

# UNIVERSIDAD TÉCNICA DEL NORTE



**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN SISTEMAS  
COMPUTACIONALES**

**TRABAJO DE GRADO PREVIO LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN SISTEMAS COMPUTACIONALES**

## **TEMA**

**“ESTUDIO DEL FRAMEWORK DE DESARROLLO  
MULTIPLATAFORMA TITANIUM APPCELERATOR.**

## **APLICATIVO**

**PROTOTIPO DE APLICACIÓN MÓVIL PARA EL CONTROL DE  
ÓRDENES EN RESTAURANTES.”**

## **AUTOR**

**IVÁN JAVIER GONZÁLEZ PLACENCIA**

## **DIRECTOR**

**ING. MAURICIO XAVIER REA**

**IBARRA – ECUADOR**

**2015**





# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN

#### A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	1712873890		
<b>APELLIDOS Y NOMBRES:</b>	GONZÁLEZ PLACENCIA IVÁN JAVIER		
<b>DIRECCIÓN:</b>	HERNÁN GONZÁLEZ DE SAA 24145 Y CORI CORI		
<b>EMAIL:</b>	ivan_gonza@hotmail.com		
<b>TELÉFONO FIJO:</b>	062650069	<b>TELÉFONO MÓVIL:</b>	0979375063

DATOS DE LA OBRA	
<b>TÍTULO:</b>	Estudio del Framework de Desarrollo Multiplataforma Titanium Appcelerator Aplicativo.- Prototipo de Aplicación Móvil para el control de órdenes en restaurantes.”
<b>AUTOR (ES):</b>	Iván Javier González Placencia
<b>FECHA: AAAAMMDD</b>	2015/07/15
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniería en Sistemas Computacionales
<b>ASESOR /DIRECTOR:</b>	Ing. Mauricio Xavier Rea

#### 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, **Iván Javier González Placencia**, con cédula de identidad Nro. **1712873890**, en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la

Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

### 3. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 15 días del mes de Julio de 2015

**EL AUTOR:**



(Firma).....

Nombre: Iván Javier González Placencia



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO**  
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, **Iván Javier González Placencia**, con cédula de identidad Nro. **1712873890**, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: **ESTUDIO DEL FRAMEWORK DE DESARROLLO MULTIPLATAFORMA TITANIUM APPCELERATOR** con el aplicativo **“PROTOTIPO DE APLICACIÓN MÓVIL PARA EL CONTROL DE ÓRDENES EN RESTAURANTES.”**, que ha sido desarrollado para optar por el título de: Ingeniería en Sistemas Computacionales en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 15 días del mes de Julio de 2015

(Firma) .....

Nombre: **Iván Javier González Placencia**

Cédula: 1712873890



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN**

Ibarra, 15 de julio de 2015

Certifico que el trabajo de grado **“ESTUDIO DEL FRAMEWORK DE DESARROLLO MULTIPLATAFORMA TITANIUM APPCELERATOR”** con el aplicativo **“PROTOTIPO DE APLICACIÓN MÓVIL PARA EL CONTROL DE ÓRDENES EN RESTAURANTES.”**, ha sido realizada en su totalidad por el Sr. Iván Javier González Placencia; portador de la cédula de identidad número 1712873890, bajo mi supervisión para lo cual firmo en constancia.

---

Ing. Mauricio Xavier Rea  
DIRECTOR



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CONSTANCIAS**

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asumo la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

En la ciudad de Ibarra, 15 julio de 2015

El autor,

A handwritten signature in blue ink, appearing to read "Iván Javier González Placencia", is written over a horizontal line.

Firma

Nombre: Iván Javier González Placencia

Cédula.: 1712873890



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**DECLARACIÓN**

Yo **Iván Javier González Placencia**, con cédula de identidad Nro. 1712873890, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte - Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

En la ciudad de Ibarra, 15 julio de 2015

El Autor,

---

Firma

Nombre: Iván Javier González Placencia

Cédula.: 1712873890



## **DEDICATORIA**

Con todo mi cariño y amor para la persona que me dio la vida y dedicó la suya a cuidarme y apoyarme en todas las metas y objetivos que me he planteado, por extenderme tu mano cuando me sentía desfallecer.

Para ti madrecita por siempre mi corazón y agradecimiento.

Iván

## **AGRADECIMIENTO**

Al cumplir un objetivo muy importante en mi vida, quiero expresar mi agradecimiento profundo y sincero a Dios por ser el guía de mi camino.

Mi gratitud especial al cuerpo docente de la Universidad Técnica del Norte por compartir sus conocimientos y experiencias, en especial a mi asesor, Ing. Mauricio Rea, quien más que un asesor un amigo, por el apoyo y compromiso mostrados durante desarrollo de este proyecto.

A Bypro's Asesores Informáticos, empresa en la que trabajo, por permitir el uso de equipos y herramientas que facilitaron el desarrollo de este trabajo, con una mención especial al Ing. Byron Padilla por su apoyo y colaboración desinteresada.

A todos mis familiares y amigos que de una u otra manera estuvieron presentes en mi vida en los momentos en que más los necesité.

A todos y cada uno mi eterno agradecimiento por contribuir en la culminación de este objetivo.

Iván

## RESUMEN

El presente proyecto nace de la necesidad de conocer el uso de nuevas herramientas multiplataforma para el desarrollo de aplicaciones móviles.

Una de estas herramientas es Appcelerator, que permite codificar aplicaciones móviles en un solo lenguaje de programación como es JavaScript y compilar esta codificación a aplicaciones nativas soportadas por los diferentes tipos de equipos móviles con plataformas IOs y Android que son las plataformas más comercializadas a nivel mundial, lo que se traduce en una reducción de los costos y tiempo que conlleva desarrollar una aplicación en los lenguajes naturales de cada plataforma móvil.

Los capítulos que han dado lineamiento al presente proyecto son:

**Capítulo I.** Que incluye una introducción de la problemática actual y se plantea los objetivos, alcance y justificación del proyecto.

**Capítulo II.** Que contempla el marco teórico, dando a conocer información general acerca de que es una aplicación móvil, herramientas disponibles para el desarrollo de aplicaciones móviles, también se explica sobre lenguajes de codificación web y bases de datos, y los formatos de comunicación entre un equipo móvil y una base de datos externa.

**Capítulo III.** Este capítulo está dedicado a explicar el uso de la herramienta Appcelerator, desde su instalación, hasta la generación de una aplicación móvil, topando aspectos de desarrollo como patrones de desarrollo, uso de bases de datos, uso de redes sociales y servicios de geo localización.

**Capítulo IV.** Aquí se describe la metodología Mobile-D, utilizada en el desarrollo del proyecto. Especificando detalladamente cada una de las etapas en las que se divide el desarrollo de una aplicación.

**Capítulo V.** Contempla el desarrollo de la aplicación. Enmarcada con la aplicación de la metodología utilizada, se describe todo el proceso desarrollo del aplicativo, desde el levantamiento de requisitos, configuración de herramientas a utilizar, desarrollo, pruebas y lanzamiento.

**Capítulo VI.** En este capítulo se dan a conocer conclusiones y recomendaciones a las que se ha llegado una vez finalizado el proyecto, incluye además la bibliografía utilizada y muestra un glosario de términos.

## SUMMARY

This project stems from the need to know the use of new tools for cross-platform mobile application development.

One of these tools is Appcelerator, which allows coding mobile applications in one programming language such as JavaScript and compile this encoding to native applications supported by the different types of mobile devices with IOS and Android platforms are the platforms most traded worldwide, resulting in a reduction of costs and time involved in developing a natural language application of each mobile platform.

The chapters have given this draft guideline are:

**Chapter I.** It includes an introduction to the current problems and the objectives, scope and rationale for the project is proposed.

**Chapter II.** Which provides the theoretical framework, presenting an overview of which is a mobile application development tools available for the mobile application, also explained about web coding languages and databases, and formats of communication between a mobile device and external database.

**Chapter III.** This chapter is devoted to explaining the use of Appcelerator tool, since its installation, to the generation of a mobile application, bumping aspects of development and development patterns, use of databases, use of social networks and services of geo localization.

**Chapter IV.** Here the Mobile-D methodology used in the project is described. Specifying in detail each of the stages in the development of an application is divided.

**Chapter V.** It includes the development of the application. Framed with the application of the methodology used, the entire application development process is described, since the lifting of requirements, configuration tools to use, development, testing and release.

**Chapter VI.** This chapter disclose findings and recommendations that the project has come once completed, also includes the bibliography and a glossary of terms shows.



## ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	¡ERROR! MARCADOR NO DEFINIDO.
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	¡ERROR! MARCADOR NO DEFINIDO.
CERTIFICACIÓN.....	IV
CONSTANCIAS .....	V
DECLARACIÓN .....	VI
DEDICATORIA.....	VII
AGRADECIMIENTO .....	VIII
RESUMEN.....	IX
SUMMARY .....	XI
ÍNDICE DE CONTENIDOS.....	XIII
ÍNDICE DE FIGURAS .....	XXV
ÍNDICE DE TABLAS.....	XXIX
<b>CAPÍTULO I .....</b>	<b>1</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1. PROBLEMA.....	2
1.1.1. Antecedentes .....	2
1.1.2. Situación Actual.....	3
1.1.3. Prospectiva.....	3
1.1.4. Problema .....	3
1.2. OBJETIVOS .....	3
1.2.1. Objetivo General.....	3
1.2.2. Objetivos Específicos .....	4

1.3.	ALCANCE.....	4
1.3.1.	De la investigación.....	4
1.3.2.	Del aplicativo .....	4
1.4.	JUSTIFICACIÓN .....	6
1.4.1.	De la investigación.....	6
1.4.2.	Del aplicativo .....	7
1.5.	CONTEXTO .....	7
<b>CAPÍTULO II.....</b>		<b>9</b>
<b>MARCO TEÓRICO.....</b>		<b>9</b>
2.1.	APLICACIONES MÓVILES.....	10
2.1.1.	Definición de APP .....	10
2.1.2.	Características de una APP .....	10
2.1.3.	Tipos de APPs.....	10
2.1.3.1.	Aplicaciones nativas.....	10
2.1.3.2.	Aplicaciones Web.....	11
2.1.3.3.	Aplicaciones Híbridas .....	11
2.1.4.	Comparación entre los tipos de APPS .....	12
2.1.4.1.	Multiplataforma.....	12
2.1.4.2.	Acceso a funcionalidades nativas.....	13
2.2.	HERRAMIENTAS DE DESARROLLO MÓVIL HIBRIDAS .....	13
2.2.1.	Alfa anywhere.....	14
2.2.2.	App Press .....	14
2.2.3.	AppArchitect.....	15
2.2.4.	Form.com.....	15
2.2.5.	IBuildApp .....	16
2.2.6.	QuickBase.....	16
2.2.7.	Salesforce1 .....	16



2.2.8.	ViziApps .....	17
2.2.9.	Mobile Chrome Development Kit .....	17
2.2.10.	PhoneGap.....	18
2.2.11.	Appcelerator.....	18
2.3.	LENGUAJES DE CODIFICACIÓN WEB .....	19
2.3.1.	Lenguaje de programación PHP .....	19
2.3.1.1.	Breve historia .....	19
2.3.1.2.	¿Qué es PHP? .....	20
2.3.1.3.	Mezclar PHP y HTML .....	21
2.3.2.	Lenguaje JavaScript.....	22
2.3.2.1.	Breve historia .....	22
2.3.2.2.	¿Qué es JavaScript?.....	23
2.4.	HERRAMIENTAS Y TECNOLOGÍAS VARIAS .....	24
2.4.1.	ScriptCase .....	24
2.4.2.	JSON.....	25
2.4.2.1.	Constitución de JSON .....	25
2.4.2.2.	Estructura JSON.....	26
2.4.2.3.	Soporte de JSON .....	29
2.5.	MYSQL .....	29
2.5.1.	Características .....	30
2.5.1.1.	Abstracción de la información .....	30
2.5.1.2.	Independencia.....	30
2.5.1.3.	Redundancia mínima.....	30
2.5.1.4.	Consistencia.....	30
2.5.1.5.	Seguridad.....	30
2.5.1.6.	Integridad.....	31
2.5.1.7.	Control de la concurrencia .....	31

2.5.1.8. Aprovecha la potencia de sistemas multiprocesador.....	31
2.5.2. Arquitectura .....	32
<b>CAPITULO III .....</b>	<b>33</b>
<b>ESTUDIO DE LA HERRAMIENTA TITANIUM APPCELERATOR .....</b>	<b>33</b>
3.1. HISTORIA Y DEFINICIÓN .....	34
3.1.1. Appcelerator Titanium: un servicio completo e integral .....	34
3.1.1.1. Creación de aplicaciones con Titanium.....	34
3.1.1.2. Pruebas automatizadas .....	35
3.1.1.3. Servicios en la nube.....	35
3.1.1.4. Analítica web.....	35
3.1.1.5. Formación y apoyo.....	36
3.1.2. Principales características de Appcelerator .....	36
3.1.3. Tipos de Aplicaciones que genera .....	37
3.2. INSTALACIÓN Y CONFIGURACIÓN.....	38
3.2.1. Pre requisitos.....	38
3.2.1.1. Memoria .....	38
3.2.1.2. Sistema Operativo .....	38
3.2.1.3. Software necesario .....	39
3.2.2. Instalación.....	40
3.2.3. Configuración .....	41
3.2.4. Tipos de Proyectos.....	43
3.2.4.1. Proyectos Alloy .....	44
3.2.4.2. Proyectos clásicos.....	45
3.2.5. Creación del Proyecto .....	46
3.2.6. Estructura de Archivos de los proyectos.....	47
3.2.6.1. Estructura de archivos de Proyectos Básicos .....	48
3.2.6.2. Proyectos Alloy .....	50

3.3.	API DE TITANIUM .....	52
3.3.1.	Titanium UI.....	52
3.3.1.1.	Vistas .....	52
3.3.1.2.	Controles .....	53
3.3.1.3.	Ventanas .....	53
3.3.1.4.	Optimización .....	53
3.3.1.5.	Contexto global y sub hilos .....	54
3.3.1.6.	Portabilidad .....	54
3.3.1.7.	Eventos .....	55
3.3.1.8.	Forma de utilización.....	55
3.3.1.9.	Principales controles.....	57
3.4.	USO DE BASE DE DATOS Y ARCHIVOS LOCALES .....	59
3.4.1.	Propiedades .....	59
3.4.1.1.	Lectura y escritura de propiedades .....	60
3.4.1.2.	Almacenar objetos JS como JSON en propiedades.....	61
3.4.2.	Base de datos .....	61
3.4.2.1.	Creación e instalación de bases de datos .....	63
3.4.2.2.	Manipulación de datos.....	64
3.4.3.	Sistema de Archivos .....	66
3.4.3.1.	Ubicación de almacenamiento.....	67
3.4.3.2.	Operaciones con archivos.....	68
3.5.	INTERACCIÓN CON DATOS REMOTOS .....	71
3.5.1.	Peticiones GET .....	72
3.5.2.	Peticiones POST .....	73
3.5.3.	Trabajar con datos JSON .....	73
3.5.4.	Trabajar con datos XML.....	75
3.6.	SERVICIOS DE LOCALIZACIÓN.....	77

3.6.1.	Consideraciones del desarrollo .....	78
3.6.1.1.	Consideraciones del desarrollo IOs .....	78
3.6.1.2.	Consideraciones del desarrollo Android .....	78
3.6.1.3.	Consideraciones del desarrollo Web Móvil .....	78
3.6.2.	El uso de los servicios de localización en la App .....	79
3.6.2.1.	Determinar si los servicios de localización están habilitados .....	79
3.6.2.2.	Configurar la precisión de los servicios .....	79
3.6.2.3.	Obtener la posición actual .....	83
3.6.2.4.	Monitorear continuamente la posición GPS.....	84
3.7.	EVALUACIÓN DE TITANIUM APPCELERATOR .....	85
3.7.1.	Criterios de la evaluación .....	85
3.7.2.	Funcionalidad.....	85
3.7.2.1.	Evaluación .....	85
3.7.3.	Fiabilidad .....	86
3.7.3.1.	Evaluación .....	87
3.7.4.	Usabilidad .....	90
3.7.4.1.	Evaluación .....	90
3.7.5.	Eficiencia .....	91
3.7.5.1.	Evaluación .....	91
3.7.6.	Mantenibilidad.....	92
3.7.6.1.	Evaluación .....	92
3.7.7.	Portabilidad.....	92
3.7.7.1.	Evaluación .....	93
3.8.	VENTAJAS Y DESVENTAJAS.....	93
3.8.1.	Ventajas .....	93
3.8.2.	Desventajas .....	94
	<b>CAPÍTULO IV.....</b>	<b>95</b>

<b>METODOLOGÍA DE DESARROLLO MOBILE-D .....</b>	<b>95</b>
4.1. HISTORIA Y DEFINICIÓN .....	96
4.1.1. Definición .....	96
4.1.2. Motivación .....	97
4.1.3. Principios básicos .....	97
4.2. FASES .....	98
4.2.1. Exploración .....	98
4.2.1.1. Motivación.....	99
4.2.1.2. Objetivos .....	99
4.2.1.3. Etapas .....	99
4.2.2. Inicialización.....	103
4.2.2.1. Motivación.....	103
4.2.2.2. Objetivos .....	103
4.2.2.3. Etapas y tareas .....	104
4.2.3. Producción .....	108
4.2.3.1. Motivación.....	108
4.2.3.2. Objetivos .....	108
4.2.3.3. Etapas y tareas .....	109
4.2.4. Estabilización.....	114
4.2.4.1. Motivación.....	114
4.2.4.2. Objetivos .....	114
4.2.4.3. Etapas y tareas .....	115
4.2.5. Pruebas del sistema .....	116
4.2.5.1. Motivación.....	116
4.2.5.2. Objetivos .....	117
4.2.5.3. Etapas y tareas .....	117
4.3. ELEMENTOS DE LA METODOLOGÍA .....	118

4.4.	JUSTIFICACIÓN .....	119
<b>CAPÍTULO V .....</b>		<b>121</b>
<b>DESARROLLO DEL APLICATIVO .....</b>		<b>121</b>
5.1.	EXPLORACIÓN .....	122
5.1.1.	Establecimiento de los grupos de interés .....	122
5.1.2.	Definición del alcance .....	122
5.1.2.1.	Colección de requisitos iniciales .....	122
5.1.2.2.	Planificación del proyecto inicial .....	123
5.1.2.3.	Definición de Usuarios .....	124
5.1.3.	Establecimiento del proyecto .....	125
5.1.3.1.	Selección del entorno .....	125
5.1.3.2.	Asignación del personal .....	125
5.1.3.3.	Definición de la Arquitectura de la Aplicación .....	126
5.2.	INICIALIZACIÓN .....	127
5.2.1.	Configuración del proyecto .....	127
5.2.1.1.	Configuración del Ambiente .....	127
5.2.1.2.	Formación.....	132
5.2.2.	Planificación inicial .....	132
5.2.2.1.	Análisis Inicial de requerimientos .....	132
5.2.2.2.	Planificación de Iteraciones.....	135
5.3.	PRODUCCIÓN .....	145
5.3.1.	Diagrama Relacional de Base de datos .....	147
5.3.2.	Producción de Iteraciones .....	149
5.3.2.1.	Diseño Menú Principal.....	149
5.3.2.2.	Pantalla de inicio .....	150
5.3.2.3.	Menú principal .....	151
5.3.2.4.	Manejo de usuarios.....	151

5.3.2.5. Manejo de Grupos .....	152
5.3.2.6. Asignación de Permisos a Grupos .....	154
5.3.2.7. Cambiar contraseña .....	154
5.3.2.8. Manejo de Lugares de Preparación .....	155
5.3.2.9. Manejo de categoría de productos .....	155
5.3.2.10. Manejo de Productos .....	156
5.3.2.11. Manejo de Ambientes .....	156
5.3.2.12. Manejo de Mesas .....	157
5.3.2.13. Asignación de Productos a Ambientes .....	158
5.3.2.14. Asignación de Usuarios a Ambientes .....	159
5.3.2.15. Asignación de Usuarios a Lugares de Preparación .....	160
5.3.2.16. Manejos de estados de Comandas .....	161
5.3.2.17. Reportes de Comandas .....	161
5.3.2.18. Elaboración de Servicios Web .....	162
5.3.2.19. Diseño de Aplicación Móvil .....	163
5.3.2.20. Inicio de Sesión .....	164
5.3.2.21. Lista de mesas .....	165
5.3.2.22. Presentación de la Mesa .....	166
5.3.2.23. Agregar Productos .....	167
5.3.2.24. Entregar productos .....	170
5.3.2.25. Preparación de Productos .....	171
5.4. ESTABILIZACIÓN .....	172
5.4.1. Día de planificación .....	172
5.4.2. Día de trabajo .....	172
5.4.3. Documentación .....	172
5.5. PRUEBAS DEL SISTEMA .....	172
5.5.1. Día de Planificación .....	172

5.5.2.	Día de trabajo .....	173
5.5.3.	Correcciones .....	173
5.5.4.	Análisis de Resultados Módulo de Administración.....	174
5.5.4.1.	Inicio de Sesión .....	174
5.5.4.2.	Definición Grupos .....	175
5.5.4.3.	Definición de Usuarios.....	175
5.5.4.4.	Asignación de permisos a Grupos .....	175
5.5.4.5.	Cambio de contraseña .....	176
5.5.4.6.	Lugares de preparación .....	176
5.5.4.7.	Categorías de productos .....	176
5.5.4.8.	Productos.....	177
5.5.4.9.	Ambientes.....	177
5.5.4.10.	Asignación de Productos a Ambientes.....	177
5.5.4.11.	Mesas.....	178
5.5.4.12.	Asignación de Meseros a Ambientes .....	178
5.5.4.13.	Finalizar comanda .....	178
5.5.4.14.	Resumen de comandas .....	179
5.5.5.	Análisis de Resultados Servicios Web.....	179
5.5.5.1.	Inicio de sesión.....	179
5.5.5.2.	Lista de Mesas y productos .....	179
5.5.5.3.	Agregar Comandas .....	180
5.5.5.4.	Listar Comandas.....	180
5.5.5.5.	Modificar Comanda.....	180
5.5.5.6.	Cambiar Comanda.....	181
5.5.6.	Análisis de Resultados aplicación móvil .....	181
5.5.6.1.	Iniciar sesión.....	181
5.5.6.2.	Ingreso de Comanda.....	181



