la herramienta sea fácil de implementar y de asimilar, *Phonegap* conoce de esto, por lo cual permite que el usuario desarrollador pueda tener distintas maneras de compilar su aplicación, es decir de generar un instalador para la aplicación sin ningún problema. La herramienta, llamada CLI (*Command-Line Interface* o Interfaz de Línea de Comando) proporciona la comodidad que generalmente un desarrollador necesita, sin interfaces complejas, solo una ventana de comandos y opciones que son ejecutadas con un simple “*Enter*”, lo cual integra de manera sencilla la estructuración de una aplicación inicial. Además de ello, *Phonegap* provee otras opciones en la elaboración de la *App*, como utilizar el IDE que el usuario prefiera o un editor de texto.

4.1.2.1. *Phonegap build environment*

El ambiente de construcción de *Phonegap* brinda la oportunidad de que el usuario pueda construir o codificar la aplicación desde el área local, es decir desde su propio computador, donde gracias a este ambiente evitará los problemas de instalación individual con todo tipo de aditamentos que necesita *Phonegap* para generar un instalador, ya que otorga la facilidad de compilar

desde la nube, un servicio que Adobe otorga a los usuarios por paga. Además de ello cuenta con herramientas como Weinre (*Web Inspector Remote Debugging*), que es una de las utilidades necesarias para la programación, permitiendo ejecutar paso a paso dicha codificación y encontrar de forma más rápida todo tipo de errores o cambios necesarios en la aplicación.

4.1.2.2. *Phonegap developer app*

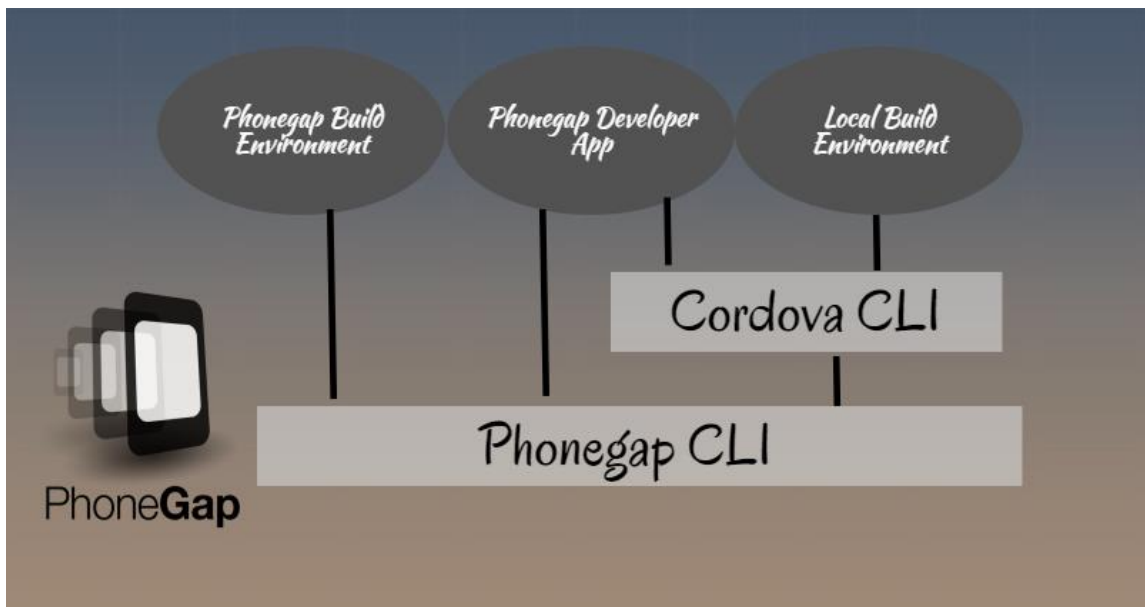
Dicha aplicación de *Phonegap* permite generar en tiempo real la ejecución de las aplicaciones dentro del dispositivo móvil, no es necesario la instalación del proyecto, básicamente funciona como un ambiente de pruebas donde se visualiza todo cambio que se genere a la aplicación desde un dispositivo, para ello no se genera un ejecutable, ya que esta aplicación realiza todo el trabajo. Beneficio de esta es la privacidad, ya que funciona desde la red local, donde es necesario que el dispositivo se encuentre conectado a una red *wifi* por donde se comunicará con el computador. No es obligatorio instalar algún aditamento más que solo la aplicación de *Phonegap*. La única desventaja de esta es que solo pueden correr los *plugins* principales de *Phonegap*, por lo cual las aplicaciones que se ejecuten a través de ella no deben ser tan complejas, o bien, no deben utilizar componentes de terceros para que no existan problemas al poner a prueba la aplicación.

