



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

**BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE  
DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS**

**Erick Rolando Vidal Bazini**

Asesorado por el Ing. Daniel Caciá Rivas

Guatemala, noviembre de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE  
DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**ERICK ROLANDO VIDAL BAZINI**

ASESORADO POR EL ING. DANIEL CACIÁ RIVAS

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, NOVIEMBRE DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADORA	Inga. Susan Verónica Gudiel Herrera
SECRETARIO	Ing. Hugo Humberto Rivera Pérez.

## 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:

### **BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha agosto de 2014.

A handwritten signature in black ink, appearing to read 'Erick Rolando Vidal Bazini', enclosed within a circular scribble.

**Erick Rolando Vidal Bazini**



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería

Guatemala, 25 de septiembre de 2014

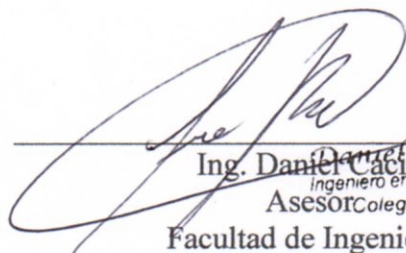
Ingeniero  
Marlon Antonio Pérez Turk  
Director  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala

Ingeniero Pérez Turk:

Me complace saludarle, haciendo referencia al trabajo de graduación titulado "BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS" desarrollado por el estudiante universitario **Erick Rolando Vidal Bazini** con numero de carne 200915388, que como asesor apruebo el contenido del mismo.

Para su conocimiento y efecto, sin otra particular, me es grato suscribirme.

Atentamente,

  
Ing. Daniel Cacia Rivas  
Ingeniero en Ciencias y Sistemas  
Asesor Colegiado No. 8882  
Facultad de Ingeniería



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

Guatemala, 22 de Octubre de 2014

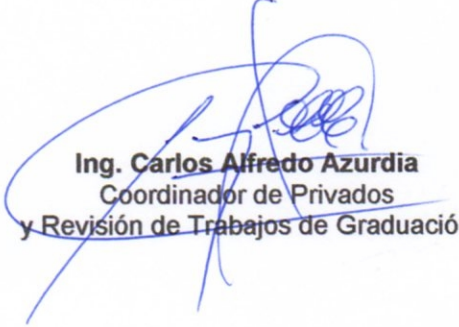
Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **ERICK ROLANDO VIDAL BAZINI** con carné **2009-15388**, titulado: **"BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS"**, 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





E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE 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 **“BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS”**, realizado por el estudiante ERICK ROLANDO VIDAL BAZINI, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

A handwritten signature in blue ink, appearing to read "Martín Antonio Pérez Türk".



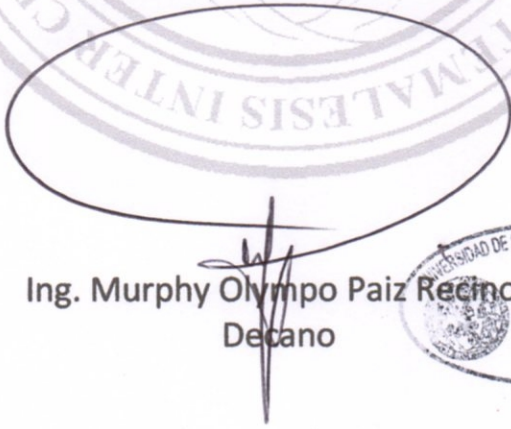
*Ing. Martín Antonio Pérez Türk  
Director, Escuela de Ingeniería en Ciencias y Sistemas*

*Guatemala, 13 de noviembre 2014*



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: **BENEFICIOS DE USAR TECNOLOGÍA MÓVIL PARA LA INDUSTRIA DE DISTRIBUIDORAS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS**, presentado por el estudiante universitario **Erick Rolando Vidal Bazini**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. Murphy Olympo Paiz Recinos  
Decano



Guatemala, 19 de noviembre de 2014

/gdech



## **ACTO QUE DEDICO A:**

- Dios** Por estar en todo momento, y brindarme los éxitos que he logrado cosechar hasta ahora.
- Mis padres** Rolando Vidal y Gisela Bazini de Vidal, por el apoyo brindado a lo largo de estos años, permitiéndome así alcanzar esta meta.
- Mis hermanos** Stefany y Jose Vidal Bazini, por los momentos que compartimos cuando fuimos pequeños, y por crecer a mi lado.
- Mi amiga** Isiris Córdova, por ayudarme y motivarme en este tiempo y brindarme su apoyo incondicional.

## **AGRADECIMIENTOS A:**

<b>Dios</b>	Por darme la sabiduría y entendimiento para poder completar esta meta.
<b>Mi familia</b>	Por su apoyo y esfuerzo para brindarme la oportunidad de finalizar esta etapa de mi vida.
<b>Mi asesor de trabajo de graduación</b>	Ing. Daniel Caciá Rivas, por sus consejos y su apoyo incondicional brindado para finalizar este trabajo.
<b>Mi asesor de EPS</b>	Ing. Heberth Stuardo Campos Hernández, por la paciencia, consejos y enseñanza brindada durante la ejecución del proyecto.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO.....	IX
RESUMEN.....	XI
OBJETIVOS .....	XIII
INTRODUCCIÓN.....	XV
1. ¿QUE SON <i>SMARTPHONE</i> Y APLICACIONES MÓVILES? .....	1
1.1. Historia .....	1
1.2. Fundamentos de móviles.....	2
1.2.1. ¿Qué es un <i>smartphone</i> ? .....	2
1.2.2. Características.....	3
1.3. Evolución de las aplicaciones móviles .....	5
1.4. Canales de movilidad .....	7
1.5. Tipos de aplicaciones móviles .....	8
1.5.1. Nativas.....	8
1.5.1.1. Ventajas .....	9
1.5.1.2. Desventajas .....	10
1.5.2. Web .....	11
1.5.2.1. Ventajas .....	11
1.5.2.2. Desventajas .....	12
1.5.3. Híbridas .....	13
1.5.3.1. Ventajas .....	14
1.5.3.2. Desventajas .....	15
1.6. Principales Sistemas operativos móviles en el mercado actual.....	15

1.6.1.	IOs de Apple.....	16
1.6.2.	Android de Google.....	18
1.6.3.	Windows Phone de Microsoft .....	19
2.	USO Y BENEFICIOS DE APLICACIONES MÓVILES EN PEQUEÑAS Y MEDIANAS EMPRESAS .....	21
2.1.	Análisis de la empresa .....	21
2.1.1.	Pequeña y media empresa.....	21
2.1.2.	Empresas distribuidoras .....	23
2.1.2.1.	Rutas de distribución .....	24
2.1.2.2.	Características del cliente de las empresas distribuidoras.....	25
2.1.2.3.	Características del producto a ser transportado .....	26
2.1.2.4.	Preventa .....	26
2.1.2.5.	Proceso de distribución físico .....	27
2.2.	Tecnología actual y su uso en la empresa .....	29
2.2.1.	Uso del <i>handheld</i> .....	30
2.2.2.	Uso del GPS.....	30
2.2.3.	Uso del POS.....	31
2.3.	¿Por qué mediana y pequeña empresa?.....	31
2.3.1.	Costo operativo .....	32
2.3.2.	Productividad.....	32
2.3.3.	Servicio al cliente.....	32
2.3.4.	Retorno de inversión .....	32
2.3.5.	Pequeña y mediana empresa .....	33
2.4.	Beneficios de utilizar aplicaciones móviles en la pequeña y media empresa distribuidoras .....	34
2.4.1.	Ahorros operacionales.....	34

2.4.2.	Aumento de la productividad.....	35
2.4.3.	Mejora del servicio al cliente .....	35
2.4.4.	Mayor control de los repartidores .....	35
3.	ARQUITECTURA DE ADF MOBILE.....	37
3.1.	Componentes de la arquitectura .....	37
3.1.1.	Patrón de diseño MVC .....	38
3.1.1.1.	Modelo .....	39
3.1.1.2.	Vista .....	40
3.1.1.3.	Controlador .....	41
3.1.1.4.	Java runtime.....	42
3.1.1.5.	Interacción de los componentes .....	42
3.1.2.	Elementos de la arquitectura.....	43
3.1.2.1.	Device-Native Container .....	45
3.1.2.2.	Web View .....	45
3.1.2.3.	Server HTML.....	45
3.1.2.4.	Local HTML.....	46
3.1.2.5.	ADF Controller .....	46
3.1.2.6.	ADF Mobile AMX Views .....	47
3.1.2.7.	Java .....	47
3.1.2.8.	Managed Beans .....	47
3.1.2.9.	ADF Model .....	48
3.1.2.10.	Application Configuration .....	48
3.1.2.11.	Credential Management and Access Control .....	48
3.1.2.12.	Apache Cordova .....	50
3.1.2.13.	Local data .....	51
3.1.2.14.	Configuration Server .....	52
3.1.2.15.	ADF Mobile Browser .....	52



4.	COMPARATIVA DE ADF MOBILE FRENTE A OTROS FRAMEWORKS.....	55
4.1.	Comparativa de aplicaciones híbridas frente a aplicaciones nativas y web .....	55
4.1.1.	Características de aplicaciones nativas .....	56
4.1.2.	Características de aplicaciones web.....	57
4.1.3.	Características de aplicaciones híbridas.....	59
4.1.4.	Cuadro comparativo de aplicaciones móviles .....	61
4.2.	Comparativa de ADF Mobile frente a otros <i>frameworks</i> híbridos .....	61
4.2.1.	Sencha Touch .....	62
4.2.1.1.	Arquitectura de Sencha Touch.....	62
4.2.1.2.	Principales características .....	65
4.2.1.3.	Dispositivos y navegadores soportados.....	66
4.2.1.4.	Licenciamiento.....	67
4.2.1.4.1.	Proyectos Open Source .....	67
4.2.1.4.2.	Proyectos comerciales .....	67
4.2.2.	jQuery Mobile .....	68
4.2.2.1.	Arquitectura de JQuery Mobile.....	68
4.2.2.2.	Principales características .....	70
4.2.2.3.	Dispositivos y Navegadores Soportados .....	71
4.2.2.4.	Licenciamiento.....	72
4.2.3.	SproutCore .....	72
4.2.3.1.	Arquitectura de Sprout Core .....	73
4.2.3.1.1.	MVC+SDR .....	74

4.2.3.2.	Principales características .....	75
4.2.3.2.1.	Model Layer .....	76
4.2.3.2.2.	View Layer .....	76
4.2.3.2.3.	Controller Layer.....	77
4.2.3.2.4.	Respond Layer .....	77
4.2.3.3.	Dispositivos y navegadores soportados .....	78
4.2.3.3.1.	Dispositivos soportados ...	78
4.2.3.3.2.	Navegadores soportados .....	78
4.2.3.4.	Licenciamiento .....	78
4.3.	¿Por qué usar ADF Mobile, frente a otros <i>frameworks</i> ? .....	79
4.3.1.	Beneficios, frente a otros <i>frameworks</i> .....	80
4.3.1.1.	Desarrollo Integrado.....	80
4.3.1.2.	Uso de patrones de diseño.....	81
4.3.1.3.	Se trabaja de forma declarativa.....	81
4.3.1.4.	Seguridad.....	82
4.3.1.5.	Fácil manejo de Java .....	82
4.4.	Cuadro comparativo de <i>frameworks</i> móviles híbridos .....	82
5.	PROPUESTA DE LA APLICACIÓN MOVIL EN EL MERCADO PARA PEQUEÑAS Y MEDIANAS EMPRESAS DISTRIBUIDORAS .....	85
5.1.	Definición del problema .....	85
5.2.	Análisis de la aplicación.....	87
5.2.1.	Requerimientos de la aplicación .....	88
5.2.2.	Casos de uso .....	90
5.2.3.	Módulos propuestos para la solución .....	94
5.2.3.1.	Módulo Mapa .....	94
5.2.3.2.	Módulo ruta crítica.....	95

5.2.3.3.	Módulo inventario .....	95
5.2.3.4.	Módulo puntos y zonas críticas .....	96
5.2.3.5.	Módulo de usuarios .....	96
5.2.4.	Tecnología propuesta para la solución .....	96
5.3.	Diseño de la aplicación .....	98
5.3.1.	Modelo conceptual .....	98
5.3.2.	Definición de la arquitectura de la aplicación .....	99
5.4.	Propuesta de valor .....	100
5.4.1.	Beneficios tecnológicos aportados a la pequeña y mediana empresa distribuidora .....	101
CONCLUSIONES .....		103
RECOMENDACIONES .....		105
BIBLIOGRAFÍA .....		107

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Arquitectura de Sistema Operativo iOS.....	17
2.	Arquitectura de Android Os .....	18
3.	Clasificación de empresas en Guatemala .....	22
4.	Interacción entre empresas distribuidoras y las demás empresas.....	24
5.	Proceso de distribución .....	29
6.	Modelo vista controlador .....	39
7.	Arquitectura de ADF Mobile .....	44
8.	ADF Mobile Definición de Flujos en JDeveloper.....	46
9.	Página de login utilizando Credential Management Access Control .....	49
10.	Interacción PhoneGap.....	51
11.	Conección URL.....	53
12.	Arquitectura de Sencha Touch .....	64
13.	Arquitectura SproutCore.....	75
14.	Modelo conceptual de la aplicación de monitoreo .....	99
15.	Arquitectura de la aplicación de monitoreo.....	100

## TABLAS

I.	Cuadro comparativo aplicaciones nativas, web e híbridas .....	61
II.	Roles principales jQuery Mobile .....	69
III.	Cuadro comparativo de <i>frameworks</i> móviles híbridos .....	83
IV.	Caso de uso–ingresar puntos de preventa .....	90
V.	Caso de uso–mostrar ruta de distribución .....	91

VI.	Caso de uso–ingresar ruta crítica .....	92
VII.	Caso de uso–ingresar/extraer inventario .....	92
VIII.	Caso de uso–ingresar horas pico .....	93
IX.	Caso de uso–ingresar zonas roja .....	93
X.	Caso de uso–consultar panel de control .....	94



## GLOSARIO

<b>API</b>	Interfaz de programación de aplicaciones, básicamente es un conjunto de métodos y funciones para poder comunicarse con diferentes partes de un software en específico.
<b>APP</b>	Aplicación móvil.
<b>APP STORE</b>	Tienda en línea donde se pone a disposición del usuario cientos de aplicaciones las cuales pueden tener o no un costo asociado para su descarga.
<b>CSS</b>	Hoja de estilo en cascada, es una hoja de estilos en la cual se centraliza los diseños de una aplicación web.
<b>GPS</b>	Sistema de posicionamiento global, el cual por medio de la latitud, longitud y altitud, puede dar una ubicación precisa.
<b>GUI</b>	Interfaz de usuario en una aplicación de software, permite que el usuario interactúe con la aplicación.
<b>HTML</b>	Lenguaje de programación que contiene marcas de hiper texto, y permite la creación de páginas para ambiente web.

<b>JSON</b>	Es una estructura para el intercambio de datos, en el se escriben objetos tipo JavaScript, puede ser utilizado como alternativa del lenguaje XML por su simplicidad.
<b>Librerías</b>	Conjunto de métodos y funciones agrupados para el uso en otros programas de software.
<b><i>Plugins</i></b>	Es un complemento de una aplicación, es una aplicación la cual se incorpora al programa central para adicionar alguna función específica.
<b><i>Widget</i></b>	Es una pequeña aplicación, usualmente con una interfaz gráfica la cual permite fácil acceso a determinadas funciones de la aplicación o software central.

## RESUMEN

En el presente trabajo se brinda una solución alternativa a la problemática del monitoreo de vehículos repartidores para pequeñas y medianas empresas repartidoras, se propone la utilización de la tecnología móvil por medio de los *smartphone* para el monitoreo de los repartidores.

Existen diferentes tipos de empresas: las grandes tienen procesos estandarizados como el GPS, *handheld* y el apoyo del punto de ventas para realizar el monitoreo. Por otro lado, se tienen las pequeñas y medianas empresas de distribución que en la actualidad aún no cuentan con este tipo de tecnología debido a que la implementación es alta.

Se presenta una comparativa entre las características de las aplicaciones nativas, web e híbridas. Así como también las características y arquitectura de cuatro diferentes *frameworks* para desarrollo híbrido los cuales son: Sencha Touch, JQuery Mobile, SproutCore y ADF Mobile.

La tecnología propuesta es ADF Mobile debido a los beneficios tecnológicos que este marco de trabajo brinda frente a los demás *frameworks*, comparados en este trabajo realizando un análisis y diseño de la posible solución a dicho problema de monitoreo.



## **OBJETIVOS**

### **General**

El propósito de la investigación es dar a conocer los beneficios de la implementación de la tecnología móvil, como una alternativa de herramienta de monitoreo orientado a las pequeñas y medianas empresas distribuidoras. Asimismo, dar a conocer las diferentes soluciones de desarrollo y los beneficios tecnológicos que una plataforma específica puede traer a estas empresas.

### **Específicos**

1. Proveer una alternativa a las pequeñas y medianas empresas mediante la implementación de aplicaciones móviles en los teléfonos inteligentes, para el proceso de monitoreo y de esta forma lograr una reducción de costos frente a otras posibles soluciones.
2. Presentar a las aplicaciones híbridas como una solución para el desarrollo de aplicaciones móviles.
3. Promover la implementación de aplicaciones móviles para optimizar los tiempos de ruta de distribución en pequeñas y medianas empresas.
4. Presentar una solución a las pequeñas y medianas empresas distribuidoras por medio del cual puedan tener un mejor control en sus procesos de distribución.





## INTRODUCCIÓN

Las empresas que están en la industria distribuidora necesitan tener un monitoreo constante de sus repartidores, los cuales tienen que cumplir una ruta establecida. La tecnología moderna permite que por medio del GPS se pueda monitorear al piloto, y llevar un control de donde se encuentra en todo momento.

La mayoría de empresas utilizan tecnología de costos elevados para realizar este monitoreo, el cual puede resultar inaccesible para una pequeña o mediana empresa que se encuentra iniciando. Por esta razón se propone el uso de la tecnología móvil, en la cual a través del uso de un *smartphone* se puede llevar a cabo el proceso de monitoreo de ruta, reduciendo los costos a un nivel accesible.

La tecnología actual presenta una gama de teléfonos inteligentes con diversas cualidades, los cuales pueden adquirirse adaptándose a las necesidades del usuario, asimismo, existen diferentes propuestas de desarrollo para la implementación de aplicaciones en un dispositivo móvil, en el presente trabajo se habla de tres ramas principales para el desarrollo de estas aplicaciones las cuales se describen a continuación:

Aplicaciones nativas: estas son desarrolladas con el JDK provisto por cada plataforma, este es completamente dependiente del dispositivo donde se está desarrollando.

Aplicaciones *web*: únicamente es código HTML combinado con CSS y JavaScript, no se tiene acceso a las funciones nativas del dispositivo debido a que estas aplicaciones corren en un *web view*.

Aplicaciones híbridas: esta es una combinación de las dos anteriores, en donde se encuentra una parte web (HTML, CSS y JavaScript) con una parte nativa, esta última parte mapea los elementos nativos del teléfono como lo son GPS, cámara, acelerómetro, y lo envía a la parte web esto permite tener acceso a las características propias del dispositivo en un ambiente web.

El coste de desarrollo va disminuyendo según el tipo de aplicación, el coste más elevado lo posee la aplicación nativa, posteriormente la aplicación híbrida y la de menor costo es la alternativa web.

En esta investigación se presenta la alternativa híbrida, como una posible solución para la reducción de costos de implementación para las pequeñas y medianas empresas interesadas en el monitoreo de rutas de distribución.

Este trabajo muestra también las diferentes plataformas que se pueden utilizar para desarrollar aplicaciones híbridas y posee una propuesta para el desarrollo de una aplicación, la cual puede ser implementada por cualquier pequeña y mediana empresa distribuidora, que desee realizar actividades de monitoreo para cada uno de sus procesos de distribución a través de un *smartphone* y de esta forma, tener un mayor control de rutas y distribución de productos para lograr optimizar y mejorar las mismas, con una reducción de costos significativa frente a la implementación de otras posibles soluciones.