5.5.6.3. Modificar comandas .....	182
5.5.6.4. Iniciar sesión.....	182
5.5.6.5. Listar Comandas.....	182
5.5.6.6. Cambiar Comanda.....	183
<b>CAPÍTULO VI.....</b>	<b>185</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>185</b>
6.1. CONCLUSIONES .....	186
6.2. RECOMENDACIONES.....	187
6.3. IMPACTOS .....	187
6.4. POSIBLES TEMAS PARA TRABAJOS DE GRADO.....	188
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>189</b>
<b>GLOSARIO.....</b>	<b>194</b>
<b>ANEXOS .....</b>	<b>200</b>



## ÍNDICE DE FIGURAS

Figura 1. Esquema de funcionamiento de la aplicación .....	5
Figura 2. Diferencias entre apps nativas, web apps y apps híbridadas. ....	12
Figura 3. Estructura JSON.....	26
Figura 4. Estructura de Arreglo JSON .....	27
Figura 5. Estructura de valor en JSON .....	27
Figura 6. Cadenas en JSON.....	28
Figura 7. Representación de números en JSON .....	28
Figura 8. Navegadores que soportan JSON.....	29
Figura 9. Arquitectura de MySQL.....	32
Figura 10. Descarga de Appcelerator Studio.....	41
Figura 11. Inicio de Sesión Appcelerator .....	42
Figura 12. Configuración para SDK Android .....	43
Figura 13. Tipos de Proyectos .....	44
Figura 14. Configuración inicial de un Proyecto.....	46
Figura 15. Estructura de archivos de Proyecto Básico .....	48
Figura 16. Estructura de Archivos de un Proyecto Alloy.....	50
Figura 17. Creación de Objetos de Titanium.....	55
Figura 18. Como agregar un control a su contenedor.....	56
Figura 19. Utilización de addEventListener .....	56
Figura 20. Creación de una Base de datos .....	63
Figura 21. Creación de una tabla .....	64
Figura 22. Instalación de una base de datos .....	64
Figura 23. Ingreso de datos en base de datos .....	65
Figura 24. Consulta de datos a una base de datos .....	65
Figura 25. Actualización de datos en una base de datos .....	66
Figura 26. Eliminación de datos en una base de datos .....	66
Figura 27. Obtención de identificador de archivo .....	68
Figura 28. Lectura de archivos .....	69
Figura 29. Escritura de archivos .....	69
Figura 30. Método para agregar datos a un archivo .....	70
Figura 31. Creación de un archivo.....	70
Figura 32. Método para copiar un archivo. ....	71

Figura 33. Esquema básico para HTTPClient .....	72
Figura 34. Envío de datos a través de POST .....	73
Figura 35. Recepción de datos en formato JSON.....	74
Figura 36. Envío de datos en formato JSON.....	74
Figura 37. Manejo de respuesta con formato XML. ....	75
Figura 38. Ejemplo de manejo de documento XML. ....	77
Figura 39. Determinar servicios de localización .....	79
Figura 40. Configuración de geo localización en IOs. ....	81
Figura 41. Configuración de geo localización en Android.....	82
Figura 42. Configuración de geo localización en web móvil.....	83
Figura 43. Obtención de la posición actual .....	84
Figura 44. Monitoreo continuo de la posición.....	84
Figura 45. Fases de desarrollo Mobile-D .....	98
Figura 46. Etapas de la Fase de Exploración.....	98
Figura 47. Etapas de la Fase de Inicialización.....	103
Figura 48. Etapas de la Fase de Producción .....	108
Figura 49. Etapas de la Fase de Estabilización.....	114
Figura 50. Etapas de la Fase de Pruebas del sistema.....	116
Figura 51. Arquitectura de la Aplicación .....	126
Figura 52. Instalación de MySql.....	128
Figura 53. Página de Inicio de ScriptCase.....	129
Figura 54. Entorno de trabajo Appcelerator .....	130
Figura 55. Pantalla Principal de Genymotion.....	131
Figura 56. Caso de Uso Roles o Grupos.....	135
Figura 57. Caso de Uso Usuarios .....	136
Figura 58. Caso de uso Permisos de Grupos .....	136
Figura 59. Caso de uso Cambio de Contraseña .....	137
Figura 60. Caso de uso Lugares de Preparación.....	138
Figura 61. Caso de uso Categoría de Productos .....	138
Figura 62. Caso de uso Productos. ....	139
Figura 63. Caso de Uso Ambientes .....	140
Figura 64. Caso de Uso Mesas .....	140
Figura 65. Caso de Uso Asignación productos a ambientes .....	141
Figura 66. Caso de Uso Asignación de usuarios a Ambientes .....	142

Figura 67. Caso de Uso Finalizar Comanda .....	142
Figura 68. Caso de Uso Recepción de Comanda .....	143
Figura 69. Caso de Uso Reportes .....	144
Figura 70. Caso de Uso Preparación Comanda .....	144
Figura 71. Diagrama Relacional de Base de Datos .....	148
Figura 72. Diseño Menú Principal.....	149
Figura 73. Pantalla de Inicio.....	150
Figura 74. Pantalla de Recuperación de Contraseña .....	150
Figura 75. Pantalla de Menú Principal .....	151
Figura 76. Pantalla de Manejo de Usuarios.....	151
Figura 77. Pantalla de Nuevo Usuario.....	152
Figura 78. Pantalla de Actualizar Usuario.....	152
Figura 79. Pantalla de manejo de Grupos.....	152
Figura 80. Pantalla de Nuevo Grupo .....	153
Figura 81. Pantalla de Actualizar Grupo .....	153
Figura 82. Pantalla de Seguridad de Grupos .....	154
Figura 83. Pantalla de Permisos de Grupos.....	154
Figura 84. Pantalla de cambio de contraseña .....	154
Figura 85. Pantalla de Lugares de Preparación .....	155
Figura 86. Pantalla de Manejo de Categorías de Productos .....	155
Figura 87. Pantalla de manejo de Productos.....	156
Figura 88. Pantalla de manejo de Ambientes .....	156
Figura 89. Pantalla de Manejo de Mesas.....	157
Figura 90. Pantalla de Selección de Ambiente.....	158
Figura 91. Pantalla de configuración de Lugar de Preparación de Productos.....	158
Figura 92. Pantalla de Selección de Usuario .....	159
Figura 93. Pantalla de Asignación de Ambientes a Usuario. ....	159
Figura 94. Pantalla de Selección de Usuario .....	160
Figura 95. Pantalla de Asignación de Lugares de Preparación a Usuario.....	160
Figura 96. Pantalla de Manejo de Comandas .....	161
Figura 97. Pantalla de Reporte de Comandas.....	161
Figura 98. Pantalla de filtros para Reportes. ....	161
Figura 99. Esquema de Servicios Web de la Aplicación Móvil.....	162
Figura 100. Diseño Aplicación de Lugares de Preparación .....	163

Figura 101. Diseño de Inicio de Sesión.....	164
Figura 102. Lista de Mesas.....	165
Figura 103. Presentación del Menú .....	166
Figura 104. Presentación de Categorías de productos.....	167
Figura 105. Presentación de productos.....	168
Figura 106. Agregar producto a pedido.....	169
Figura 107. Listado de productos por entregar.....	170
Figura 108. Lista de productos por preparar.....	171
Figura 109. Modelo para realización de Pruebas .....	173
Figura 110. Hoja de Control de Correcciones. ....	174

## ÍNDICE DE TABLAS

Tabla 1.	Ventajas y desventajas de PHP .....	21
Tabla 2.	Ventajas y Desventajas de JavaScript .....	24
Tabla 3.	Tipos de Aplicaciones y plataforma requerida.....	37
Tabla 4.	Requisitos de memoria Appcelerator Studio.....	38
Tabla 5.	Sistemas operativos compatibles con Appcelerator .....	38
Tabla 6.	JDK compatibles con Appcelerator.....	39
Tabla 7.	Compatibilidad de Node.js con Appcelerator .....	40
Tabla 8.	Descripción del sistema de archivos de un proyecto básico .....	49
Tabla 9.	Descripción de Estructura de Archivos Alloy.....	51
Tabla 10.	Características de Controles de Titanium.....	59
Tabla 11.	Métodos de Escritura y Lectura de Propiedades .....	60
Tabla 12.	Funciones utilizadas por tipo de respuesta de Servicios Web.....	73
Tabla 13.	Métodos para analizar documentos XML .....	76
Tabla 14.	Definición de usuarios.....	124
Tabla 15.	Asignación de personal .....	125
Tabla 16.	Elementos preparados para el desarrollo de la Solución.....	131
Tabla 17.	Análisis de Requisitos .....	134
Tabla 18.	Planificación de Iteraciones .....	145
Tabla 19.	Resultados de Iteración Inicio de Sesión.....	174
Tabla 20.	Resultados de Iteración Definición de Grupos.....	175
Tabla 21.	Resultados de Iteración Definición de Usuarios .....	175
Tabla 22.	Resultados de Iteración Asignación de Permisos a Grupos .....	175
Tabla 23.	Resultados de Iteración cambio de Contraseña.....	176
Tabla 24.	Resultados de Iteración Lugares de Preparación.....	176
Tabla 25.	Resultados de Iteración Categoría de Productos .....	176
Tabla 26.	Resultados de Iteración Productos .....	177
Tabla 27.	Resultados de Iteración Ambientes .....	177
Tabla 28.	Resultados de Iteración Asignación de Productos a Ambientes .....	177
Tabla 29.	Resultados de Iteración Mesas .....	178
Tabla 30.	Resultados de Iteración Asignación de Meseros a Ambientes.....	178
Tabla 31.	Resultados de Iteración Finalizar Comanda.....	178
Tabla 32.	Resultados de Iteración Resumen de Comandas.....	179

Tabla 33. Resultados de Iteración Inicio de Sesión.....	179
Tabla 34. Resultados de Iteración Lista de Mesas y Productos .....	179
Tabla 35. Resultados de Iteración Agregar Comandas .....	180
Tabla 36. Resultados de Iteración Listar Comandas .....	180
Tabla 37. Resultados de Iteración Modificar Comanda .....	180
Tabla 38. Resultados de Iteración Cambiar Comanda .....	181
Tabla 39. Resultados de Iteración Iniciar Sesión .....	181
Tabla 40. Resultados de Iteración Ingreso de Comanda .....	181
Tabla 41. Resultados de Iteración Modificar Comanda .....	182
Tabla 42. Resultados de Iteración Iniciar Sesión .....	182
Tabla 43. Resultados de Iteración Listar Comandas .....	182
Tabla 44. Resultados de Iteración Cambiar Comanda .....	183
Tabla 45. Análisis de Impactos .....	187



## CAPÍTULO I



### INTRODUCCIÓN

---

- Problema
  - Objetivos
  - Alcance
  - Justificación
  - Contexto
-

## 1.1. PROBLEMA

### 1.1.1. Antecedentes

En el mercado actual se ha proliferado la venta de Smartphones<sup>1</sup> y PDA (Personal Digital Assistant – Asistente Personal Digital), que dependiendo del modelo cuentan con su propio sistema operativo; los equipos con sistema operativo<sup>2</sup> Android<sup>3</sup> son los más vendidos en un 80% de la cuota mundial del mercado, seguido por IOs<sup>4</sup> en un 15%, dejando el resto para equipos con sistemas operativos como BlackBerry<sup>5</sup>, Symbian<sup>6</sup>, Windows Phone<sup>7</sup> y otros.

Cada uno de los sistemas operativos para equipos móviles cuenta con sus propios entornos de desarrollo y sus propios lenguajes de programación de aplicaciones, lo que dificulta el trabajo de los programadores, que deben conocer varios lenguajes de programación para el desarrollo de estas aplicaciones y estas a su vez estén disponibles en todos los tipos de Smartphones y Tabletas.

Por esta razón es necesario minimizar la dificultad en el proceso de programación y desarrollo de aplicaciones móviles, adicionalmente que estas aplicaciones sean portables a la mayoría de los sistemas operativos mencionados.

---

<sup>1</sup> Smartphone: Teléfono móvil inteligente

<sup>2</sup> Sistema Operativo – Operating System: Conjunto de órdenes y programas que controlan los procesos básicos de una computadora o dispositivo móvil y permiten el funcionamiento de otros programas.

<sup>3</sup> Android: sistema operativo para dispositivos móviles basado en Linux, de libre distribución.

<sup>4</sup> IOs: sistema operativo diseñado por Apple para sus productos, iPhone, iPad, iPod, etc.

<sup>5</sup> BlackBerry: sistema operativo diseñado por BlackBerry para sus productos.

<sup>6</sup> Symbian: sistema operativo diseñado por Nokia para sus productos.

<sup>7</sup> Windows Phone: sistema operativo diseñado por Microsoft.

### **1.1.2. Situación Actual**

La tecnología actual ha puesto al alcance de los desarrolladores un sinnúmero de entornos de desarrollo de aplicaciones móviles y web multiplataforma, para cumplir y satisfacer en el menor tiempo posible las necesidades del cliente. Muchos de estos entornos son poco conocidos en nuestro medio; no se conoce sus ventajas y desventajas, ni cuando utilizarlos, lo que conlleva a no utilizarlos.

### **1.1.3. Prospectiva**

Por lo tanto la investigación sobre el uso de nuevas herramientas de desarrollo de aplicaciones móviles es importante tanto para profesionales, estudiantes y docentes ya que estas herramientas de programación son versátiles y generan código para cualquier plataforma de Sistema Operativo.

### **1.1.4. Problema**

Hace falta la investigación sobre el uso de nuevas herramientas existentes para el desarrollo de aplicaciones móviles multiplataforma.

## **1.2. OBJETIVOS**

### **1.2.1. Objetivo General**

Analizar la herramienta de desarrollo Titanium Appcelerator para determinar las bondades, ventajas y desventajas que este presenta en el desarrollo de una aplicación móvil multiplataforma.

### **1.2.2. Objetivos Específicos**

- Conocer los requisitos necesarios para utilizar la herramienta de desarrollo Titanium Appcelerator.
- Determinar los requerimientos de los restaurantes para la implementación de la aplicación.
- Emplear el framework de desarrollo Titanium Appcelerator con consumo de servicios web JSON y base de datos MySQL en el desarrollo de una aplicación móvil.
- Desarrollar un prototipo de aplicación móvil para el control de órdenes en un restaurante.

## **1.3. ALCANCE**

### **1.3.1. De la investigación**

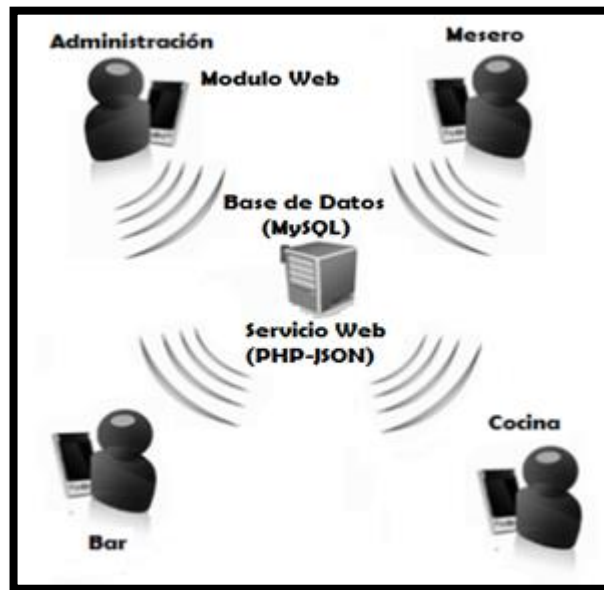
El presente estudio analiza una de las herramientas disponibles en la actualidad para el desarrollo de aplicaciones móviles y web; Titanium Appcelerator, en lo referente a sus características, bondades y desventajas al utilizarla en el desarrollo de una aplicación móvil que estará disponible para su utilización en diferentes equipos con sistemas operativos IOS y Android. Se dará a conocer la forma en cómo esta herramienta se comunica con bases de datos y ocupa servicios web, además de cómo se integra para el uso de servicios de geo localización.

### **1.3.2. Del aplicativo**

El aplicativo corresponde a una aplicación móvil para el control de órdenes de consumo en un restaurante que permitirá la sincronización de actividades entre los

meseros encargados de tomar la orden a los clientes y los cocineros encargados de preparar los platos solicitados.

Para el desarrollo de esta aplicación móvil se utilizara la metodología de desarrollo Mobile-D, que es una metodología enfocada en el desarrollo ágil de aplicaciones móviles.



*Fuente Propia*

*Figura 1. Esquema de funcionamiento de la aplicación*

En la figura 1 se muestran los procesos con los que contará el aplicativo y se detallan a continuación.

### **Servidor Web**

Para el almacenamiento de datos se utilizará un servidor web con una base de datos MySQL<sup>8</sup>, en cuanto a la comunicación con la aplicación móvil se utilizara el consumo de servicios web generados con PHP<sup>9</sup> y con formato JSON<sup>10</sup>.

<sup>8</sup> MySQL es un sistema de gestión de bases de datos relacional.

<sup>9</sup> PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico (ConceptoWeb, s.f.).

<sup>10</sup> JSON: acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos

## **Administración**

Este proceso consiste en un módulo web desde donde se realizará la configuración de la aplicación, ingresando información de las mesas con la que cuenta el restaurante, los usuarios con sus respectivos roles, los productos y precios que tiene disponible el establecimiento.

## **Mesero**

En este proceso se realizará la toma de órdenes de acuerdo a la mesa que ocupen los clientes y se enviará un mensaje a la cocina o bar dependiendo de la situación.

## **Cocina y Bar**

En este proceso se mostraran las órdenes realizadas por los clientes, con un criterio de primero en entrar primero en salir, para que se proceda a su elaboración, en el momento de terminar la elaboración de la orden, se notificara al Mesero para que proceda a la entrega de los productos solicitados.

## **1.4. JUSTIFICACIÓN**

### **1.4.1. De la investigación**

Con el estudio de este Framework se pretende generar el conocimiento necesario para la utilización de esta herramienta y poder desarrollar aplicaciones móviles multiplataforma; a partir del desarrollo en un único lenguaje como es JavaScript.

Esta herramienta facilita el trabajo de los desarrolladores de aplicaciones móvil y web, al no programar la misma aplicación en diferentes lenguajes de programación,

reduciendo el tiempo y los costos generados en la culminación de un proyecto, además, que al ser una herramienta de software libre no genera costos adicionales para su obtención.

#### **1.4.2. Del aplicativo**

En la actualidad, se puede decir que casi todos los restaurantes operan de una forma tradicional. Estos procesos tradicionales incluyen la toma de pedidos de una forma manual con el uso de papel y lápiz por parte de los meseros, quienes son encargados de coordinar con la cocina, bares y caja la preparación y posterior cobro de dichos pedidos; procesos que se pueden mejorar con el uso de tecnología.

La elaboración de esta aplicación pretende introducir el uso de la tecnología en este tipo de negocios para mejorar el proceso de comunicación entre las diferentes áreas que intervienen en las actividades cotidianas de un restaurante y por ende mejorar la calidad con la que se atiende a las personas utilizan sus servicios.

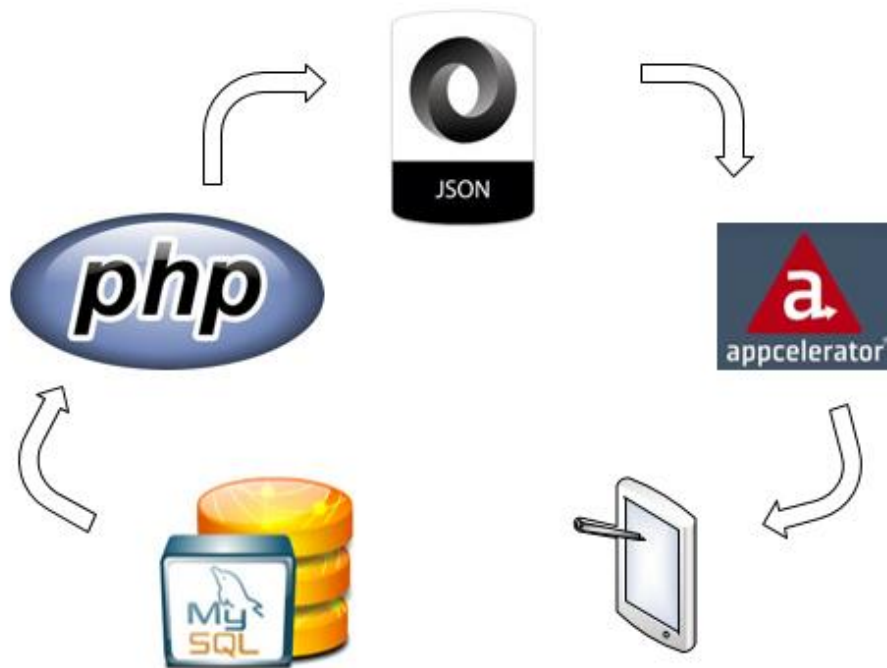
#### **1.5. CONTEXTO**

Luego de revisar el listado de proyectos realizados en la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte se puede determinar que no hay estudios similares.





## CAPÍTULO II



## MARCO TEÓRICO

- 
- Aplicaciones Móviles
  - Herramientas de desarrollo móvil híbridas
  - Lenguajes de Codificación Web
  - Herramientas y Tecnologías Varias
  - MySQL
-

## **2.1. APLICACIONES MÓVILES**

### **2.1.1. Definición de APP**

(Deusto Formación, 2014) El término App es una abreviatura de la palabra en inglés *application*, lo que significa que una APP no deja de ser un programa informático pero con unas características especiales, que afectan a la capacidad, rendimiento, diseño, experiencia de navegación, dispositivos utilizados, proceso de creación, programación, desarrollo y costes.

### **2.1.2. Características de una APP**

Las APPs o aplicaciones tienen unas características especiales y diferenciales, que las sitúan en un nivel superior a las páginas webs tradicionales tanto en enfoque, rendimiento y capacidad, como en funcionalidades.

Para hablar de las características de las APPs es necesario aclarar que no todas tienen las mismas características, ni son del mismo tipo.

### **2.1.3. Tipos de APPs**

Los tipos de aplicaciones móviles que se conocen son, nativas, web e híbridas.

#### **2.1.3.1. Aplicaciones nativas**

Una App nativa es una aplicación desarrollada bajo un lenguaje y entorno de desarrollo específico, lo cual permite, que su funcionamiento sea muy fluido y estable para el sistema operativo que fue creada. Estas APPs pueden acceder y aprovechar el funcionamiento de los sensores típicos de los dispositivos móviles como el geoposicionamiento, brújula, cámara, etc...

### 2.1.3.2. Aplicaciones Web

Las aplicaciones web son aquellas desarrolladas usando lenguajes para el desarrollo web como lo son HTML<sup>11</sup>, CSS<sup>12</sup> y JavaScript<sup>13</sup> y un framework para el desarrollo de aplicaciones web, como por ejemplo jQuery Mobile<sup>14</sup>, Sencha, Kendo UI, entre otros..