4.1.2.3. *Local build environment*

Un ambiente de construcción local deja a un lado todas las ventajas que brinda la versión en la nube, ya que en este el usuario desarrollador tiene el control total de las instalaciones de componentes para la compilación, además de ello, tiene en sus manos las actualizaciones de dichas utilidades. En estos casos es común utilizar IDE's para instalar fácilmente todas las partes

necesarias que permiten ejecutar una aplicación *Phonegap*, aun cuando sea una interfaz *web* y pueda verse previamente desde el computado, sin necesidad más que de solo un navegador. Es indispensable que los *plugins* que controlen el móvil sean comprobados desde uno, sea simulado o real. Principalmente, la ventaja es un total control, pero con ello se define también que el ente responsable de la manipulación debe poseer el conocimiento necesario para evitar cualquier tipo de incidente, ya que una mala instalación de versiones podría provocar que algunos *plugins* no funcionen correctamente.

Figura 4. **Esquema de comunicación del *Phonegap* CLI con sus distintos ambientes de construcción**



Fuente: SHOTTS, Kerri. *Phonegap for enterprise*. P. 10. Consulta: 25 de mayo del 2016.

Phonegap CLI es el paso para acceder a cada uno de los ambientes, ya que Cordova es un renombramiento de *Phonegap*. Este ofrece algunas accesibilidades para los nuevos ambientes otorgados, como lo son la *App* y el

local seleccionado por parte del usuario. Con esta parte *Phonegap* da vida a la multiplataforma, aun cuando es necesario realizar cambios en las versiones de los *plugins* en cada sistema. El costo de esto se reduce, a diferencia de si reconstruye la misma aplicación para el uso en otra plataforma.

4.1.3. Estructuración

Como es bien sabido *Phonegap* depende de la elaboración del proyecto desde código HTML conjunto con sus otras herramientas, como lo son CC3 y Javascript. A pesar de eso, internamente *Phonegap* no hace una directa comunicación entre el *software* y el dispositivo, el lenguaje nativo se ve sumergido entre estos enlaces, ya que cambiar el proceso de producción de un sistema operativo desde cero es casi nulo. Es necesario que el lenguaje nativo esté introducido, pero no hace mal alguno en el desarrollo, ya que el usuario depende solo de conocer bien los ambientes *web*. De esta forma *Phonegap* presenta una estructuración base para que el proyecto sea elaborado de una forma ordenada y así evitar la mayor parte de errores por mal direccionamiento de componentes en el sistema.

La primera parte se concentra en la carpeta “WWW”, que básicamente es la generada por toda aplicación *web*, en ella se ve envuelto todo aquello relacionado con la interfaz del usuario, así como la lógica que permite su funcionamiento. En los sistemas *web* estas estructuras pueden expandirse más dependiendo de la arquitectura utilizada (mencionada en capítulos anteriores), pero en el caso de *Phonegap* no es necesario ir más allá de ello. Dentro de esta carpeta se definen los archivos “.JS”, que son pertenecientes a lenguajes Javascript. En estos se produce la lógica que permite que un botón o algún cuadro de texto efectúe las acciones que le son otorgadas.

Se incluyen los archivos “.CSS”, que son los estilos o decoraciones que se pueden implementar en una página HTML, básicamente hacen ver de una mejor forma y ordenan los complementos visuales que se aplican a las páginas que se utilizarán. En el caso de *Phonegap*, se encarga del estilo de las aplicaciones, desde colores, formas de botones, estilo de letras, etc.

Finalmente se encuentran los archivos “.HTML”, que son la presentación de la interfaz del usuario, en estos se introducirán todos aquellos componentes que hacen a la aplicación cumplir su función de forma visual, que son la cara de todo el funcionamiento del sistema, desde páginas de validación de usuarios hasta tablas que muestre algún conjunto de información. Son los ejemplos de lo que puede construirse en esta, fácilmente son todas las distintas caras de la aplicación en *Phonegap*.

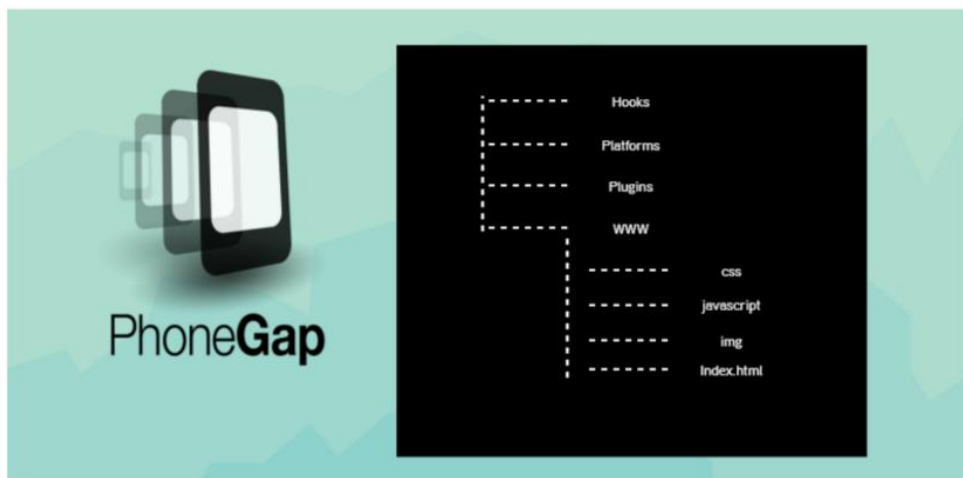
La carpeta WWW forma parte del ambiente de todo *framework* que sea de construcción híbrida. Ahora bien, *Phonegap* provee su propia estructura además de la habitual HTML, pues existe la carpeta “HOOKS”, que es de ámbito opcional, ya que el uso depende de que el usuario desee implementar procesos dentro del CLI de *Phonegap*, es decir, ejecuciones de porciones de código mientras el CLI es ejecutado. En estas pueden entrar levantamiento de servidores, compilaciones, etc. Este proceso procede a ser algo más minucioso si se desea utilizar.