# 1. ¿QUE SON *SMARTPHONE* Y APLICACIONES MÓVILES?

## 1.1. Historia

La industria de la tecnología en su primer intento por agregarle al teléfono móvil “algo más”, surgió en 1993, cuando IBM lanzó su primer teléfono el cual denominó “Simon” fue comercializado por BellSouth, en este se incorporaron diversos servicios, entre los cuales se contaba con voz y datos, entonces el equipo además de funcionar como un teléfono móvil, era un asistente digital personal (debido a que tenía calendario, libreta de direcciones, reloj mundial, calculadora, bloc de notas, correo electrónico y juegos), e incluso como una máquina de fax.

Este teléfono móvil contaba con una pantalla táctil para la marcación de números telefónicos y el texto se ingresaba mediando un pequeño teclado *qwerty*, este artefacto tuvo un precio en aquella época de 900 dólares.

El primer teléfono que se llamó realmente *smartphone* fue desarrollado por la empresa Ericsson, en 1997, ellos le dieron el nombre de “Ericsson GS88” apodado “Pamela”, este dispositivo disponía del sistema operativo Geos de GeoWorks el cual tenía una arquitectura de 16 bit. Entre los servicios que disponía se encontraba el correo electrónico POP3, mensajes de texto, reloj mundial, navegador entre otros, tenía la funcionalidad de “Modo Vuelo” en donde se desactivaban todas las funciones inalámbricas. Tenía manos libres integrado, módem integrado, puerto de infrarrojos, conexión al pc por medio de RS232 (conocido popularmente como cable serie) y teclado *qwerty* físico.

Uno de los cambios radicales hacia la tecnología de los *smartphone* fue el lanzamiento de iPhone de Apple en el 2007, este vendió millones de unidades, en parte gracias a su pantalla táctil, y a la experiencia que este les daba a sus usuarios en el acceso a internet.

Ese mismo año, Google presento su sistema operativo Android, no fue tan explosivo como el de Apple, pero fue introduciendo cambios, como búsquedas por voz, aplicaciones en Android Market, cambios estéticos, hasta llegar a la versión actual.

Luego del *smartphone* en el 2010 se lanza al mercado un nuevo dispositivo portátil el cual vendría a revolucionar los dispositivos móviles, estos fueron las *tablets* por las cuales se tenía una pantalla más grande, y siempre con las funcionalidades de los teléfonos *smartphone* debido a que traían el sistema operativo similar.

## **1.2. Fundamentos de móviles**

A continuación se presentan los fundamentos de móviles.

### **1.2.1. ¿Qué es un *smartphone*?**

Un teléfono inteligente o *smartphone* por su nombre en inglés, es un dispositivo que posee más características que un teléfono móvil convencional, es una computadora de bolsillo, el cual realiza actividades semejantes a una computadora de escritorio. El término “inteligente”, viene dado para su comercialización debido a que en la actualidad, ninguna computadora o microprocesador puede pensar o razonar únicamente sigue las instrucciones que el ser humano le da a través de la programación.

Estos teléfonos poseen un sistema operativo que se asemeja al de las computadoras, permitiendo instalar aplicaciones de terceros con el fin de que el teléfono se adapte más a las necesidades de los usuarios, a diferencia de los teléfonos móviles convencionales en los que el usuario debía adaptarse al dispositivo.

El *smartphone* generalmente posee un teclado frontal en forma *qwerty*, así como también una pantalla táctil, cabe mencionar que ninguno de estas funcionalidades es obligatoria, para que un dispositivo sea clasificado como un teléfono inteligente.

En la actualidad el *smartphone* se ha convertido en un accesorio empresarial, así como también para algunas personas un objeto indispensable para continuar sus labores en la oficina, debido a que en él se encuentra una lista de tareas a realizar en el día, los clientes a los que se debe llamar para ofrecer algún servicio; esto hace que las personas no solo utilicen el móvil en su vida privada sino en su entorno profesional.

### **1.2.2. Características**

Estas son algunas de las características que tienen los dispositivos *smartphone*, cabe mencionar que en la actualidad estas características son muchas más que las que se presentan a continuación y además cambian dependiendo del dispositivo.

- Acceso a internet: lo cual permite tener acceso a las redes sociales, así como al correo electrónico, esto también permite acceso a servicios en la nube como la sincronización de archivos a través de DropBox o Google Drive, entre otras cosas.

- Correo electrónico: permite estar al pendiente de cualquier mensaje que se reciba durante el día, por lo general los *smartphones* permiten sincronizarse con el correo electrónico del usuario, así como también se pueden vincular varias direcciones de correo electrónico por ejemplo la dirección personal, del trabajo, etc.
  
- Pantalla táctil: esta es una función la cual permite controlar todas las funciones del dispositivo con solo ubicar la mano en la pantalla. Existen dispositivos como los iPhone en los cuales interpretan gestos de la mano, lo cual permite manipular con los cinco dedos de la mano diferentes característica que posee una aplicación.
  
- Servicios de oficina
  - Contactos: permite almacenar las diferentes formas de contacto de las personas asociadas a la agenda de contactos ya sea de la oficina y personales.
  
  - Agenda personal: la cual lleva el control de las tareas, actividades, citas importantes, en las cuales se debe activar alguna alerta.
  
  - Sincronización: al conectar el *smartphone* con la computadora o al estar conectado a internet, permite la sincronización de contactos, tareas, calendario, para que se tenga informa actualizada entre todos los dispositivos, de manera que todo se encuentre al día.
  
- GPS: sistema de posicionamiento global, el cual por medio de la latitud, longitud y altitud, puede dar una ubicación precisa.

- Reconocimiento de voz: en las últimas versiones de estos dispositivos se permite darle órdenes por medio de la voz humana, como por ejemplo si se desea tomar una foto o poder llamar a una persona en específico.

### **1.3. Evolución de las aplicaciones móviles**

Las aplicaciones móviles han ido cambiando a lo largo del tiempo, las primeras aplicaciones surgieron en los 90s con los primeros teléfonos inteligentes, estas aplicaciones eran básicamente libreas de contacto, agendas, juegos, etc.

Todas estas aplicaciones eran desarrolladas únicamente por las empresas dueñas del dispositivo, no permitían que terceros programaran sus propias aplicaciones. Con la llegada del WAP (*Wireless Application Protocol* por sus siglas en inglés), se pudo crear un único navegador en el cual por medio de los celulares se podía acceder a internet, por supuesto no como en un ordenador, debido a que en la mayoría de los dispositivos móviles de esa época, eran pantallas monocromáticas.

En el 2007 con la llegada del iPhone de Apple, puso a disposición inicialmente 500 aplicaciones en su tienda de aplicaciones (App Store). Lo novedoso que introdujo Apple, es que en lugar de ellos desarrollar todas las aplicaciones pusieron a disposición un API, en la cual tercero podían desarrollar aplicaciones y subirlas a la tienda.

A finales del 2008 con el lanzamiento del SO Android de Google nace con tan solo 50 aplicaciones en su tienda (Android Market), de igual forma permitiendo a terceros poder crear aplicaciones a través de su API.



Con el lanzamiento del SO Android se marca un punto muy importante en la historia de la tecnología móvil debido a que este sistema es un SO gratuito, con lo cual muchas empresas pueden adaptarlo fácilmente a su dispositivo y hacer uso del mismo sin agregar un costo adicional.

Durante el 2010 comienza una nueva era, la y de las *tablets*, Apple lanza en abril el primer IPAD con el sistema operativo IOS, este artefacto con una pantalla táctil de 9,2 pulgadas, incluía diversas funciones como lo era reproducción de música, GPS, redes sociales, así como el acceso a las diferentes aplicaciones que existían en el App Store. A finales de este mismo año, Samsung lanza la primera *tablet* con el sistema operativo Android en estados unidos esta *tablet* poseía una pantalla de 7 pulgadas, las aplicaciones para estos dispositivos vino siendo similar a la de los *smartphone* pero cambio la resolución de pantalla de las mismas, ahora se debía adaptar a una pantalla más grande que la de un *smartphone* convencional.

En la actualidad se tienen diferentes dispositivos, así como también diferentes sistemas operativos, anteriormente el desarrollo de aplicaciones se hacía de forma nativa esto quiere decir se desarrollaba para una plataforma en específico.

Con la llegada de HTML y los navegadores a los dispositivos se comenzó el desarrollo para las diferentes plataformas, pero aunque se creía que se tenía un desarrollo uniforme este tipo de aplicaciones tenía muchas limitantes, debido a que en los primeros prototipos se debía scrollear mucho la pantalla para poder visualizar el contenido. Para solventar este problema las nuevas aplicaciones contenían una GUI más pequeña, lo cual redujo demasiado la visibilidad.

Las aplicaciones Web debido a estos problemas y al hecho que no se podía tener el uso de los componentes nativos del dispositivo (cámara, libreta de contactos, GPS, etc.) no cubrieron las expectativas.

Luego de tener estos dos conceptos aplicaciones nativas, y aplicaciones web, se creó una mezcla de ambas, y con esto nació las aplicaciones híbridas, en las están diseñadas básicamente de HTML, CSS y JavaScript pero poseen un componente adicional en el permite el uso de los componentes nativos del dispositivo como lo son cámara, libreta de contactos, GPS, etc.

#### **1.4. Canales de movilidad**

Existen diferentes canales de movilidad, los cuales se tornan diferentes por medio del dispositivo, los canales móviles más conocidos o con un mayor alcance que se encuentran actualmente en el mercado son *smartphones* y *tablets*.

Así también existen nuevos canales emergentes, tales como los relojes inteligentes y las *smartTV*; estos nuevos canales permiten a su vez descargar aplicaciones, así como también permiten realizar búsquedas y navegación, acceso a redes sociales además de la convergencia con *smartphones* y *tablets*, por ejemplo controlar la televisión a través de una *tablet*, o poder realizar una llamada a través del reloj el cual se conecta con el *smartphone*, etc.

Para lograr esta sincronización se deben tener aplicaciones las cuales se adaptan a estos dispositivos, con el fin de lograr la convergencia de todos los dispositivos, y poder dar una experiencia al usuario grata y des estresada, así como hacer más eficiente su tiempo.

## **1.5. Tipos de aplicaciones móviles**

Como se mencionaba anteriormente las aplicaciones móviles se dividen esencialmente en tres tipos las cuales son aplicaciones nativas, web e híbridas. Al comienzo las aplicaciones móviles eran únicamente nativas, posteriormente surgió la tecnología web, y con lo que se llega a la actualidad que es la tecnología híbrida.

### **1.5.1. Nativas**

Este desarrollo se realiza de forma específica para determinado sistema operativo o dispositivo. Como se depende del sistema operativo, se debe programar en diferentes lenguajes, dependiendo de la plataforma donde se está desarrollando, estas aplicaciones corren de forma más eficiente debido a que su código está optimizado, específicamente para la plataforma donde serán instaladas.

Estas aplicaciones permiten hacer uso de las diferentes características que están instaladas en los *smartphone* (cámara, acelerómetro, GPS, entre otras), así como también hacer uso de otras aplicaciones instaladas en el dispositivo sin tener que hacer una configuración exhaustiva debido a que la API proporcionada por el proveedor de la plataforma tiene un conjunto de librerías de software para poder realizar esta tarea.

Para desarrollar sobre una aplicación nativa se debe de considerar que cada sistema operativo tiene diferentes lenguaje de programación para ser desarrollado, por ejemplo para el SO Android se utiliza el lenguaje de programación JAVA, para el SO IOS se utiliza el lenguaje de programación Objective C, para el SO Windows Phone se utiliza el lenguaje C Sharp ( C#), y

así cada plataforma tiene su propio lenguaje a utilizar, con lo cual se tiene un costo adicional al intentar desarrollar multiplataforma.

#### **1.5.1.1. Ventajas**

- Ejecución rápida sobre el dispositivo: varias de las aplicaciones nativas traen elementos precargados, así como también cada aplicación esta optimizada para correr en la plataforma específica, con lo cual da una experiencia al usuario de mayor velocidad.
- Acceso a las características nativas: una fuerte ventaja de las aplicaciones nativas es el uso de las características nativas de cada dispositivo, por ejemplo la aplicación Instagram la cual hace uso de la cámara del dispositivo, así como también de las diferentes aplicaciones de redes sociales instaladas en el dispositivo para poder vincular las cuentas entre estas redes sociales.
- Se puede utilizar en línea o fuera de línea: al ser una aplicación nativa no es necesario que se tenga una conexión permanente a una red de datos. Si no que se puede utilizar de forma fuera de línea.
- Almacenamiento local de información: como el dispositivo posee un espacio de almacenamiento ya sea interno o externo se tiene acceso a esta característica de forma directa.
- Tiendas de aplicaciones: el usuario puede encontrar fácilmente la aplicación debido a que cada plataforma comercializa las aplicaciones a través de una tienda (Google Play para Android, App Store para iOS, etc.)

### 1.5.1.2. Desventajas

- No soporta multiplataforma: es necesario desarrollar específicamente para cada plataforma ya que se debe utilizar el *framework* de cada una de ellas, con lo cual se tiene diferentes lenguajes de programación por plataforma.
- Costo desarrollo elevado: debido a que se debe desarrollar en diferentes lenguajes de programación para cada plataforma es necesario contratar a diferentes programadores que desarrollen la aplicación.
- Desarrollo no homogéneo: cómo se desarrolla una aplicación por cada plataforma es probable que mismas aplicaciones construidas en diferentes plataformas posean características diferentes.
- Aprendizaje de desarrollo alto: debido a que se debe desarrollar en lenguajes específicos de cada plataforma, la curva de aprendizaje es mayor en comparación de aprender a desarrollar en una sola herramienta, por ejemplo para Android se necesita el aprendizaje del lenguaje Java.
- Proceso de aprobación en las tiendas de aplicaciones: cuando se desea publicar una aplicación en las diferentes tiendas como lo son Google Play y App Store, se debe pasar por un proceso de aprobación, y pagar un costo adicional por la estadía de esta app, en la tienda.
- Actualización manual: este proceso debe realizarse manualmente por el usuario se puede notificar de una actualización disponible pero siempre se deberá descargar los instaladores al dispositivo del usuario.

## **1.5.2. Web**

Estas aplicaciones son páginas web, las cuales corren en un navegador web, en el caso de los móviles sobre un navegador web móvil. Su ventaja principal es la portabilidad que poseen dichas aplicaciones, debido a que son independientes de la plataforma por ejemplo, se puede visualizar el contenido del sitio web en un dispositivo con sistema operativo Android, así como también un sistema iPhone o Black Berry, sin que exista mayor cambio en el código principal de la aplicación.

Estas aplicaciones están compuestas básicamente por HTML y CSS, una de las principales desventajas es que estas aplicaciones tienen limitante del uso de los componentes internos del *smartphone*. Por ejemplo, al querer hacer uso de la agenda para sincronizar alguna actividad, esta tarea se torna imposible en este tipo de aplicaciones.

### **1.5.2.1. Ventajas**

- Portabilidad: permite desarrollar una sola vez para todos los dispositivos debido que para poder visualizar la aplicación únicamente se debe hacer uso del navegador web.
- Actualización automática: ya que estas aplicaciones residen en un servidor es fácil poder lanzar una nueva versión sin que el usuario tenga que intervenir o se deba bajar el instalador al dispositivo, las actualizaciones son transparentes al usuario, esto quiere decir que él no percibe cuando se realiza.

- Costo de desarrollo bajo: debido a que se desarrolla una sola vez para las diferentes plataformas del mercado, y se hace sobre un lenguaje que es común y sencillo de aprender (HTML y CSS), el costo de desarrollo es bajo pudiendo abarcar con una sola aplicación todo el mercado.
- Fácil publicación: estas aplicaciones no pasan por el proceso de aprobación en ninguna tienda, con lo cual luego de ser desarrolladas y probadas pueden salir al mercado inmediatamente.

#### **1.5.2.2. Desventajas**

- No se tiene acceso a características nativas: estas aplicaciones corren sobre un navegador web, el cual no tiene acceso a las diferentes características que tiene un *smartphone* como por ejemplo: cámara, acelerómetro, libreta de contactos.
- No se puede acceder *offline*: se necesita una conexión de internet para poder visualizar el contenido de la aplicación.
- No permiten guardar datos localmente: estas aplicaciones solo se puede guardar información en el servidor donde residen, como por ejemplo información de sesión, nombre del usuario, correo electrónico, etc. Con lo cual se tiene una cantidad de espacio limitada a la cuota que ofrezca el servidor.
- Velocidad dependiente de la conexión: la velocidad de la aplicación es dependiente de la conexión de internet que se disponga en el plan de datos. En ocasiones puede ser rápida, así como también en otras puede

ser lenta y esto conlleva a una mala experiencia al usuario ya que jamás cargara la información que él está requiriendo.

### **1.5.3. Híbridas**

Son la mezcla de las dos anteriores web e híbrido, con la llegada de las nuevas tecnologías web como lo son HTML5 y CSS3, viene este nuevo tipo de aplicaciones al mercado, la cual tiene lo mejor de los dos mundos.

Estas aplicaciones pueden ser instaladas como una aplicación nativa esto quiere decir que cuenta con un código fuente y puede ser descargada desde los diferentes tiendas de aplicaciones para cada plataforma, así como también posee estándares de la tecnología web como lo son css3 y java script, para proveer más funcionalidades.

Además este tipo de aplicaciones tiene un componente adicional el cual es llamado contenedor nativo o *native container*, este contenedor básicamente mapea las características nativas del dispositivo y ponerlas a disposición de la siguiente capa de la aplicación que sería navegador o *Browser*.

Las principales características de este tipo de aplicaciones son las siguientes:

- Optimizadas para móviles y por lo mismo se tiene una aplicación optimizada para el dispositivo.
- Posibilidad de poder acceder de forma local al almacenamiento del dispositivo móvil, así como también a los servicios nativos del dispositivo, como lo son GPS, cámara, Libreta de contactos.



- La posibilidad de reusar código, debido a que se dispone de diferentes librerías que ya fueron desarrolladas anteriormente, así como la creación de nuevas librerías, para ponerlas a disposición del negocio.

### **1.5.3.1. Ventajas**

- **Multiplataforma:** estas aplicaciones pueden ser instaladas en las diferentes plataformas que se encuentran en el mercado. Esto quiere decir que la misma aplicación puede ser utilizada en diferentes sistemas operativos móviles.
- **Bajo costo de desarrollo:** se desarrolla una vez para todas las plataformas con las cuales tiene compatibilidad el *framework* híbrido, y además se utiliza un lenguaje que es sencillo de aprender (HTML5 y CSS3), el costo de desarrollo es relativamente menor comparado con el desarrollo nativo.
- **Acceso a las características nativas:** este tipo de aplicaciones poseen el contenedor nativo, con el cual permite poder mapear las características como cámara, libreta de contactos, etc.
- **Se puede utilizar en línea o fuera de línea:** al ser una aplicación diseñada con HTML5 no es necesario que se tenga una conexión constante a internet. Permitiendo trabajar.
- **Almacenamiento local de información:** como el dispositivo posee un espacio de almacenamiento ya sea interno o externo se tiene

- Tiendas de aplicaciones: el usuario puede encontrar fácilmente la aplicación debido a que cada plataforma comercializa las aplicaciones a través de una tienda (Google Play para Android, App Store para iOS, etc.)

#### **1.5.3.2. Desventajas**

- Proceso de aprobación en Apps Store: si se desea publicar la aplicación en las diferentes tiendas como lo son Google Play (para la plataforma Android) y Apple App Store (para la plataforma iOS) , se debe pasar por el proceso de aprobación, así como pagar un costo agregado para la estadía de la aplicación en la tienda.
- Disminución en el rendimiento: debido a que una parte de la aplicación es renderizada por el *native container*, la parte grafica puede tener un poco de menos nitidez.
- Dependencia de un *framework*: debido a que desde el diseño de la aplicación se debe elegir en que *framework* se desarrollara la aplicación, es difícil poder migrar de *framework* una vez se ha comenzado el desarrollo, la dependencia del *framework* es fuertemente ligado a la aplicación.

#### **1.6. Principales Sistemas operativos móviles en el mercado actual**

A continuación se presentan los principales sistemas operativos móviles en el mercado actual.