Se podría decir que este tipo de aplicaciones es muy usada para brindar accesibilidad a la información desde cualquier dispositivo, sin importar el sistema operativo, ya que solo se necesita contar con un navegador para acceder a esta (deideaaapp.org, 2015).

### 2.1.3.3. Aplicaciones Híbridas

(DeIdeaAapp, 2014) Las aplicaciones híbridas, como su nombre lo indica tienen un poco de cada tipo de las aplicaciones ya nombradas. Este tipo de aplicaciones se desarrolla utilizando lenguajes de desarrollo web y un framework dedicado para la creación de aplicaciones híbridas, como por ejemplo phonegap, Titanium Appcelerator, Steroids, entre otros. La facilidad que brinda este tipo de desarrollo es que no hay un entorno específico el cual hay que utilizar para su desarrollo y la mayoría de las herramientas son de uso gratuito, también pudiendo integrarlo con las herramientas de aplicaciones nativas.

---

<sup>11</sup> *Html siglas de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web (Artedínamico, s.f.).*

<sup>12</sup> *CSS siglas en inglés de cascading style sheets (hojas de estilo en cascada) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML (Alonso, 2012)*

<sup>13</sup> *JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado.*

<sup>14</sup> *jQuery Mobile es un Framework optimizado para dispositivos táctiles (también conocido como Framework móvil) que está siendo desarrollado actualmente por el equipo de proyectos de jQuery.*

## 2.1.4. Comparación entre los tipos de APPS

	APP's nativas	APP's Webs	APP's Híbridas
Acceso a Hardware	Completo	Bajo	Parcial
Velocidad	Muy rápida	Rápida	Rápida
Coste	Elevado	Moderado	Moderado
Tienda aplicaciones	Disponible	No	Disponible
Necesidad aprobación	Obligatorio	No	Obligatorio
Requiere Internet	Según funcionalidad	Obligatorio	Obligatorio
Multiplataforma	No	Sí	Parcial

Fuente:

<http://www.tauazul.com/wp-content/uploads/2015/03/comparativatipoapp.png>  
Figura 2. Diferencias entre apps nativas, web apps y apps híbridas.

A continuación se ira analizando los diferentes campos de confrontación entre los diferentes tipos de desarrollo, comparando las ventajas y desventajas de estos.

### 2.1.4.1. Multiplataforma

Hay que tener en cuenta que no todos los Smartphones funcionan bajo la misma plataforma. Es fundamental llegar al máximo número de plataformas, o por lo menos a las más importantes como son IOs y Android que entre ellas dos suponen casi el 90% de los Smartphones en todo el mundo. Por detrás están BlackBerry a y Windows Phone cuya cuota de mercado está en decadencia.

Una de las principales ventajas de las web app con respecto a una app nativa es que por definición son originalmente multiplataforma gracias a ejecutarse en cualquier navegador que soporte HTML5. Con tan sólo unos retoques para cada plataforma,

cualquier usuario disfrutará de tu web app en su dispositivo. Por el contrario, en el caso de app nativas, habrá que hacer un desarrollo distinto para cada sistema operativo, lo que supondrá un incremento del coste de desarrollo de la aplicación (Qode, 2014). En el caso de las apps híbridas dependen del entorno de desarrollo en el que se esté trabajando.

#### **2.1.4.2. Acceso a funcionalidades nativas**

Una de las principales ventajas de una app nativa con respecto a una web app es la de poder hacer uso de los accesorios integrados dentro del propio teléfono como la cámara, el GPS, la brújula, etc..

Gracias a estas funcionalidades las apps nativas se convierten en preciadas herramientas para los usuarios, pudiendo hacer uso de mapas y geo localización en tiempo real, funcionalidades de realidad aumentada, subir fotos directamente desde la cámara y las miles posibilidades que ofrecen los Smartphone (Qode, 2014).

#### **En resumen... ¿Web app, App Nativa o App Híbrida?**

Es evidente que una aplicación nativa es una opción más completa que una web app pero también es notablemente más cara. Por ello, es vital analizar los requisitos, las necesidades reales y el presupuesto disponible para la app, para saber cuál es la mejor opción (Qode, 2014).

## **2.2. HERRAMIENTAS DE DESARROLLO MÓVIL HIBRIDAS**

Mientras en el mundo se continúa con el debate sobre qué tipo de aplicación móvil brinda una mejor solución al problema de las empresas, lo único cierto es que todos ellos ofrecen argumentos y planteamientos convincentes.

(Heller, 2014) La dificultad y el costo del desarrollo de las aplicaciones móviles no han escapado a la atención de las empresas innovadoras. Aquí presentamos algunas herramientas con poco o sin código para construir aplicaciones móviles. Algunas abarcan más de una plataforma móvil, algunas abarcan también aplicaciones Web, pero todas tienen como objetivo hacer que el proyecto móvil de su organización funcione rápidamente.

### **2.2.1. Alfa anywhere**

Un constructor de extremo a extremo de poco código, rápido, basada en asistente, con un IDE basado en Windows, Alpha Anywhere (99 dólares/mes) soporta muchas bases de datos y se dirige a las aplicaciones Web, móviles (IOs, Android y de Windows Phone), y a las aplicaciones de escritorio. Las aplicaciones HTML pueden ser construidas usando un diseñador basado en componentes y para que se adapten a los tamaños de pantalla desde cuatro pulgadas a cuatro pies. Alfa Anywhere se integra con PhoneGap y PhoneGap Build de Adobe, lo que permite la fácil creación de aplicaciones móviles híbridas, sin requerir que el desarrollador instale varios entornos de desarrollo nativos o compre una Mac. Actualmente, la compañía está probando una solución única para aplicaciones móviles que se conectan ocasionalmente y que están basadas en bases de datos remotas.

### **2.2.2. App Press**

App Press (30-460 dólares por mes) es un creador de aplicaciones sin código basado en la Web dirigido al iPhone, iPad, y a aplicaciones de Android. Orientado para diseñadores, App Press utiliza una interfaz de usuario parecida al Photoshop, desde pantallas montadas hasta activos visuales usando capas. En la parte final, App Press es un servicio y una plataforma basada en la nube de Amazon. La compañía afirma que los diseñadores pueden producir su primera aplicación en un día, por lo que los diseñadores

de experiencia pueden crear cinco aplicaciones al día, y los diseñadores experimentados pueden entrenar a los nuevos diseñadores en la plataforma.

### **2.2.3. AppArchitect**

AppArchitect (beta libre) es un constructor sin código de arrastre y pegado basado en la web y también es una plataforma para aplicaciones nativas del iPhone y la iPad, las cuales se pueden pre visualizar en la AppArchitect Preview App de la aplicación. Se puede descargar en la App Store de iTunes, y los binarios terminados se pueden descargar para presentarlos a la App Store. AppArchitect reúne plugins de bloques de construcción que se escriben en Objective-C, y un SDK AppArchitect estará disponible para extender las capacidades del producto. La compañía tiene previsto ampliar el producto para generar aplicaciones web y Android en el futuro, y planea cobrar de 40 a 100 dólares por mes una vez que se libere el producto.

### **2.2.4. Form.com**

Form.com (412 dólares por mes o más, dependiendo del alcance y los requisitos de proyecto) es una plataforma empresarial basada en web para soluciones web, con un constructor de formularios de arrastre y pegado y tecnología flexible de back-end. El constructor puede crear nuevos formularios o replicar los existentes de papel, configurar el flujo de trabajo y la integración de la API, los procesos específicos y transiciones lógicas, permitir la captura de imágenes dentro de los formularios, las firmas digitales, y permitir el llenado automático de campo. Los formularios móviles terminados pueden recopilar información cuando se desconectan y transferirlos cuando se reestablece la conexión.

### **2.2.5. IBuildApp**

IBuildApp es un constructor web que ofrece plantillas personalizables para iPhone, iPad, y aplicaciones de Android, además promete que puede crear una aplicación en cinco minutos. Su aplicación puede ser gratuita si acepta iBuildApp branding y límites muy estrechos a la cantidad de usuarios y visitas. Las visitas ilimitadas de usuarios y aplicaciones de tableta cuestan 299 dólares al mes, y hay varios planes entre los extremos. Para los tipos de aplicaciones comunes, los sistemas basados en plantillas como iBuildApp a veces puede producir resultados útiles, siempre y cuando la selección de widgets incluya la funcionalidad que necesita.

### **2.2.6. QuickBase**

QuickBase (299 dólares por mes y más) es un constructor en línea y una plataforma de aplicaciones de base de datos para web y móviles. Ofrece más de 300 plantillas de aplicaciones personalizables, incluyendo el Complete Manager que se muestra en la diapositiva. Los usuarios pueden crear aplicaciones "desde cero" a partir de un diseño de datos, y todos los sitios web QuickBase también pueden ser vistos como sitios web para móviles. Si bien Mobile QuickBase no está disponible actualmente en forma de aplicación, el sitio web para móviles es eminentemente utilizable.

### **2.2.7. Salesforce1**

Salesforce1 (desde US\$ 25 dólares por usuario al mes) le da la capacidad de acelerar el desarrollo y despliegue de aplicaciones móviles HTML5, IOs y Android, así como aplicaciones web. En el modelo más simple, utiliza una aplicación para móviles, sitios web o genérica de visor Salesforce con el fin de trabajar con la aplicación web Force.com. Un paso adelante respecto a esto es crear una aplicación móvil jQuery Mobile (que se



muestra en la diapositiva), Angular.js, Backbone.js o Knockout HTML5, utilizando un paquete móvil de Salesforce. En el nivel más complicado, puede crear aplicaciones nativas o híbridos para IOs y Android utilizando el Salesforce Mobile SDK para la plataforma móvil combinado con las herramientas nativas de SDK. Estas aplicaciones se comunican con todas la parte trasera a través de una aplicación conectada en Salesforce.

### **2.2.8. ViziApps**

ViziApps (desde US\$ 15 hasta US\$ 149 por mes, además de la tasa de publicación) combina un diseñador visual en línea y muestras de aplicaciones personalizables con generación de código para la web móvil, así como para IOs y aplicaciones nativas de Android. El diseñador ViziApps tiene campos de formulario y gráficos, 60 fondos y cuatro mil imágenes. Es compatible con mapas, vídeos, sonido, barras de navegación, y los paneles de navegación, y tiene muchas personalizaciones y extensiones de JavaScript. Las aplicaciones de plantilla muestran cómo se utilizan los campos, las acciones y las interfaces de datos.

### **2.2.9. Mobile Chrome Development Kit**

El 'Kit de Desarrollo Móvil Chrome' de Google (gratis), lanzado recientemente, como una herramienta de pre visualización del desarrollo basado en Apache Córdoba, toma un estrategia de aplicación híbrida. Un único proyecto está dirigido a IOs, Android y aplicaciones de Chrome. Como se puede ver en la diapositiva, la interface de usuario es el estándar HTML y CSS, que está integrado en las herramientas nativas de Android (muestra) e IOs. Si bien esto no es definitivamente una herramienta sin código, puede hacer muchas cosas utilizando cualquier diseñador de páginas HTML. Una vez que tenga que agregar el código móvil, tiene API Chrome y API Córdoba a su disposición, desde

JavaScript para mejorar la aplicación sin tener que bajar a nivel de código nativo dependiente de la plataforma.

#### **2.2.10. PhoneGap**

(<http://phonegap.com/>, s.f.)PhoneGap es una implementación de código abierto de estándares abiertos y gratis. Eso significa que los desarrolladores y las empresas pueden utilizar PhoneGap para aplicaciones móviles que están libres, comerciales, de código abierto, o cualquier combinación de éstos.

PhoneGap es un marco de desarrollo de aplicaciones móviles, basado en el proyecto de código abierto Apache Córdoba. Se le permite escribir una aplicación una vez con HTML, CSS y JavaScript, y luego implementarlo en una amplia gama de dispositivos móviles sin perder las características de una aplicación nativa. PhoneGap Build es un servicio basado en la nube construido en la cima del marco PhoneGap. Te permite crear fácilmente esas mismas aplicaciones móviles en la nube.

#### **2.2.11. Appcelerator**

Appcelerator combina los servicios de fondo en la nube y múltiples marcos IDE, SDK, en un sistema de nivel empresarial para el desarrollo móvil. El SDK Titanium le permite desarrollar aplicaciones Web nativas, híbridas, y móviles desde una única base de código.

Titanium Studio es una IDE extensible, basada en Eclipse para la construcción de aplicaciones Titanium y Web, y Appcelerator Cloud Services proporciona un conjunto de características de la red a escala de forma automática y objetos de datos para su aplicación. El marco Alloy es un marco de Appcelerator diseñado para desarrollar rápidamente aplicaciones de Titanium, basados en la arquitectura MVC y que contiene soporte

integrado para Backbone.js y Underscore.js. Si bien Appcelerator no es una solución sin código, proporciona herramientas basadas en JavaScript para iOS, Android, Tizen, BlackBerry y aplicaciones Web móviles en un solo lugar.

Appcelerator es libre para desarrollar; Appcelerator Platform tiene precios a niveles empresariales.

En el capítulo III de este documento se profundizará en la descripción de esta herramienta.

## **2.3. LENGUAJES DE CODIFICACIÓN WEB**

Actualmente existen diferentes lenguajes de programación para desarrollar en la web que han surgido debido a las tendencias y necesidades de las plataformas, desde los inicios surgiendo diferentes demandas de los usuarios y se dieron soluciones mediante lenguajes de programación, acción que dio lugar a desarrollar lenguajes de programación para la web dinámica que permitieran interactuar con los usuarios y utilizar sistemas de bases de datos.

### **2.3.1. Lenguaje de programación PHP**

#### **2.3.1.1. Breve historia**

(Heurtel, 2011) El lenguaje PHP (Personal Home Page históricamente, oficialmente acrónimo recursivo de PHP: Hypertext Preprocessor) fue diseñado en 1994 por Rasmus Lerdorf para estas necesidades personales antes de su lanzamiento a principios de 1995.

En 1995 se publicó una nueva versión completamente reescrita bajo el nombre de PHP/FI versión 2. Esta versión, capaz de manejar formularios y de acceder a la base de datos SQL, permite al lenguaje crecer rápidamente.

En 1997, el desarrollo del lenguaje recae en un equipo liderado por Rasmus Lerdorf y conduce al lanzamiento de la versión 3.

En 2000, el analizador PHP se migra al motor de análisis Zend para proporcionar un mejor rendimiento y admitir un mayor número de extensiones: se trata de la versión 4 de PHP.

En 2004 nace la versión 5. Esta nueva versión basada en la versión del motor Zend 2, aporta varias características nuevas, la mayoría de ellas relacionadas con la programación orientada a objetos.

A día de hoy, los analistas creen que más de 20 millones de sitios Web utilizan PHP en el mundo (en número de dominios).

#### **2.3.1.2. ¿Qué es PHP?**

PHP es un lenguaje de script que se ejecuta del lado del servidor, el código PHP se incluye en una página HTML normal. Por lo tanto se puede comparar con otros lenguajes de script que se ejecutan según el mismo principio: ASP (Active Server Pages), JSP (Java Server Pages) o PL/SQL Server Pages (PSP).

Lo que diferencia a PHP de un lenguaje como JavaScript, donde el código se ejecuta del lado del cliente (en el explorador), es que el código PHP se ejecuta del lado del servidor. El resultado de esta ejecución se incrusta en la página HTML, que se envía al navegador. Este último no tiene conocimiento del procesamiento que se llevó a cabo en el servidor. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez ofrece muchas características avanzadas para los

programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

Aunque el desarrollo de PHP está centrado en programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas y combinación con otros lenguajes de vista como JavaScript, jQuery, CSS. (PHP, 2014)

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Multiplataforma</li> <li>• Manejo de excepciones</li> <li>• Biblioteca nativa de funciones</li> <li>• Permite técnicas de programación orientada a objetos.</li> <li>• Amplia documentación en su página oficial</li> <li>• Destacada conectividad con MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.</li> <li>• Es libre.</li> <li>• Muy fácil de aprender.</li> <li>• Se caracteriza por ser un lenguaje muy rápido.</li> <li>• Capacidad de expandir su potencial utilizando módulos.</li> </ul>	<ul style="list-style-type: none"> <li>• Se necesita instalar un servidor web.</li> <li>• Todo el trabajo lo realiza el servidor y no delega al cliente.</li> <li>• La legibilidad del código puede verse afectada al mezclar, sentencias HTML y PHP, Es muy difícil de optimizar.</li> <li>• Dificulta la modularización.</li> <li>• Dificulta la organización por capas de la aplicación.</li> <li>• No posee adecuado manejo de Unicode.</li> </ul>

Fuente: Autor y

<http://blogs.utpl.edu.ec/disenowebymultimedia/2009/07/23/ventajas-y-desventajas-de-php-2>

Tabla 1. Ventajas y desventajas de PHP

### 2.3.1.3. Mezclar PHP y HTML

Existen varios enfoques para mezclar PHP y HTML, sin embargo, estos enfoques se basan en dos principios simples:

- ✓ La página puede tener una o varias inclusiones de código PHP
- ✓ El código PHP genera el “texto” que se integra en la página HTML que se envía al explorador. Por lo tanto cualquier “Texto” que sea comprensible para el

navegador puede generarse por código PHP: texto simple, código HTML, código JavaScript.

No existen reglas para mezclar PHP y HTML. Un enfoque utilizado por los programadores es utilizar código PHP únicamente en la parte verdaderamente dinámica de la página, y escribir el resto directamente en HTML. Esta técnica hace que el código sea menos pesado y permite localizar inmediatamente donde se encuentra la lógica de la aplicación.

### **2.3.2. Lenguaje JavaScript**

#### **2.3.2.1. Breve historia**

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó compatibilidad con la tecnología Java en su navegador web Netscape Navigator en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

JavaScript es una marca registrada de Oracle Corporation. Es usada con licencia por los productos creados por Netscape Communications y entidades actuales como la Fundación Mozilla.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers 'Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado

como un estándar ECMA, con el nombre de ECMAScript. Poco después también como un estándar ISO.

### 2.3.2.2. ¿Qué es JavaScript?

JavaScript se presenta como un lenguaje de desarrollo de aplicaciones Cliente-Servidor a través de internet. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

El programa en JavaScript reconoce eventos, son Acciones de JavaScript (uno de los eventos es OnClick, que se refiere a que al pulsar con el cursor sobre la zona a la cual está asociado ese evento nos realiza una determinada acción), creados por el usuario, definiendo así un sistema interactivo. Podemos por ellos crear formularios que verifiquen la validez de la información e interpreten esta en el mismo programa contenido en el documento HTML sin necesidad de comunicación por la red. (Sánchez Maza, 2001)

Ventajas	Desventajas
<ul style="list-style-type: none"><li>• Es un lenguaje sencillo</li><li>• Es un lenguaje de programación muy liviano</li></ul>	<ul style="list-style-type: none"><li>• Código visible por cualquier usuario.</li></ul>

- 
- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• Se recomienda para la creación de aplicaciones web</li> <li>• Utiliza poca memoria</li> <li>• Útil para el desarrollo de páginas web dinámicas</li> <li>• Tiene gran cantidad de efectos visuales</li> <li>• Fácil manejo de datos</li> <li>• Es soportado por los más populares navegadores</li> <li>• Puede ser usado en los servidores para hacer guiones de alta calidad</li> <li>• Ligero de carga</li> <li>• Fácil de integrar</li> <li>• Cientos de aplicaciones disponibles para uso</li> <li>• Puede agregar interactividad a elementos web</li> <li>• Compatible con la gran mayoría de los navegadores modernos incluyendo iPhone, móviles &amp; PS3</li> </ul> | <ul style="list-style-type: none"> <li>• El código debe descargarse completamente.</li> <li>• Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS.</li> <li>• Sus recursos no son tan extensos</li> <li>• Sus opciones 3d son limitadas</li> <li>• En sistemas no tan complejos puede generar errores.</li> <li>• Los usuarios pueden desactivar JavaScript en su navegador HTML.</li> <li>• La simplicidad y el estándar abierto</li> </ul> |
|---|---|
- 

Fuente: <http://clubensayos.com/Temas-Variados/JAVA-SCRIPT-VENTAJAS-Y-DESVENTAJAS/222066.html>

Tabla 2. Ventajas y Desventajas de JavaScript

## 2.4. HERRAMIENTAS Y TECNOLOGÍAS VARIAS

Las herramientas RAD (acrónimo en inglés de *Rapid Application Development*) desarrollo rápido de aplicaciones, permiten gestionar el desarrollo de una aplicación de una forma segura y rápida.

### 2.4.1. ScriptCase

ScriptCase Generador de PHP es una herramienta poderosa para aumentar la productividad del desarrollo web, ahorrando tiempo y aumentando las ganancias. ScriptCase Generador de PHP puede construir sistemas completos y crear informes personalizados seguros y rápidos. Es una herramienta de desarrollo rápido web disponible en el mercado.



Por medio de ScriptCase se puede desarrollar sistemas completos de PHP y reportes personalizados con gran rapidez. Se pueden crear aplicaciones de una forma intuitiva y rápida, como los informes de gestión, formularios de inscripción, gráficos personalizables en tiempo de ejecución, autenticación de usuarios, menús dinámicos, calendarios, dashboards y mucho más.

ScriptCase puede trabajar con cualquier navegador web, ya sea en su red local, o a través de Internet y permite a varios desarrolladores para trabajar simultáneamente en el mismo proyecto. Permite conexiones a la base de datos favorita (MySQL, PostgreSQL, Oracle, SQL Server, etc.) para generar aplicaciones que se ejecutan independientemente de ScriptCase, y puede ser publicado en cualquier servidor web con PHP (ScriptCase, 2015).