“*Platforms*” es otra sección donde *Phonegap* le dice al compilador para qué sistemas se está trabajando. Cuando el sistema desee generar un instalador irá directamente a consultar a este directorio para conocer qué tipo de extensión creará, de esta forma se accede a los complementos necesarios de dicho sistema operativo. En este irán todos aquellos que *Phonegap* tiene disponibles: *Android, IOS, Symbian, etc.*

El directorio “*plugins*”, como su nombre lo denota, es donde se almacenan todos los *plugins* que en el proyecto se vayan instalando; su funcionalidad no se extiende más. El siguiente es el archivo “*config.xml*”, en el cual se almacenan las configuraciones del ambiente, así como la descripción de los componentes que se están integrando a la aplicación. Puede decirse que es un archivo maestro donde se podrá conocer todo lo referente a la aplicación que se esté construyendo, concentrando toda la información necesaria.

Finalmente, dependiendo de la versión en la cual se trabaje, algunos archivos pueden integrarse pero su funcionamiento no va más allá de lo que realmente se necesita. Básicamente estas son las funcionalidades que *Phonegap* entrega al crear un proyecto nuevo, corriendo desde su archivo “*Index.html*”, que es donde inicia toda la aplicación, de ahí se ramifica en sus distintas vistas o ventanas en las cuales se desarrollará todo el sistema.

Figura 5. **Estructuración de un proyecto en *Phonegap***



Fuente: elaboración propia.

Para establecer cómo ha de contemplarse existe otro directorio denominado “img”. Este realmente solo juega el trabajo de almacenar todo aquel contenido estático de la aplicación, que puede estar formado por imágenes, videos, música, etc., generalmente todo el contenido multimedia. *Phonogap* hace entrega de un marco de trabajo bastante fácil de comprender, por lo cual beneficia al querer trabajar con este. El resultado es un tipo de aplicaciones que no pueden ser fácilmente diferenciadas de una buena aplicación nativa.

4.1.4. Beneficios

Ya que se ha mostrado lo que *Phonogap* otorga como *framework* para la elaboración de aplicaciones móviles, es momento de ver qué beneficios muestra como tal:

- Dentro de los beneficios que habitualmente tiene todo *framework* para *Apps* híbridas, la facilidad en el desarrollo cuenta como uno de sus principales, así como simplificar el aprendizaje de lenguajes nativos complicados a lo liviano del HTML.
- Por ser un *framework* que tiene experiencia en la generación de aplicaciones móviles por su larga historia desde que fue creado, permite establecer una mayor seguridad.
- La posibilidad de compilación a través de un servicio en la nube. Dejar a un lado las cuestiones de versionamiento o instalación de paquetes, para que genere el instalador de la aplicación, permite establecer la libertad que conlleva restarle peso al desarrollo, teniendo resultados más rápidos.
- Crear una proyecto rápidamente a través de simples comandos con ayuda del CLI *Phonogap*.

- Un *framework* gratuito, sin ningún costo para su completo funcionamiento.
- Accesibilidad a distintos componentes de *hardware* pertenecientes al dispositivo móvil.
- Ya que *Phonegap* es una opción *open-source* permite la posibilidad de poder crear nuevos *plugins* que mejoran sus capacidades.
- El *kit* que *Phonegap* presenta para la creación de sus aplicaciones tiene considerado el nivel empresarial que puede surgir desde dichas *Apps*, por tanto las expectativas para este tipo de sistemas son altos.
- Adobe, como empresa, entrega un conjunto de herramientas dedicadas a las empresas que se dedican a la interacción con aplicaciones móviles, en ello se incluye aditamentos y soporte para *Phonegap*.
- *Phonegap*, por ser uno de los *frameworks* más conocidos, entrega una alta documentación entre toda la red.
- Finalmente el esfuerzo establecido en utilizar *Phonegap* es mucho menos que lo que conlleva realizar una aplicación nativa. Se concentra en los menores costos de dinero, tiempo y recursos.