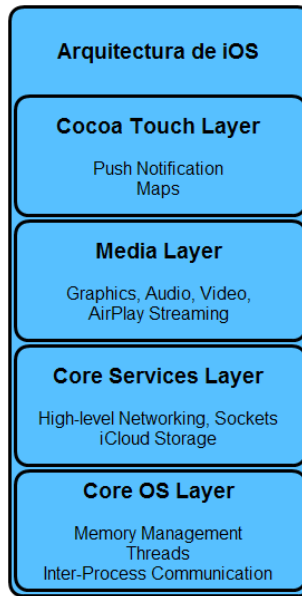
### 1.6.1. IOs de Apple

Este es el sistema operativo para móviles de la compañía Apple Inc. Está instalado en todos los dispositivos de esta empresa desde Smartphones hasta *tablets*. Fue diseñado para dispositivos touch, y está basado en XNU Kernel de MAC OS X.

Su primer lanzamiento fue en el 2007, fue el primer SO que soportaba pantalla multi touch, siendo esto una de sus más principales características que lo distinguen en el mercado.

En su arquitectura se distinguen 4 capas principales las cuales son: Core OS Layer, Core Services Layer, Media Layer y Cocoa Touch Layer. Cada capa se asocia con distintos *frameworks*. A continuación se muestra una representación gráfica de las capas, y los componentes principales de cada una.

Figura 1. **Arquitectura de Sistema Operativo iOS**

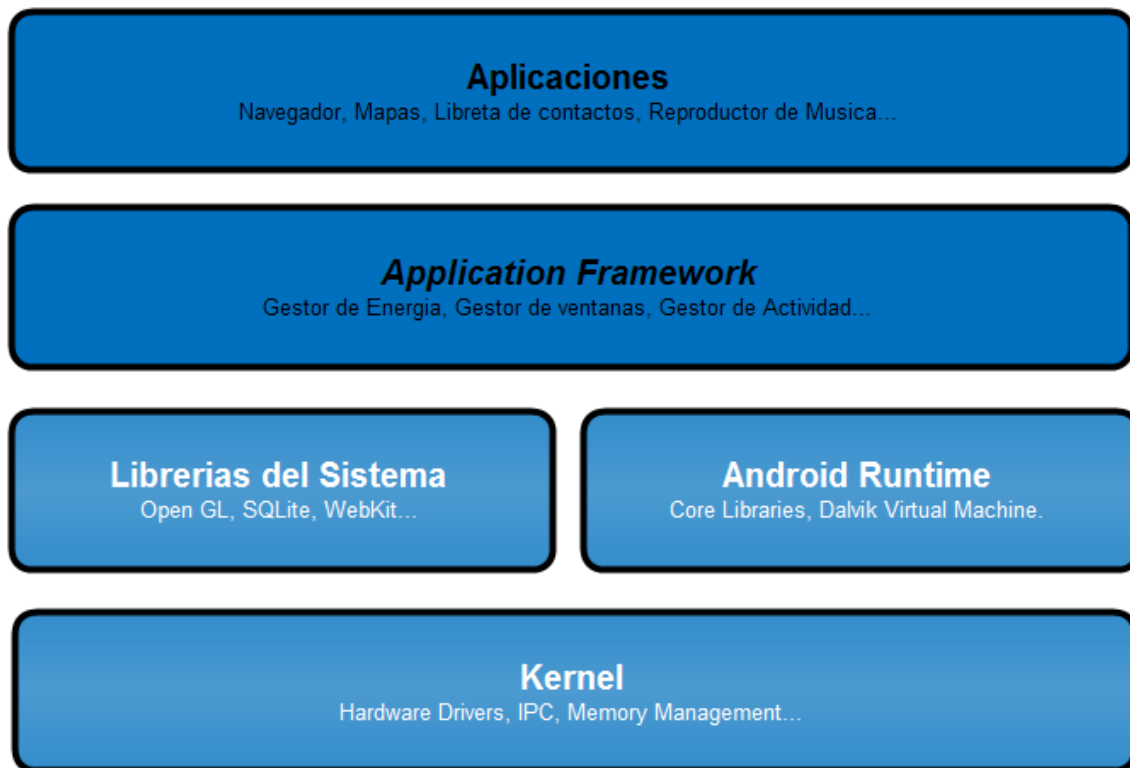


Fuente: elaboración propia.

### 1.6.2. Android de Google

Android es una plataforma móvil libre que actualmente se encuentra siendo desarrollada por la empresa Google, está bajo la licencia GNU (General Public License) y Apache 2,0. Esta plataforma está basada en el *kernel* de Linux. Actualmente es la plataforma con mayor demanda en el mundo. A continuación se presenta una serie de componentes por los cuales está compuesta la arquitectura de Android.

Figura 2. **Arquitectura de Android Os**



Fuente: elaboración propia.

### **1.6.3. Windows Phone de Microsoft**

Windows Phone es la plataforma móvil que Microsoft pone al mercado, inicialmente comenzó Windows Mobile, en el 2007 con la llegada de iOS y Android OS, Microsoft da un giro radical a este sistema operativo móvil para poder competir contra estos dos grandes OS móviles. En el 2010 se lanza Windows Phone, con varias diferencias sobre el antiguo Windows Mobile, una de ellas fue que Windows Phone fue lanzado hacia el público en general, en lugar de solo ser dirigido para el mercado empresarial.

Es importante mencionar que otro aspecto importante es que con Windows Phone Microsoft introduce Metro UI, que es un lenguaje de diseño, tiene como fin que sean señales visuales.



## **2. USO Y BENEFICIOS DE APLICACIONES MÓVILES EN PEQUEÑAS Y MEDIANAS EMPRESAS**

Para poder presentar el uso y beneficios de aplicaciones móviles en pequeñas y medianas empresas es necesario definir la palabra uso y beneficio según el diccionario de la RAE, la palabra “usar” significa: “Hacer servir una cosa para algo” esto quiere decir utilizar o hacer uso constante o acostumbrado de algo para obtener un bien. Así como define la palabra beneficio como: “Bien que se hace o se recibe” con lo cual se tiene que es obtener un bien o producto del cual se pueda sacar provecho.

A continuación se presenta el análisis de la empresa así como también la definición de las empresas distribuidoras y el flujo de operación de las mismas.

### **2.1. Análisis de la empresa**

A continuación se presenta el análisis de la empresa.

#### **2.1.1. Pequeña y media empresa**

Para poder definir pequeña y mediana empresa antes se define como empresa a la entidad integrada por el capital y el trabajo, como factores de la producción y dedicada a actividades industriales mercantiles o de prestación de servicios con fines lucrativos y en beneficio de la sociedad.

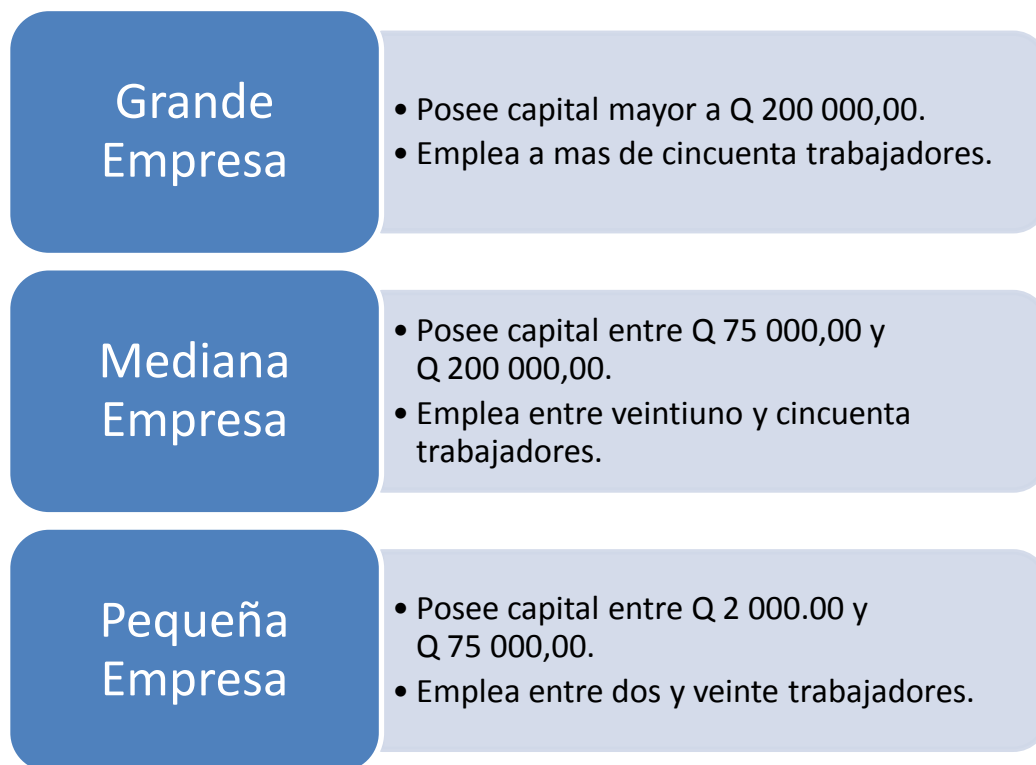
La pequeña y media empresa básicamente es una entidad independiente que a pesar de tener gran influencia en el mercado comercial, quedan excluidas



del mercado industrial debido a que para poder entrar a este mercado se necesita realizar una inversión más fuerte.

En Guatemala existen diferentes definiciones de como clasificar este tipo de empresas. En la siguiente figura se muestra como clasifica el Instituto Nacional de Estadística a las empresas en el territorio guatemalteco.

Figura 3. **Clasificación de empresas en Guatemala**



Fuente: elaboración propia.

En Guatemala la Cámara de la Industria tiene dos diferentes definiciones para las pequeñas y medianas empresas, la primera está basada en el Programa de Bonos, la cual sigue un criterio de número de empleados y la otra

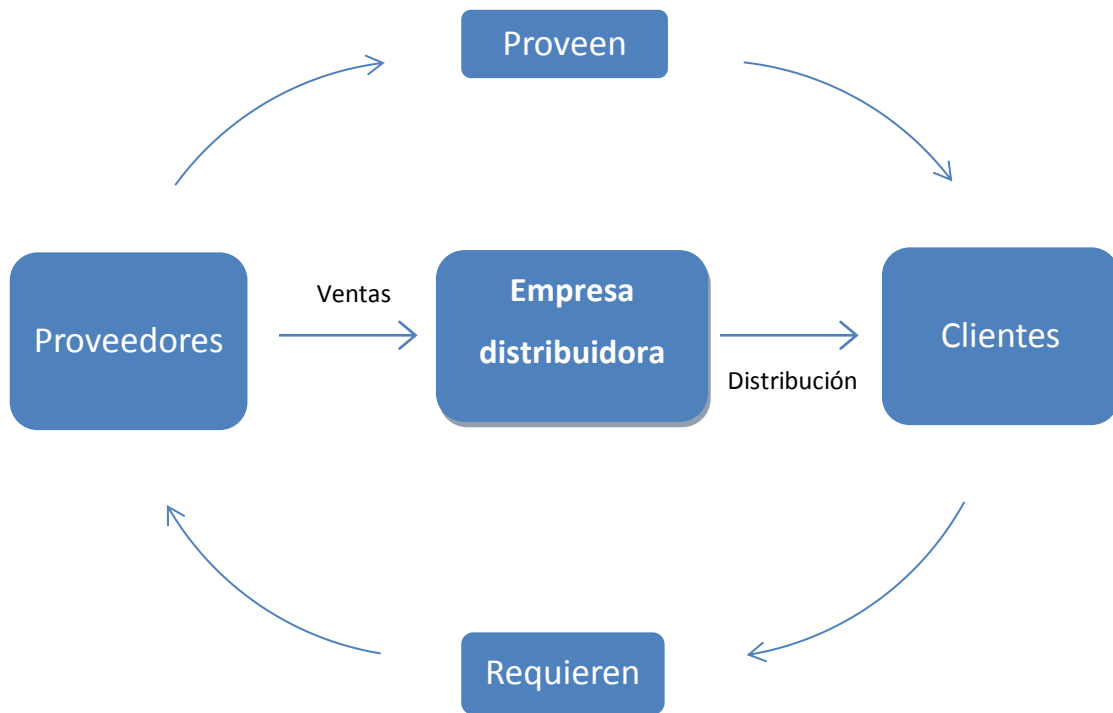
definición está basada a definir empresas industriales, utilizando el criterio de cantidad de empleados, ventas anuales y los activos totales. El Ministerio de Economía tiene su propia definición, utilizando como criterio la cantidad de empleados que posee la empresa.

### **2.1.2. Empresas distribuidoras**

La empresa distribuidora básicamente es la encargada de entregar un producto o un servicio al consumidor o al usuario de negocio, esta empresa puede ser intermediaria entre los proveedores y el consumidor o punto de venta.

A continuación se presenta el diagrama de la interacción entre las empresas distribuidoras con las otras entidades, las cuales son el cliente, y el proveedor.

Figura 4. **Interacción entre empresas distribuidoras y las demás empresas**



Fuente: elaboración propia.

### 2.1.2.1. **Rutas de distribución**

Para las empresas que transportan el producto de la línea de ensamblaje o el servicio hacia el consumidor o punto de venta, es necesario especificar un origen y un destino, en este caso el origen está donde es fabricado el producto o servicio, y el destino está en la ubicación del consumidor o punto de venta, cabe mencionar que como destino también puede ser el almacenamiento del producto, por lo cual un destino válido también puede ser la bodega.

Esta se construye básicamente por el análisis de mapas o gráficas del territorio que se está investigando. Inicialmente se debe establecer un origen y un destino, para posteriormente poder realizar el respectivo análisis de tiempos y la viabilidad de la ruta.

#### **2.1.2.2. Características del cliente de las empresas distribuidoras**

El cliente es uno de los factores a considerar por dichas empresas debido a que ellos son las entidades que recibirán el producto o servicio. Del cliente importan varias características los cuales se presentan a continuación:

- **Ubicación geográfica:** la ubicación es un punto importante debido a que es en donde se encuentra localizado el cliente, en qué departamento, municipio y/o aldea. Esta ubicación debe tener las coordenadas de latitud y longitud.
- **Modo de enlace geográfico:** para poder llegar a la ubicación establecida por medio de las coordenadas de GPS se debe considerar ciertos aspectos geográficos como lo son: carreras, ríos, calles, etc.
- **Características comerciales:** las características comerciales básicamente definen el tipo de cliente al que se le llevara el servicio. Ya sea fabricante, comercializador, etc.

### **2.1.2.3. Características del producto a ser transportado**

El producto es otro factor que debe ser considerado debido a las diferentes características que este posee, ya que no todos los productos llevan el mismo proceso de transporte, a continuación se presentan las características más relevantes a considerar sobre un producto:

- Tipo de producto: existen diferentes tipos de producto dependiendo del giro del negocio, por ejemplo si el producto a transportar es perecedero o no perecedero, ya sea que el producto sea de naturaleza agrícola, industrial, tecnológica, etc.
- Aspectos comerciales: los aspectos comerciales del producto se refiere a las condiciones del mercado donde se distribuirá y como se realizará esta acción.
- Características físicas: cada producto tiene un conjunto de aspectos y atributos propios del mismo, como por ejemplo: peso, dimensiones, forma, naturaleza biológica, entre otros aspectos.

### **2.1.2.4. Preventa**

La preventa es el proceso por el cual se puede visualizar el terreno de ventas a posibles clientes, la preventa tiene como finalidad poder visitar a los clientes para ofrecerles nuevos productos, así como también poder adquirir nuevos clientes, con la preventa se tiene un prevendedor el cual es el encargado de visitar a los clientes que le fueron asignados de acuerdo a su sector.

La preventa suele utilizarse para la creación o adaptación de la ruta de distribución hacia los nuevos clientes, esta preventa permite trazar una ruta de distribución preliminar para posteriormente ser analizada por el Departamento de Logística, permitiendo tener un mayor control en el territorio asignado a cada ruta.

La preventa se realiza regularmente en vehículos livianos, el prevendedor ocupa su tiempo vendiendo únicamente, está enfocado en la atención al cliente, por lo cual lleva únicamente muestras del producto a vender o fotografías del mismo.

Cabe mencionar que antes de implementar la preventa en empresas distribuidoras se debe realizar un análisis costo/beneficio, debido a que no aplica a todos los tipos de mercado, ya que este proceso depende mucho del territorio donde se estará repartiendo el producto, el tipo de vehículo que se utilizará para repartir entre otras variables a considerar.

#### **2.1.2.5. Proceso de distribución físico**

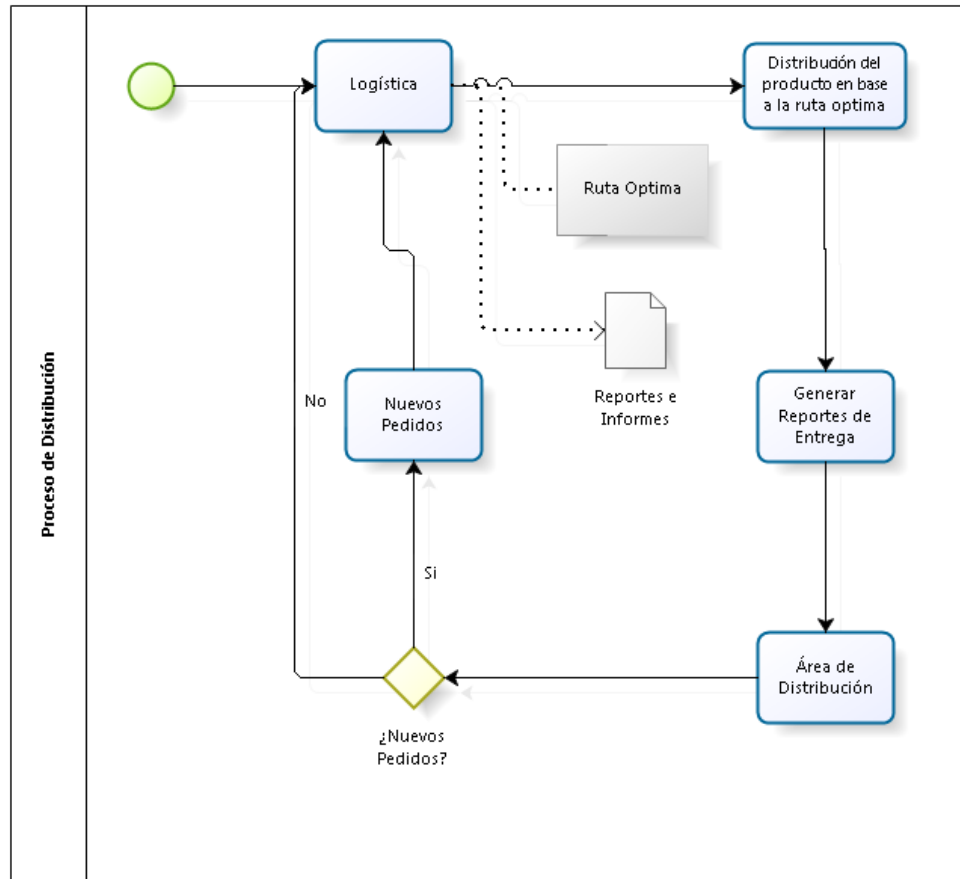
El proceso de distribución físico es transportar un producto o servicio utilizando la menor cantidad de recursos posible así como hacerlo de la forma más rápida posible. El producto o servicio es transportado desde la línea de producción hasta el consumidor o punto de venta.

En este flujo se puede incluir diferentes empresas las cuales son independientes como distribuidores minoristas, mayoristas, el fabricante entre otras empresas. Todos estos pasan a ser un canal de distribución diferente.

La distribución va ligada estrechamente al método de transporte que se utilice para el traslado de la mercadería. El costo de transporte está sujeto a diferentes factores, como lo son: el tráfico, la infraestructura, regulaciones del gobierno, variación en los costos de los combustibles, así como cualquier otro factor que influya directamente al costo de traslado.

El flujo básico de transporte es el que se presenta en la siguiente figura, donde se puede encontrar una unidad de logística, la cual es la encargada de validar los informes que fueron entregados de las previas distribuciones y en base a ellos calcular nuevamente la ruta.

Figura 5. **Proceso de distribución**



Fuente: elaboración propia.

## 2.2. **Tecnología actual y su uso en la empresa**

Actualmente se disponen de diferentes mecanismos para la distribución de productos, por lo cual este proceso se lleva a cabo a través de las siguientes herramientas: *handheld*, GPS y POS.