#### **2.4.2. JSON**

**JSON**, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

JSON está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edición - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

##### **2.4.2.1. Constitución de JSON**

JSON está constituido por dos estructuras:

Una colección de pares de nombre/valor. En varios lenguajes estos son conocidos como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

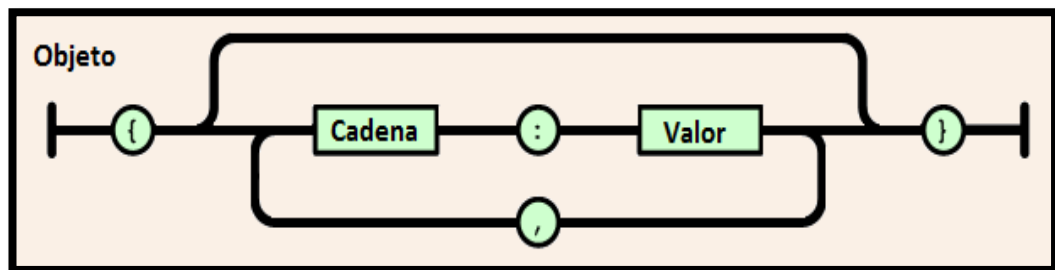
Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementado como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

#### 2.4.2.2. Estructura JSON

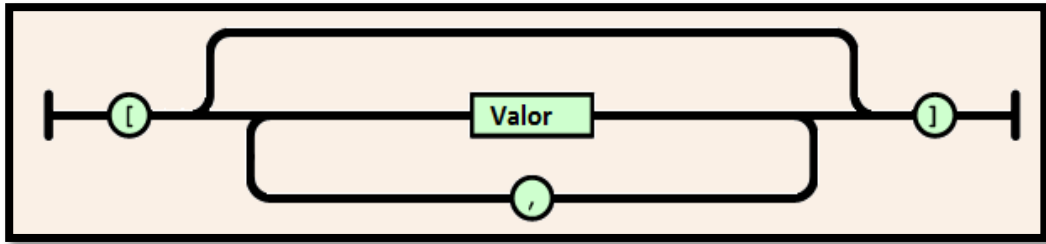
(Json., 2015)En JSON, se presentan de estas formas:

Un objeto es un conjunto desordenado de pares nombre/valor. Un objeto comienza con “{” (llave de apertura) y termine con “}” (llave de cierre). Cada nombre es seguido por “:” (dos puntos) y los pares nombre/valor están separados por “,” (coma).



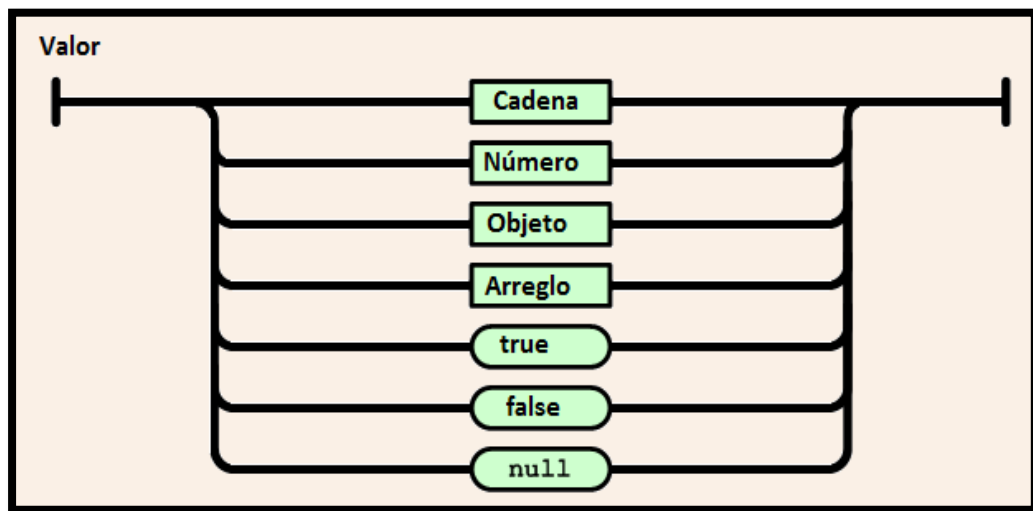
Fuente: <http://json.org/object.gif>  
Figura 3. Estructura JSON

Un arreglo es una colección de valores. Un arreglo comienza con “[” (corchete izquierdo) y termina con “]” (corchete derecho). Los valores se separan por “,” (coma).



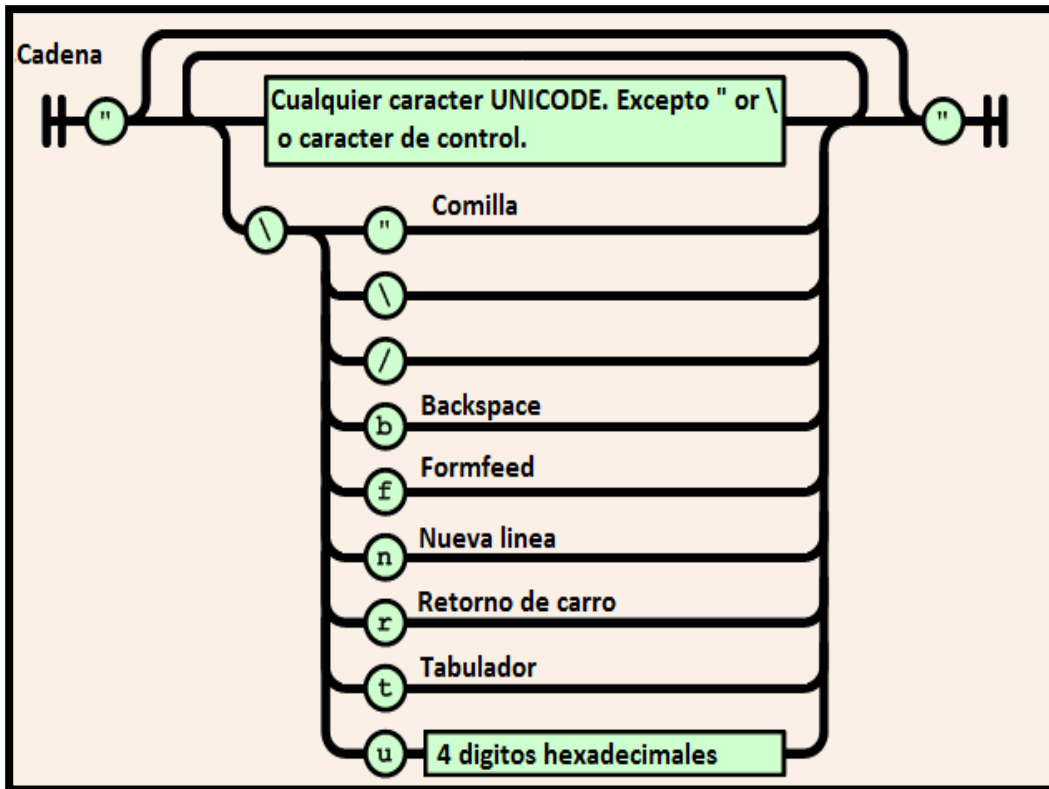
Fuente: <http://json.org/array.gif>  
 Figura 4. Estructura de Arreglo JSON

Un valor puede ser una cadena de caracteres con comillas dobles, o un número, o “true” o “false” o “null”, o un objeto o un arreglo. Estas estructuras pueden anidarse.



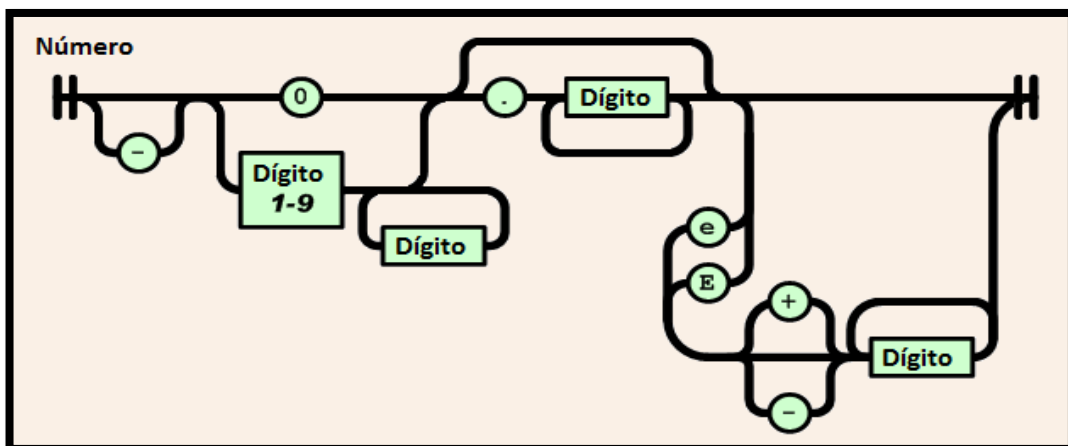
Fuente: <http://json.org/value.gif>  
 Figura 5. Estructura de valor en JSON

Una cadena de caracteres es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter. Una cadena de caracteres es parecida a una cadena de caracteres C o Java.



Fuente: <http://json.org/string.gif>  
 Figura 6. Cadenas en JSON

Un número es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.



Fuente: <http://json.org/number.gif>  
 Figura 7. Representación de números en JSON

### 2.4.2.3. Soporte de JSON

En la actualidad la mayoría de Navegadores de Internet ofrecen o aceptan el formato JSON.

IE	Firefox	Cromo	Safari	Ópera	iOS Safari *	Opera Mini *	Navegador de Android *	Chrome para Android
		31						
		36						
		37						
		39					4.1	
8	31	40					4.3	
9	36	41					4.4	
10	37	42	7	28	7.1		4.4.4	
11	38	43	8	29	8.3	8	40	42
Borde	39	44		30				
	40	45		31				

Fuente:

<http://www.campusmv.es/recursos/image.axd?picture=Soporte-CORS-navegadores.png>

Figura 8. Navegadores que soportan JSON

## 2.5. MYSQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo (Santillán, 2004)

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos

deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

### **2.5.1. Características**

#### **2.5.1.1. Abstracción de la información**

Ahorran al usuario detalles acerca del almacenamiento físico de los datos ya que este hecho se hace transparente al usuario así se definen varios niveles de abstracción.

#### **2.5.1.2. Independencia**

Consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se la utilizan.

#### **2.5.1.3. Redundancia mínima**

Un buen diseño de una base de datos logra evitar la aparición de información repetida o redundante, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

#### **2.5.1.4. Consistencia**

En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente es decir, que los datos repetidos se actualicen de forma simultánea.

#### **2.5.1.5. Seguridad**

La información almacenada en la base de datos puede llegar a tener un gran valor, el gestor de datos debe garantizar que la información se encuentra segura frente a usuarios malintencionados que intenten leer información privilegiada; frente a ataques que deseen

manipular o destruir la información o simplemente ante la impericia de algún usuario autorizado.

#### **2.5.1.6. Integridad**

Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados; es decir, proteger los datos ante fallos de hardware, datos introducidos por usuarios o cualquier otra circunstancia capaz de corromper la información almacenada.

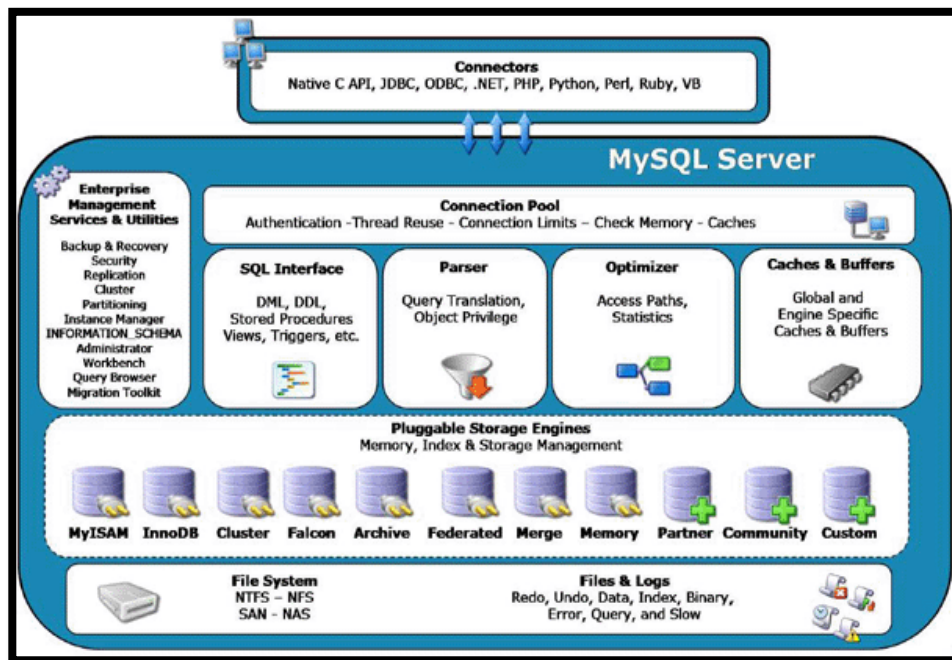
#### **2.5.1.7. Control de la concurrencia**

En la mayoría de entornos (excepto quizás el doméstico) lo más habitual es que sean varios usuarios que acceden a los datos bien para recuperar información, almacenar o también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un gestor permite controlar dichos accesos concurrentes a la información que podría derivar en inconsistencias.

#### **2.5.1.8. Aprovecha la potencia de sistemas multiprocesador**

Gracias a su implementación multi hilo. Soporta gran cantidad de tipos de datos para las columnas. Dispone de API's en gran cantidad de lenguajes. Gran portabilidad entre sistemas. Soporta hasta 32 índices por tabla. Gestión de usuarios y passwords manteniendo buen nivel de seguridad en los datos. Condición de Open Source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad. Se puede descargar el código fuente esto ha favorecido en el desarrollo y actualizaciones.(Cherry, 2011)

## 2.5.2. Arquitectura



Fuente:

<http://www.oracle.com/ocom/groups/public/@otn/documents/digitalasset/146799.gif>

Figura 9. Arquitectura de MySQL



## CAPITULO III



### ESTUDIO DE LA HERRAMIENTA TITANIUM APPCELERATOR

---

- Historia y definición
  - Instalación y configuración
  - API de Titanium
  - Uso de base de datos y archivos locales
  - Interacción con datos remotos
  - Servicios de localización
  - Evaluación de la herramienta
  - Ventajas y desventajas
-

### **3.1. HISTORIA Y DEFINICIÓN**

Appcelerator Titanium es una plataforma de código libre creada por la empresa Appcelerator Inc. para el desarrollo de APPs nativas para móviles (Smartphones y Tablets) programadas en lenguaje JavaScript. Desde su lanzamiento en el año 2008, la plataforma ha tenido una excelente acogida, contando en la actualidad con más de 560.000 usuarios registrados que han desarrollado más de 50.000 aplicaciones. Su oferta se completa con una gran cartera de servicios en la nube.

#### **3.1.1. Appcelerator Titanium: un servicio completo e integral**

Actualmente, la creación y desarrollo de aplicaciones móviles demanda una serie de requerimientos, necesidades y servicios que van mucho más allá de la propia construcción de la APP.

Los nuevos estándares de funcionamiento hacen imprescindible un servicio integral, donde se interconecten las distintas fases y aspectos relacionados con el desarrollo de aplicaciones: estrategias previas, construcción, testeo, pruebas, integración con redes sociales, mediciones y análisis.

La plataforma Appcelerator Titanium ofrece respuesta a todas estas necesidades con un servicio integral que incluye las siguientes herramientas, funciones y servicios:

##### **3.1.1.1. Creación de aplicaciones con Titanium**

Con el programa Titanium y sin apenas conocimientos en programación se pueden construir fácilmente aplicaciones nativas, que funcionan en todo tipo de móviles y tabletas con una base única de código JavaScript.

La plataforma cuenta con un amplio surtido de herramientas que permiten la creación de APPs de gran calidad, capaces de dar respuestas a las exigencias actuales de rendimiento, diseño, capacidad y funcionalidades integradas con las tecnologías de última generación, como la geo localización o los proyectos de realidad aumentada.

#### **3.1.1.2. Pruebas automatizadas**

El sistema permite realizar prototipos y pruebas de la aplicación y testearlas con usuarios reales antes de su salida mercado, asegurando el óptimo funcionamiento del producto desde el primer momento.

#### **3.1.1.3. Servicios en la nube**

Appcelerator es una plataforma que ofrece, entre otros, los siguientes servicios en la nube:

- Arquitectura segura, fiable y escalable donde guardar y gestionar todo tipo de datos (bases de datos, fotos, videos, posts, ratings, likes) y en gran cantidad.
- Servicios propios de Backend as a Service (BaaS), entre los que destacan: notificación de inserciones, actualizaciones de estado, almacenamiento de fotos e integración con las principales redes sociales.
- Posibilidad de extender los servicios estándar y configurarlos a medida.

#### **3.1.1.4. Analítica web**

La plataforma ofrece al desarrollador una completa visión en tiempo real de todas sus aplicaciones, que incluye, con opciones de personalización, todos los indicadores y métricas que le permitirán obtener información de gran utilidad como:

- Comprobar si se están cumpliendo los objetivos de visitas y descargas.

- Calcular el Retorno de la Inversión (ROI).
- Conocer el perfil o target de usuario de la APP: características personales, como sexo o edad, hábitos de navegación, tecnología utilizada, expectativas, necesidades, etc.
- Detectar errores y áreas de mejora.

#### **3.1.1.5. Formación y apoyo**

Manuales en pdf, video tutoriales, webinars, servicio técnico on-line y telefónico, así como el apoyo de una comunidad de casi medio millar de desarrolladores muy activa y colaboradora a la hora de intercambiar, ideas, consejos y experiencias, conforman una base muy sólida y útil para crecer y mejorar constantemente, así como para resolver las dudas que puedan surgir durante el proceso de trabajo con Titanium (Formacion, 2014).

#### **3.1.2. Principales características de Appcelerator**

Las características más importantes de este programa se encuentran en sus funcionalidades o, dicho de otro modo, lo que permite hacer y la forma de hacerlo:

- Con Titanium se pueden crear aplicaciones nativas de gran calidad, válidas y adaptables en una única versión para los principales sistemas operativos de móviles y Tablets.
- Se trata de una plataforma en constante desarrollo, lo que hace que sus posibilidades vayan en aumento. Por ejemplo, en sus inicios sólo permitía crear aplicaciones para los dispositivos de Apple y Android, extendiéndose después al sistema BlackBerry.
- Utiliza el lenguaje de programación JavaScript, que es el propio de las páginas web, y posteriormente traduce al sistema nativo de cada aplicación.

- El interfaz o entorno de programación está basado en Eclipse (Aptana Studio) y es muy fácil, visual e intuitivo, por lo que no es necesarios tener conocimientos de programación.
- Tienen una extensibilidad ilimitada del propio framework Titanium, lo que permite que constantemente se añadan nuevos módulos con más posibilidades y recursos.
- Máxima interconexión con servicios en la nube, lográndose altísimos niveles de capacidad y rendimiento en la aplicación.

### 3.1.3. Tipos de Aplicaciones que genera

Con Appcelerator Studio se pueden generar aplicaciones móviles funcionales en cualquier plataforma móvil Android, IOs y Windows Phone desde la versión 4.0 de Appcelerator, además de aplicaciones Web.

La generación de las aplicaciones móviles depende además de la plataforma en que Appcelerator se encuentra instalada.

Tipo de Aplicación Generada	Appcelerator instalado en Sistema Operativo
Android	<ul style="list-style-type: none"> <li>• Linux</li> <li>• Windows</li> <li>• Mac OSx</li> </ul>
IOs	<ul style="list-style-type: none"> <li>• Mac OSx. Las aplicaciones para iPhone e IPad únicamente se desarrollan desde entornos MAC.</li> </ul>
Windows Phone	<ul style="list-style-type: none"> <li>• Se requiere Windows 8.1</li> </ul>
Web Móvil	<ul style="list-style-type: none"> <li>• Linux</li> <li>• Windows</li> <li>• Mac OSx</li> </ul>

Fuente: <https://docs.appcelerator.com>

Tabla 3. Tipos de Aplicaciones y plataforma requerida.

## 3.2. INSTALACIÓN Y CONFIGURACIÓN

### 3.2.1. Pre requisitos

Para poder instalar y utilizar la herramienta Appcelerator Titanium es necesario cumplir con ciertos requisitos.

#### 3.2.1.1. Memoria

Los requerimientos mínimos para la memoria RAM se definen en la siguiente tabla:

Producto	Sistema Operativo	RAM
Appcelerator Studio	Todos	1 Gb
Último SDK Android	Mac OS X	1,5 Gb
Último SDK Android	Windows	1 Gb
Último SDK Android	Linux Ubuntu	1,5 Gb

Fuente: <https://docs.appcelerator.com/platform/latest/#!/guide/Prerequisites>

Tabla 4. Requisitos de memoria Appcelerator Studio

Como regla general 4 Gb de memoria RAM instaladas son generalmente adecuadas para la ejecución de toda la plataforma Appcelerator.

#### 3.2.1.2. Sistema Operativo

Appcelerator es compatible con los siguientes sistemas operativos

Sistema Operativo	Versión
Mac OS X	10.9.4 y posteriores (Mavericks)
Windows	Windows 7 Windows 8 Windows 8.1 se requiere para el desarrollo de Windows Phone
Linux desktop	Ubuntu 12.04

Fuente: <https://docs.appcelerator.com/platform/latest/#!/guide/Prerequisites>

Tabla 5. Sistemas operativos compatibles con Appcelerator

### 3.2.1.3. Software necesario

El SDK de Titanium requiere JDK y Node.js de Oracle.

#### a) Kit de Desarrollo de Java JDK

Titanium es compatible con las siguientes versiones de JDK para cada sistema operativo.

Sistema Operativo	Versión JDK mínima	Versión JDK máxima	Paquete Arco Versión	Ubicación Descargar	Notas
OS X	7	8 última revisión	64-bit	Sitio Web Oficial	En Mac OS X 10.6 (Snow Leopard) y versiones anteriores, el JDK está preinstalada. En Mac OS X 10.7 (Lion) y más tarde, el sistema operativo debe pedir que lo instale cuando sea necesario.
Windows	7	8 última revisión	<b>32 bits</b> (x86 / i586)	Sitio Web Oficial	La versión de 32 bits del JDK es necesario independientemente de si Titanium se está ejecutando en un sistema Windows de 32 bits o 64 bits. Studio ofrece para instalar JDK 7 si no se encuentra JDK.
Linux Desktop	7	8 última revisión	32-bit y 64-bit	Sitio Web Oficial	Este paquete se debe instalar para lanzar Estudio en sistemas Linux. Si se utiliza el JDK de 64 bits, el paquete ia32-libs es requerido.

Fuente: <https://docs.appcelerator.com/platform/latest/#!/guide/Prerequisites>  
Tabla 6. *JDK compatibles con Appcelerator.*

## b) Node.js

Para ejecutar todos los componentes de Appcelerator, debe tener Node.js 0.10.13 o posterior.

En Mac OS X y Windows, si ha seleccionado para instalar las actualizaciones de Titanium, Studio le pide que instale Node.js. En Linux, es necesario instalar manualmente Node.js.