4.2. ***Ionic***

4.2.1. **Descripción**

Ionic es la muestra de lo que se puede realizar con la ayuda de grandes herramientas que permiten un menor esfuerzo y un resultado favorable. Es un *framework* dedicado al desarrollo de aplicaciones móviles híbridas, el cual tiene como objetivo el mejor rendimiento y una estética balanceada, elimina todos aquellos componentes que producen una sobrecarga en el móvil, dando libertad a que la fluidez de las aplicaciones se pueda notar.

Nació en el 2013 como proyecto de Drifty Co, empresa que se dedica al desarrollo de herramientas de *software*, *frameworks* y aplicaciones *web*, donde gracias a la ayuda que otorgó el *feedback* de todos sus usuarios de aplicaciones móviles, se logró constituir un proyecto que dio vida a lo que se conoce actualmente como *Ionic*. Después de varias pruebas, en el 2015 se lanzó una versión estable del *framework*. Actualmente *Ionic* es un sistema *open-source*, así como lo es *Phonegap*, que permite que sus utilidades sean gratuitas. Como parte importante cabe mencionar que *Ionic* funciona a través de Cordova, este permite que toda la comunicación entre el *framework* y el móvil esté dada de forma estable. *Ionic* funciona actualmente a través de HTML5, CSS y Javascript, además de estos *Ionic* utiliza una herramienta bastante poderosa como AngularJS, que permite que las aplicaciones *web* puedan ser más sencillas de desarrollar, reduce a pequeños trozos de código lo que antes podría extenderse a muchos archivos en Javascript o funciones complicadas, y a ello se suma el dinamismo que entrega a las páginas *web*, en este caso a las aplicaciones móviles.

AngularJS se basa en una arquitectura MVC dentro del ambiente del usuario, esto permite que las acciones estén ordenadas de manera que el desarrollador tenga previsto en lo que se está trabajando. Ya que AngularJS es una de las fuentes principales de *Ionic*, es necesario que se pueda entender correctamente si se desea sacar todo el potencial del *framework*. Otro de los puntos donde se concentra *Ionic* es en hacer sentir al usuario la familiarización con la aplicación y sus componentes, que constituyen la interfaz gráfica y se adaptan de manera sencilla, haciéndola ver como una aplicación nativa. La idea es transmitir a la comunidad el control de todas las aplicaciones, respectivamente en los sistemas *IOS*, que son rigurosos con cada parámetro que deben contener sus *Apps*. Imágenes, botones, colores, todo aspecto CSS,

Ionic lo tiene preparado para que con unas simples líneas de código sea implementado.

Como todo buen *framework*, *Ionic* ofrece una gama de utilidades que permiten que el ambiente de desarrollo sea más cómodo de conducir, desde inclusión de publicidad de manera sencilla, hasta la implementación de su propio editor para la creación de interfaces con un simple “*drag and drop*”, sin necesidad de tocar código. *Ionic* es un potente *framework* que saca provecho de lo que actualmente revoluciona los ambientes *web* y con ello las aplicaciones móviles.

4.2.2. Análisis

Es de importancia conocer que *Ionic* trabaja por medio de Cordova, que es el que potencia sus capacidades con el sistema móvil, por ello no es necesario repasar nuevamente los *plugins* que habitualmente Cordova ofrece, ya que estos fueron descritos anteriormente con *Phonegap*, lo que cabe destacar es la unión entre AngularJS con Cordova. *Ionic* ha hecho un gran trabajo complementando dos grandes herramientas y convirtiéndolo en ngCordova, el cual revoluciona el control entre la programación y la funcionalidad de los dispositivos, integra nuevas variantes en los móviles más actuales, como lo son la huella digital, el *loge* entre varias redes sociales y contenido multimedia, y aun cuando ngCordova no tienen en su totalidad los *plugins* posibles, cuenta con los más necesarios para que toda aplicación estándar pueda funcionar.

Cordova, por conceder sus atributos a *Ionic*, le ha permitido el acceso a gran variedad de sistemas operativos, con ello se indica que para cada sistema es necesario el desarrollo en un ambiente específico, es decir en una plataforma que cumpla los requerimientos para que se pueda ejecutar dicha

aplicación. A continuación se presenta la tabla de sistemas en los que se debe desarrollar cada aplicación:

Tabla IV. **Distribución de sistemas operativos para el desarrollo de aplicaciones móviles en sistemas operativos móviles**

	Windows	Mac	Linux
Android	sí	sí	sí
Windows Phone	sí	no	no
IOS	no	sí	no
Blackberry 10	no	sí	sí
Amazon Fire OS	sí	sí	sí
Navegador	sí	sí	sí
Ubuntu	no	no	sí
Firefox OS	sí	sí	sí

Fuente: *Manual de introducción a Ionic*.

https://ajgallego.gitbooks.io/ionic/content/basicos_primer_proyecto.html. Consulta: 30 de mayo de 2016.