### **2.2.1. Uso del *handheld***

El *handheld* surgió como un dispositivo de ayuda portátil para poder tomar información, es como una pequeña computadora portátil, el objetivo es que pueda utilizarse para las diferentes funciones de la empresa.

Una de las ventajas de este dispositivo es la portabilidad, ya que para ello fue diseñado, además que permite una captura de información fácil y eficiente, así como también el diseño de aplicaciones que se adaptan a las necesidades del negocio. Estos dispositivos suelen ser muy resistentes debido a que fueron pensados para el uso en diferentes ramas de la industria.

Para el caso de las empresas distribuidoras se utiliza para el control de inventarios, así como para la verificación y entrega de mercadería a los diferentes clientes, ya que en ellos se puede acceder a la información de los clientes, y poder validar que pedido realizaron y con la frecuencia que se debe entregar dicho pedido. En algunas empresas se utiliza también para realizar el pago electrónicamente desde este dispositivo.

### **2.2.2. Uso del GPS**

Sistema Global de Navegación por Satélite o GPS por sus siglas en inglés, es un aparato que permite determinar las coordenadas de latitud y longitud de un objeto en cualquier parte del mundo.

En las empresas actualmente se utiliza para marcar la ruta de distribución del producto, y así poder llevarlo a los diferentes clientes. La ruta es trazada por un programa de logística para posteriormente ser cargada al GPS, así como también este puede brindar una localización del vehículo en cualquier momento,

con esto poder visualizar donde se encuentra el vehículo y validar que el vehículo se mantenga en la ruta de distribución establecida.

Una de las ventajas que permite el GPS es poder seguir la ruta sin problemas, así como también algunos disponen comando de voz, estos aparatos son colocados en el vehículo de distribución.

### **2.2.3. Uso del POS**

Terminal de punto de venta o POS por sus siglas en inglés, es una terminal de computadora que se encuentra instalada en el establecimiento afiliado, esta se comunica vía teléfono al servidor central donde se realizará el consumo. Este pequeño aparato se encarga de leer las tarjetas con cinta magnética

El uso en la empresa radica en el cobro del producto, debido a que únicamente se desliza la tarjeta, se ingresa el monto a debitar y se realiza el cobro automáticamente, por medio electrónico.

### **2.3. ¿Por qué mediana y pequeña empresa?**

Esta sección del capítulo está dividida principalmente en dos pequeñas áreas, en la primera se explican algunas definiciones, las cuales son necesarias para entender los alcances a grandes rasgos de una pequeña y mediana empresa en Guatemala, luego se presenta basado en los puntos anteriores el porqué de la pequeña y mediana empresa.

### **2.3.1. Costo operativo**

Los costos operativos son todos aquellos costos que surgen de las actividades que se realizan en la empresa como lo son: sueldos de los trabajadores, energía eléctrica, renta, así como también el mantenimiento de un sistema instalado y funcionando.

### **2.3.2. Productividad**

La productividad es la relación que existe entre los recursos que están disponibles en la empresa he invierte en su operación y los beneficios que se obtiene de la misma, la productividad es una métrica que se debe considerar para poder analizar el estado en que se encuentra la empresa.

### **2.3.3. Servicio al cliente**

Es la forma en que la empresa se relaciona con sus clientes, son todas aquellas actividades que satisfacen la petición del cliente. Cada empleado dentro de la empresa que tiene contacto con el cliente, debe estar consciente de brindar un buen servicio al cliente, ya que esta es una estrategia eficaz.

### **2.3.4. Retorno de inversión**

Retorno de la inversión o ROI por sus siglas en inglés, es un indicador financiero por el cual se puede medir la rentabilidad, esto quiere decir que se compara la utilidad obtenida en relación con la inversión realizada.

### 2.3.5. Pequeña y mediana empresa

Una de las interrogantes sobre esta investigación es ¿por qué se centra en pequeñas y medianas empresas?, esto es debido a que las grandes empresas ya tienen procesos estandarizados para el cálculo de ruta y gestión de inventarios, así como también ya realizan todo el proceso de distribución con los diferentes recursos como lo son la *handheld*, GPS, etc.

Las pequeñas y medianas empresas poseen un proceso no estandarizado, con el cual no llevan el control de todos sus distribuidores. Una pequeña o media empresa no posee el capital para poder implementar estos softwares de logística para poder gestionar una ruta óptima.

Una de las ventajas de la tecnología móvil, es que se puede adquirir un GPS, una *handheld* por un precio bastante económico. Debido a que la tecnología ha avanzado a un punto en el que se tienen *smartphone* a bajos precios.

Por ejemplo una empresa que tiene dos mensajeros, cada mensajero tiene diferentes puntos de entrega, el vehículo que utilizará cada mensajero es una motocicleta, con lo cual implica que si la empresa quisiera poder monitorear a ambos mensajeros así como también poderle proporcionar una ruta óptima para las diferentes entregas que debe realizar en el día; la empresa deberá de colocar en cada motocicleta un GPS esto implica un costo operativo demasiado alto, sin contar algunos otros factores como la seguridad.

## **2.4. Beneficios de utilizar aplicaciones móviles en la pequeña y media empresa distribuidoras**

A continuación se presentan los beneficios significativos que se encontraron durante la investigación, para lo cual se pueden adaptar las tecnologías móviles para empresas distribuidoras.

### **2.4.1. Ahorros operacionales**

Por medio de una aplicación móvil, diseñada para el cálculo de rutas se puede reducir los costos operativos de los vehículos y aumentar la productividad de la empresa.

- Se reducirá el número de kilómetros recorridos y los gastos de combustible, así como también menos depreciación a los vehículos ya que no se está perdiendo recursos en recorrer distancias innecesarias.
- Se podrá sincronizar a cada distribuidor, en las diferentes rutas de distribución que se deben cumplir, por lo cual se evita cualquier pérdida de recurso innecesaria abarcando un mismo territorio por dos diferentes repartidores.
- En la actualidad existen en Guatemala zonas que están catalogadas como Zona Roja la cual indica que es riesgoso transitar por estas calles, se puede colocar algún medio de seguridad marcando una ruta como peligrosa para que al calcular la ruta óptima se pueda hacer en base a estas métricas y obviar este tipo de trayectoria.

#### **2.4.2. Aumento de la productividad**

Por medio de la aplicación móvil permitiría aumentar la productividad y optimizar el uso del tiempo de los distribuidores o repartidores.

- Reducción significativa de los tiempos de planificación.
- Optimización del cargamento a ser transportado por los vehículos y lograr un ajuste ideal en los volúmenes de repartición.
- El tiempo muerto entre las diferentes reparticiones se reduce con una buena planificación.

#### **2.4.3. Mejora del servicio al cliente**

A través de la aplicación móvil se puede realizar pedidos, así como poder ubicar a clientes potenciales, por medio de la aplicación móvil también se podrá realizar encuesta del servicio de los distribuidores, para poder notar si el tiempo de entrega fue el adecuado.

- Se podrá cumplir con el día y horario programado con el cliente.
- Se podrá adelantar el pedido, ya que el mismo cliente puede realizarlo a través de su dispositivo móvil.

#### **2.4.4. Mayor control de los repartidores**

Al poder proporcionar a cada repartidor un *smartphone* se podrá saber en dónde se encuentra en cierto momento el repartidor debido a que se tiene la ruta trazada, el repartidor deberá continuar con esta ruta. Si existiera algún percance podrá monitorearse por que el vehículo se quedó cierto tiempo estacionado.

- Se lanzará alertas con base en la ruta de distribución que debe cumplir cada repartidor, al momento de salirse de la ruta poder validar hacia donde se está dirigiendo el vehículo.
- Medir tiempo de entrega de producto en cada cliente.

### 3. ARQUITECTURA DE ADF MOBILE

ADF Mobile es un *framework*, para el desarrollo de aplicaciones para dos plataformas las cuales son iOS y Android, este *framework* se basa en un componente híbrido, con el cual se logra el mapeo de los componentes nativos del *smartphone* (GPS, agenda de contactos, etc.) y se logra utilizar en los otros diferentes componentes de dicho *framework*.

Debido a que ADF Mobile se integra a estas dos plataformas se pueden encontrar diferentes componentes los cuales representan diferentes funcionalidades de la aplicación móvil, dichos componentes tienen una similitud con los componentes de una aplicación híbrida, pero Oracle estableció otras capas para garantizar la integración con la plataforma para el desarrollo de aplicaciones *Enterprise Edition* de Java.

ADF Mobile permite establecer características establecidas por el usuario, pero a la vez permite establecer un estilo estándar para el desarrollo de cualquier aplicación con lo cual tendrá un repositorio de características estandarizado, así como también se pueden agrupar estas características clasificándolas por medio de su funcionalidad.

#### 3.1. Componentes de la arquitectura

En este capítulo se presenta cada uno de los componentes involucrados en la arquitectura de ADF Mobile, primero se comenzará describiendo el Modelo Vista Controlador, que es la base de la arquitectura ADF Mobile a continuación se especifica cada uno de los elementos de esta arquitectura los



cuales permiten el correcto funcionamiento para las plataformas de IOs y Android.

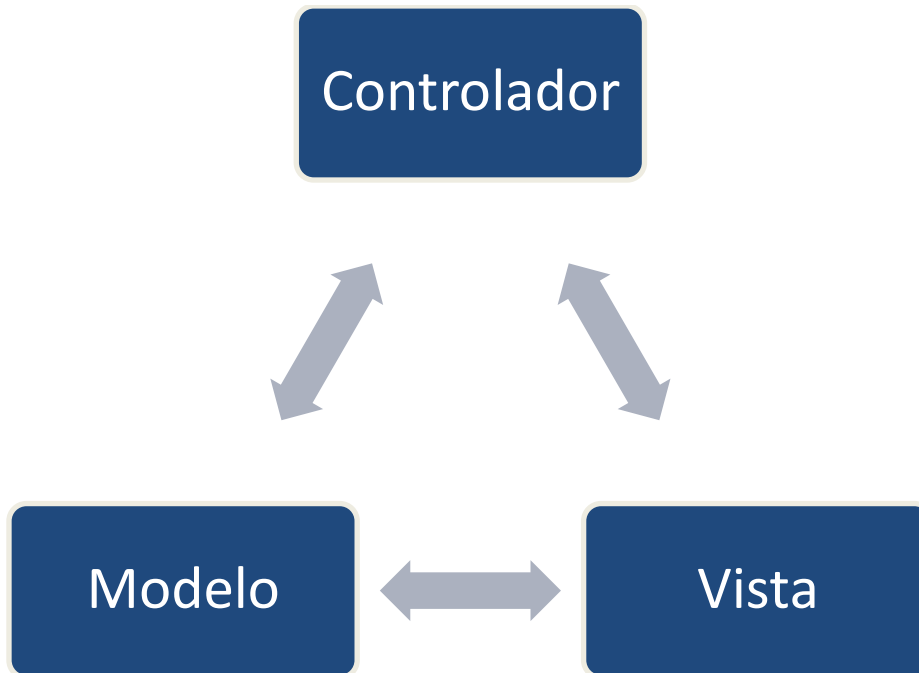
### **3.1.1. Patrón de diseño MVC**

El patrón de arquitectura de software Modelo Vista Controlador es probablemente el patrón más citado en la programación web, y a la vez es el encargado de separar los datos y la lógica del negocio de la interfaz de usuario, asimismo, es que tiene la tarea de gestionar los eventos y las comunicaciones.

Desde este punto de Vista MVC propone la utilización de tres componentes distintos los cuales son el modelo, la vista y el controlador los cuales se estarán describiendo más adelante, además ADF Mobile agrega el Java Runtime el cual se encuentra incrustado en la aplicación.

A continuación se presenta la gráfica la cual muestra la forma en la que se comunican cada uno de los tres componentes de MVC.

Figura 6. **Modelo vista controlador**



Fuente: elaboración propia.

#### **3.1.1.1. Modelo**

Este componente es la representación de la información con la que el sistema se encuentra operando, esta parte es la encargada de gestionar todos los accesos a esta información, consultas, actualizaciones y la implementación de los privilegios de acceso que se han descrito en la lógica del negocio. Asimismo, es el encargado de enviar a la vista la información solicitada por el usuario, cada petición de dicho usuario llegan al modelo por medio del controlador. El modelo es el responsable de los siguientes aspectos:

- Acceder a la capa en la cual se encuentran almacenados los datos, la buena práctica es que el modelo sea independiente del sistema de almacenamiento de información.
- Llevar un registro de las vistas y controladores involucrados al sistema.
- Si se está trabajando en un modelo activo notificará a cada una de las vistas los cambios que han sido generados por agentes externos.

### **3.1.1.2. Vista**

La vista es la presentación del modelo (información y lógica del negocio) en un formato acorde para poder interactuar, normalmente es la interfaz de usuario y busca que la información que se encuentra en el modelo sea eficaz y confiable para poder presentarla al usuario como una salida en el formato presentado en la interfaz. Este componente es el responsable de las siguientes funciones:

- Recibir los datos enviados por el modelo y presentarlos al usuario de forma entendible.
- Posee al igual que el modelo un registro, pero en este caso es del controlador asociado a la vista, ya que es con quien le toca interactuar para poder realizar las peticiones de información.
- Es el encargado de enviar las peticiones realizadas por el usuario al controlador para que este pueda interactuar con el modelo.

### 3.1.1.3. Controlador

El controlador es el que responde a los eventos los cuales usualmente son peticiones de los usuarios, en estos eventos se invoca al modelo cuando se hace una solicitud de información por ejemplo la actualización de un registro que se encuentra en la base de datos. También se pueden enviar a través del controlador comandos a la vista asociada por ejemplo, si en la vista se hizo una petición al modelo y la información retornada por este es una cantidad de registros bastante grande, el controlador puede pedirle a la vista que presente los datos en una sola pantalla y que implemente un *scroll* para poder apreciar toda la información. Siendo el controlador el responsable de las siguientes funciones:

- Recibir los eventos de entrada enviados a través de la vista
- Contiene las reglas que rigen cada uno de los eventos, por ejemplo si se realiza la acción X entonces ejecutar Z, siempre recordando que estas acciones pueden incluir peticiones tanto a la vista como al modelo
- El controlador es el encargado de comunicar la vista con el modelo

#### **3.1.1.4. Java runtime**

Como se mencionó con anterioridad dependiendo de la implementación el Modelo Vista Controlador puede tener variaciones en cuanto a los componentes que interactúan con él, en este caso para ADF Mobile el MVC reside en el dispositivo móvil, y representa una re implementación del ADF Modelo Vista Controlador y cada una de sus capas. La Java virtual machine se encuentra incrustada en cada aplicación e interactúa con el controlador para poder realizar las diferentes acciones solicitadas por el usuario.

Al Java Runtime se le puede describir como un conjunto de utilidades que permite la ejecución de programas Java, ya que posee un conjunto de bibliotecas Java y otros componentes que son necesarios para que una aplicación escrita en lenguaje java pueda ser ejecutado correctamente.

#### **3.1.1.5. Interacción de los componentes**

Existe una variedad de representaciones del Modelo Vista Controlador, pero en general la interacción de estos tres componentes con el Java runtime se explica a continuación, aclarando nuevamente que dependiendo de los requerimientos puede tener variaciones en alguna de las capas.

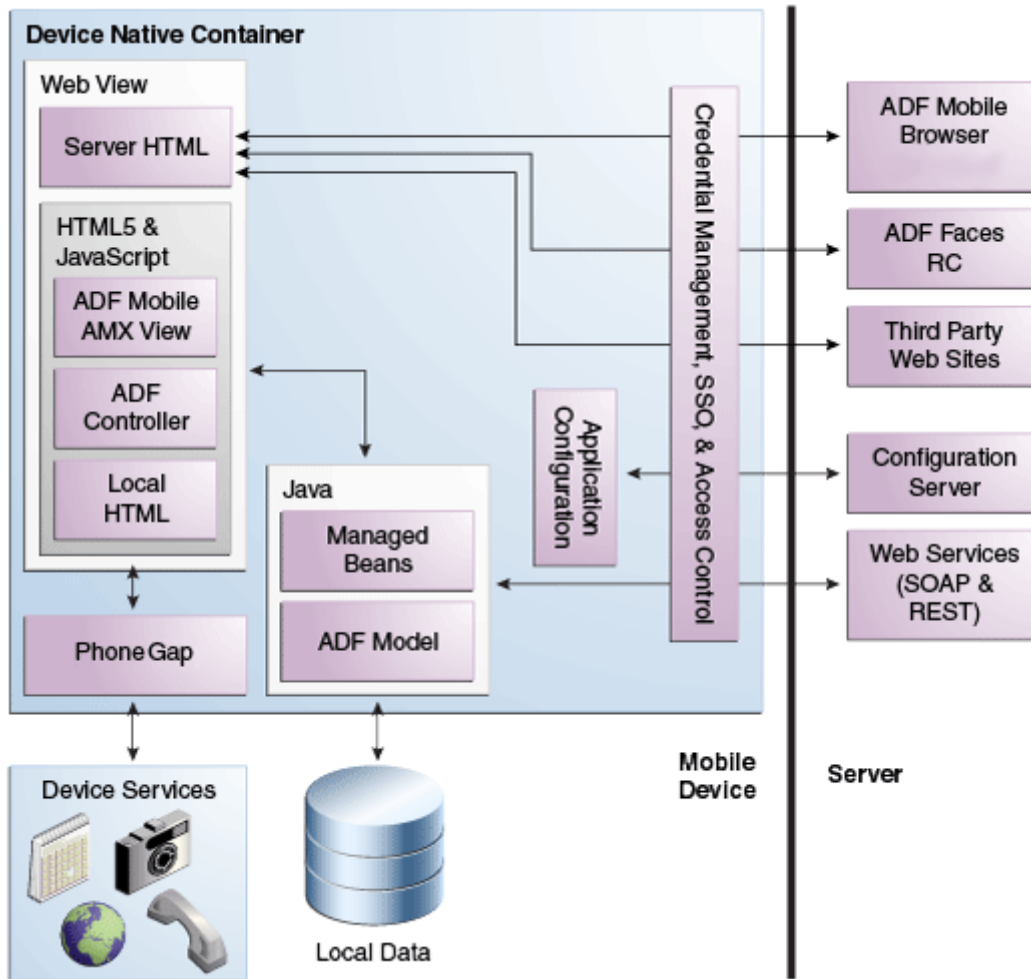
- El usuario interactúa con la interfaz de usuario, ya sea presionando en algún enlace o en un botón el cual genera una petición.
- El controlador recibe la petición y se encarga de gestionar y validar la acción que se debe realizar de acuerdo a dicha petición, en este caso el controlador también interactúa con el Java runtime para poder validar y gestionar las diferentes peticiones.

- El controlador accede al modelo dependiendo de la acción puede solamente consultar información o modificar esta misma.
- El controlador entrega a la vista los datos solicitados, y esta se encarga de desplegarlos al usuario de una forma apropiada y entendible. Como se observa el modelo no tiene conocimiento directamente sobre lo que la vista hará con los datos solicitados, se pueden implementar patrones de diseño para poder involucrar indirectamente el modelo y la vista, pero estos seguirán manteniendo cierto grado de independencia.
- Nuevamente la vista queda a la espera de una nueva acción generada por el usuario para empezar nuevamente el ciclo.

### **3.1.2. Elementos de la arquitectura**

En esta parte del capítulo se proporciona a detalle cada uno de los elementos de ADF Mobile, los cuales están destinados a proveer una aplicación segura y confiable además de las virtudes que se mencionan en los siguientes capítulos, los cuales hacen de ADF Mobile un *framework* robusto. El diagrama de la aplicación se presenta a continuación y se encuentra dividido principalmente en tres áreas las cuales son el Device Native Container, el Server y el Mobile Device.

Figura 7. Arquitectura de ADF Mobile



Fuente: Oracle® Fusion Middleware Mobile Developer's Guide for Oracle Application Development Framework 11g Release 2 (11.1.2.3.0). Part Number E24475-01.

[http://docs.oracle.com/cd/E35521\\_01/doc.111230/e24475/intro.htm](http://docs.oracle.com/cd/E35521_01/doc.111230/e24475/intro.htm). Consulta: 8 de septiembre de 2014.