Sistema Operativo	Nombre del paquete	Versión del paquete	Ubicación descarga
OS X	Node	Más reciente	<a href="#">Sitio Web Oficial</a>
Windows	Node	Más reciente	<a href="#">Sitio Web Oficial</a>
Linux Desktop	Nodejs	Más reciente	<a href="#">Página oficial</a>

Fuente: <https://docs.appcelerator.com/platform/latest/#!/guide/Prerequisites>  
*Tabla 7. Compatibilidad de Node.js con Appcelerator*

### 3.2.2. Instalación

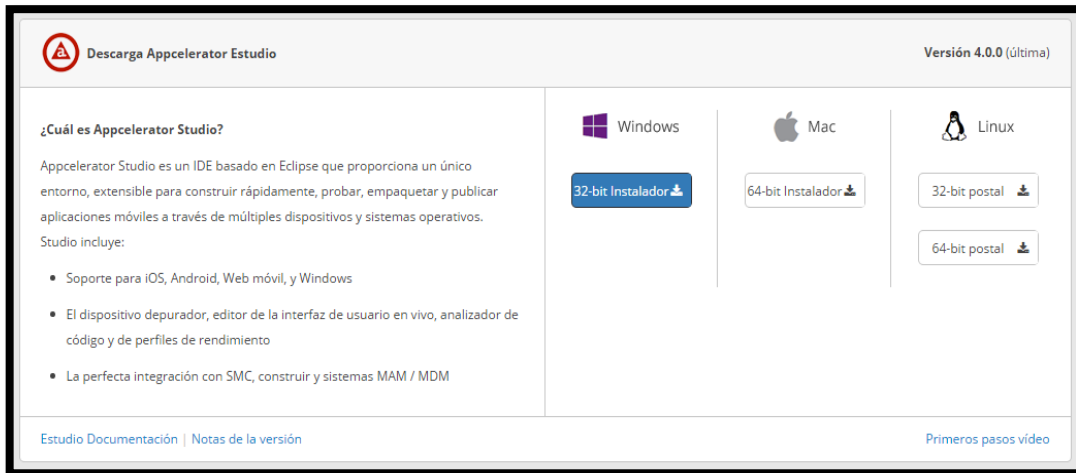
Para obtener el instalador de la herramienta, primero se debe crear una cuenta de usuario en el sitio oficial de la plataforma [www.appcelerator.com](http://www.appcelerator.com), una vez que inicie una sesión en la plataforma podrá tener acceso al instalador de acuerdo al sistema operativo en el que vaya a instalarlo.

Una vez descargado, instale Appcelerator Studio:

Para los sistemas Mac OS X y Windows, ejecute el programa de instalación.

Para los sistemas Linux de escritorio, descomprimir el archivo ZIP.

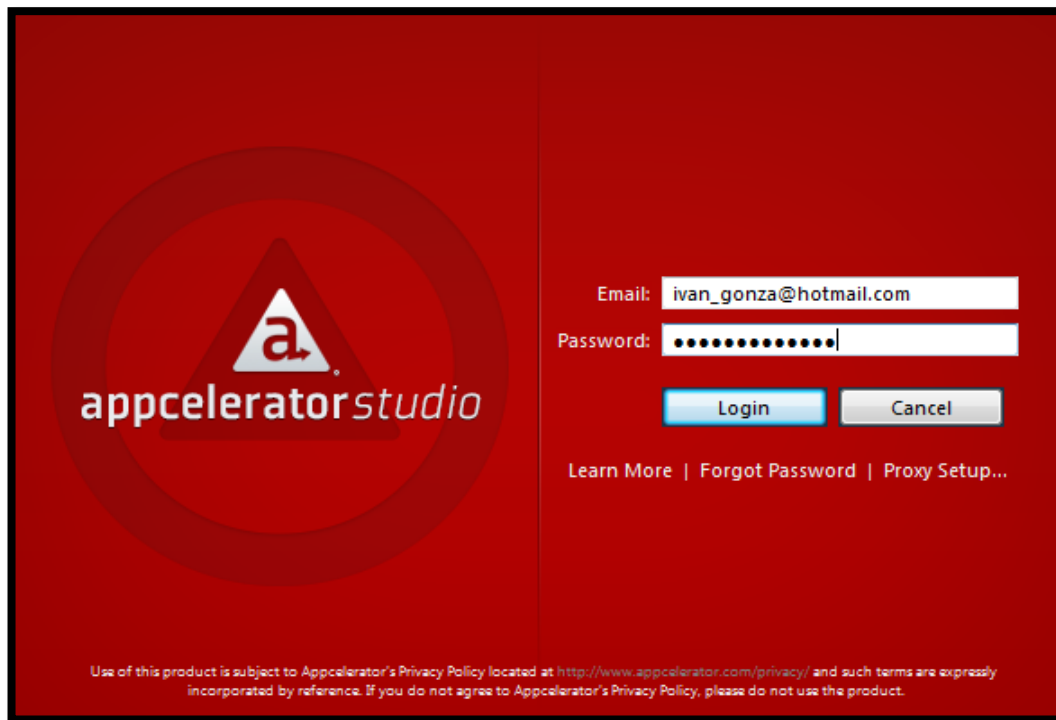




*Fuente: <https://web.appcelerator.com/product/studio>  
Figura 10. Descarga de Appcelerator Studio*

### 3.2.3. Configuración

Quando inicie Appcelerator Studio por primera vez, se debe iniciar sesión con su nombre de usuario y contraseña de la Plataforma Appcelerator. Studio le pide que seleccione un espacio de trabajo, que es simplemente una carpeta donde Studio coloca todos los archivos del proyecto y datos de preferencias IDE. Si está detrás de un proxy, tendrá que configurar la configuración del proxy en Studio antes de iniciar sesión.



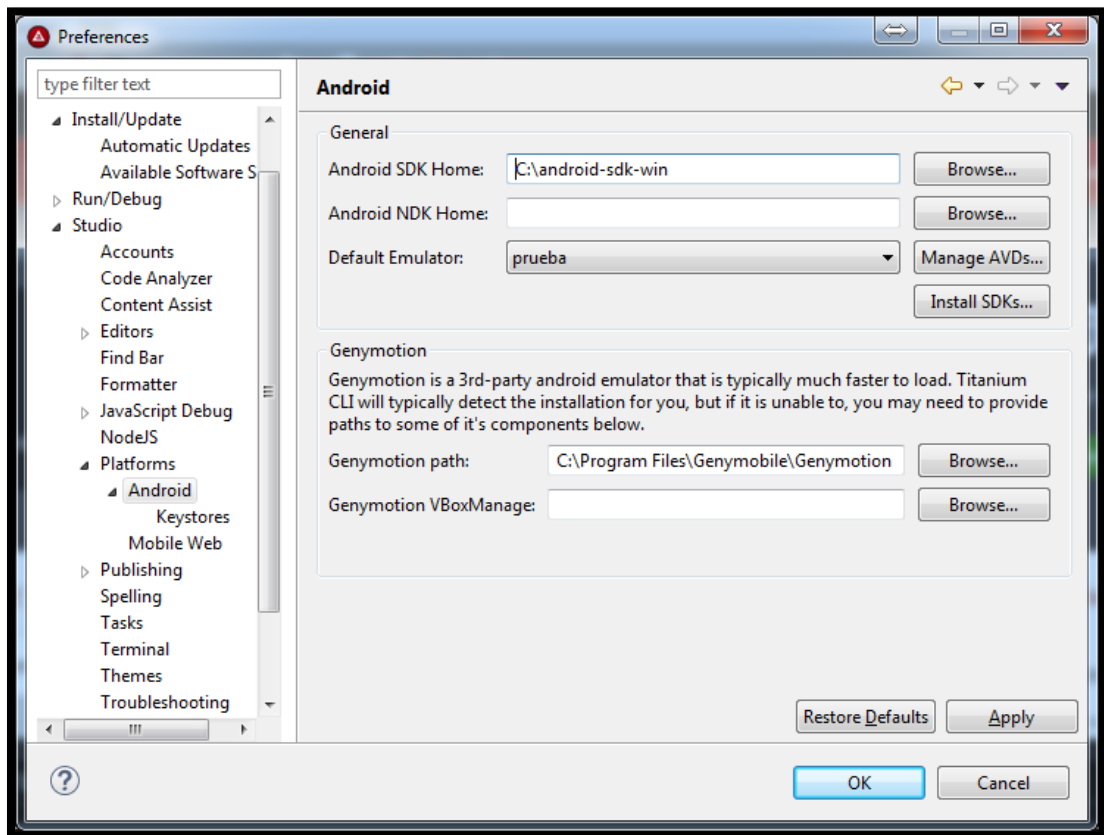
*Fuente: Propia*  
*Figura 11. Inicio de Sesión Appcelerator*

Cuando Appcelerator se ejecuta por primera vez, se inicia automáticamente la instalación de la última versión del SDK de Titanium, módulos de servicio Appcelerator y herramientas Appcelerator de línea de comandos adicionales. Una vez que se termine esto, confirme que no hay más actualizaciones pendientes y que Studio se encuentra en la versión más reciente.

En la barra de menú, seleccione Ayuda→Comprobar actualizaciones Appcelerator SDK para verificar que todos los componentes oficiales Appcelerator están instalados y repita este paso hasta que no haya más actualizaciones disponibles.

Para desarrollar aplicaciones nativas con Titanium se necesita los SDK y herramientas para esas plataformas nativas instaladas en su sistema. Por ejemplo, para el desarrollo de aplicaciones Android se necesita el SDK de Android.

Esto se configura en la opción Windows→Preferences. Que de acuerdo a cada Plataforma se establecen las configuraciones adecuadas.

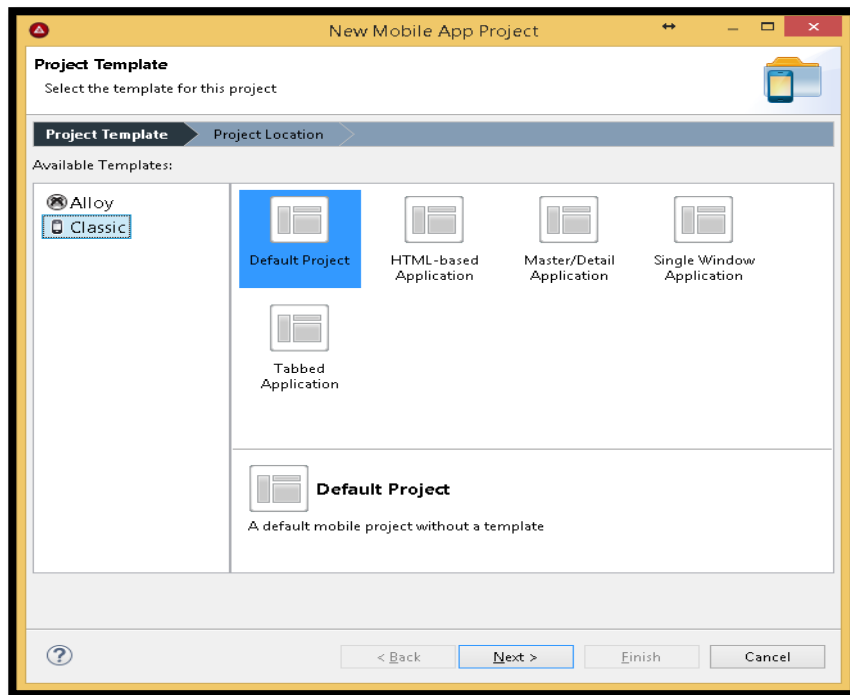


Fuente: Propia

Figura 12. Configuración para SDK Android

### 3.2.4. Tipos de Proyectos

En el framework Appcelerator se pueden crear básicamente dos tipos de Proyectos de aplicaciones móviles:



*Fuente: Propia*  
 Figura 13. Tipos de Proyectos

### 3.2.4.1. Proyectos Alloy

Alloy es un marco de aplicación construida sobre el SDK de Titanium diseñado para ayudar a desarrollar rápidamente aplicaciones de alta calidad y reducir el mantenimiento.

Alloy utiliza la arquitectura modelo – vista – controlador para separar la aplicación en tres componentes:

- **Modelos** proporcionan los datos de la aplicación.
- **Vistas** proporcionan los componentes de interfaz de usuario para interactuar con la aplicación, por escrito utilizando el formato XML y Titanium hojas de estilo (TSS), que abstrae los componentes de interfaz de usuario de la API de Titanium.
- **Los controladores** proporcionan la capa de enlace entre los componentes del modelo y de vista, así como la lógica de aplicación adicional utilizando la API de Alloy y la API de Titanium.

### 3.2.4.2. Proyectos clásicos

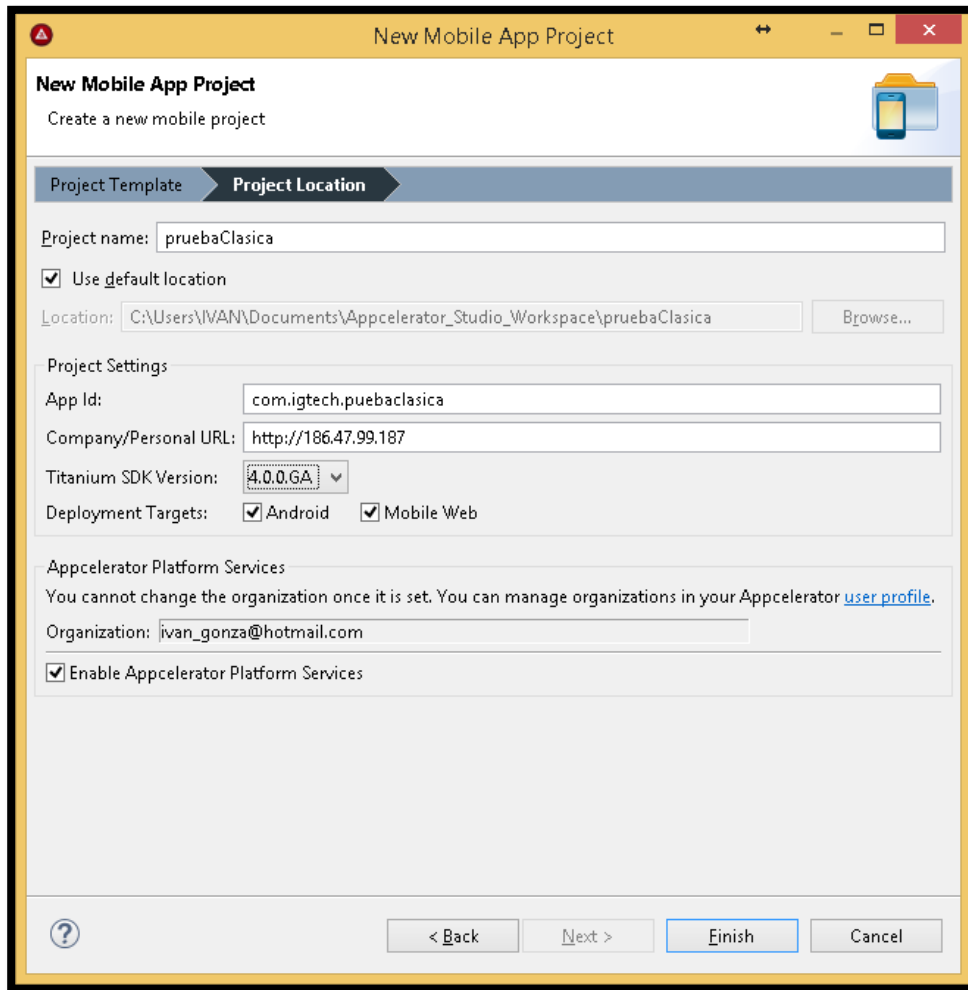
Titanium es el marco de desarrollo clásico de la plataforma Appcelerator, está basado en programación con lenguaje JavaScript.

Titanium utiliza el patrón de fábrica para la construcción de objetos y un patrón de nomenclatura general de las API. Por ejemplo, para construir una Alerta de diálogo, se llama al método *createAlertDialog*.

Para crear un *TextArea*, se llama al método *createTextArea*.

Una vez que se crea un objeto, éste estará disponible hasta que el ámbito sea cerrado.

### 3.2.5. Creación del Proyecto



*Fuente: Propia*

*Figura 14. Configuración inicial de un Proyecto*

Una vez que se selecciona el tipo de proyecto a desarrollar es necesario ingresar algunas características necesarias para la creación del proyecto.

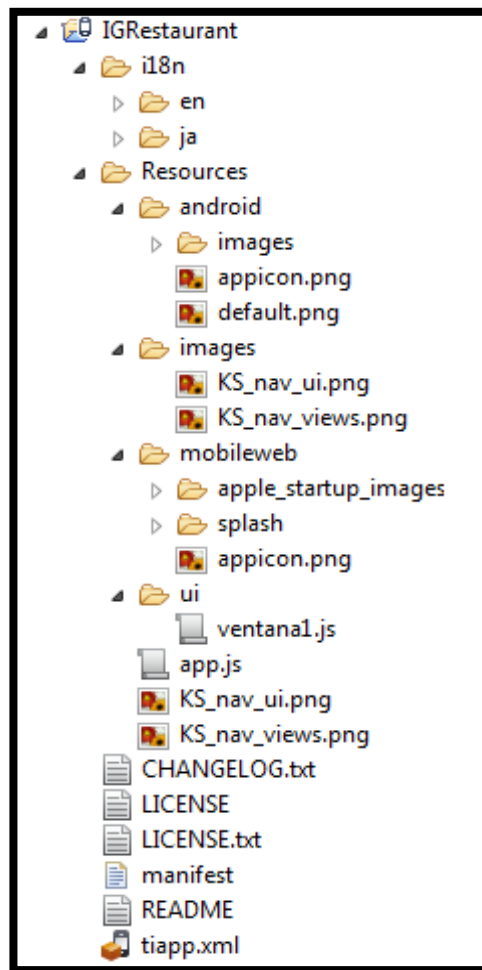
- **Project name:** es el nombre que identifica al proyecto.
- **App Id:** es el identificador único a nivel mundial de la aplicación que sirve para identificar nuestra aplicación al momento de publicarla en las tiendas de aplicaciones. Como sugerencia se establece escribir el identificador de la aplicación con el formato de una dirección web invertida.

- ***Company/Personal URL:*** se especifica una dirección URL de la persona o compañía que desarrolla la aplicación.
- ***Titanium SDK versión:*** se selecciona la versión de Titanium se va utilizar en el desarrollo, dado que de acuerdo a la versión del SDK se contara con más o menos funcionalidades de la API de Titanium.
- ***Deployment Targets:*** donde se especifica para que plataformas se va a poder compilar nuestro proyecto.

### **3.2.6. Estructura de Archivos de los proyectos**

Dependiendo del tipo de proyecto que se crea, se genera un sistema de carpetas y archivos que organiza el desarrollo de la aplicación.

### 3.2.6.1. Estructura de archivos de Proyectos Básicos



*Fuente: Propia*

*Figura 15. Estructura de archivos de Proyecto Básico*

Como se aprecia en la figura 15, un proyecto básico está compuesto de varias carpetas y archivos que se describen a continuación:

Nombre	Tipo	Descripción
i18n	Carpeta	Se utiliza para guardar archivos de configuración para generar aplicaciones en múltiples idiomas.
Resources	Carpeta	En esta carpeta se colocan todos los archivos de imagen y código fuente en JavaScript que utiliza la aplicación.

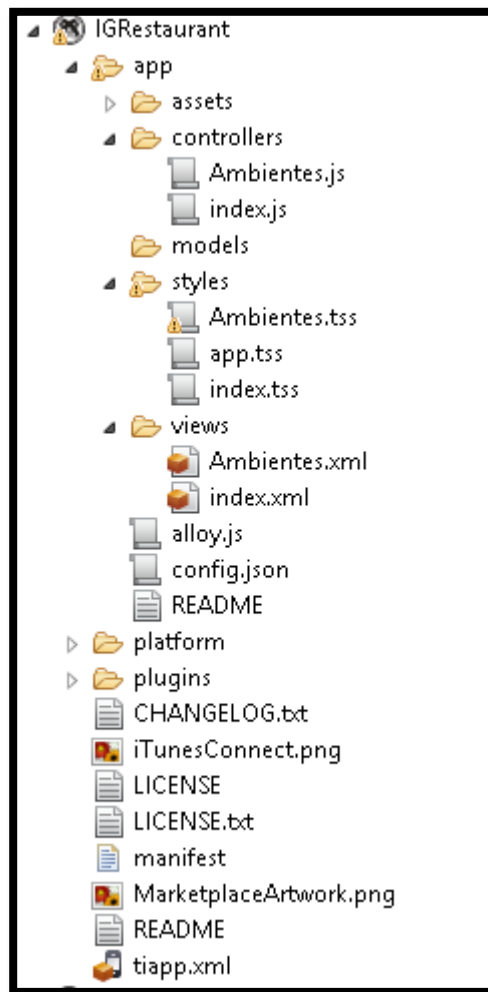


android	Carpeta	Esta carpeta contiene los archivos que se utilizaran al momento de compilar la aplicación para un sistema Android.
images	Carpeta	Utilizada para colocar todas las imágenes que se utilizan en el proyecto.
movilweb	Carpeta	Esta carpeta contiene los archivos que se utilizaran al momento de compilar la aplicación para un sistema movilweb.
ui	Carpeta	En esta carpeta se colocan los archivos JS que contienen toda la codificación de programación.
App.js	Archivo	Este archivo es donde se escribe la codificación principal de la aplicación.
tiApp.xml	Archivo	Este archivo es el que contiene la información básica que se ingresó al momento de crear el proyecto.

Fuente: Propia

*Tabla 8. Descripción del sistema de archivos de un proyecto básico*

### 3.2.6.2. Proyectos Alloy



*Fuente: Propia*

Figura 16. Estructura de Archivos de un Proyecto Alloy

Los proyectos Alloy tienen un sistema de archivos muy diferente a los proyectos clásicos, a continuación se describen cada uno de estos.

Nombre	Tipo	Descripción
app	Carpeta	Es una carpeta donde se colocaran todos los archivos necesarios para compilar la aplicación.
Assets	Carpeta	En esta carpeta se colocan todos los archivos de imagen que utiliza la aplicación.

controllers	Carpeta	Esta carpeta contiene los archivos js encargados de las funciones de controlador del patrón MVC.
Models	Carpeta	Esta carpeta contiene los archivos js encargados de las funciones de modelo del patrón MVC que son los accesos a datos.
Styles	Carpeta	En esta carpeta se colocan los archivos que definen los estilos visuales de cada elemento definido dentro de las vistas.
Views	Carpeta	En esta carpeta se colocan los archivos mxl que contienen toda la codificación que genera el aspecto visual de la aplicación.
Alloy.js	Archivo	Este archivo es donde se escriben funciones globales para la aplicación.
tiApp.xml	Archivo	Este archivo es el que contiene la información básica que se ingresó al momento de crear el proyecto.
Plataform	Carpeta	<p>Contiene subcarpetas por cada tipo de Plataforma a la que se pueda compilar el proyecto (Android, IOs, Móvil Web, etc.)</p> <p>Estas subcarpetas contendrán los archivos necesarios para cada plataforma.</p>

Fuente: Propia  
 Tabla 9. Descripción de Estructura de Archivos Alloy

### 3.3. API DE TITANIUM

El módulo de Titanium ofrece la API Titanium Mobile, permitiendo a los desarrolladores acceder a las funciones nativas de cada entorno de destino. En la actualidad, Android, IOs y entornos Web Móvil (beta) son compatibles.