Los sistemas operativos son el enfoque de las aplicaciones híbridas. La necesidad de la variedad hace ser a un *framework* el mejor ante los demás cuando sabe llevar bien los cambios entre los ambientes, es por ello que *Ionic* busca a su manera que el cambio no sea brusco, tanto para el usuario final como para el desarrollador, y esto se hace ver con ayuda de sus componentes gráficos. La facilidad con la que los componentes se adaptan al sistema operativo son ventajas que *Ionic* demuestra a grandes rasgos, sin necesidad de hacer mayores modificaciones a sus archivos realiza un enmascarado de sus objetos visuales a la medida de la plataforma, haciendo que un botón se observe tan natural en iOS como en *Android*. La contribución de estas características permite que el desarrollador se preocupe solamente por las posiciones de los ítems, siguiendo respectivamente el patrón de diseño de cada sistema operativo y dejando a un lado el maquillado que es el trabajo de un diseñador en CSS. El principal objetivo es la concentración en el funcionamiento de las aplicaciones. Sumergiéndose más en lo que *Ionic* puede ofrecer, se encuentran sus distintas utilidades, que permiten un desarrollo mejor elaborado, este es el caso de *Ionic Lab*.

Ionic Lab es una aplicación que permite la creación de *Apps* por medio de su interfaz gráfica, la cual deja a un lado lo complejo de codificar componentes y ubicarlos, ya que con su hábil *drag and drop* concede el control al desarrollador de los resultados en el aspecto gráfico del producto. Además de ello, *Ionic Lab* concede vistas previas de las aplicaciones, para realizar pruebas de las mismas por medio de sus simuladores y, finalmente, crear los instaladores de las *App* en los sistemas *Android* y iOS. *Ionic Lab* es la herramienta que deja a un lado lo complejo del desarrollo y lo convierte en sencillos pasos que todo usuario sin mayor conocimiento puede realizar.

Ionic CLI es otra de las utilidades que *Ionic* presenta. Al igual que la terminal de *Phonegap*, *Ionic* CLI permite que con el conjunto de comandos predeterminados pueda crearse aplicaciones con unos cuantos pasos. Esta función está dedicada a aquellos desarrolladores que se encuentran familiarizados con las terminales de Linux o para quienes prefieren un mejor control en la elaboración de sus aplicaciones. *Ionic* CLI es el *core* de *Ionic*, ya que otorga el acceso al usuario de todas las acciones que se pueden producir con este.

Ionic conoce que es importante la aprobación de sus aplicaciones, por lo cual ofrece una plataforma alternativa que permite la publicación de las *Apps*, donde clientes, usuarios u otras personas pueden obtenerlas; esta es llamada *Ionic View*. El papel principal de *Ionic View* es dar a conocer los productos antes de su lanzamiento final para con ello obtener con facilidad las opiniones de los clientes. Juega parte importante en todo el proceso de desarrollo, ya que pueden realizarse los cambios necesarios antes de que se lance a producción para así asegurar que la aceptación en la población sea la esperada.

Ionic no se conforma con que los productos que se elaboren dentro de su marco de trabajo sean sencillos o poco presentables, por ello proporciona el *Ionic Market*, que es una tienda donde otorga distintas plantillas y temas muy bien diseñados para iniciar aquellos proyectos que necesitan ser elaborados en corto tiempo. Así deja a un lado la preocupación por el diseño y se concentra en que la lógica del negocio sea implementada. *Ionic Market* cuenta con un conjunto de *plugins* elaborados por la comunidad que reducen más el trabajo en la implementación de distintos aspectos de las *Apps*, como lo son las barras de menús, las pantallas de bloqueo y muchos más.

Ionic también toma en cuenta las etapas precodificación de las aplicaciones, es decir que considera las fases de diseño. Con la ayuda de *Ionic Creator* el usuario puede elaborar prototipos de sus proyectos con el objetivo de ser presentados a sus clientes, dándoles un vistazo de lo que pretende ser su producto final. En el desarrollo es importante que el cliente sea parte de la elaboración del producto, cuestión mencionada en capítulos anteriores para las metodologías de desarrollo. *Ionic* conoce de qué depende un buen sistema, por lo cual ha creado esta utilidad.

Finalmente, pero no menos importante, se encuentra *Ionic Platform*, que son los servicios en la nube que otorga *Ionic*. Cada utilidad está pensada para que el programador no gaste tiempo en instalación de librerías o paquetes que son necesarios para la compilación u otro proceso, como lo son el despliegamiento e incluso la autenticación de usuarios en la redes sociales, prueba de esto es el *Native Build*, el cual, como parte de *Ionic Platform*, permite el compilado de las *Apps* en sus respectivos sistemas operativos para poder generar posteriormente los instaladores para su producción.