### **3.1.2.1. Device-Native Container**

Es uno de los componentes más importantes y representa el contenedor de aplicaciones o plantilla compilada como una aplicación binaria de un dispositivo nativo. Este contenedor provee el ambiente de ejecución para que la aplicación de ADF Mobile pueda ejecutarse en el dispositivo móvil como una aplicación nativa. Además de acoger los componentes del lado del cliente, provee las herramientas de navegación las cuales permiten el acceso particular a las características de la aplicación.

### **3.1.2.2. Web View**

Como se ha mencionado ADF Mobile soporta contenido web, el Web View como su nombre lo indica es la parte del contenedor nativo que utiliza el motor web del dispositivo móvil para mostrar y procesar el contenido basado en web.

### **3.1.2.3. Server HTML**

Representa una interfaz de usuario basada en web que es generada en el servidor y es entregada como una página web de ADF Mobile. El código HTML de la aplicación, la lógica del negocio, y el flujo de la página son generados en un servidor remoto. El servidor HTML puede acceder a los servicios nativos así como a la cámara a través del API Javascript soportado por PhoneGap, mientras se está ejecutando por una aplicación de ADF Mobile. Opciones comunes para las páginas de servidores basados en HTML son el ADF Mobile browser y el Oracle ADF Faces.



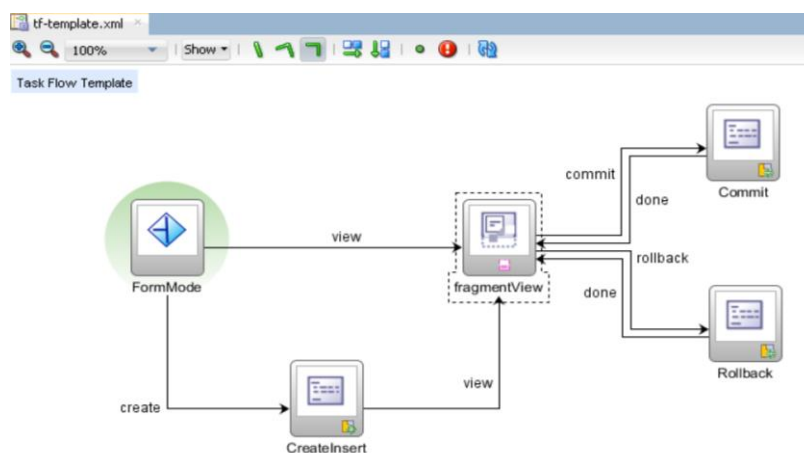
### 3.1.2.4. Local HTML

Representa el desarrollo de las páginas web usando JDeveloper o herramientas de terceros que directamente están integrados en el ADF Mobile. Las páginas son entregadas como parte de la aplicación de ADF Mobile. Los archivos de HTML locales pueden ser accedidos como funciones nativas del dispositivo a través del API de Javascript soportado por PhoneGap.

### 3.1.2.5. ADF Controller

Está representado por una versión móvil del Oracle ADF Controller que admite un subconjunto de componentes de flujo disponibles para una aplicación Oracle ADF basada en el servidor. Los flujos de tareas limitados e ilimitados son soportados por ADF Oracle, así como un conjunto de eventos son apoyados por el ADF basado en el servidor.

Figura 8. ADF Mobile Definición de Flujos en JDeveloper



Fuente: BARANOVSKIS, Andrejus . *ADF Task Flow Template Improvements in 12c*.  
<http://andrejusb.blogspot.com/2013/08/adf-task-flow-template-improvements-in.html>. Consulta: 8 de agosto de 2014.

### **3.1.2.6. ADF Mobile AMX Views**

Es la tecnología basada en el ADF Mobile AMX que brinda una experiencia de desarrollo JSF parecida a una interfaz de usuario basada en HTML5. Las vistas de ADF Mobile AMX son definidas usando la interfaz de usuario y el editor de códigos provisto por JDeveloper. Estas vistas están embebidas en las aplicaciones del ADF Mobile y desplegadas en el dispositivo móvil. En el tiempo de ejecución el motor de Javascript brinda en la vista web las definiciones de vistas de ADF Mobile en componentes de HTML5. De los enfoques de implementación proporcionados por ADF Mobile las características de las aplicaciones que utilizan los componentes de ADF Mobile AMX presentan una auténtica experiencia al usuario a través del soporte de animaciones y gestos.

### **3.1.2.7. Java**

Provee un ambiente de ejecución Java para las aplicaciones de ADF Mobile, esta es la JVM (Java Virtual Machine) la cual es implementada en el código nativo del dispositivo y es embebido o compilado en cada instancia de la aplicación ADF Mobile. La JVM es basada en las especificaciones de JavaME Connected Device Configuration (CDC).

### **3.1.2.8. Managed Beans**

Son las clases de Java creadas para ampliar las capacidades del ADF Mobile, tales como el procesamiento de los datos de la lógica del negocio devueltos por el server. Managed Beans son ejecutados por el soporte embebido de Java y por lo tanto debe ajustarse a las especificaciones de la JavaME CDC.

### **3.1.2.9. ADF Model**

En una aplicación de ADF Mobile se soporta un subconjunto de componentes de la lógica del negocio disponibles para una aplicación Oracle ADF basada en el servidor. El ADF Model contiene la capa de enlace que proporciona una interfaz entre los componentes de lógica de negocio y la interfaz de usuario, así como la lógica de ejecución para invocar el resto o servicios *web* basados en SOAP.

### **3.1.2.10. Application Configuration**

En este punto se hace referencia a los servicios que permiten la configuración de aplicaciones que van a ser descargadas o actualizadas. Por ejemplo la URL de un Web Services, o la URL de una conexión remota.

### **3.1.2.11. Credential Management and Access Control**

Esta parte se enfoca en los servicios del lado del cliente que proporcionan servicios relacionados con la seguridad de una aplicación ADF Mobile, como por ejemplo, las credenciales de almacenamiento local que son almacenadas de forma segura en caché para luego ser accedidas de manera *offline* o los servicios para mostrar u ocultar características de la aplicación basadas en los permisos de usuario.

ADF Mobile utiliza JavaScript y un *plugin* de PhoneGap, así como manejadores de comandos nativos que conjuntamente manejan la interacción con la página de inicio de sesión de usuario.

Figura 9. **Página de login utilizando Credential Management Access Control**



Fuente: *Oracle® Fusion Middleware Mobile Developer's Guide for Oracle Application Development Framework 11g Release 2 (11.1.2.3.0). Part Number E24475-01.*

[http://docs.oracle.com/cd/E35521\\_01/doc.111230/e24475/security.htm#CDDHGJAG](http://docs.oracle.com/cd/E35521_01/doc.111230/e24475/security.htm#CDDHGJAG). Consulta:

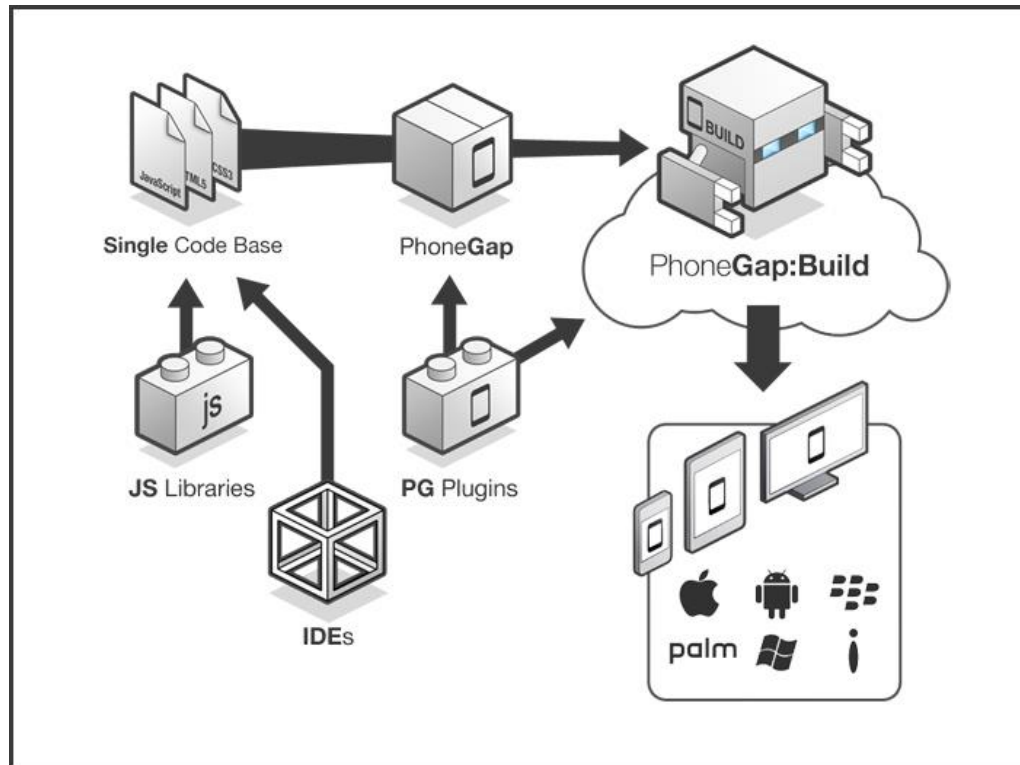
06 de junio de 2014.

### **3.1.2.12. Apache Cordova**

Es una biblioteca de código abierto que proporciona una API de JavaScript para acceder a diversos servicios del dispositivo móvil tales como la cámara. PhoneGap provee la mayoría de los servicios de integración del dispositivo para ADF Mobile. La API de JavaScript esta abstraída como controles de datos en el diseñador de JDeveloper de ADF Mobile AMX basado en vistas, permitiendo la integración de servicios del dispositivo simplemente arrastrando y colocando los controles de datos en la vista del ADF Mobile AMX.

A continuación se presenta un diagrama el cual muestra a grandes rasgos la interacción de PhoneGap con otros sistemas nativos, así como la interacción con JavaScript.

Figura 10. Interacción PhoneGap



Fuente: Agency Blueleaf. *¿Qué es excelente sobre PhoneGap?*. <http://www.blueleaf.co.uk/what-is-phonegap/>. Consulta: 15 de agosto de 2014.

### 3.1.2.13. Local data

Se refiere a la data que se encuentra almacenada en el dispositivo móvil, en ADF Mobile, esta se encuentra encriptada utilizando el gestor de bases de datos SQLite, todas las operaciones tales como crear, eliminar, modificar son compatibles con este almacenamiento de datos local a través de la capa que provee JDBC API de Java.

Es importante mencionar que para mantener la concurrencia a nivel local, el SQLite permite en un tiempo de ejecución solo una única instancia de escritura-lectura o varias instancias de solo lectura.

#### **3.1.2.14. Configuration Server**

Se refiere a un servidor WebDav (Web Distributed Authoring and Versioning) basado en archivos de configuración de *hosts* utilizados por los servicios de configuración de la aplicación. El servidor de configuración se entrega como una implementación de referencia cualquier servicio WebDav es comúnmente alojado en el servidor J2EE (Java 2 Platform Enterprise Edition).

WebDav se refiere al protocolo el cual proporciona las funcionalidades para crear, editar y mover documentos en un servidor remoto, con el fin de hacer de la World Wide Web un medio legible y editable.

Es importante aclarar que el J2EE es una plataforma de programación para ejecutar y desarrollar aplicaciones de software en el lenguaje Java, a la vez permite utilizar arquitecturas de N capas.

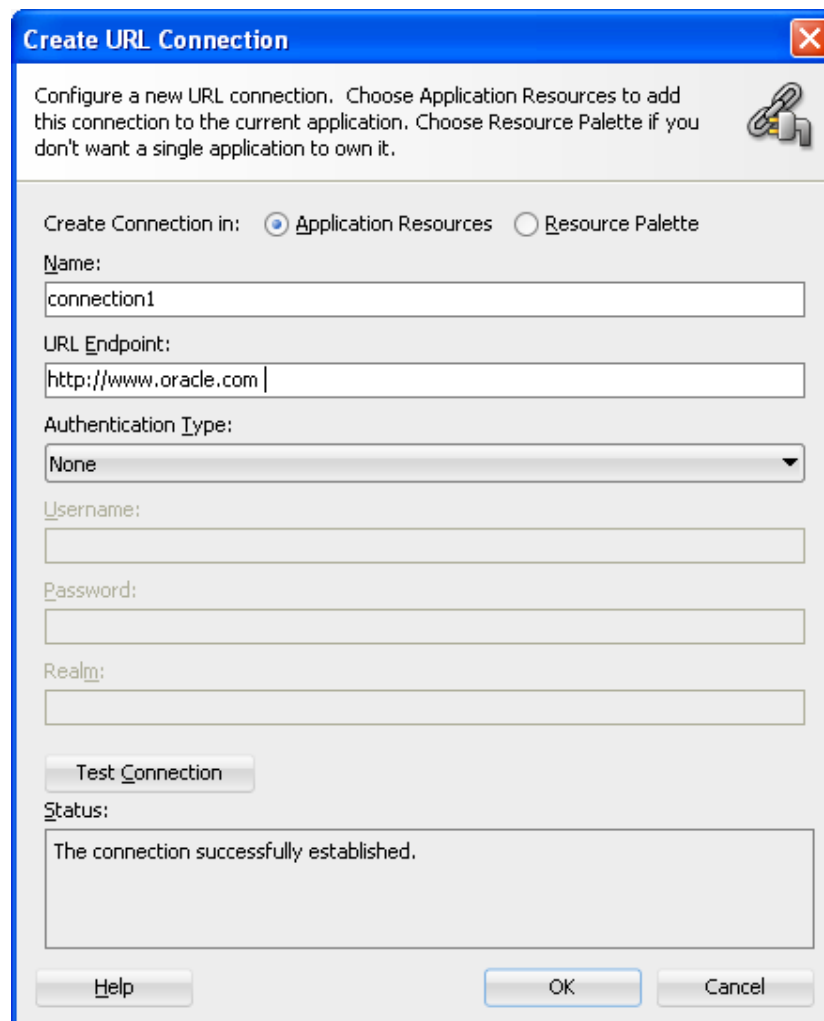
#### **3.1.2.15. ADF Mobile Browser**

Del lado del servidor se encuentra que el ADF Mobile browser el cual se refiere al *framework* para el desarrollo de aplicaciones del servidor que se puede utilizar para la implementación de la función de URL remotas en ADF Mobile.

Aplicaciones basadas en el navegador para teléfonos inteligentes son creadas para permitir la presentación de una variedad de características propias

de cada teléfono inteligente. A continuación se muestra una imagen en la cual se puede ver como se puede agregar una URL remota para navegarla a través del ADF Mobile Browser.

Figura 11. **Conección URL**



Fuente: *Oracle® Fusion Middleware Mobile Developer's Guide for Oracle Application Development Framework 11g Release 2 (11.1.2.3.0).*

[http://docs.oracle.com/cd/E35521\\_01/doc.111230/e24475/define\\_features.htm#CIHBGAFI](http://docs.oracle.com/cd/E35521_01/doc.111230/e24475/define_features.htm#CIHBGAFI).

Consulta: 08 de septiembre de 2014.





## **4. COMPARATIVA DE ADF MOBILE FRENTE A OTROS FRAMEWORKS**

En la actualidad existen diferentes plataformas para teléfonos móviles como lo son Android OS de Google, iOS de Apple, entre otras plataformas que se encuentran en el mercado. ADF Mobile como se mencionó en los capítulos anteriores es un *framework* de desarrollo híbrido, el cual permite el desarrollo para diferentes plataformas con poco esfuerzo.

La primera comparación que se presenta es frente a estas diferentes formas de desarrollar aplicaciones móviles, las ventajas y desventajas que se tienen frente a las aplicaciones nativas y aplicaciones web. Posteriormente se realizará una comparativa entre otros *frameworks* híbridos los cuales tienen características similares a las de ADF Mobile.

### **4.1. Comparativa de aplicaciones híbridas frente a aplicaciones nativas y web**

En el capítulo 1, se observó y definió las diferentes aplicaciones que existen en el mercado, las cuales son nativas, web e híbridas así como las ventajas y desventajas que presentan cada una de ellas. En esta sección se realiza la comparación entre cada una de ellas con el fin de poder validar el *framework* ADF Mobile, y las ventajas que conlleva su implementación.

#### **4.1.1. Características de aplicaciones nativas**

Las aplicaciones nativas poseen características muy propias del dispositivo y del sistema operativo que se corre en el mismo, por esta razón las hacen muy óptimas al momento de correr gráficos y mucho más rápidas que los otros tipos de aplicaciones, sus principales características son:

- Lenguaje propio de desarrollo: el lenguaje de la aplicación depende de cada sistema operativo donde se está desarrollando, por ejemplo, para Android OS se utiliza el lenguaje Java así como las diferentes librerías que el SDK conlleva.
- Rendimiento y velocidad: la característica anterior, permite desarrollar a nivel específico para la plataforma y el dispositivo teniendo así un alto rendimiento al momento de correr la aplicación en el dispositivo.
- Tiempo de desarrollo: para poder desarrollar este tipo de aplicaciones se debe tener conocimiento de los lenguajes de cada plataforma a la que se desea lanzar la aplicación, por lo cual se debe de estimar un tiempo de desarrollo para cada plataforma específica, debido a que cada aplicación lanzada al mercado es independiente una de otra.
- Ciclo de vida de la aplicación: como se mencionó en la característica anterior, se debe estimar un tiempo de desarrollo para cada plataforma, por esta razón se tiene un ciclo de vida extenso, al igual que para lanzar una actualización se debe manejar diferente para cada una de las plataformas.

- Portabilidad: la portabilidad en aplicaciones nativas no existe debido a que no se puede utilizar nada del código fuente escrito de en una plataforma, y pasarlo hacia otra.
- Soporte de características nativas: las aplicaciones nativas soportan todas las características nativas del dispositivo sobre el cual se están ejecutando, debido a que el SDK fue optimizado para poder utilizar estas características.
- Soporte online/offline: este tipo de aplicaciones puede utilizarse tanto fuera de línea (offline) como en línea (online). Debido a que el almacenamiento de información se puede hacer de forma local, y la aplicación se encuentra instalada en el dispositivo.
- Aprendizaje de desarrollo: se debe aprender el lenguaje de la plataforma que se deseé desarrollar. Esto lleva un tiempo de aprendizaje alto.
- Costo de desarrollo: el costo de desarrollo es alto, debido a que se debe tener un equipo diferente para cada plataforma que se desea liberar la aplicación. Como se mencionó en las características anteriores el tiempo de desarrollo es más alto

#### **4.1.2. Características de aplicaciones web**

- Lenguaje propio de desarrollo: el lenguaje en que se desarrollan las aplicaciones web es HTML y CSS
- Alto rendimiento y velocidad: debido a que estas aplicaciones corren en el navegador, depende del recurso que este tenga asignado, por esta

razón pueden correr lentos los gráficos de dichas aplicaciones, sin mencionar que la conexión a internet depende mucho de la velocidad para cargar las páginas de estas aplicaciones.

- Tiempo de desarrollo: un tiempo más reducido frente a las aplicaciones nativas e híbridas, debido a que se lanza una aplicación para todos los dispositivos.
- Ciclo de vida de la aplicación: el ciclo de vida para las aplicaciones web es corto, incluso las actualizaciones se hacen automáticas, sin que el usuario se dé cuenta que se aplicaron a la aplicación.
- Portabilidad: la portabilidad es alta, se desarrolla una vez y se ejecuta en todos los dispositivos que se desea.
- Soporte de características nativas: La deficiencia mayor en este tipo de aplicaciones es en el uso de GPS, cámara, etc. Debido a que no se puede utilizar ninguno de estas características, el posicionamiento lo pueden realizar por medio de la IP asignada al dispositivo pero es totalmente impreciso.
- Soporte online/offline: este tipo de aplicaciones necesitan tener conexión a internet para poder trabajar.
- Aprendizaje de desarrollo: el lenguaje HTML es común entre los programadores el aprendizaje es mínimo para el desarrollo de una aplicación web.

- Costo de desarrollo: comparado con las aplicaciones nativas, el costo de desarrollo es bajo por que únicamente se lanza una aplicación para todas las plataformas.