La API de espacio de nombres Titanium completo es accesible desde el espacio de nombres *Titanium*, pero, por conveniencia y brevedad, se utiliza el alias *Ti*, se recomienda siempre utilizar *Ti* en el código.

#### 3.3.1. Titanium UI

(Appcelerator, 2015) El módulo de interfaz de usuario (UI) es el responsable de los componentes de interfaz de usuario nativa y de la interacción dentro de Titanium. El objetivo del módulo de interfaz de usuario es proporcionar una experiencia nativa junto con un rendimiento nativo mediante la compilación de código JavaScript en sus contrapartes nativas, como parte del proceso de construcción.

El módulo de interfaz de usuario se divide en tres áreas importantes

##### 3.3.1.1. Vistas

Son contenedores que alojan los elementos visuales como los controles u otros puntos de vista. Vistas pueden tener sus propiedades personalizadas, tales como su color de borde y el radio, puede disparar eventos como eventos toques, y pueden contener opcionalmente una jerarquía u otras vistas. En Titanium, la mayoría de los puntos de vista son especializados para llevar a cabo tanto una función visual y un conjunto de comportamientos de interacción como la vista de tablas. Las Vistas siempre se nombran con el sufijo *view*.

### **3.3.1.2. Controles**

Son elementos visuales tales como deslizadores, botones y etiquetas. Proporcionan un elemento visual que tiene un comportamiento definido y típico tiene una configuración especial y eventos especiales. Los Controles mismos en sí mismos son *vistas* y también heredan propiedades, funciones y eventos de vistas.

### **3.3.1.3. Ventanas**

Son típicamente construcciones visuales de alto nivel que son la parte principal de su interfaz. Una aplicación siempre tendrá al menos una ventana y las ventanas pueden tomar diferentes formas y tamaños, pueden tener propiedades de visualización e interacción tales como pantalla completa o modal y pueden ser modificadas para requisitos particulares, por ejemplo, cambiar su opacidad o color de fondo. Las ventanas son ellos mismos puntos de vista y también heredan propiedades, funciones y eventos de una vista. Hay Ventanas especializadas, como una ficha Grupo, que ofrecen un comportamiento adicional más allá de la ventana básica.

Las ventanas se nombran con el sufijo *Window*.

### **3.3.1.4. Optimización**

Los objetos de UI son optimizados por Titanium para ser mostrados únicamente cuando se los necesita; es decir, se puede crear objetos, establecer sus características y añadirlos a las jerarquías sin tener que preocuparse por la memoria o el rendimiento.

Cuando la superficie de dibujo nativo necesita hacer una vista o control específico, Titanium creará automáticamente la vista según sea necesario. Además, Titanium está optimizado para liberar también la memoria una vez que ya no se necesita la vista, en la

pantalla o en situaciones de poca memoria. Sin embargo, es una buena idea ayudar en ciertos casos en los que ya no utilice objetos. Por ejemplo, debe llamar *Close* en una instancia *Windows* cuando ya no lo esté utilizando, y puede llamar de forma segura *Open* en el *Windows* para volver a abrirlo.

#### **3.3.1.5. Contexto global y sub hilos**

Se debe tener cuidado con los objetos que se crean en el archivo *app.js* y sólo se utilizan una vez. Ya que el *app.js* es de contexto global y por lo general se liberan recursos hasta el cierra la aplicación.

Los Objetos *Windows* se abren con la propiedad *url* en otro archivo JavaScript, esto proporciona una buena manera de descomponer su aplicación en unidades más pequeñas.

Además, los objetos *Windows* creados con una *url* funcionan en un contexto JavaScript separado. Mientras que todo el procesamiento de la interfaz de usuario se realiza en el hilo principal de la interfaz de usuario, otro tipo de elaboración dentro de una ventana o del *app.js* que no tiene interacción de interfaz de usuario se ejecutará en su propio hilo.

Otra ventaja de utilizar la propiedad *url* es que cuando la ventana está cerrada, los recursos asignados en el contexto de la ventana se pueden limpiar de inmediato, lo que permite el ahorro de recursos como la memoria y la CPU.

#### **3.3.1.6. Portabilidad**

Los componentes de Titanium están diseñados para ser portables a través de la mayor cantidad de plataformas que ya que soporta.

Sin embargo, hay casos en que un dispositivo no admite una característica específica o necesita una funcionalidad adicional. Para los casos en que el sistema operativo del dispositivo admite capacidades que otras plataformas no lo hacen, Titanium trata de colocar esas capacidades en un espacio de nombres separado, como *Titanium.UI.iPhone*. Sin embargo, en los casos en que el control está en un espacio de nombres común y compatible con las funciones adicionales, se coloca esa funcionalidad directamente en el objeto.

### 3.3.1.7. Eventos

Las acciones de los eventos deben ser definidas antes de que el evento en si ocurra, caso contrario no se garantiza su correcta activación.

Las definiciones de las acciones se deben hacer lo más tarde posible, teniendo en cuenta lo mencionado anteriormente, esto puede mejorar significativamente la capacidad de respuesta de la aplicación. Por ejemplo un detector de eventos para un evento *click* se puede definir después de que la ventana padre ha sido abierta.

### 3.3.1.8. Forma de utilización

Para la utilización de la API de Titanium lo primero es la asignación de la vista, ventana o control a una variable.

```
var titulo = Ti.UI.createLabel({
  color:'white',
  text:'Inicio de Sesión',
  top:0,
  width: Ti.UI.SIZE,
  height: Ti.UI.SIZE
});
```

*Fuente: Propia*

*Figura 17. Creación de Objetos de Titanium*

A cada uno de los elementos se les puede asignar los valores de sus propiedades como color, posición, título, etc. Los valores de las propiedades se establecen dentro de los símbolos de llaves “{}” con un formato de “clave:valor” donde *clave* es el nombre de la propiedad y *valor* es el valor que se asigna a esta propiedad, y separados por una coma “,”.

Para que los controles sean visibles al usuario se deben agregar al contenedor principal haciendo uso de la propiedad *add*.

```
ventana1.add(titulo);
```

*Fuente: Propia*

*Figura 18. Como agregar un control a su contenedor*

Se puede determinar las acciones ante determinados eventos con la propiedad *addEventListener* que tiene como parámetros en evento al que va a responder y la función con la respuesta a dicho evento.

```
titulo.addEventListener('click',function (e){/*codigo de la función */});
```

*Fuente: <https://docs.appcelerator.com>  
Figura 19. Utilización de *addEventListener**



### 3.3.1.9. Principales controles

Control	Descripción	Método de Creación
Button	Se utiliza para la creación de botones.	Titanium.UI.CreateButton
ButtonBar	La barra de botones es un conjunto de botones unidas en un solo control.	Titanium.UI.createButtonBar
ImageView	Vista para mostrar una sola imagen o serie de imágenes animadas.	Titanium.UI.createImageView
Label	Se utiliza para crear una Etiqueta.	Titanium.UI.createLabel
ListItem	Es un ítem que pertenece a un ListItem	No tiene un método de creación propio, se utiliza la propiedad setItem del control ListSection para agregar un Item.
ListSection	Es un contenedor que organiza una lista de Items. Se utilizada dentro de un ListView.	Titanium.UI.createListSection
ListView	Se utiliza para presentar información, organizada en secciones y artículos, en una vista vertical de desplazamiento.	Titanium.UI.createListView
Tab	Es una instancia de una Pestaña que se utilizara dentro de un TabGroup.	Titanium.UI.createTab
TabGroup	Contenedor de una o más pestañas definidas con el Objeto Tab.	Titanium.UI.createTabGroup, para agregar pestañas se utiliza el método addTab.
TableView	Se utiliza para presentar información, organizada	Titanium.UI.createTableView

	en secciones y filas, en una vista vertical de desplazamiento.	
TableViewRow	Es un elemento individual en una tabla. Se pueden añadir de forma explícita a un TableViewSection y mostrarse a través de un TableView.	Titanium.UI.createTableViewRow
TextArea	Define un campo de texto de varias líneas que admita la edición y desplazamiento.	Titanium.UI.createTextArea
TextField	Define un campo de texto de una sola línea.	Titanium.UI.createTextField
View	Es una superficie de dibujo vacía o contenedor.  Define la base para los widgets de interfaz de usuario.	Titanium.UI.createView
WebView	Es un control que permite cargar una página HTML ya sea local o remota. Es muy pesada ya que carga el navegador HTML.	Titanium.UI.createWebView
Window	Es una superficie de dibujo vacía o contenedor.  Es un contenedor de nivel superior al que se le puede agregar otro tipo de controles y vistas.	Titanium.UI.createWindow

---

Puede ser abierto o cerrado.

---

Fuente: propia y docs.appcelerator.com  
Tabla 10. Características de Controles de Titanium.

### 3.4. USO DE BASE DE DATOS Y ARCHIVOS LOCALES

Titanium ofrece diversos medios para guardar los datos al dispositivo móvil del usuario; para utilizar el almacenamiento local de un dispositivo, se necesitan los siguientes objetos:

- *Titanium.App.Properties* es ideal para almacenar los ajustes relacionados con las aplicaciones.
- *Titanium.Database* da acceso a bases de datos SQLite3 locales.
- *Titanium.Filesystem* facilita la manipulación de archivos y directorios.

#### 3.4.1. Propiedades

El API *Titanium.App.Properties* proporciona un almacén de claves / valor de peso ligero. Titanium proporciona métodos para la lectura y la escritura de cadena, entero, booleano, y los valores de la matriz de propiedades. Cualquier dato que puede ser serializado en JSON puede ser almacenado en una propiedad de aplicación. Las propiedades son un gran lugar para almacenar pequeñas cantidades de datos, como los datos de configuración para su aplicación. Pero si tiene una gran cantidad de datos para almacenar sería mejor ponerlo en una base de datos.

Tanto IOs como Android guardan sus archivos de propiedades en diferentes lugares, Titanium ofrece una forma uniforme para establecer y obtener las propiedades de aplicaciones a través de la API *Titanium.App.Properties*.

No hay un límite específico de la cantidad de datos que puede almacenar en propiedades. Sin embargo, los datos de la propiedad de una aplicación se cargan en la memoria cuando la aplicación se inicia, y permanece hasta que se cierre. Esto permite un acceso muy rápido a él, pero a costa del aumento de memoria utilizada por la aplicación.

### 3.4.1.1. Lectura y escritura de propiedades

*Titanium.App.Properties* cuenta con diferentes métodos de escritura y lectura de propiedades de acuerdo al tipo de dato.

Tipo de Dato	Método de Lectura	Método de Escritura	Ejemplo
Cadena	getString	setString	Ti.App.Properties.setString ('miCadena', 'Esto es una cadena'); Ti.App.Properties.getString ('miCadena', cadena por defecto');
Entero	getInt	setInt	Ti.App.Properties.setInt ( 'myInt' , 10);
Decimal	getDouble	setDouble	Ti.App.Properties.getDouble ( 'myInt' , 20);
Booleano (verdadero o falso)	getBool	setBool	Ti.App.Properties.setBool ( 'myBool' , true ); Ti.App.Properties.getBool ( 'myBool' , false );
Matrices	getList	setList	Ti.App.Properties.setList ( 'myList' , myArray); Ti.App.Properties.getList ( 'myList' ));

Fuente:

<https://wiki.appcelerator.org/display/guides2/Lightweight+Persistence+with+the+Properties+API>

Tabla 11. Métodos de Escritura y Lectura de Propiedades

Los métodos *get* reciben dos parámetros, el primero que representa el nombre de la propiedad a buscar y el segundo el valor por defecto que devolverá en caso de no encontrar dicha propiedad.

Los métodos *set* tiene dos parámetros, el primero que es el nombre de la propiedad y el segundo el valor que se asignara a esta propiedad.

#### **3.4.1.2. Almacenar objetos JS como JSON en propiedades**

Si se desea guardar un objeto JavaScript compleja, puede convertirlo en una cadena JSON usando *JSON.stringify ()* proporcionado por *Titanium.JSON*, lo que le permitirá almacenar en la base de datos utilizando el método *Titanium.App.Properties.setString ()*.

Esta cadena JSON almacenado posteriormente se puede convertir de nuevo a un objeto JavaScript usando *JSON.parse()*.

#### **3.4.2. Base de datos**

Tanto IOs y Android incluyen un motor de base de datos SQLite3. En pocas palabras, SQLite es un sistema de gestión de base de datos simplificada que soporta la mayor parte de la especificación SQL92, incluyendo la mayor parte de las sentencias SQL familiares e incluso transacciones.

La base de datos es el lugar apropiado para almacenar gran cantidad de datos estructurados o cuando se necesita soporte transaccional. Puede definir tablas y columnas, y hacer cumplir las relaciones entre tablas del mismo modo que lo haría con un SQL Server o MySQL. Los datos persisten en la base de datos hasta que el usuario desinstala la aplicación o hasta que se sobrescriban o eliminen los datos.

SQLite3 es la versión 3 del sistema basado en SQL de SQLite gestión de bases de datos relacionales (RDBMS), elegido por Apple, Google y RIM para proporcionar almacenamiento local de datos en sus dispositivos móviles. SQLite es actualmente la base de datos más utilizado del mundo. Su popularidad se debe a su código fuente abierto, y como no se basa en ningún servicios instalados a diferencia de casi todos los otros DBMS, su simplicidad de instalación y mantenimiento.

Hay algunas cosas a tener en cuenta la primera vez que trabajo con SQLite, que pueden influir en la forma en que desarrolla con ella:

Los datos de SQLite se almacenan en un archivo de texto simple. No hay privilegios de seguridad o de usuario granulares para los datos. Cualquier persona con acceso al sistema de archivos que puede leer su contenido.

Sólo hay cinco tipos de datos subyacentes; Texto, Numérico, Entero, Real, Null.

Objetos binarios (BLOB) se almacenan como representaciones de texto, así el acceso a los BLOB no es óptima. Es recomendable almacenar los objetos binarios en el sistema de archivos y almacenar la ruta de sistema de archivos en la base de datos.

SQLite apoya el acceso de lectura concurrente, pero refuerza el acceso de escritura secuencial. Esto se debe a un bloqueo de sistema de ficheros se coloca en el archivo durante operaciones de escritura. Este es un punto importante a tener en cuenta con las aplicaciones de subprocesos múltiples.

La integridad referencial no está habilitado de forma predeterminada.

Hay apoyo ALTER TABLE limitada; las columnas no se pueden modificar ni eliminar.

Titanium ofrece acceso a SQLite a través de su módulo *Titanium.Database* que devuelve un Objeto *Titanium.Database.DB*, para la instalación de bases de datos, la apertura y ejecución de consultas y un objeto *Titanium.Database.ResultSet*, que contiene los datos devueltos por las consultas.

### 3.4.2.1. Creación e instalación de bases de datos

Al momento de crear o instalar una base de datos para que la aplicación la use, se cuenta con dos opciones.

- Crear una base de datos vacía y definir su estructura y contenidos a través de sentencias de SQL incorporado dentro de su aplicación.
- Instalar una base de datos predefinida (con o sin datos) que se suministra con su aplicación.

#### a) Creación de una base de datos

Para la creación de una base de datos se hace uso del método *Titanium.Database.open()*, si la base de datos especificada no existe, Titanium la creará automáticamente. Esta llamada devolverá un identificador de conexión que se utilizará en las operaciones posteriores.

```
var db = Ti.Database.open ( 'weatherDB' );
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 20. Creación de una Base de datos

Una vez creada una base de datos, el siguiente paso sería construir su estructura, esto se consigue ejecutando sentencias SQL normales como *Create Table*.

```
// Arrancar la base de datos
var db = Ti.Database.open ( 'TiBountyHunter' );
db.Execute ( 'CREATE TABLE IF NOT EXISTS fugitivos (id INTEGER PRIMARY KEY, nombre TEXTO, capturado INTEGER, TEXTO url, capturedLat REAL, capturedLong REAL);' );
db.Close ();
```

Fuente:<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 21. Creación de una tabla

Nótese en la figura que se ejecuta *db.Close()* que cierra la conexión, esto es recomendable hacerlo por razones de rendimiento.

## b) Instalación de una Base de Datos

Para la instalación de una base de datos se hace uso del método *Titanium.Database.install()*, que copia una base de datos preexistente desde el directorio *Resources* del Proyecto Titanium a una subcarpeta adecuada de la aplicación en el equipo.

```
var db = Ti.Database.install ( '/ mydata / weatherDB' , 'weatherDB' );
```

Fuente:<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 22. Instalación de una base de datos

Para tener la base de datos original se necesitará algún tipo de herramienta para crear la base de datos SQLite, como el Administrador de SQLite de Firefox.

### 3.4.2.2. Manipulación de datos

SQLite (y por extensión Titanium) es compatible con una lista bastante completa de las sentencias SQL. Estas sentencias se ejecutan utilizando el método *execute()*.



### a) Ingreso de datos

Para el ingreso de datos basta con ejecutar una sentencia SQL de *Insert* a través del método *execute()*.

```
db.execute('INSERT INTO city (name,continent,temp_f,temp_c,condition_id) VALUES (?,?,,?,?)',
          importName, importContinent, importTempF, importTempC, dbConditionId);
```

Fuente:<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 23. Ingreso de datos en base de datos

### b) Consulta de datos

Al igual que en el ingreso de datos, para la consulta basta con ejecutar una sentencia SQL *Select* tradicional.

```
var cityWeatherRS = db.execute('SELECT id,name,continent FROM city');
while (cityWeatherRS.isValidRow())
{
    var cityId = cityWeatherRS.fieldByName('id');
    var cityName = cityWeatherRS.fieldByName('name');
    var cityContinent = cityWeatherRS.fieldByName('continent');
    Ti.API.info(cityId + ' ' + cityName + ' ' + cityContinent);
    cityWeatherRS.next();
}
cityWeatherRS.close();
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 24. Consulta de datos a una base de datos

En este caso Titanium retorna un Objeto *Titanium.Database.ResultSet*, que cuenta con varias propiedades para el acceso a la información en sí.

### c) Actualización de datos

Para la actualización de una fila en una tabla se utilizaría una sentencia *Update* con los parámetros necesarios.

```
db.execute('UPDATE condition SET icon=? WHERE id=?',importIcon,dbConditionId);
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 25. Actualización de datos en una base de datos

### d) Eliminación de datos

Se puede eliminar una o más filas de una tabla utilizando una sentencia *Delete*.es importante recalcar la importancia de enviar los parámetros adecuados para estas operaciones.

```
db.Execute ( "DELETE FROM ciudad WHERE id =? ' , CityId);
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Working+with+a+SQLite+Database#WorkingwithaSQLiteDatabase-Creatingadatabase>

Figura 26. Eliminación de datos en una base de datos

SQLite da soporte además a la sentencia *Drop table* si se desea eliminar toda una tabla. Además no cuenta con soporte para *Truncate*.

### 3.4.3. Sistema de Archivos

*Titanium.Filesystem* permite leer y escribir archivos en el dispositivo del usuario. En términos generales, su aplicación se limita a leer y escribir sus propios archivos y no se puede acceder a los archivos creados por otras aplicaciones. En los dispositivos Android,

su aplicación puede guardar archivos en el almacenamiento interno o externo (tarjeta SD). Los archivos son un gran lugar para almacenar imágenes u otros datos binarios.

Dependiendo de las necesidades de la aplicación, el almacenamiento de datos como un archivo podría tener más sentido que el almacenamiento de la información en una base de datos o en una propiedad.

### 3.4.3.1. Ubicación de almacenamiento

Antes de conocer la mecánica del manejo de archivos es necesario entender los lugares a los que se puede acceder para manejar archivos.

Los siguientes lugares son potencialmente accesibles por la aplicación:

- ***Ti.Filesystem.applicationDataDirectory***: Un directorio de lectura / escritura accesible por su aplicación. Coloque sus archivos específicos de la aplicación de este directorio. El contenido de este directorio persisten hasta que elimine los archivos o hasta que el usuario desinstala la aplicación.
- ***Ti.Filesystem.resourcesDirectory***: Un directorio de solo lectura donde se encuentran los recursos de la aplicación; este directorio se corresponde con el directorio *Resources* en el proyecto de Studio. El contenido de este directorio persisten hasta que el usuario desinstala la aplicación.
- ***Ti.Filesystem.tempDirectory***: Un directorio de lectura y escritura en su aplicación puede colocar archivos temporales. El contenido de este directorio persisten hasta que su aplicación se cierra totalmente, momento en que el sistema operativo puede borrar sus archivos.
- ***Ti.Filesystem.externalStorageDirectory***: Un directorio de lectura y escritura en el dispositivo de almacenamiento externo (tarjeta SD) accesible por su aplicación, si

existe un lugar así. Compruebe primero con *Ti.Filesystem.isExternalStoragePresent()* (que devuelve un booleano). Sólo está disponible para la plataforma Android.

- ***Ti.Filesystem.applicationCacheDirectory***: Un directorio de lectura y escritura en su aplicación puede almacenar en caché los datos. El contenido de este directorio persisten después de que su aplicación se cierra totalmente, pero a discreción del sistema operativo. Para la plataforma Android, el caché se limita a 25 MB y los archivos permanecen durante toda la vida de la aplicación. Para la plataforma IOs, no hay límite de tamaño, pero los datos sólo permanece allí hasta IOs limpia el directorio si se requiere el espacio de disco.