Otro componente es el *Live Deploys*, el cual permite la actualización de las aplicaciones en las tiendas sin necesidad de volver a subir todo el proyecto. Problemas muy habituales que surgen en los despliegues son solventados por esta utilidad. Existe también el *User Authentication*, que recicla las típicas pantallas de inicio de sesión y las convierte en lo que actualmente se utiliza como autenticación por medio de las redes sociales, en las que con unas pocas líneas de código puede implementarse esta parte esencial de una *App*.

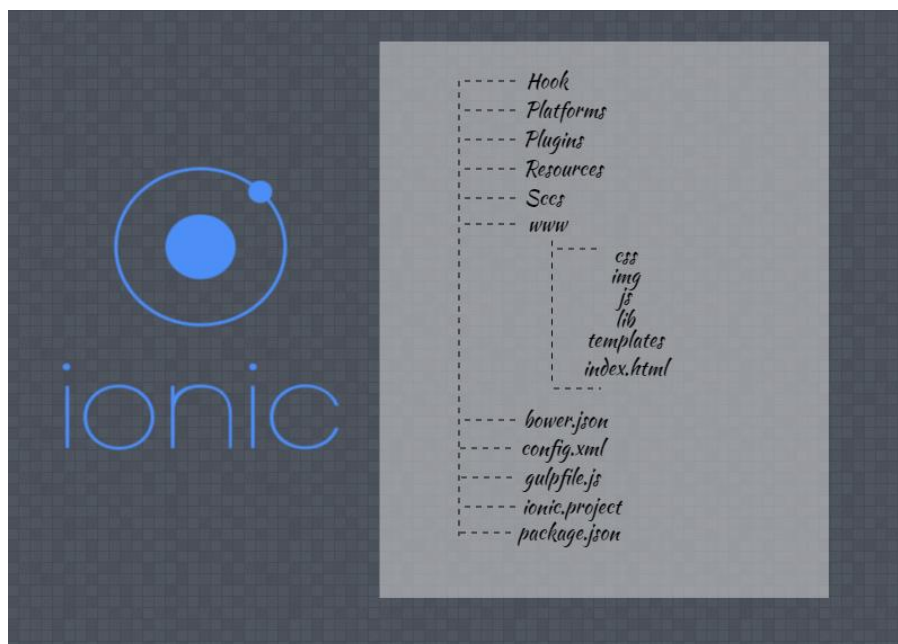
Como una función final, *Ionic* entrega el *Push Notification*, la cual es la tecnología que permite a un servidor enviar mensajes directos a sus clientes sin necesidad que estos configuren previamente la funcionalidad. Es de mencionar

que esta habilidad no es exclusiva de *Ionic*, ya que otros sistemas lo han implementado, pero *Ionic* facilita con este *plugin* que este procedimiento sea más sencillo de producir. Además, demuestra que considera todos los aspectos posibles en el desarrollo, y puede ser catalogado como buen *framework*, ya que hace el trabajo de abarcar los detalles de forma minuciosa, generando una producción de *software* amigable. Indica ser un gran resultado de un buen equipo de trabajo y de la dedicación por revolucionar la construcción de aplicaciones móviles híbridas.

4.2.3. Estructura

Cordova es la base de *Ionic*, por ello la estructuración de sus proyectos es, en la mayoría de rasgos, idéntica a la de *Phonegap*. *Ionic* incluye ligeramente algunos archivos extras para trabajar sus sistemas.

Figura 6. Estructuración de un proyecto en *Ionic*



Fuente: elaboración propia.

- *Hook*: esta carpeta está destinada para el almacenamiento de archivos de rutinas o *scripts*, lo cuales se ejecutarán en algún tiempo de la compilación, al inicio o al finalizar, esto depende de las configuraciones del usuario.
- *Platforms*: son todas las configuraciones dadas hacia la plataforma para la cual se compilará, pueden ser Android, iOS, etc.
- *Plugins*: directorio donde es almacenado todos los archivos necesarios para los *plugins* que se instalen dentro del proyecto de Ionic.
- *Resources*: son aquellos recursos que dependen de la plataforma a utilizar y de los funcionamientos que el sistema operativo necesite.
- *Sccs*: archivos referentes a la construcción de Saas en Ionic.
- *www*: todo lo referente a la construcción de la aplicación, pantallas, detalles, diseños, imágenes, etc.
- *Bower.json*: archivo en formato json donde se almacena todo lo referente a componentes y dependencias desde el gestor de paquetes Bower.
- *Config.xml*: contiene las configuraciones de Cordova.
- *Gulpfile.js*: archivo creado para realizar distintas tareas en la construcción de las páginas definidas por el sistema Gulp.
- *Ionic.project*: configuraciones de Ionic.
- *Package.json*: definición de todas las dependencias de *software* y paquetes definidos para el proyecto en construcción.