#### **4.1.3. Características de aplicaciones híbridas**

- Lenguaje propio de desarrollo: en este tipo de aplicaciones se utiliza HTML5, CSS3 y JavaScript.
- Alto rendimiento y velocidad: debido a que estas aplicaciones hacen uso de un componente de JavaScript corren más eficientemente que las aplicaciones web, pero para la parte gráfica es más eficiente una aplicación nativa, debido a que está realizada específicamente para el dispositivo.
- Tiempo de desarrollo: debido a que se puede lanzar para varias plataformas, el tiempo de desarrollo comparado con las aplicaciones nativas es corto.
- Ciclo de vida de la aplicación: para poder lanzar una actualización menor se puede realizar desde el servidor sin que el usuario perciba dicho cambio, para actualizaciones del contenedor nativo si se necesitara la aprobación del usuario.
- Portabilidad: este tipo de aplicaciones son altamente portables, debido a que se pueden ejecutar en varias plataformas utilizando el mismo código fuente.

- Soporte de características nativas: el soporte de características nativas es relativamente alto, debido a que depende del *framework* que se esté utilizando, pero en relación con las aplicaciones web es alto, ya que la mayoría de *frameworks* permite la utilización de las características nativas más relevantes como lo son: GPS, cámara, libreta de contactos, acelerómetro, entre otras
- Soporte online/offline: permiten soporte online tanto como offline esta es una característica de HTML5. Además se puede guardar la información del usuario localmente para cuando se posea conexión poder sincronizar con el servidor central de ser necesario
- Aprendizaje de desarrollo: el aprendizaje es relativamente corto comparado con las aplicaciones nativas, debido a que como se mencionó en las aplicaciones web, HTML y CSS es conocido en el mundo de los desarrolladores, y es técnicamente fácil de aprender, la mayor complejidad de estas aplicaciones depende de las librerías JavaScript
- Costo de desarrollo: el costo de desarrollo es menor, que una aplicación nativa, debido a que se puede abarcar varias plataformas con el mismo código fuente.

#### 4.1.4. Cuadro comparativo de aplicaciones móviles

Luego de identificar las diferentes características de cada tipo de aplicación móvil se puede resumir en el cuadro presentado a continuación.

Tabla I. Cuadro comparativo aplicaciones nativas, web e híbridas

Característica	Nativa	Web	Híbrida
Lenguaje de desarrollo	Depende de la Plataforma	HTML y CSS	HTML5, CSS3 y JavaScript
Rendimiento	Alta	Medio	Medio
Tiempo de desarrollo	Alto	Bajo	Bajo
Ciclo de vida de la APP	Largo	Corto	Corto
Soporte de características nativas (GPS, contactos, cámara, etc.)	Alto	Ninguno	Medio
Portabilidad	Baja	Alta	Alta
Soporte <i>online/offline</i>	Alto	Nulo	Alto
Aprendizaje de desarrollo	Alto	Bajo	Bajo

Fuente: elaboración propia.

#### 4.2. Comparativa de ADF Mobile frente a otros *frameworks* híbridos

Existen diferentes *frameworks* para el desarrollo híbrido, hasta el momento se ha presentado únicamente ADF Mobile y la arquitectura del mismo, a continuación se presenta otros *framework* con los cuales se puede desarrollar aplicaciones híbridas dichos *frameworks* son Sehncha Touch, JQuery Mobile y



Sprout Core, y así poder comprar los beneficios que se dispone con las diferentes herramientas.

De los tres anteriores *frameworks* presentados se dará una visión general de las características más importantes de cada uno de ellos, como lo es la arquitectura, y los componentes más sobresalientes de cada uno, así como el tipo de licenciamiento que tienen.

#### **4.2.1. Sencha Touch**

Sencha touch es un *framework* JavaScript basado en el patrón MVC, fue diseñado para crear aplicaciones móviles para dispositivos con la capacidad de *touchscreen*.

Sencha touch es un *framework* muy extenso, a pesar de estar basado en JavaScript usa muy pocas librerías de JQuery o Prototype. Una de las grandes características de la versión Sencha Touch 2 que permitía el desarrollo de aplicaciones híbridas, posteriormente a esta versión únicamente permitía el desarrollo de aplicaciones web.

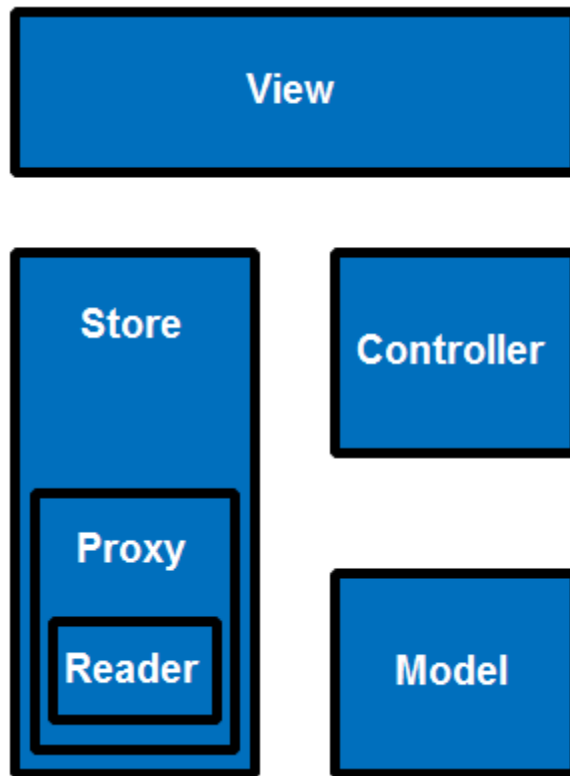
##### **4.2.1.1. Arquitectura de Sencha Touch**

La arquitectura utilizada por este *framework* es MVC con algunas variaciones, las cuales se detallan en breve. Sencha Touch provee un alto nivel orientado a objetos una parte provista por el lenguaje JavaScript y la otra por el propio *framework*. Con lo cual permite que los desarrolladores puedan crear clases, y poder instanciar cada una de estas clases para la creación de objetos.

En esta sección se presenta una figura en la cual se muestra la arquitectura de Sencha Touch básicamente es una arquitectura MVC con la adición de dos capas más las cuales son Stores y Proxies.

- Stores: es una abstracción para almacenar la información, básicamente describe como poder enlazar los datos de la aplicación, así como poder almacenarlos de mejor manera, ya sea por medio de listas o árboles de datos. Esta capa se encarga de ordenar, clasificar y agrupar la información a ser almacenada por la aplicación.
- Vistas: las vistas que están enlazadas con esta capa, son actualizadas automáticamente luego de una actualización de información. Existen dos tipos de *stores* ya sea lineal o jerárquico.
- Proxy: esta es una entidad abstracta la cual encapsula la conexión hacia un mecanismo de almacenamiento, puede ser local o remoto. Básicamente la data puede ser manipulada por este componente sin importar que este almacenada en la misma aplicación, o en algún servidor remoto. Este componente permite redireccionar la información si esta fuera cambiada de destino, realizando un cambio totalmente transparente para la aplicación, debido a que esto solo se realiza del lado del Proxy. Existen dos tipos de Proxy uno local y el otro remoto.

Figura 12. **Arquitectura de Sencha Touch**



Fuente: elaboración propia.

Los otros componentes vista, modelo y controlador de la arquitectura se comportan de forma similar a los explicados en el capítulo 3. Por esta razón solo se detallaron los nuevos que proporcionó el *framework* de Sencha Touch.

#### 4.2.1.2. Principales características

Sencha Touch es un *framework* orientado a la creación de utilidades y aplicaciones de productividad, actualmente es una plataforma completa con las siguientes características:

- Posee un *widget* con una gran librería de interfaz de usuario, inspirado en iOS tanto en diseño como en funcionalidad.
- Incluye un motor de renderizado rápido basado en CSS, que puede ser acelerado dependiendo del hardware en las últimas versiones de dispositivos móviles.
- Arquitectura bien definida reforzada desde el inicio en el modelo MVC.
- Conectores incorporados para el servicio de datos de red, algunos ejemplos son REST (*Representational State Transfer*) la cual es una técnica para describir cualquier interfaz web simple que utiliza XML y HTTP, así como para dar soporte para aplicaciones web fuera de línea.
- Un mecanismo avanzado de carga de clases el cual impone pautas de nomenclatura y de la arquitectura de MVC para mayor facilidad del usuario.
- Posee un sistema de línea de comandos el cual dirige la reducción de líneas de código de la aplicación, así como la compilación de aplicaciones nativas para Android y iOS.

- Extensa documentación disponible como un conjunto de páginas HTML dinámicas incluyendo búsqueda y filtrado de características sin requerir otra arquitectura del lado del servidor.
- Sencha Touch provee una clase principal que es llamada *class system*, la cual permite una mejor abstracción de la aplicación que se está desarrollando, también ayuda a los desarrolladores a la creación de código más portátil y la reutilización del mismo.

#### 4.2.1.3. Dispositivos y navegadores soportados

Sencha Touch en la actualidad soporta solamente las siguientes plataformas móviles:

- IOS desde la versión 3
- Android desde la versión 2.3
- BlackBerry desde la versión 6 (solamente para dispositivos con plataforma para navegadores *Web-kit powered*).
- Windows Phone desde la versión 8

Sencha Touch es un *framework* basado en el navegador y como tal es independiente del servidor, Sencha Touch puede implementar sus aplicaciones utilizando cualquier tecnología del lado del servidor tales como: PHP, Java, Ruby on Rails, .Net o cualquier otra elección.

#### **4.2.1.4. Licenciamiento**

Sencha Touch está disponible bajo un sistema de licenciamiento bastante complejo, hasta este momento los desarrolladores pueden usar el *framework* de la siguiente manera:

##### **4.2.1.4.1. Proyectos Open Source**

- Si se planea distribuir la aplicación y revelar el código fuente hay una licencia de Sencha Touch distribuida a través de la licencia GPLv3
- Si no se desea usar la licencia GPLv3 existe una versión Free Libre Open Source (FLOSS) la cual posee una licencia también disponible

##### **4.2.1.4.2. Proyectos comerciales**

- Se puede utilizar Sencha Touch sin ningún recargo ya sea por aplicación, por usuario o por desarrollo.
- Por aplicaciones embebidas se puede utilizar Sencha Touch hasta 5,000 instalaciones.
- Finalmente para las empresas que deseen una licencia comercial OEM está disponible para compañías que deseen distribuir Sencha Touch como parte de sus propias aplicaciones comerciales o servicios.

## 4.2.2. jQuery Mobile

jQuery Mobile es un conjunto de *plugins* y *widgets* de jQuery que tienen como objetivo proporcionar una API multiplataforma para crear aplicaciones móviles, en cuanto a ejecución de código jQuery mobile es muy similar a la interfaz de usuario de jQuery.

jQuery mobile se creó para ayudar a los diseñadores y desarrolladores a crear aplicaciones móviles interactuando fácil con un ambiente web y además un entorno multiplataforma, personalizable y con código discreto.

### 4.2.2.1. Arquitectura de JQuery Mobile

jQuery utiliza un enfoque muy simple y potente para definir el contenido de la *webapp*. EL *framework* utiliza un enfoque discreto lo cual significa que los documentos HTML funcionarán incluso sin que jQuery Mobile cargue correctamente.

La unidad principal del *framework* es la página, no de forma igual a los archivos HTML normales la diferencia es que en este caso una página es un solo elemento *div* con un rol específico. Un documento HTML puede alojar una página o múltiples páginas dentro del mismo archivo, esto a veces es un nuevo concepto para la mayoría de los diseñadores de páginas. Se pueden vincular páginas dentro del mismo documento HTML o páginas en documentos HTML externos usando un HTML *markup* como una etiqueta.

La habilidad de incorporar más de una página dentro del mismo documento ha estado desde hace bastante tiempo atrás un buen ejemplo de esto es la obsoleta Wireless Markup Language (WML) este estándar tenía la

habilidad de insertar varias páginas visuales dentro del mismo documento con la meta de reducir la latencia y el tiempo de descarga, jQuery Mobile sigue esta misma estructura implementada sobre JavaScript y HTML.

Cuando se implementaba WML world cada página fue llamada una tarjeta y cada documento WML fue llamado una cubierta. En un fichero WML usa la etiqueta de la tarjeta para definir una página dentro de un documento mientras que en jQuery mobile se utiliza una etiqueta *div* con un role específico.

Los roles de una página en jQuery Mobile usa el formato HTML estándar, tal como la etiqueta *div*, esta se utiliza para establecer lo que el *framework* debe hacer con ese *div* se define un rol. A continuación una lista de los principales roles aceptados en jQuery Mobile.

Tabla II. Roles principales jQuery Mobile

Rol	Descripción
Page	Define una página y es utilizado para mostrar contenido
Header	Establece el encabezado de la página
Content	En esta sección se indica cual será el contenido de la página
Footer	Sección a pié de página
Navbar	Define una barra de navegación, normalmente a nivel de encabezado
Button	Para crear un botón visual
Controlgroup	Para crear un componente



Continuación de la tabla II.

<b>Collapsible</b>	<b>Es un panel el cual puede contraer el contenido dentro de una página</b>
<b>Collapsible-set</b>	Grupo de paneles de contenido que pueden contraer el mismo
<b>Fieldcontain</b>	Contenedor de campos
<b>Listview</b>	Contenedor de múltiples ítems de una lista
<b>Dialog</b>	Diálogo de una página
<b>Slider</b>	Control para visualizar valores booleanos
<b>Nojs</b>	Elemento que se aculatará en navegadores compatibles con jQuery Mobile

Fuente: elaboración propia.

#### 4.2.2.2. Principales características

jQuery Mobile comenzó en agosto de 2010 como un *framework* moderno incluyendo muchos patrones y prácticas para el desarrollo multiplataforma, las principales características del *framework* son las siguientes:

- Plataforma cruzada, dispositivo cruzado y navegador cruzado
- Interfaz de usuario utilizada para dispositivos táctiles
- Temas y diseños personalizables
- El uso de la semántica de HTML5 de manera no intrusa a nivel de código solamente, sin la necesidad de JavaScript, CSS o API de desarrollo.
- Las llamadas Ajax para cargar automáticamente el contenido dinámico

- Ejecución del núcleo de jQuery
- Tamaño ligero 120kb comprimido
- Mejora progresiva
- Soporte de accesibilidad

#### 4.2.2.3. Dispositivos y Navegadores Soportados

jQuery Mobile es un *framework* para dispositivos táctiles, incluyendo teléfonos inteligentes, *tablets* y dispositivos multimedia. La lista de compatibilidad puede ir cambiando con el tiempo, pero hasta este momento se menciona la versión de jQuery Mobile 1.0 la cual es compatible con los siguientes navegadores:

- iOS: Safari para iPhone, iPad Touch, y iPad para iOS 3.2
- Android OS: navegador Android para teléfonos inteligentes y *tablets*
- Blackberry OS: navegador Blackberry para para teléfonos inteligentes para la versión 5.0 y *tablets*
- Symbian: para navegadores Nokia con dispositivos touch
- webOS: navegador para webOS para la versión 1.4
- Bada: compatible también con el navegador de Bada
- MeeGo: para el micronavegador y el navegador Nokia incluido en el teléfono Nokia N9
- Windows Phone: Internet Explorer para Windows Phone Mobile 6.5 y Windows Phone 7.0
- Kindle: para el navegador de Kindle 3

Cabe mencionar que jQuery también es compatible con el siguiente trío de navegadores:

- Opera minitotalmente soportado para la versión 5.0 en la mayoría de los dispositivos móviles.
- Opera Mobile totalmente soportado para la versión 10.0 en la mayoría de los dispositivos móviles.
- Firefox Mobile
- Chrome, Firefox, Safari, Opera e Internet Explorer para escritorio también son compatibles con jQuery Mobile.

#### **4.2.2.4. Licenciamiento**

El uso de jQuery Mobile (y del núcleo de jQuery) es totalmente gratuito ya que es de código abierto, con licencia dual, MIT originada en Massachusetts Institute of Technology o la licencia GPL versión 2. La versión MIT es la recomendada para la mayoría de proyectos y no necesita nada adicional del usuario. Lo único importante es recordar no cambiar nada respecto a la información de copyright en la parte superior de los archivos.

#### **4.2.3. SproutCore**

En el sentido totalmente técnico SproutCore es un *framework* de código abierto para el desarrollo de aplicaciones web. EL *framework* posee un extenso grupo de bibliotecas, las cuales poseen características que proporcionan funciones auxiliares y útiles para desarrollar sitios web dinámicos.

En pocas palabras se puede decir que SproutCode es una plataforma de código abierto para construir aplicaciones en la nube de rápido acceso en la web, SproutCore ha desarrollado las herramientas necesarias para construir aplicaciones a gran escala las cuales lleguen a millones de usuarios a través de muchos dispositivos.

#### **4.2.3.1. Arquitectura de Sprout Core**

Las aplicaciones de SproutCore siguen un patrón de diseño clásico del cual ya se ha hablado en este trabajo, el Modelo Vista Controlador (MVC), el cual divide el código en tres áreas principales: el modelo, la vista y el controlador.

En SproutCore el modelo implementa la lógica del negocio y la aplicación del mismo, acá se definen los contactos y eventos como las relaciones entre ambos, adicionalmente en esta parte del código se validan los datos para asegurarse de que los datos siempre sean coherentes. El modelo también contiene código de interacción con el servidor que sincroniza los datos en memoria con almacenamiento local explorador y servidor.

La vista en SproutCore muestra la interfaz de usuario de la aplicación maneja también a bajo nivel el ratón táctil y los eventos del teclado y los convierte en acciones superiores de modo que la aplicación pueda responder. En SproutCore la visualización real de las vistas se queda en el nivel del navegador utilizando HTML, CSS, SVG, Canvas y en algunos casos Flash.

El controlador en SproutCore es el encargado del traslado de datos entre el modelo y las vistas, el controlador es el encargado de elegir el modelo de objetos visible en cada vista. En las aplicaciones SproutCore la capa del controlador se divide a menudo en diversos controladores que trasladan peticiones de datos y respuestas que responden a acciones de petición por parte de la vista.

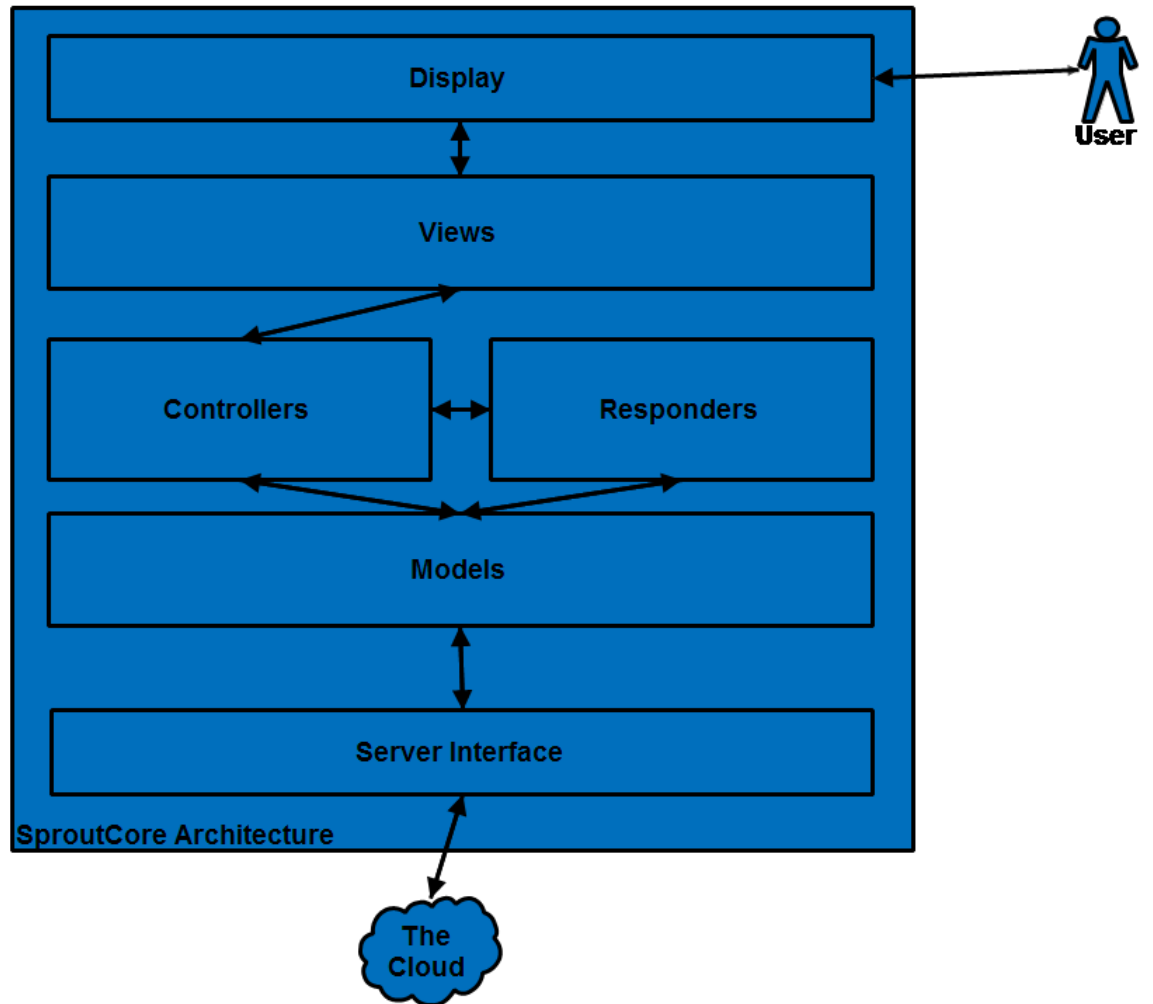
#### 4.2.3.1.1. MVC+SDR

Aunque MVC ha sido un patrón de diseño muy acertado desde hace aproximadamente 20 años, SproutCore presenta algunas características para apoyar este modelo, SproutCore adiciona tres capas más a su aplicación: Server Interface, Display and Responders.