### 3.4.3.2. Operaciones con archivos

#### a) Obtener identificador de archivo

Lo primero que se debe hacer a la hora de manipular archivos es obtener una referencia del archivo a utilizar, esto se logra con el *método* *Ti.Filesystem.getFile()*, pasando a ella la ruta y el nombre del archivo, este método retorna una instancia del objeto *Ti.Filesystem.File*.

```
var f = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'yourfile.txt' );
```

*Fuente:*

<https://wiki.appcelerator.org/display/guides2/FileSystem+Access+and+Storage#FileSystemAccessandStorage-StorageLocations>

*Figura 27. Obtención de identificador de archivo*

#### b) Lectura

Una vez que se obtiene una referencia a un archivo, el proceso de lectura es bastante simple, se utiliza el método *read()*. El contenido del archivo retorna encapsulada, lo que

es ideal si se trata de un archivo binario; pero, si es un archivo de texto se debe acceder al contenido de la capsula con la propiedad *text*, o se puede saber el tipo de archivo con la propiedad *mimeType*.

```
var f = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'yourfile.txt' );
var contenidos f.read = ();
Ti.API.info ( 'salida como una mancha:' + contenidos); // útil si los contenidos son binarios
Ti.API.info ( "texto de salida del archivo: ' contenidos.text);
Ti.API.info ( 'Salida el archivo \ ' s tipo MIME: ' contenidos.mimeType); // por ejemplo text
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Filesystem+Access+and+Storage#FilesystemAccessandStorage-StorageLocations>

Figura 28. Lectura de archivos

### c) Escritura

El proceso de escritura es igual de sencillo, basta con utilizar en método *write()*. Ahora lo que puede ser un poco más complicado es generar los datos que se pretenden escribir.

```
var f = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'emptyfile.txt' );
f.write ( "El archivo ya no está vacía! " ); // escritura al archivo
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Filesystem+Access+and+Storage#FilesystemAccessandStorage-StorageLocations>

Figura 29. Escritura de archivos

No existe un método definido para añadir datos a un archivo existente pero se puede utilizar el método *write()*, enviando como segundo parámetro un valor *true* (verdadero).

```
var log = Ti.Filesystem.getFile(Ti.Filesystem.applicationDataDirectory, 'logfile.txt');
log.write('My log file\n');
for(var i=0; i<10; i++) {
    log.write(i+' : new log statement\n', true);
}
alert(log.read().text);
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/FileSystem+Access+and+Storage#FileSystemAccessandStorage-StorageLocations>

Figura 30. Método para agregar datos a un archivo

#### d) Creación y copia de archivos

Para la creación de un archivo, en el momento en que se hace uso de la obtención de un identificador de archivo, Titanium busca el archivo y si no lo encuentra lo crea automáticamente.

Existen casos en los que sea necesario crear un archivo manualmente, utilizando código, para esto Titanium pone a disposición el método *createFile()*.

```
var f = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'nonexistent_file.txt' );
si (f.exists () === false ) {
    // No es necesario hacer esto, pero usted podría ...
    f.createFile ();
}
f.write ('escribir en el archivo sería suficiente para crearlo');
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/FileSystem+Access+and+Storage#FileSystemAccessandStorage-StorageLocations>

Figura 31. Creación de un archivo

Titanium no proporciona un método específico para copiar un archivo, en lugar de ello, se puede hacer uso de una combinación de lectura y escritura.

```
var oldfile = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'old.txt' );  
var newfile = Ti.Filesystem.getFile (Ti.Filesystem.applicationDataDirectory, 'nuevo.txt' );  
newfile.write (oldfile.read ()); // tanto old.txt y nuevo.txt existe ahora
```

*Fuente:*

<https://wiki.appcelerator.org/display/guides2/Filesystem+Access+and+Storage#FilesystemAccessandStorage-StorageLocations>

*Figura 32. Método para copiar un archivo.*

### 3.5. INTERACCIÓN CON DATOS REMOTOS

Titanium interactúa con servidores remotos a través de HTTP utilizando el objeto *Ti.Network.HTTPClient*.

Es posible utilizar HTTPClient para interactuar con muchos tipos populares de los servicios web, pero la forma más fácil de trabajar es con servicios web de estilo REST. la aplicación móvil va a interactuar con estos datos en la web usando peticiones HTTP GET o POST, aunque la gama completa de verbos HTTP que son compatibles con HTTPClient son GET, POST, PUT y DELETE.

```

1  var url = " https://www.appcelerator.com " ;
2  var xhr = Ti.Network.createHTTPClient ({
3      onload: función (e) {
4          // Esta función se llama cuando se devuelven datos desde el servidor y disponible para su uso
5          // This.responseText celebra el regreso de texto en bruto del mensaje (utilizado para el texto / JSON)
6          // This.responseXML tiene cualquier regresaron XML (incluyendo SOAP)
7          // This.responseData ejerce en cualquier dato binario devuelto
8          Ti.API.debug ( este .responseText);
9          alerta ( "éxito" );
10     },
11     onerror: función (e) {
12         // Esta función se llama cuando se produce un error, incluyendo un tiempo de espera
13         Ti.API.debug (e.error);
14         alerta ( "error" );
15     },
16     timeout: 5000 / * en milisegundos * /
17 });
18 xhr.open ( "GET" , url);
19 xhr.send (); // solicitud se envía en realidad con esta declaración

```

Fuente:

<https://wiki.appcelerator.org/display/guides2/HTTPClient+and+the+Request+Lifecycle#HTTPClientandtheRequestLifecycle-GETrequests>

Figura 33. Esquema básico para HTTPClient

### 3.5.1. Peticiones GET

Para realizar una petición existen tres pasos básicos:

1. Crear un cliente HTTP. (línea 2 en la figura 33)
2. Abrir una conexión con un recurso específico (línea 18 en la figura 33)
3. Envío de una petición HTTP. (línea 19 en la figura 33)

La mayoría de veces el simple envío de la petición de la solicitud no es útil para la aplicación, es necesario acceder a los datos que el servidor envía en el cuerpo de su respuesta. Para acceder a estos datos se pueden crear funciones específicas que se ejecutaran de acuerdo a respuesta del servidor. En el ejemplo de la figura 33, *onload* se ejecuta si se recibe una respuesta con éxito y *onerror* en caso de que exista algún error.

De acuerdo al tipo de respuesta del servidor se utiliza las funciones:



Tipo de Respuesta	Función
Texto sin formato	this.responseText
Texto en formato XML	this.responseXML
Datos binarios	this.responseData

*Fuente: <https://wiki.appcelerator.org/display/guides2/HTTPClient+and+the+Request+Lifecycle#HTTPClientandtheRequestLifecycle-GETrequests>*

*Tabla 12. Funciones utilizadas por tipo de respuesta de Servicios Web*

### 3.5.2. Peticiones POST

Generalmente es necesario enviar datos al servidor en el cuerpo de una solicitud tal como se hace en un formulario HTML estándar, esto se consigue a través de una solicitud POST. En Titanium esto se logra fácilmente enviando un objeto JavaScript en el método *send*.

```
var xhr = Ti.Network.createHTTPClient ();

xhr.onload = función (e) {
    // Respuesta mango, que como mínimo será un código de estado HTTP
};

xhr.open ( 'post' , ' http://www.myblog.com/post.php ' );
xhr.send ({
    Título: "Mi increíble el blog ' ,
    cuerpo: 'Hoy me encontré con Susy en la lavandería. Mejor día REVA \! '
});
```

*Fuente:*

*<https://wiki.appcelerator.org/display/guides2/HTTPClient+and+the+Request+Lifecycle#HTTPClientandtheRequestLifecycle-GETrequests>*

*Figura 34. Envío de datos a través de POST*

### 3.5.3. Trabajar con datos JSON

El mejor formato de transporte de datos para su uso con JavaScript (y por tanto de Titanium) es JavaScript Object Notation o JSON. JSON es un gran beneficio para aplicaciones de JavaScript, ya que se puede serializar muy fácilmente. Además, dado que

JSON es un formato de transporte liviano, se tarda menos tiempo y consume menos ancho de banda para transferir datos, que puede llegar a ser importante en redes de datos de baja velocidad.

Si se accede a un servicio web que tiene un formato JSON como respuesta, los datos de la respuesta se pueden obtener con la función *JSON.parse()*.

```
var url = " http://example.com/json.txt " ;
var json;

var xhr = Ti.Network.createHTTPClient ({
  onload: la función () {
    // Analizar los datos recuperados, convirtiéndolo en un objeto JavaScript
    json = JSON.parse ( este .responseText);
    // ...
  }
});
```

Fuente: <https://wiki.appcelerator.org/display/guides2/Working+with+JSON+Data>  
Figura 35. Recepción de datos en formato JSON

Para el envío de datos se puede utilizar la función *JSON.stringify()*, aunque Titanium hace esta conversión automáticamente por lo que no es necesario.

```
var blogpost = {
  Título: "Mi increíble el blog " ,
  cuerpo: 'Hoy me encontré con Susy en la lavandería. Mejor día REVA \!' '
};

var xhr = Ti.Network.createHTTPClient ({
  onload: la función () {
    // Manejar la respuesta
  }
});

xhr.open ( 'post' , ' http://www.myblog.com/post.php ' );
// Opcional:
// Blogpost = JSON.stringify (blogpost);
xhr.send (blogpost);
```

Fuente: <https://wiki.appcelerator.org/display/guides2/Working+with+JSON+Data>  
Figura 36. Envío de datos en formato JSON.

### 3.5.4. Trabajar con datos XML

Aunque JSON es el formato ideal para trabajar con Titanium, existen varios servicios web que utilizan XML para trabajar con aplicaciones cliente. Titanium ofrece un conjunto de métodos para trabajar con XML dentro de su estación de nombre *Titanium.XML*.

Si un servicio web tiene como respuesta datos en formato XML, Titanium de serializa automáticamente esta respuesta para su uso.

```
xhr.onload = función (e) {  
    var doc = this.responseXML.documentElement;  
    // Este es el objeto de documento XML  
  
    // Usar la API DOM para analizar el documento  
    var elementos = doc.getElementsByTagName ( "SOMETAG" );  
};
```

Fuente: <https://wiki.appcelerator.org/display/guides2/Working+with+XML+Data>  
Figura 37. Manejo de respuesta con formato XML.

Un documento XML está formado por nodos, Algunos tipos de nodos pueden tener nodos secundarios, mientras que otros son nodos hoja que no tienen hijos. Para los documentos XML y HTML, estos son algunos de los tipos de nodos compatibles:

- **Documento:** representa todo el documento XML y contiene un máximo de un nodo Elemento
- **Elemento:** contiene el resto de los nodos en el documento, esto es generalmente el nodo se trabaja con que el nivel superior de la jerarquía
- **Attr:** representa los atributos de un nodo y contiene los nodos de texto
- **Texto:** un tipo de nodo hoja sin hijos
- **CDATA:** Sección sin hijos

➤ **Entidad:** contiene nodos de varios tipos

Titanium proporciona los siguientes métodos para analizar un documento XML.

Método	Descripción
getElementsByTagName(nodo)	Método que devuelve una matriz de nodos con el nombre dado
item()	Método que se utiliza para seleccionar un nodo específico en una matriz.
getAttribute()	Método que se utiliza para recuperar el valor de un atributo con el nombre dado
text, nodeValue	que recuperan los valores de nodo hoja asociados con el nodo

Fuente: <https://wiki.appcelerator.org/display/guides2/Working+with+XML+Data>  
*Tabla 13. Métodos para analizar documentos XML*

A continuación un ejemplo de la utilización de estas funciones, que recibe datos de un blog de noticias.

```

xhr.onload = function() {
    // Los datos se devuelve desde el blog, iniciar el análisis
    var xml = this.responseXML;
    // Título del blog es en un nodo denominado "canal"
    var channel = xml.documentElement.getElementsByTagName ( "canal" );

    // Comenzar recorriendo las entradas del blog
    var data = [];
    // entradas del blog están en nodos denominan "ítem"
    var items = xml.documentElement.getElementsByTagName("item");
    for (var i=0;i<items.length;i++) {
        data.push({
            postTitle: items.item(i).getElementsByTagName("title").item(0).text
            postLink: items.item(i).getElementsByTagName("link").item(0).text
        });
    }
    // Desencadenar un evento de nivel de aplicación para notificar a la interfaz
    Ti.App.fireEvent('net:rssDataReturned',{
        blogTitle: channel.item(0).getElementsByTagName("title").item(0).text,
        blogPosts: data
    });
};

```

Fuente: <https://wiki.appcelerator.org/display/guides2/Working+with+XML+Data>  
 Figura 38. Ejemplo de manejo de documento XML.

Un punto muy importante al momento de analizar un documento XML es que se debe conocer la estructura jerárquica de los nodos que componen el documento.

### 3.6. SERVICIOS DE LOCALIZACIÓN

Las API de posición y rumbo son parte del módulo *Ti.Geolocation*, que contiene todas las funciones, propiedades y eventos necesarios para manejar la información de ubicación. Ese espacio de nombres se divide en los espacios de nombres *Ti.Geolocation.Android* y *Ti.Geolocation.MobileWeb*, que proporcionan características específicas de la plataforma.

### **3.6.1. Consideraciones del desarrollo**

#### **3.6.1.1. Consideraciones del desarrollo IOs**

A los usuarios de IOs se les solicita conceder o denegar el permiso cuando la aplicación intenta utilizar la información de geo localización. De acuerdo con las directrices de Apple, debe proporcionar un mensaje personalizado para contar con mayor claridad a los usuarios de por qué estás solicitando su ubicación. Debe establecer el *Ti.Geolocation.purpose* propiedad igual a la cadena que se mostrará a los usuarios.

#### **3.6.1.2. Consideraciones del desarrollo Android**

En general, las pruebas de código de geo localización se deben hacer en un dispositivo físico de modo que se pueda probar la aplicación en un entorno cercano a un escenario de uso del mundo real. Si se va a probar el código en el emulador, debe asegurarse de que la AVD (del inglés Android Virtual Device, Dispositivo Virtual Android) incluye emulación de GPS, y usted necesita enviar su ubicación simulando coordenadas a ese AVD.

#### **3.6.1.3. Consideraciones del desarrollo Web Móvil**

Para trabajar geo localización en web móvil, el usuario debe ejecutar un navegador que soporte la API de geo localización de W3C. La aplicación efectiva de geo localización, y la precisión proporcionada, dependen del navegador del usuario, así como el dispositivo en el que se están ejecutando ese navegador. Al igual que con IOs, las aplicaciones web móviles le sugerirán usuarios conceder permiso para acceder a su ubicación. Sin embargo, no se puede cambiar el mensaje que se muestra a los usuarios. Ese mensaje depende del navegador se utiliza.

### 3.6.2. El uso de los servicios de localización en la App

El uso de los servicios de localización general tiene los siguientes pasos:

1. Determinar si los servicios de localización están habilitados
2. Configurar la precisión de los servicios
3. Habilitar una sola escucha o un monitoreo constante.
4. Con una ubicación-oyente, gestionar activamente el oyente en coordinación con el ciclo de vida de la aplicación.

#### 3.6.2.1. Determinar si los servicios de localización están habilitados

Para determinar si los servicios de localización estarán disponibles para usted en el dispositivo móvil actual, sólo hay que comprobar la propiedad booleana *Ti.Geolocation.locationServicesEnabled*.

```
if (Ti.Geolocation.locationServicesEnabled) {  
    // Realizar otras operaciones con Ti.Geolocation  
} else {  
    alert( "Por favor, activa los servicios de ubicación " );  
}
```

*Fuente:*

<https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>  
*Figura 39. Determinar servicios de localización*

#### 3.6.2.2. Configurar la precisión de los servicios

La configuración de la precisión depende de la plataforma para cual se realiza la aplicación.

## a) Configuración en IOs

En IOs la configuración de la precisión de la geo localización está dada por el valor de la propiedad *Ti.Geolocation.accuracy*. De acuerdo a la configuración de esta propiedad IOs automáticamente hace un ajuste para determinar la frecuencia con la que se actualizara la ubicación.

Las propiedades que se pueden configurar son las siguientes:

- **Accuracy**: representa la precisión que debe tener la ubicación, puede tomar los valores determinados en la siguientes constantes:
  - **ACCURACY\_BEST** - Los datos de localización serán de la máxima precisión posible de que el dispositivo es capaz
  - **ACCURACY\_HUNDRED\_METERS** - Los datos de localización serán precisos dentro de los 100 metros
  - **ACCURACY\_KILOMETER** - Los datos de localización serán exactos a 1 kilómetro
  - **ACCURACY\_NEAREST\_TEN\_METERS** - Los datos de localización serán precisos dentro de los 10 metros
  - **ACCURACY\_THREE\_KILOMETERS\_** - Los datos de localización serán exactos a 3 kilómetros
- **distanceFilter** - El cambio mínimo de la posición (en metros) antes de que el evento de actualización se active. El valor predeterminado es 0, lo que significa que los eventos de localización se generan continuamente.
- **headingFilter** - El cambio minio de la posición (en grados) antes de que el evento de actualización se active. El valor predeterminado es 0, lo que significa que los eventos se generan continuamente.



➤ **preferredProvider** - Le permite especificar el método preferido para la recepción de una ubicación. Los siguientes constantes representan sus posibles opciones:

- PROVIDER\_NETWORK - Dar a la preferencia a la ubicación basada la red.
- PROVIDER\_GPS - Dar la preferencia de localización GPS.

```
if (Ti.Geolocation.locationServicesEnabled) {
    Ti.Geolocation.purpose = 'Get Current Location';
    Ti.Geolocation.accuracy = Ti.Geolocation.ACCURACY_BEST;
    Ti.Geolocation.distanceFilter = 10;
    Ti.Geolocation.preferredProvider = Ti.Geolocation.PROVIDER_GPS;

    Ti.Geolocation.addEventListener('location', function(e) {
        if (e.error) {
            alert('Error: ' + e.error);
        } else {
            Ti.API.info(e.coords);
        }
    });
} else {
    alert('Please enable location services');
}
```

*Fuente:*

<https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>

Figura 40. Configuración de geo localización en IOs.

## b) Configuración en Android

Android ofrece un modelo de geo localización mucho más rica, con varios proveedores de localización, filtros de distancia, frecuencias de actualización, y así sucesivamente. Existen tres modo de servicio de ubicación diferente compatibles con Android: legado, manual, y simple.

➤ **El modo tradicional** es el modo que existía antes de la versión 2.0. El modo tradicional se utiliza cuando se establece la propiedad precisión a una de las constantes

EXACTITUD IOs (ver arriba). Este modo está en desuso en Android y no se debe utilizar para un nuevo desarrollo.

- **El modo manual** ofrece a los desarrolladores un control de bajo nivel de cambios de ubicación, inclusive permitiendo a los proveedores de localización individuales y filtrar las actualizaciones, para la mejor combinación de precisión y duración de la batería. Se utiliza cuando la propiedad *Titanium.Geolocation.Android.manualMode* se establece en verdadero. En el modo manual, la propiedad exactitud no se utiliza, y toda la configuración se realiza a través del módulo *Titanium.Geolocation.Android*.
- **Modo simple** proporciona un modo de compromiso que ofrece un apoyo adecuado a aplicaciones poco exigentes de localización sin necesidad de que los desarrolladores escribir mucho código. La configuración de *Ti.Geolocation.accuracy* a ACCURACY\_HIGH o ACCURACY\_LOW activa el modo simple. En este modo la plataforma se encarga de activar y desactivar los proveedores de ubicación y actualizaciones de ubicación de filtrado.

```
// demonstrates manual mode:
var providerGps = Ti.Geolocation.Android.createLocationProvider({
  name: Ti.Geolocation.PROVIDER_GPS,
  minUpdateDistance: 0.0,
  minUpdateTime: 0
});
Ti.Geolocation.Android.addLocationProvider(providerGps);
Ti.Geolocation.Android.manualMode = true;
var locationCallback = function(e) {
  if (!e.success || e.error) {
    Ti.API.info('error:' + JSON.stringify(e.error));
  } else {
    Ti.API.info('coords: ' + JSON.stringify(e.coords));
  }
};
Titanium.Geolocation.addEventListener('location', locationCallback);
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>  
Figura 41. Configuración de geo localización en Android

### c) Configuración en web móvil

Los servicios de ubicación en Web Móvil operan de manera similar a las operaciones del modo simples en Android. Configurando propiedad *accuracy* a `ACCURACY_HIGH` produce las mejores actualizaciones de ubicación disponibles, con el consumo de energía más alta. Usando `ACCURACY_LOW` ofrece actualizaciones de ubicación de menor calidad con menor consumo de energía. Además de la configuración de la precisión, hay varias opciones de configuración específica en la Web móvil.

```
if (Ti.Geolocation.locationServicesEnabled) {
  Ti.Geolocation.accuracy = Ti.Geolocation.ACCURACY_HIGH;
  Ti.Geolocation.MobileWeb.maximumLocationAge = 15000; // in milliseconds
  Ti.Geolocation.addEventListener('location', function(e) {
    if (e.error) {
      alert('Error: ' + e.error);
    } else {
      Ti.API.info(e.coords);
    }
  });
} else {
  alert('Please enable location services');
}
```

Fuente:

<https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>  
Figura 42. Configuración de geo localización en web móvil.

#### 3.6.2.3. Obtener la posición actual

Asumiendo que la aplicación está configurada para utilizar el nivel apropiado de la configuración específica de la plataforma de geo localización, se puede acceder a la posición actual con el método *Titanium.Geolocation.getCurrentPosition* que devuelve los datos en su propiedad *coords*.

```

if (Ti.Geolocation.locationServicesEnabled) {
    Titanium.Geolocation.purpose = 'Get Current Location';
    Titanium.Geolocation.getCurrentPosition(function(e) {
        if (e.error) {
            Ti.API.error('Error: ' + e.error);
        } else {
            Ti.API.info(e.coords);
        }
    });
} else {
    alert('Please enable location services');
}

```

Fuente: <https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>  
 Figura 43. Obtención de la posición actual

#### 3.6.2.4. Monitorear continuamente la posición GPS

A menudo es necesario conocer la ubicación del dispositivo continuamente, el ejemplo más común es en aplicaciones para navegación en direcciones de conducción o manejo.

Para tener esto en Titanium simplemente se necesita registrar el evento *location* en el módulo *Ti.Geolocation*.