4.2.4. Beneficios

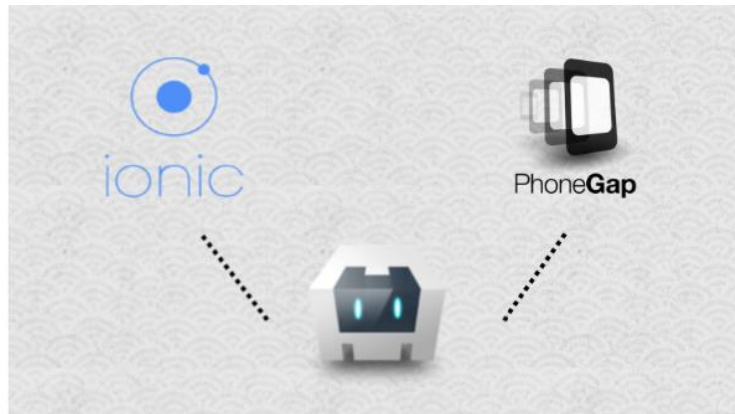
- Su principal utilidad radica en la simplificación y optimización de tareas con los componentes del móvil gestionado a través de AngularJs.
- Código *open-source*.

- Los medios que presenta para la producción de aplicaciones son amplios en comparación con otros *frameworks*.
- Utilidades que permiten pasar una aplicación híbrida por una nativa, gráficamente y sin mayor complicación.
- Accesible a la mayoría de sistemas operativos para móviles.
- Despliegues por medio de un sistema sencillo sin afrontar las complicaciones de las tiendas de aplicaciones.
- Gratuito.
- Una gran cantidad de *plugins* a disposición.
- *Framework* utilizado con gran frecuencia actualmente, por ello cuenta con una documentación muy bien definida.
- Gran cantidad de *templates* para poder iniciar la mayor parte de proyectos.
- Su simplicidad se limita al aprendizaje de utilizar sus herramientas para la construcción de interfaces.
- Bases confiables para la construcción.
- Servicios de compilación a través de la nube.

4.3. Comparación

Ionic y *Phonegap* son dos grandes opciones para poder iniciar en el desarrollo de *Apps* híbridas, dejando a un lado sus diferencias son *frameworks* confiables y de gran rendimiento que permitirán siempre obtener un resultado satisfactorio.

Figura 7. **Representación de estructura Cordova-Ionic/Phonegap**



Fuente: elaboración propia.

Tabla V. **Tabla comparativa Ionic vs Phonegap**

PHONEGAP	IONIC
Sistema con base en Cordova.	Sistema basado en Cordova.
Con un extenso historial de construcción que ha permitido establecer una sólida base en aplicaciones híbridas.	Su elaboración se ha constituido en unos pocos años, por lo cual aún pueden definirse ajustes que mejoren el sistema.
Libertad de utilización entre todos los recursos que permiten el control de los componentes del móvil disponibles en Cordova.	ngCordova es la combinación de tecnologías que facilita el manejo de la mayor parte de utilidades en el móvil, aquellas que no están incluidas deben implementarse de la forma habitual.
La construcción de los sistemas basados en <i>Phonegap</i> tiende a ser más elaborada y toma más tiempo que la habitual forma de desarrollo.	Presenta un mayor número de herramientas que dejan en claro la facilidad en la construcción del <i>software</i> .

Continuación de tabla V.

<p>Es necesario conocimiento en CSS, HTML Y Javascript.</p>	<p>Es necesario conocimiento en CSS, HTML, Javascript y AngularJs, este último es necesario para sacar el mejor provecho del <i>framework</i>.</p>
<p>Las similitudes visuales con las aplicaciones nativas dependen del diseñador y no del <i>framework</i>.</p>	<p>Todos sus componentes están elaborados para adaptarse al sistema en el que se aloje y tomar el parecido a sus aplicaciones nativas.</p>
<p>Brinda un conjunto de servicios en la nube, como por ejemplo el compilado de aplicaciones.</p>	<p>Servicios en la nube, además de implementación de nuevas tendencias de <i>software</i> como el <i>Push Notification</i>.</p>
<p>El <i>framework</i> puede limitarse solo al desarrollo de la aplicación.</p>	<p>Toma en cuenta, dentro del desarrollo del proyecto el diseño, la elaboración y el mantenimiento.</p>
<p>Existe una comunidad muy extensa de habla hispana.</p>	<p>Su documentación puede limitarse mayormente al idioma ingles</p>
<p><i>Framework</i> dependiente de Cordova.</p>	<p><i>Framework</i> que toma provecho de todas las facultades de <i>Phonegap</i> y Cordova.</p>
<p>Su objetivo es la creación de aplicaciones móviles híbridas de manera en que pueda controlarse la mayor parte de <i>hardware</i> del móvil haciéndole tener las capacidades de un nativo.</p>	<p>Su objetivo no solo se centra en el control de <i>hardware</i> sino en la optimización y la ligereza de la aplicación para un total funcionamiento en los móviles.</p>

Continuación de tabla V.