- Server Interface: es la capa que se encarga de movilizar los datos entre el servidor web y la capa de modelo de datos, opcionalmente usando el almacenamiento del lado del cliente para soportar el modo fuera de conexión
- Display: es la capa que pinta la interfaz de usuario. Incluye el navegador web y cualquier biblioteca DOM de bajo nivel así como jQuery o Prototype Javascript
- Responders: es la capa que controla el estado general de la aplicación. Aquí es donde se coloca el código de nivel superior que configura los modelos, vistas y controladores basado en el estado de la carga y otros factores.

La capa Display en lugar de pintar y cambiar el tamaño de los objetos en SproutCore crea vistas HTML que son representadas en el navegador web, de esta forma el navegador web es el encargado de redibujar, cambiar el tamaño e incluso a veces animar el contenido que se está mostrando. La figura que se muestra a continuación muestra las partes principales de la arquitectura de SproutCore:

Figura 13. **Arquitectura SproutCore**



Fuente: elaboración propia.

#### 4.2.3.2. Principales características

A continuación se presentan las principales características de SproutCore las cuales se describen por nivel de las capas y la forma en la cual cada una de ellas interactúa.

#### **4.2.3.2.1. Model Layer**

La mayor parte de la lógica del negocio reside en esta capa, acá se definen las clases que representan los tipos de datos con los que se trabajará tales como contactos, eventos y fotografías entre otras cosas. Cuando se ejecuta la aplicación va a crear instancias de esas clases para gestionar los datos reales en la aplicación. Estos objetos del modelo pueden tener relaciones de uno a otro por ejemplo un evento puede tener asistentes la cual sería una matriz de contactos los cuales interactúan entre sí, estas clases y sus relaciones se llaman esquemas.

El esquema se encargará automáticamente de los datos JSON (JavaScript Object Notation) que envía y recibe desde el server de objetos de JavaScript que se puede utilizar para manipular el contenido. El almacenamiento de datos también gestionará el ciclo de vida de estos objetos controlar por ejemplo cuando son creados, modificados y eliminados.

Permite agregar al modelo de clases los eventos requeridos de acuerdo a las necesidades de la lógica del negocio por ejemplo si se desea enviar invitaciones a los invitados de un evento, se puede crear el método `enviarInvitacion()`, el cual tendrá las características creadas por el usuario. La importancia de este modelo es que debe ser capaz de ejecutarse aún si no se estuviera en el navegador web.

#### **4.2.3.2.2. View Layer**

Controla la presentación de una solicitud y es la primera capa en responder a los eventos del usuario tales como un clic, toques en la pantalla o pulsaciones de teclas. Generalmente posee propiedades las cuales controlan

la apariencia o reflejan el estado actual de la aplicación. Cuando se cambian estas propiedades la vista cambiará automáticamente y la pantalla reflejará el cambio. Asimismo, cada vez que se haga clic, o que se toque una parte de la pantalla, o se presione una tecla una vista obtendrá este evento y a la vez actualizará alguna propiedad y llamará algún método encargado de realizar la solicitud.

SproutCore viene con un conjunto bastante completo de vistas incorporadas, aunque el número de vistas en comparación con otros *frameworks* es pequeño son vistas flexibles y a menudo se pueden adaptar a cualquier necesidad.

#### **4.2.3.2.3. Controller Layer**

Los controladores conectan los modelos y las vistas, en SproutCode los controladores dependen en gran medida de las propiedades de los dispositivos los enlaces conectan una propiedad de un objeto a otra propiedad de otro objeto, se pueden escribir métodos que pueden modificar las propiedades de otros controladores y actualizar de esta forma las vistas. Este diseño permite que evolucione la capa de las vistas sin tener que volver a reescribir gran parte de su código.

#### **4.2.3.2.4. Respond Layer**

Controla el estado general de la aplicación, por ejemplo la aplicación puede mostrar un tipo de interfaz de usuario cuando el usuario aún se encuentra cargando sus datos, otra cuando el usuario se encuentre navegando para ver estos, y otra cuando el usuario desee modificar los datos. En esta capa



es donde se implementa el código para las transacciones en estos estados, es decir detectar que acciones puede realizar el usuario en cada estado.

### **4.2.3.3. Dispositivos y navegadores soportados**

La lista de dispositivos y navegadores con los cuales SproutCore trabaja no es tan amplia en comparación con otros *frameworks*, pero a continuación se presenta el listado de compatibilidad.

#### **4.2.3.3.1. Dispositivos soportados**

- iPhone OS
- Android OS
- Palm Pre

#### **4.2.3.3.2. Navegadores soportados**

- Firefox
- Internet Explorer
- Opera
- Safari
- Mobile Safari
- Opera Mini

### **4.2.3.4. Licenciamiento**

SproutCore maneja el tipo de licencia MIT, esta licencia otorga el permiso de obtener una copia gratuitamente del software y los archivos de documentación asociados al mismo, incluyendo los derechos para modificar,

copiar, distribuir, sub licenciar o inclusive vender el software siempre bajo los siguientes términos:

Se deja claro que SproutCore no se hace cargo de cualquier demanda o reclamo de ningún tipo, tampoco se hace cargo de ninguna infracción que sea llevada sobre el producto final para el cual haya sido utilizado el código de SproutCore.

#### **4.3. ¿Por qué usar ADF Mobile, frente a otros *frameworks*?**

En esta sección se presenta las características que hacen a ADF Mobile la mejor opción frente a otros *frameworks*, a continuación se resume los beneficios tecnológicos de usar ADF mobile, para posteriormente detallar cada uno de ellos.

- El *framework* aprovecha los beneficios de la máquina virtual de Java para permitir a los desarrolladores construir aplicaciones móviles para dispositivos inteligentes.
- El *framework* a su vez ofrece un conjunto de más de 50 componentes los cuales son utilizados para definir interfaces de usuario que implementan HTML5 en los dispositivos.
- Además se incluye un control el cual incorpora una capa de seguridad de datos encriptados que utiliza la base de datos SQLite y la integración de las características del dispositivo (cámara, GPS, contactos, entre otros).

- Soporta notificaciones nativas para las aplicaciones que son capaces de registrarse y recibir notificaciones directamente al dispositivo desde los servidores.
- Soporte de credencialización ya que añade íconos de notificación a la aplicación.
- Vista de archivos, permite la visualización de varios archivos utilizando los visores nativos del dispositivo.
- Empaquetamiento de aplicaciones para reutilizar y distribuir los ISVs.
- Soporte para nuevos servicios SDK.
- Gama amplia de componentes de interfaz de usuario.
- Documentación estable y actualizada.

#### **4.3.1. Beneficios, frente a otros *frameworks***

Adicionalmente de los beneficios tecnológicos presentados anteriormente en la siguiente sección se detalla las bases de ADF Mobile las cuales lo hacen la mejor elección al momento de desarrollar aplicaciones híbridas.

##### **4.3.1.1. Desarrollo Integrado**

Una de las más convincentes razones por la cual trabajar con ADF Mobile es la capacidad que posee el *framework* para desarrollar aplicaciones para móviles y aprovechar cada funcionalidad que este posee. ADF Mobile provee la

fácil integración con las propiedades del dispositivo tales como la cámara, los servicios de localización base, aplicaciones que manejan los contactos y de esa misma forma con las bondades del dispositivo.

Todo esto es desarrollado desde la herramienta JDeveloper el cual es un IDE muy robusto que posee diferentes características. ADF Mobile puede ser fácilmente integrado también con una aplicación Web ADF desarrollada en Java Enterprise Edition.

#### **4.3.1.2. Uso de patrones de diseño**

ADF Mobile es una herramienta de tipo RAD (Rapid Application Developer) es decir enfocada al desarrollo ágil y rápido, por ello mismo extiende una gama de patrones de diseño los cuales presentan la solución a varios problemas que se repiten en el ámbito de la programación, estos patrones de diseño son soluciones exitosas a varios problemas comunes que se presentan día a día y el uso de estos patrones al momento de desarrollar representan una ventaja competitiva de ADF Mobile frente a otros *frameworks*.

#### **4.3.1.3. Se trabaja de forma declarativa**

Las arquitecturas que se encuentran basadas en Java presentan entornos de trabajo basados en componentes robustos la cual permite trabajar con menos decodificación ya que se trabaja de forma declarativa, pero ¿Qué son los lenguajes declarativos?, son los lenguajes en los cuales se le indica a la máquina que es lo que se desea hacer de una forma fácil y rápida sin necesidad de escribir grandes porciones de código, ya que ADF Mobile contiene sentencias y estructuras que proveen esta facilidad en el entorno de desarrollo.

#### **4.3.1.4. Seguridad**

ADF Mobile está integrado con la seguridad del negocio y la infraestructura del mismo por lo cual provee servicios de autenticación y acceso a datos. Toda la información el almacenamiento de credenciales y la comunicación entre los canales se encuentra cifrado. Esto provee mayor seguridad a los usuarios en cuanto a ataques y pérdida de información o filtración de la misma.

#### **4.3.1.5. Fácil manejo de Java**

ADF Mobile permite reutilizar herramientas existentes desarrolladas en Java y basadas en tecnología Web, ventaja es que los desarrolladores no deben aprender otro lenguaje de programación o el uso de otras APIs, ya que se pueden crearse clases Java para ampliar la funcionalidad y capacidades del ADF Mobile dependiendo de la lógica del negocio en cada caso.

ADF Mobile posee además una pequeña máquina virtual de java (JVM por sus siglas en inglés) incrustada dentro de la arquitectura, por lo cual permite cargar y ejecutar verdadero código Java, sin necesidad de ninguna transcripción.

#### **4.4. Cuadro comparativo de *frameworks* móviles híbridos**

Luego de exponer las principales características y arquitectura de los *frameworks* híbridos Sencha Touch, jQuery Mobile, SproutCore y ADF Mobile se puede sintetizar esta información en el siguiente cuadro comparativo.

Tabla III. **Cuadro comparativo de *frameworks* móviles híbridos**

<b>Característica</b>	<b>Sencha Touch</b>	<b>jQuery Mobile</b>	<b>Sprout Core</b>	<b>ADF Mobile</b>
<b>Arquitectura</b>	MVC + Store Layer	MVC	MVC +SDR	MVC
<b>Soporta iOS de Apple</b>	Si	Si	Si	Si
<b>Soporta Android OS de Google</b>	Si	Si	Si	Si
<b>Soporta Windows Phone de Microsoft</b>	Si	Si	No	No
<b>Otros OS móviles soportados</b>	BlackBerry OS	BlackBerry OS, Symbian, WebOS, MeeGO	Palm Pre	Ninguno
<b>Tipo de Licenciamiento</b>	GPLv3	GPL v2	MTI	Privativa
<b>Documentación</b>	Buena	Regular	Buena	Buena
<b>Desarrollo Integrado</b>	Bueno	Regular	Bueno	Bueno
<b>Soporte de Java</b>	No	No	No	Si
<b>Integración con PhoneGap</b>	Integrado	No integrado	Integrado	Integrado
<b>Integración con SOA</b>	Integrado	No integrado	No Integrado	Integrado
<b>Forma de programación</b>	Imperativa	Imperativa	Imperativa	Declarativa
<b>Tiempo de Desarrollo</b>	Medio	Alto	Medio	Bajo

Fuente: elaboración propia.



## **5. PROPUESTA DE LA APLICACIÓN MOVIL EN EL MERCADO PARA PEQUEÑAS Y MEDIANAS EMPRESAS DISTRIBUIDORAS**

### **5.1. Definición del problema**

En la actualidad existen diversas empresas distribuidoras las cuales se poseen dificultades en el control de sus procesos de distribución, debido a que no tienen un sistema de monitoreo establecido. Esta situación puede ocasionar diferentes fallas en el proceso de distribución, pudiendo ser fallas operativas o deficiencia en el proceso de entrega, como consecuencia estos problemas llevan a un derroche de recursos innecesario.

Uno de los factores es el proceso no estandarizado de distribución el cual en algunos casos las pequeñas y medianas empresas no poseen dicho proceso, por esta razón envían a sus repartidores únicamente con direcciones de los lugares donde deben entregar la mercadería o el servicio sin tomar en cuenta la ruta de distribución, el tráfico, etc.

Otro factor que influye es la falta de recursos para implementar el monitoreo a través de medios tradicionales como lo es GPS, programas de logística, *handhelds*, etc. Por esta razón no implementan ningún medio de monitoreo, debido a que es más viable tener una pérdida de capital en los factores indicados anteriormente que en invertir en estos medios de monitoreo.



Otro problema que causa una pérdida de recursos de operación es el inventario; en muchas grandes empresas tienen solventado este problema por medio de *handheld* así como el cobro por medio de POS.

En las grandes empresas tiene programas para el cálculo de ruta crítica, y para gestionar todos los puntos donde se debe repartir la mercadería, programas como RoadShow son alimentados con la información brindada por la preventa que fue recolectada por las *handheld* y así este programa brinda una salida, dicha salida es la ruta que el repartidor o mensajero debe seguir para poder optimizar recursos.,

Si no se posee un programa de cálculo de ruta crítica, el repartidor podrá disponer de su experiencia para realizar la distribución tomando, la ruta que el considere más óptima, al final la ruta es elegida a la discreción del conductor, que en la mayoría de los casos será la ruta no óptima, un problema el cual es difícil de gestionar si no se posee dicho control del vehículo, al tener una ruta crítica se puede gestionar por medio de alertas, para validar que el repartidor está cubriendo la zona que le corresponde.

Uno de los factores que influyen en el consumo de tiempo y combustible, en la actualidad es el factor de saturación de vehículos en ciertas zonas de la ciudad así como áreas rurales. Este factor es influenciado por horas pico, las cuales se tiene previsto cierta saturación de vehículos en zonas específicas.

El territorio guatemalteco lastimosamente posee Zonas Rojas, las cuales son llamadas así, porque en ellas habitan delincuentes, al pasar un vehículo desconocido por dichas zonas puede estar expuesto a robos de mercadería, o del vehículo de transporte como tal.

En resumen se tienen identificado los siguientes problemas:

- Proceso de distribución no estándar
- Problema de falta de recursos
- Problema de inventarios
- Problema cálculo de ruta crítica
- Problema alertas fuera de la ruta crítica
- Problema de tráfico
- Problema de Zonas Rojas en la capital

En el presente capítulo se presenta una solución a la problemática descrita anteriormente, se presenta un bosquejo de aplicación móvil el cual puede ser desarrollado por medio de una aplicación híbrida e implementado en un teléfono inteligente.

Pero ¿Cuál es la ventaja de la implementación en un teléfono inteligente? y ¿por qué esta propuesta para la pequeña y mediana empresa?, la solución es bastante sencilla, ya que en comparativa a una *handheld* o POS un teléfono inteligente presenta un costo de adquisición menor y en comparativa de estas, otras soluciones representa una ventaja para estas empresas ya que pueden ser adquiridos de una forma más fácil y a través de la implementación de una aplicación híbrida, esto requeriría solamente el desarrollo de una aplicación no importando si el teléfono es plataforma Android o iOS.

## **5.2. Análisis de la aplicación**

En esta sección se presenta un análisis de una aplicación móvil, la cual permita solventar los problemas más fuertes que se presentaron anteriormente,

con el fin de mostrar los beneficios para las pequeñas y medianas empresas distribuidoras.

En la actualidad existen diferentes aplicaciones móviles con las cuales se puede tener el posicionamiento de un vehículo en tiempo real a través del teléfono móvil, estas aplicaciones capturan la posición que envía el *smartphone* a través del GPS que tiene incorporado en el dispositivo, cabe mencionar que en algunos dispositivos pueden tener una incertidumbre mínima, de dos o tres metros sobre la posición real.

A continuación se presenta el análisis de la aplicación iniciando por los requerimientos, seguidamente los casos de uso, los módulos propuestos para la solución del problema y la tecnología propuesta en base a la investigación realizada con anterioridad.

### **5.2.1. Requerimientos de la aplicación**

Los requerimientos planteados a continuación son requerimientos básicos para cubrir las necesidades de monitoreo en las pequeñas y medianas empresas distribuidoras. La aplicación deberá cumplir con los problemas básicos presentados anteriormente, con el fin de poder tener monitoreo tanto del producto que se está movilizandoo como el medio de transporte que se utiliza.

1. Requerimiento #1: la aplicación deber tener un mapa para poder ir verificando e ingresando los puntos en la preventa con el fin de poder posteriormente analizar la ruta crítica de los nuevos clientes y clientes potenciales.

2. Requerimiento #2: poder llevar el control del vehículo en tiempo real, por medio del GPS que está incorporado en el *smartphone*.
3. Requerimiento#3: tener un panel de control, con el cual se pueda monitorear diferentes aspecto como:
  - a. Tiempo promedio de la ruta a recorrer
  - b. Velocidad
  - c. Clientes prioritarios
  - d. Producto o servicio a distribuir
  - e. Alertas al salir de la ruta cierta distancia
  - f. Verificar zonas rojas
4. Requerimiento #4: llevar el control del inventario a través de la aplicación, cuando se entregue un pedido se descontara automáticamente del inventario que se lleva del camión para posteriormente ser sincronizado con el inventario central.
5. Requerimiento #5: permitir el ingreso de rutas críticas para evitar dichas rutas en el cálculo de la ruta crítica.
6. Requerimiento #6: permitir el ingreso de horas picos por zonas geográficas, para poder evitar transitar por estas zonas cuando se realizara la distribución del producto o servicio.

### 5.2.2. Casos de uso

A continuación se presentan las descripciones de los casos de uso que tienen como finalidad describir los eventos de un actor (el agente externo) que utiliza de la aplicación, y poder mostrar de forma gráfica los requerimientos presentados en la sección anterior.

Tabla IV. **Caso de uso–ingresar puntos de preventa**

<b>Caso de uso</b>	<b>Ingresar puntos de preventa</b>
<b>Actores</b>	Prevendedor
<b>Descripción</b>	El personal encargado de realizar la preventa registra los puntos nuevos de venta, o el cliente potencial donde se deberá entregar producto de muestra. Dichos puntos serán ingresados con las coordenadas de GPS

Fuente: elaboración propia.

Tabla V. **Caso de uso–mostrar ruta de distribución**

<b>Caso de Uso</b>	<b>Mostrar ruta de distribución</b>
<b>Actores</b>	Repartidor
<b>Descripción</b>	El repartidor podrá visualizar la ruta de distribución, esta llevará los puntos prioritarios que debe cubrir. También deberá indicar el producto o servicio que debe entregar.

Fuente: elaboración propia.

Tabla VI. **Caso de uso–ingresar ruta crítica**

<b>Caso de Uso</b>	<b>Ingresar ruta crítica</b>
<b>Actores</b>	Logística
<b>Descripción</b>	Logística deberá ingresar la ruta crítica, por medio de un archivo, indicando las coordenadas que deberán ser cubiertas, y la ruta que se deberá tomar.

Fuente: elaboración propia.

Tabla VII. **Caso de uso–ingresar/extraer inventario**

<b>Caso de Uso</b>	<b>Ingresar/Extraer inventario</b>
<b>Actores</b>	Operaciones
<b>Descripción</b>	Operaciones podrá ingresar el inventario antes que se realice la distribución, y extraer el mismo luego de finalizar.

Fuente: elaboración propia.

Tabla VIII. **Caso de uso–ingresar horas pico**

<b>Caso de Uso</b>	<b>Ingresar horas pico</b>
<b>Actores</b>	Logística
<b>Descripción</b>	El ingreso de horas pico será por medio de zonas geográficas de manera de identificar calles y/o avenidas con problemas de atascamiento de vehículos

Fuente: elaboración propia.

Tabla IX. **Caso de uso–ingresar zonas roja**

<b>Caso de Uso</b>	<b>Ingresar zonas roja</b>
<b>Actores</b>	Logística
<b>Descripción</b>	Logística podrá ingresar las zonas rojas por medio de las coordenadas de GPS, estos serán sectores marcados en el mapa, los cuales se deberá evitar el paso de mercadería por dichos puntos.

Fuente: elaboración propia.



Tabla X. **Caso de uso–consultar panel de control**

<b>Caso de Uso</b>	<b>Consultar panel de control</b>
<b>Actores</b>	Logística, repartidor
<b>Descripción</b>	El usuario ingresa al panel de control y verifica los tiempos promedios de distribución, alertas para verificar si el repartidor se encuentra fuera de la ruta crítica, o ha estado estacionado cierto tiempo en un mismo lugar.

Fuente: elaboración propia.

### **5.2.3. Módulos propuestos para la solución**

Los módulos propuestos son para una fácil abstracción de la aplicación y poder visualizar como estará compuesta, pero dependiendo de las necesidades de cada empresa específica y de la lógica propia de cada negocio se pueden realizar algunas modificaciones a esta estructura presentada.

#### **5.2.3.1. Módulo Mapa**

Este módulo será el que gestionara el mapa, no se debe crear desde cero debido a que ya existen los mapas digitalizados, por ejemplo Google Maps que puede usarse libremente a través de su API.

Por medio de este módulo se gestionará todos los puntos que deberán ser señalados por medio del GPS, también este gestionará la ruta crítica que deberá ser seguida por el repartidor.

#### **5.2.3.2. Módulo ruta crítica**

Este módulo es el encargado de la lectura de la ruta crítica y los tiempos que esta implica, así como también las alertas que este debe registrar al momento de encontrarse algún umbral sobrepasado.

El módulo de ruta crítica, debe presentar sobre el trazo que debe seguir el repartidor así como las diferentes variantes, respecto a los factores tráfico, zona roja, etc.

#### **5.2.3.3. Módulo inventario**

El módulo inventario es el encargado de registrar el inventario inicial así como también el inventario final, este módulo debe permitir realizar el inventario automáticamente luego de una entrega. Este gestiona los diferentes tipos de costos por artículo y una breve descripción del mismo.

Desde este módulo se puede crear nuevos productos, modificarlos, asignarles ubicación en bodega, agregarles subcomponentes, ingreso de petición de nuevos productos por parte de los clientes. Básicamente se maneja toda la información de los productos en este módulo.

#### **5.2.3.4. Módulo puntos y zonas críticas**

El módulo puntos y zonas críticas es el encargado de gestionar dos factores, el tráfico en horas pico y las zonas rojas, este módulo permite el ingreso de horas pico especificando el sector, calle y/o avenida y el tiempo en que se prevé estará el tráfico.

El ingreso de zonas rojas será a través de coordenadas de GPS (latitud y longitud) también se podrá gestionar las zonas rojas con un rango establecido por el usuario, en donde se pueda visualizar los sectores con más riesgo, riesgo medio y bajo riesgo.

#### **5.2.3.5. Módulo de usuarios**

Como en toda aplicación que será utilizada por diferentes usuarios, se necesita la gestión de los mismos, como iniciar sesión, enviar reporte, verificar el estado del repartidor, etc.

Este módulo será el encargado de la gestión de usuario, con el fin de poder monitorear al usuario a través del *smartphone*, este módulo permite asociar nuevos vehículos al usuario, así como verificar los tiempos promedio que el usuario se tarda en realizar la distribución.

#### **5.2.4. Tecnología propuesta para la solución**

Para este tipo de necesidad se tiene que utilizar varias características nativas del teléfono *smartphone*, por esta razón únicamente se tiene a la elección entre dos tipos de aplicaciones nativa o híbrida.

Según la investigación realizada, las aplicaciones híbridas en cuestión de costos es más bajo su desarrollo que las aplicaciones nativas, y se logra abarcar muchos mercados en un solo desarrollo.

Como la pequeña y mediana empresa no dispone del presupuesto para adquirir las herramientas de monitoreo como GPS y *handheld*. Y dado que la tecnología móvil ha avanzado a tal punto de poder cubrir las necesidades de estas dos últimas herramientas, la tecnología a utilizar es aplicaciones móviles de tipo nativo.

Al desarrollar de forma nativa se evita tener que pagar por desarrollo sobre la plataforma especificativa únicamente se realiza el gasto sobre un proyecto.

El *framework* híbrido propuesto es ADF Mobile, debido a los beneficios tecnológicos que se validaron en el capítulo 4, de todos esos beneficios los que se enfocan con mayor fuerza a este tipo de aplicación de monitoreo son los siguientes:

- Seguridad: en la actualidad grandes empresas tiene agujeros de seguridad exponiendo la información de los usuarios públicamente, aspectos del negocio tales como clientes, localización de los clientes, rutas de distribución, costo del producto, productos distribuidos tienen gran valor para la empresa y en manos equivocadas podría impactar las ventas en el mercado. Por esta razón ADF Mobile es una opción viable debido a la seguridad que posee incorporada en el propio *framework*.
- Trabajo fuera de línea: ADF mobile permite realizar trabajo fuera de línea, guardando toda la información en una base de datos SQLite, en los

dispositivos *smartphone* no siempre se dispone de conexión a internet depende de las zonas geográficas por donde se está distribuyendo el producto o servicio, la ventaja es que esta funcionalidad ya está programada y lista para usarse, la implementación es transparente totalmente para el desarrollador que está implementando la solución.

- Programación declarativa: lo cual permite un desarrollo rápido de aplicaciones, debido a que la mayor parte de código ya está realizado y empaquetado en componentes que se pueden utilizar únicamente arrastrando y pegando a la solución.

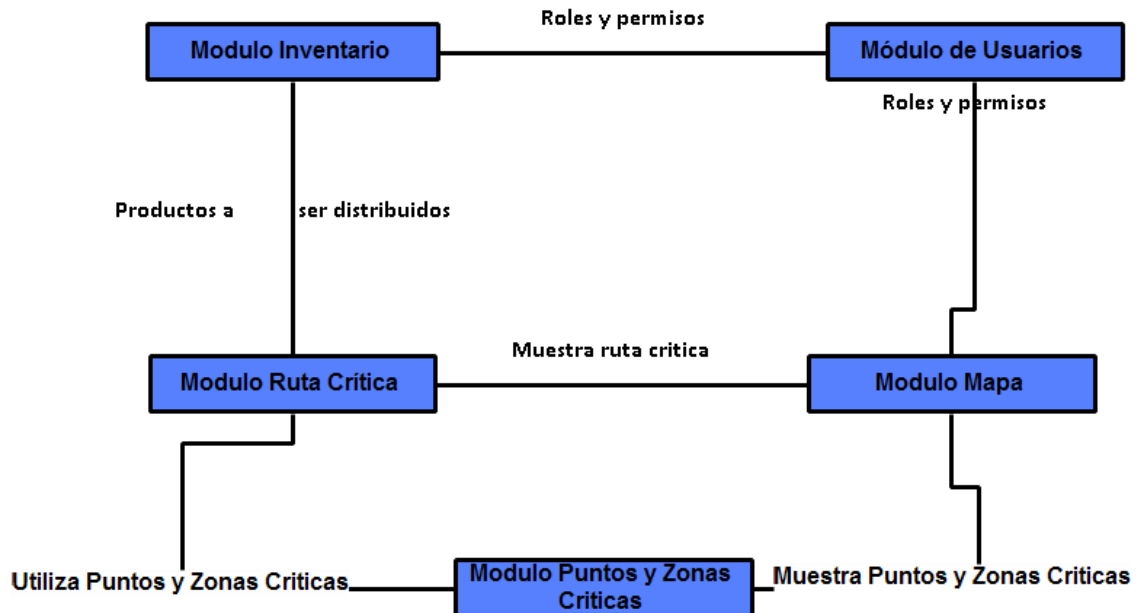
### **5.3. Diseño de la aplicación**

El diseño de la aplicación propuesta es la solución lógica de como el sistema satisface los requerimientos planteados en el análisis. El diseño presentado en esta sección se limita al modelo conceptual y la arquitectura propuesta para la solución.

#### **5.3.1. Modelo conceptual**

El diagrama conceptual es para visualizar gráficamente los conceptos de la aplicación, en este modelo se muestra la inter relación que tienen los módulos propuestos en la sección del análisis.

Figura 14. **Modelo conceptual de la aplicación de monitoreo**

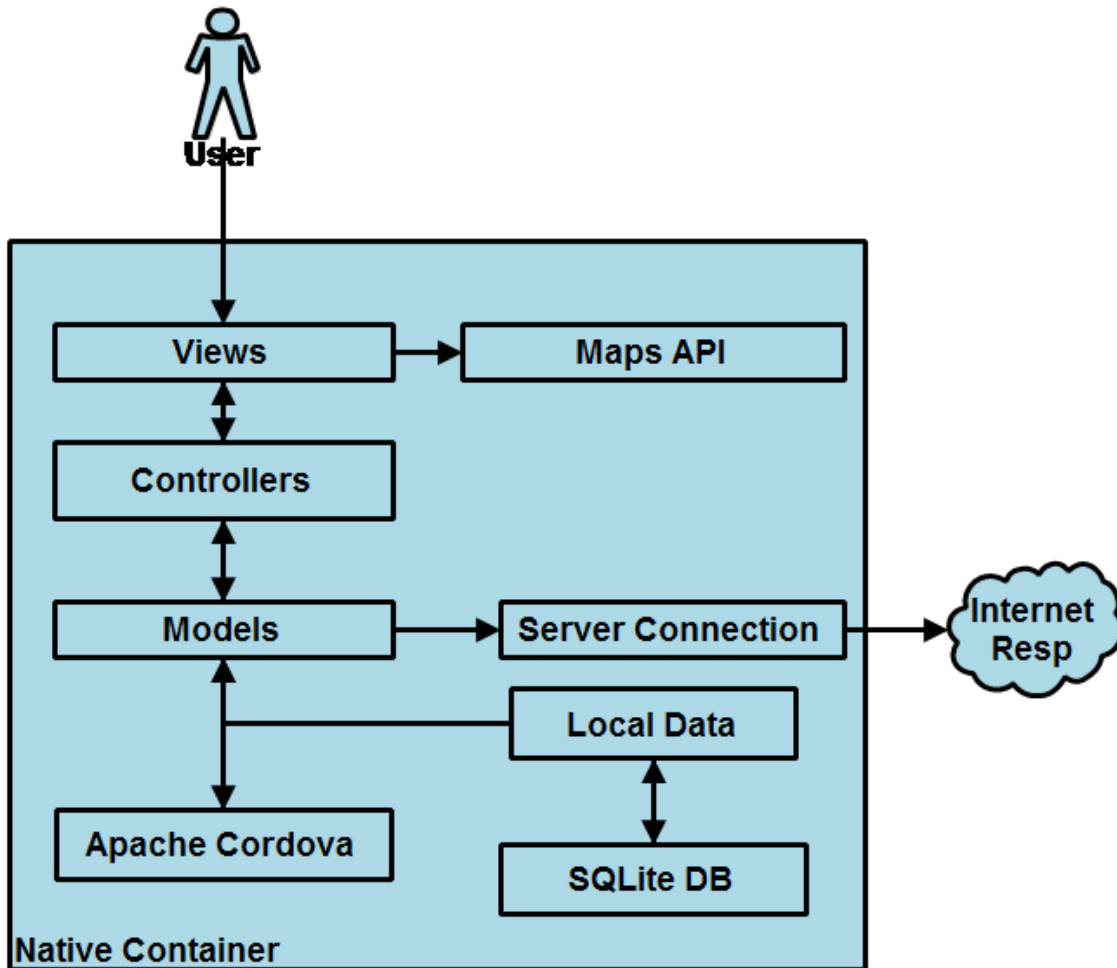


Fuente: elaboración propia.

### 5.3.2. Definición de la arquitectura de la aplicación

La arquitectura propuesta es la que se hereda de ADF Mobile, una arquitectura tradicional de un desarrollo web la cual es Modelo, Vista y Controlador (MVC).

Figura 15. **Arquitectura de la aplicación de monitoreo**



Fuente: elaboración propia.

#### 5.4. **Propuesta de valor**

Se propone una aplicación que cumpla con los requerimientos planteados anteriormente. Esta aplicación ayudará a solventar problemas operativos enfocados a la distribución de los productos y/o servicios.

Dicha aplicación traerá una optimización en la distribución de los productos por medio del seguimiento de una ruta óptima, con la cual no se desperdiciará recursos en recorrer distancias innecesarias.

#### **5.4.1. Beneficios tecnológicos aportados a la pequeña y mediana empresa distribuidora**

A continuación se presenta los beneficios tecnológicos aportados a la pequeña y media empresa:

- Optimizar recursos: se podrá optimizar los recursos tiempo, combustible, depreciación de vehículo.
- Monitoreo de los repartidores: se podrá verificar en donde se encuentra localizados los repartidores, y validar que se esté cumpliendo.
- Protección de sus activos: al tener un control de zonas rojas, se podrá evitar el paso por estos sectores, así como verificar las alertas del sistema, si algún dispositivo marca una ruta diferente a la que estaba establecida preguntar al repartidor, el porqué de esta situación.
- Inventario de productos al día: realizando el inventario automáticamente con cada entrega, se mantiene un inventario al día, sin problemas de llegar al cierre de mes.





## CONCLUSIONES

1. Se concluyó con la presente investigación que las aplicaciones móviles han tomado gran auge en el mercado en los últimos años y por esta razón, se propone la implementación de *smartphone* en pequeñas y medianas empresas distribuidoras, para que pueden aprovechar esta tecnología y lograr la reducción de costos operativos y de distribución.
2. Se presentó a las aplicaciones híbridas (las cuales permiten desarrollar un mismo código fuente para diferentes plataformas por medio de HTML5, CSS3 y JavaScript) para el uso de características nativas del teléfono *smartphone*, lo cual representa un costo de desarrollo menor frente a las aplicaciones nativas.
3. Se logró presentar como el tiempo de distribución se puede optimizar por medio del uso de la ruta crítica en las grandes empresas, se exhibió como estas empresas utilizan programas de logística para poder calcular esta ruta y por lo tanto se propuso utilizar una aplicación móvil por medio de la cual se puede visualizar esta ruta crítica.
4. Utilizando el GPS implementado en los *smartphone* y una conexión a internet se puede monitorear los diferentes vehículos incorporados en el proceso de distribución de las pequeñas y medianas empresas y lograr un mejor control de este proceso.

5. Se especificó que los teléfonos *smartphone* en la actualidad permiten el desarrollo de aplicaciones de terceros por medio del API que es proporcionado por diferentes plataformas en el mercado, lo cual hace aplicaciones versátiles tanto a nivel del dispositivo móvil y aplicaciones hechas a la medida de acuerdo a las necesidades de la empresa o de usuario, de esta forma se presentó una alternativa frente a otras soluciones para el monitoreo y control de rutas en pequeñas y medianas empresas distribuidoras.

## RECOMENDACIONES

1. Las aplicaciones móviles híbridas se ajustan a un alto nivel a los requerimientos de las aplicaciones empresariales de distribución debido a que los gráficos pueden ser catalogados a un nivel aceptable, sin embargo, si se desea realizar una aplicación que requiera un alto contenido gráfico como el desarrollo de un juego (por ejemplo que incluya ambientes en tres dimensiones extensos), se recomienda realizar una prueba de rendimiento para validar si la aplicación cubriría con los tiempos y utilización del hardware adecuados.
2. A pequeñas y medianas empresas distribuidoras que antes de comenzar cualquier desarrollo se estandaricé a medida de lo posible los procesos internos de la empresa, para garantizar el éxito de la implementación.
3. Antes de iniciar el desarrollo de la herramienta se recomienda analizar las rutas actuales, así como los tiempos medios de distribución de producto, para tener una métrica de comparación cuando se implemente la aplicación de monitoreo.



## BIBLIOGRAFÍA

1. ABDELSALAM A. HELAL, SUMI HELAL, RAJA BOSE, WENDONG LI. *PhoneGap Essentials: Building Cross-platform Mobile Apps*. EEUU: Morgan & Claypool Publishers, 2012. 122 p.
2. ADRIAN KOSMACZEWSKI. *Sencha Touch 2 Up and Running*. EEUU: O'Reilly Media, Inc., 2013. 259 p.
3. CACIÁ RIVAS, Daniel. *Arquitectura de aplicaciones J2EE basadas en el patrón MVC utilizando Oracle ADF*. Trabajo de graduación de Ing. en Ciencias y Sistemas. Universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2007. 188 p.
4. Charles Jolley. *SproutCore Documentation*. [en línea]. Basics- Architecture Overview. [http://wiki.sproutcore.com/w/page/12412844/Basics- Architecture%20Overview](http://wiki.sproutcore.com/w/page/12412844/Basics-Architecture%20Overview). [Consulta: agosto de 2014.]
5. Craig Larman. *UML y patrones: introducción al análisis y diseño orientado a objetos*. México: Félix Varela, 2004. 486 p.
6. ING. JUAN MANUEL TOLOZA. *Algoritmos y técnicas de tiempo real para el incremento de la precisión posicional relativa usando receptores GPS estándar*. Trabajo de graduación de Doctorado en Ciencias Informáticas. Universidad Nacional de La Plata, Argentina, Facultad de Informática, 2012. 213 p.

7. JOHN M. WARGO. *Mobile Platforms and Development Environments*. EEUU: Addison-Wesley Professional, 2012. 359 p.
8. MAXIMILIANO FIRTMAN. *JQuery Mobile: Up and Running*. EEUU: O'Reilly Media, Inc., 2012. 254 p.
9. Oracle and/or its affiliates. *Developing Applications with ADF Mobile*. [en línea]. ADF Academy. [http://download.oracle.com/otn\\_hosted\\_doc/jdeveloper/academy/Developing%20Applications%20with%20ADF%20Mobile/player.html](http://download.oracle.com/otn_hosted_doc/jdeveloper/academy/Developing%20Applications%20with%20ADF%20Mobile/player.html). [Consulta: junio de 2014.]
10. Oracle and/or its affiliates. *Oracle® Fusion Middleware Mobile Developer's Guide for Oracle Application Development Framework 11g Release 2 (11.1.2.3.0)*. [en línea]. Introduction to ADF Mobile. [http://docs.oracle.com/cd/E35521\\_01/doc.111230/e24475/intro.htm](http://docs.oracle.com/cd/E35521_01/doc.111230/e24475/intro.htm). [Consulta: agosto de 2014.]
11. Sam Johnson. *SproutCore Documentation*. [en línea]. Basics-Introducing SproutCore MVC. <http://wiki.sproutcore.com/w/page/12412848/Basics-Introducing%20SproutCore%20MVC>. [Consulta: agosto de 2014.]
12. TYLER KEATING. *SproutCore Web Application Development*. EEUU: Packt Publishing Ltd, 2013. 194 p.

13. VAIDES VELA, Víctor Danilo. *Implementación de un sistema de distribución denominado preventa, en una distribuidora de bebidas gaseosas*. Trabajo de graduación de Ing. Mecánico Industrial. Universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2006. 53 p.