Código <i>open source</i> y gratuito, pero con posibilidades de cobros en un futuro.	Código <i>open source</i> y gratuito.
Sistema que puede estar orientado a construcción de <i>software</i> a nivel empresarial.	Puede limitarse a un conjunto de proyectos que, por su falta de recursos, permitan extender sus capacidades más lejos de lo habitual.

Fuente: elaboración propia.

CONCLUSIONES

1. El primer factor que define el éxito de una aplicación móvil se deriva del conocimiento que se obtiene de los antecedentes que han cimentado esta tecnología. Gracias a estos conceptos básicos el aprendizaje de las partes, tanto de las positivas como de las negativas, contribuye a una mejor toma de decisiones.
2. El entorno híbrido es la manera más rápida de iniciar en el desarrollo de aplicaciones móviles, ya que ofrece una menor curva de aprendizaje y pocos tiempos de desarrollo, además de ello, distintos precursores de esta nueva tecnología la mejoran día con día para poder sacar todo el provecho posible de los móviles.
3. Los métodos de mercadeo más comunes que se implementan con la mayoría de productos suelen ser útiles para el *software* móvil, sean complejos o simples, ya que las aplicaciones son un producto con amplio mercado.
4. El construir una aplicación efectiva se deriva de implementar correctamente, en etapas de concepción, las cualidades de la misma a partir del modelo de negocio.
5. Al utilizar *Scrum* en conjunto con XP como metodologías de desarrollo en aplicaciones móviles híbridas, se obtienen mayores ventajas en comparación con otros métodos, principalmente en su diseño para abarcar proyectos con entregas rápidas y de estructuración simple.

6. A partir del análisis de cualidades dirigido a las herramientas para el desarrollo de aplicaciones móviles híbridas se observa que las ventajas dependen del objetivo que el *software* tenga. *Phonegap* demuestra que lo necesario es la estructuración de un buen proyecto que permita analizar los mejores componentes de un dispositivo y sacarle provecho a sus funcionalidades. De esta manera permite la elaboración de un producto profesional dependiente del análisis preliminar, por otra parte, *Ionic* se concentra en que la aplicación pueda tener el mayor rendimiento posible, no solo otorgando flexibilidad en su construcción sino permitiendo incluir nuevas tecnologías que serán tendencia, por lo cual el producto tiende a ser más sólido.

RECOMENDACIONES

1. Es necesario enterarse día a día de los cambios y nuevas tendencias en los sistemas de aplicaciones móviles, para tener de primera mano ideas que aumentarán las probabilidades de éxito sobre el producto.
2. El capital para el mercadeo de productos es importante, por lo cual es recomendable tener en cuenta una cantidad suficiente para atender esta parte del proceso, independientemente de la metodología de mercadeo a implementar.
3. Entender la importancia de ordenar los procesos de desarrollo sobre el *software*, ya que de ello dependerá que el producto sea entregado bajo el coste esperado.
4. Elegir el *framework* que más se acople al producto deseado, ya que cada uno de ellos ofrece utilidades que pueden ser más complejas de implementar que otras, es por ello importante determinar el alcance del producto.
5. Para realizar una buena implementación de las metodologías de desarrollo, para este tipo de proyectos deben incluirse al menos 5 individuos, para que exista desglose de responsabilidades.
6. El ambiente necesario para el desarrollo de aplicaciones es un punto importante, por eso es recomendable invertir en equipo que permita un desarrollo sin mayor complicación.

BIBLIOGRAFÍA

1. IBM Software. *El desarrollo de aplicaciones móviles, web o híbridas*. [en línea]. <
ftp://ftp.software.ibm.com/la/documents/gb/commons/27754_IBM_WP_Native_Web_or_hybrid_2846853.pdf>. [Consulta: abril de 2016].
2. IPPOLITO, Peter. *La guía definitiva de mercado digital que realmente vende en Latinoamérica*. 2016. 39 p.
3. KNIBERG, Henrik. *Scrum y XP desde las trincheras*. 2007. 116 p.
4. KUMAR, Ash. *A complete guide to mobile marketing for 2014*. Tansense. 2014. 70 p.
5. Mobile Marketing Association. *Libro blanco de las web móviles*. [en línea]. <
<http://www.mmaspain.com/wp-content/uploads/2015/12/Libro-Blanco-Webs-Moviles.pdf>>. [Consulta: marzo de 2016].
6. Mobile Marketing Association. *Libro blanco de Apps*. [en línea]. <
<http://www.mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf>>. [Consulta: marzo de 2016].
7. PALACIOS, Juan. *Gestión de proyectos Scrum manager*. v. 2.5. 2014. 97 p.

8. SHOTTS, Kerri. *Phonegap for enterprise*. Packt Publishing. 2014. 171 p.
9. WARDO, Jhon. *Apache Cordova 4, programing*. Pearson Education. 2015. 503 p.