```

if (Ti.Geolocation.locationServicesEnabled) {
    Titanium.Geolocation.purpose = 'Get Current Location';
    Titanium.Geolocation.addEventListener('location', function(e) {
        if (e.error) {
            alert('Error: ' + e.error);
        } else {
            Ti.API.info(e.coords);
        }
    });
} else {
    alert('Please enable location services');
}

```

Fuente: <https://wiki.appcelerator.org/display/guides2/Tracking+Position+and+Heading>  
 Figura 44. Monitoreo continuo de la posición.

## **3.7. EVALUACIÓN DE TITANIUM APPCELERATOR**

### **3.7.1. Criterios de la evaluación**

Para determinar los criterios de evaluación de la herramienta Titanium Appcelerator se ha utilizado los criterios establecidos en la norma ISO 9126 que es un estándar internacional para la evaluación de la calidad del software.

### **3.7.2. Funcionalidad**

Se entiende por funcionalidad, la aceptación funcional que poseen las actividades que son realizadas para conseguir el despliegue de una aplicación en un dispositivo móvil. El proceso se basa en el inicio de un nuevo proyecto, la realización de test, el proceso de compilación y la puesta en producción en un terminal móvil.

#### **3.7.2.1. Evaluación**

##### **¿Qué tan fácil es crear un proyecto en Appcelerator?**

El primer paso a realizar es la configuración de los entornos para los cuales se desea desplegar las aplicaciones, para lo cual es indispensable tener los SDK de cada una de las plataformas, este proceso es similar al que tendría que hacerse si se desea realizar un proyecto en las plataformas nativas; es decir, tener instalado Xcode para desarrollar aplicaciones de IOS o tener instalado los SDK de Android para desarrollar aplicaciones en Eclipse.

### **¿Cómo se realizar el proceso de testeo y compilación de un proyecto?**

Para la realización de testeo y compilación, Appcelerator presenta funcionalidades similares a las de muchos entornos de desarrollo de aplicaciones, facilitando procesos de depuración con puntos de interrupción.

### **¿Qué tan difícil es poner la aplicación en un terminal móvil?**

En cuanto a la puesta en producción en un terminal móvil, los procesos son establecidos por las tiendas de distribución de aplicaciones, entonces este proceso es similar al que se debe seguir si se desarrolla en los entornos nativos.

### **¿Qué grado de dificultad conlleva codificar una aplicación en Appcelerator?**

La codificación en Appcelerator se la realiza en lenguaje JavaScript, por lo que a criterio personal, es mucho más fácil aprender un solo lenguaje de programación y aplicarlo en Appcelerator para obtener aplicaciones funcionales tanto en IOs como en Android, a tener que aprender un lenguaje para cada tipo de Aplicación; Java para el caso de Android y Objective-C para el caso de IOs, si se desea codificar en los entornos nativos de las aplicaciones.

#### **3.7.3. Fiabilidad**

Se entiende por fiabilidad, la precisión con la que se pueden realizar aplicaciones en un entorno Titanium. Es decir, que interfaces gráficos se pueden representar y cuáles no. Si funcionan de igual manera en un entorno u otro.

### 3.7.3.1. Evaluación

**¿Qué tal fiables son los resultados obtenidos al momento de generar las aplicaciones con Appcelerator?**

Para dar contestación a esta interrogante se presenta un análisis de cada uno de los elementos gráficos que se ha utilizado en el desarrollo de la aplicación propuesta y su comportamiento al momento de ejecutarla en los dispositivos móviles o Emuladores de Android y iPhone.

- **ImageView**

La vista de imagen sirve para representar una imagen o un conjunto de ellas. Admite los formatos PNG y JPEG. Su funcionamiento es correcto tanto para IOs como para Android.

- **Label**

Las etiquetas tienen un funcionamiento completamente igual y funcionan correctamente, también se pueden aplicar los esteticismos en IOs o Android, como añadir bordes, redondearlos, etc.

- **TextField**

Los cuadros de texto son elementos importantes a la hora de que el usuario necesite ingresar algún tipo de información, funcionan adecuadamente en IOs y Android, además se pueden aplicar atributos para personalizar su apariencia, como bordes, tipo de teclado que se desea utilizar, o si es un texto de contraseña.

- **Picker**

El elemento Picker es un elemento de selección, en la aplicación propuesta se lo utilizar al momento de seleccionar la empresa a la que pertenece un usuario.

Su comportamiento funcional es correcto en todas las plataformas; no así su presentación en pantalla, este varía de una plataforma a otra, en IOs este control tiene las propiedades de ancho y largo con valores estáticos por lo que es necesario establecer una correcta posición dentro de la pantalla que lo contiene para evitar que se sobreponga con otros elementos utilizados, en Android es mucho más sencillo utilizarlo ya que se puede asignar las propiedades de ancho y largo, y la presentación de las opciones se realiza en otra vista separada de la principal.

- **Button**

Los botones también funcionan adecuadamente en IOs y Android, de igual manera sus estilos se pueden personalizar, si no se les establece algunas propiedades estas son heredadas de los estilos de las plataformas, por esto su presentación gráfica puede cambiar de una plataforma a otra.

- **TabGroup**

Este elemento sirve para agrupar las pestañas que el desarrollador haya creado y permite al usuario interactuar con estas, su funcionamiento es correcto en ambas plataformas, aunque en Android existen propiedades graficas como el color de fondo, que no pueden ser asignadas ya que las heredan de la propia plataforma.



- **Tab**

Cada elemento Tab representa una pestaña y sirve para establecer diferentes secciones de presentación, cada uno con sus propias características funcionales. Este elemento funciona adecuadamente en Android e IOs.

- **TableView**

Este es un elemento que sirve de contenedor de filas de una tabla, y funciona correctamente.

- **TableViewCell**

Este elemento representa una fila de una tabla y permite insertar dentro de cada una de estos elementos de texto, gráficos y botones, es necesario especificar las dimensiones de los controles insertados en cada fila para evitar que se superpongan entre sí. Funcionan adecuadamente tanto en IOs como en Android.

- **View**

Las vistas son elementos muy utilizados en Appcelerator y sirven como contenedores de otros controles, pueden ser personalizadas. Funcionan correctamente.

- **Window**

El componente Window es imprescindible para el desarrollo de aplicaciones en Titanium. Cada pestaña tiene su propia ventana en donde se agregan los demás tipos de elementos para presentar al usuario. Funcionan adecuadamente en IOs y Android.

En general los resultados obtenidos al generar aplicaciones para Android e IOs con la plataforma Appcelerator son bastante funcionales, salvo con la utilización de ciertos

controles que necesitan de especificaciones determinadas para ser utilizadas en las diferentes plataformas.

### **3.7.4. Usabilidad**

Se entiende por usabilidad, la facilidad de utilizar el entorno de desarrollo para crear aplicaciones. Para ello, se comprobará la documentación y recursos que Appcelerator pone a disposición de su comunidad de desarrolladores, así como lo usable que es.

#### **3.7.4.1. Evaluación**

**¿Existe una documentación completa a disposición de la comunidad de desarrolladores de Titanium?**

La plataforma Appcelerator pone a disposición una gran cantidad de documentación muy bien organizada en su sitio web [https://docs.appcelerator.com/platform/latest/#!/guide/Titanium\\_Development](https://docs.appcelerator.com/platform/latest/#!/guide/Titanium_Development) y que se actualiza constantemente cuando algún desarrollador encuentra fallos a la hora de implementar soluciones con esta herramienta.

Esta documentación posee un sistema de navegación muy intuitivo y bastante completo. Existe un buscador, una ruta del paquete, una lista de componentes, una lista de compatibilidades y la descripción y ejemplos del componente seleccionado.

Los ejemplos son simplistas, y están claramente diseñados para ser utilizados a modo de “copia y pega”, indicando a los desarrolladores dónde debería ir el código para ser adaptado a unas necesidades específicas.

Los componentes que se van consultando, se despliegan a través de una serie de pestañas, lo que hace ahorrar bastante tiempo cuando se necesita consultar más de un componente en la documentación.

También se pone a disposición de la comunidad de desarrolladores a través de la documentación un buscador inteligente que funciona con bastante precisión.

Esta documentación se encuentra únicamente en idioma Inglés.

También existe en google una comunidad bastante activa de desarrolladores en español, en donde se puede hacer consultas y compartir ejemplos.

### **3.7.5. Eficiencia**

Se entiende por eficiencia de Titanium los tiempos de compilación y carga de las aplicaciones desarrolladas en la plataforma.

#### **3.7.5.1. Evaluación**

En este apartado, al no contar con datos de otras herramientas o de las herramientas nativas, que permitan realizar una comparación en cuanto a los tiempos de compilación y carga, no se puede dar un criterio exacto.

Cabe indicar que Appcelerator en el proceso de compilación toma la codificación en JavaScript y genera proyectos que pueden ser abiertos en las herramientas nativas de cada tipo de aplicación móvil; esto es, un proyecto Xcode para IOs y un proyecto para eclipse en Android y luego genera las aplicaciones en si utilizando los SDK's de cada plataforma, por lo que los tiempos deben ser mayores a los obtenidos en las herramientas nativas.

A esto se debe sumar los tiempos que se demora en abrir los emuladores utilizados para hacer pruebas sobre la aplicación, que en Android es mucho más demoroso que en IOs.

### **3.7.6. Mantenibilidad**

Se entiende por mantenibilidad el proceso de actualización y de corrección de errores de la plataforma.

#### **3.7.6.1. Evaluación**

La plataforma Appcelerator pone constantemente actualizaciones a disposición de los usuarios, estas actualizaciones presentan mejoras en la utilización de los controles y herramientas, además que van a la par de nuevas actualizaciones que aparecen en los SDK´s de las plataformas nativas; es decir, genera una actualización para poder lanzar aplicaciones utilizando los nuevos SDK´s de IOs y Android.

Algunas actualizaciones también han sido un tanto problemáticas, especialmente al trabajar sobre el Sistema Operativo Windows, debido a que al momento de instalar las actualizaciones se des configuran las rutas de los SDK`s de Android y se presentan errores al momento de compilar, por lo que se debe volver a configurar estas rutas.

### **3.7.7. Portabilidad**

Se entiende por portabilidad la facilidad de instalación de la herramienta en diferentes sistemas operativos y el funcionamiento en cada uno de ellos.

### **3.7.7.1. Evaluación**

El framework de desarrollo Appcelerator se puede instalar en diferentes Sistemas Operativos como son Mac OS, Windows y Linux, y los proyectos generados son fácilmente trasladables de un sistema operativo a otro. En cada uno de estos sistemas operativos de tiene instalar también los respectivos SDK's para las diferentes plataformas móviles.

Ahora cada sistema operativo presenta ciertas limitaciones a la hora de generar las aplicaciones, tal es el caso de las aplicaciones para IOs que únicamente se pueden generar desde un sistema operativo Mac OS y en el caso de querer generar un aplicación a Windows Phone se lo debe realizar desde Windows.

## **3.8. VENTAJAS Y DESVENTAJAS**

### **3.8.1. Ventajas**

El desarrollo de aplicaciones con el framework Titanium presenta una serie de ventajas en diferentes ámbitos o áreas, como la optimización de recursos, el ahorro de costes económicos y de horas de trabajo. Con ella se consigue:

- Ahorrar directamente en licencias por tratarse de un programa gratuito.
- Crear aplicaciones que lleguen al máximo número de potenciales usuarios con el menor esfuerzo.
- Ahorrar tiempo y costes en la creación, desarrollo y mantenimiento de diferentes versiones.
- Aprovechar una amplia y variada documentación proporcionada por la plataforma como manuales, videos, webinars, etc. Así como la experiencia de la comunidad de

usuarios, que constantemente comparten consejos y trucos, además de colaborar en la resolución de errores.

- Permite crear aplicaciones nativas con las que se puede interactuar con los distintos componentes del dispositivo como el GPS, la cámara, el directorio de contactos, etc.
- Ofrece la posibilidad de tener aplicaciones extensibles, además de poder mejorar las aplicaciones constantemente.

### **3.8.2. Desventajas**

Como todo framework de desarrollo Titanium muestra algunas desventajas para su utilización, como:

- Debido a las políticas de Apple para poder generar aplicaciones para IOs, es necesario trabajar con Titanium desde un equipo con sistema operativo MAC OSx.
- En Titanium no es posible realizar una maquetación ya que no hay una página HTML inicial donde añadir los controles, sino que hay que crear las ventanas y controles directamente desde JavaScript.
- Al tener un enfoque multiplataforma es necesario conocer que componentes funcionan o no en determinadas plataformas.
- Presenta constantes actualizaciones que en algunas ocasiones generan errores al momento de utilizarlas.

## CAPÍTULO IV



### **METODOLOGÍA DE DESARROLLO MOBILE-D**

---

- Historia y Definición
  - Fases de la Metodología
  - Elementos de la Metodología
  - Justificación
-

## **4.1. HISTORIA Y DEFINICIÓN**

Se podría pensar que Mobile-D es una creación un tanto antigua, ya que se desarrolló como parte de un proyecto finlandés, ICAROS, allá por 2004. Sin embargo, vale la pena mencionarlo por dos razones.

Primera: fue creado mediante un proyecto de cooperación muy estrecha con la industria. El grueso del trabajo fue realizado por los investigadores del VTT<sup>15</sup>. Aun así la metodología de diseño se elaboró con una participación importante de las empresas de TI finlandesas. Esto consiguió que la investigación llevada a cabo no se alejara demasiado de las reglas de desarrollo de las aplicaciones comerciales.

Segundo, Mobile-D es una mezcla de muchas técnicas. Tal como se verá luego, los investigadores no dudaron en echar mano de las prácticas habituales de desarrollo software. Pero, al mismo tiempo, consiguieron crear una contribución original para el nuevo escenario del desarrollo de aplicaciones para sistemas móviles. Este ejemplo ilustra perfectamente cómo se pueden usar conjuntamente diferentes metodologías y técnicas en el contexto del desarrollo ágil.

### **4.1.1. Definición**

Mobile-D <sup>TM</sup> es la metodología de VTT para el desarrollo de software ágil. Además del desarrollo de software para móviles, es conveniente para los varios contextos, por ejemplo, seguridad, finanzas, logística y aplicaciones de simulación de productos

---

<sup>15</sup> *VTT Research Centre Ltd. es la empresa líder de la investigación y la tecnología de la región nórdica de Europa.*



### 4.1.2. Motivación

La metodología se creó en un periodo de intenso crecimiento en el terreno de las aplicaciones móviles. Por tanto, en ese momento no existían demasiados principios de desarrollo a los que acudir. Los autores de Mobile-D apuntan a la necesidad de disponer de un ciclo de desarrollo muy rápido para equipos muy pequeños. De acuerdo con sus suposiciones, Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si trabajan con el ciclo de desarrollo propuesto, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas.

### 4.1.3. Principios básicos

La aproximación de Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: eXtreme Programming (XP)<sup>16</sup>, Crystal methodologies<sup>17</sup> y Rational Unified Process (RUP)<sup>18</sup>. Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron un input muy valiosos en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida (Blanco, Camarero, Fumero , Wertesky, & Rodriguez, 2013).

---

<sup>16</sup> *eXtreme Programming (Programación extrema de ahora en adelante, XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck.*

<sup>17</sup> *Crystal methodologies familia de metodologías descritas por Alistair Cockburn y se considera un ejemplo de metodologías ágiles.*

<sup>18</sup> *Rational Unified Process (RUP) es un framework iterativo del proceso de desarrollo de software creado por el Rational Software Corporation, una división de IBM desde el año 2003.*

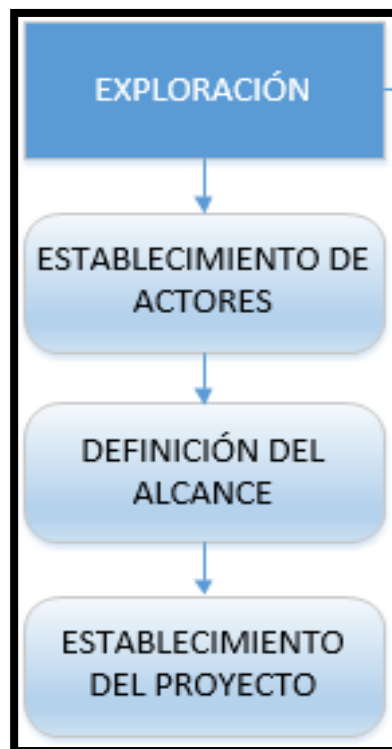
## 4.2. FASES



Fuente: Propia  
Figura 45. Fases de desarrollo Mobile-D

El desarrollo de aplicaciones siguiendo el enfoque de Mobile-D se compone de cinco fases como es muestra en la Figura 3: Exploración, Inicialización, Producción, Estabilización y Pruebas del Sistema. Cada una de estas fases tiene asociadas etapas tareas y prácticas.

### 4.2.1. Exploración



Fuente: Propia  
Figura 46. Etapas de la Fase de Exploración

#### **4.2.1.1. Motivación**

El propósito de la fase de Exploración es la planificación y el establecimiento del proyecto. "Una buena planificación es la mitad del trabajo" es un dicho para ser recordado también en el contexto de desarrollo de software. La fase de Exploración puede estar disponible oportunamente para las últimas fases de Mobile-D y también superponerse a la fase de 0 iteración. La fase de Exploración es una fase importante para establecer las bases para la implementación controlada en relación con el desarrollo del producto de software, por ejemplo, las cuestiones relacionadas con la arquitectura del producto, proceso de desarrollo de software y la selección de medio ambiente. Se necesitan diferentes grupos de partes interesadas para ofrecer su experiencia en la fase de Exploración.

#### **4.2.1.2. Objetivos**

Los objetivos de la fase de exploración son:

- Establecer los grupos de actores necesarios en la planificación y el seguimiento del proyecto de desarrollo de software
- Definir y acordar los objetivos y el alcance del proyecto de desarrollo de software, y
- Planificar el proyecto en cuanto a medio ambiente, el personal y las cuestiones de proceso.

#### **4.2.1.3. Etapas**

Las etapas individuales de la fase de Exploración son:

#### **d) Establecimiento de las partes interesadas**

El propósito de esta etapa es identificar y establecer los grupos de interés que se necesitan en diversas tareas de Explore fase, así como en el apoyo a las actividades durante el desarrollo de software - excluyendo el propio equipo de desarrollo de software. Se necesita una gran variedad de conocimientos y cooperación para obtener una aplicación controlada y una electiva implementación del producto de software.

Para el cumplimiento de este propósito Mobile-D establece un grupo de tareas que se detallan a continuación:

##### **➤ Establecimiento del cliente**

El objetivo de esta tarea es establecer el grupo de interés de los clientes que tiene la experiencia definitiva, el conocimiento del dominio y la autoridad de los requisitos para el producto de software.

##### **➤ Creación de grupos de interés**

Este procedimiento es bastante general y depende en gran medida del proyecto y el entorno donde el proyecto está a punto de ponerse en práctica. Así que en móvil-D, no hay ninguna descripción específica para este procedimiento.

#### **e) Definición del Alcance**

Definición del Alcance es una etapa en la que los objetivos y el alcance del proyecto de desarrollo de software están definidos y acordados por los grupos de interés. Esto incluye temas como los requisitos para el producto y la línea de tiempo del proyecto.

Las tareas que se desarrollan en esta etapa son las siguientes:

### ➤ **Colección de requisitos iniciales**

El propósito de esta tarea es producir una definición general inicial de alcance del producto, propósito y funcionalidad. Esto es necesario para realizar la planificación y establecimiento del proyecto (tamaño, cuestiones técnicas, arquitectura, etc.). Además, los requisitos documentados serán el punto de partida para el equipo del proyecto para construir una visión de conjunto sobre el producto en cuestión. El cliente y el grupo de dirección deben acordar y documentar la funcionalidad central del producto como se ve desde el punto de vista del cliente. Además, también otros requisitos, tales como los propios requerimientos de la organización, y las limitaciones del desarrollo del producto deben ser identificados, acordados y documentados.

### ➤ **Planificación del proyecto inicial**

El objetivo de esta tarea es establecer el plan inicial para el proyecto de desarrollo de software con respecto a la línea de tiempo, el ritmo y las inversiones del proyecto. Esto se hace a fin de permitir el establecimiento ulterior del proyecto.

### **f) Establecimiento del Proyecto**

El propósito de esta etapa es definir y asignar los recursos (tanto técnicos y humanos) necesarios para el desarrollo del proyecto de software. Asimismo, el establecimiento del proceso de la línea de base para el proyecto es una tarea importante de esta etapa. La fase de establecimiento proyecto se da con el fin de asegurarse que el equipo del proyecto puede iniciar el desarrollo de software real y sin retrasos causados por, por ejemplo, herramientas desaparecidas y una formación adecuada.

Mobile-D propone una secuencia de tareas para el cumplimiento de esta etapa.

➤ **Selección del entorno**

Es una tarea donde se planifica todo lo concerniente al entorno técnico del proyecto (incluyendo, por ejemplo, los dispositivos de destino, herramientas de desarrollo y plataformas para el producto de software para móviles), así como los recursos humanos y el medio ambiente de trabajo necesarios en el proyecto.

➤ **Asignación de personal**

Es una tarea en la que el equipo de desarrollo de software, así como el equipo de apoyo están definidos y asignados para el proyecto. Por lo tanto, la selección del ambiente puede dar una pauta de las competencias necesaria en el proyecto.

➤ **Definición de la línea de Arquitectura**

El objetivo de esta tarea es conseguir la confianza suficiente en las cuestiones arquitectónicas para que el proyecto pueda ser realizado exitosamente.

➤ **Establecimiento de procesos**

Esta tarea incluye, en primer lugar, la identificación y la definición de la formación necesaria entre el equipo del proyecto en relación con el proceso, así como las cuestiones técnicas. En segundo lugar, esta tarea incluye la adaptación de un proceso de base para que se adapte al proyecto. Además, cuestiones como la supervisión de los proyectos (incluidas las cuestiones de garantía de calidad, la documentación del proyecto, así como las prácticas, métricas y herramientas para la vigilancia) se han de definir en esta etapa.

## 4.2.2. Inicialización



*Fuente: Propia*

Figura 47. Etapas de la Fase de Inicialización

### 4.2.2.1. Motivación

El propósito de la fase de inicialización es permitir el éxito de las próximas fases del proyecto, mediante la preparación y verificación de todos los problemas críticos del desarrollo, a fin de que todas estén en plena disposición al final de la fase de implementación de los requisitos seleccionados por el cliente.

### 4.2.2.2. Objetivos

Los objetivos de la fase de inicialización son:

- Obtener una buena comprensión global del producto por parte del equipo del proyecto sobre la base de los requisitos iniciales y descripciones de línea de arquitectura.

- Preparar los recursos físicos, técnicos y humanos, así como la comunicación con el cliente, los planes del proyecto y todas las cuestiones fundamentales del desarrollo a fin de que todas estén disponibles para incorporar los requisitos seleccionados por el cliente durante las próximas fases del proyecto.

#### **4.2.2.3. Etapas y tareas**

Para el garantizar el cumplimiento de la fase de inicialización, Mobile-D establece varias etapas, que se detallan a continuación.

##### **a) Configuración del proyecto**

En esta etapa se determinan la configuración de los recursos físicos y técnicos para el proyecto, así como el ambiente para el seguimiento del proyecto, el entrenamiento al equipo del proyecto, según sea necesario, y se establecen las formas específicas para comunicarse con el grupo de clientes.

Todas las tareas de la configuración del Proyecto incluyen la participación del equipo del proyecto y se describen así:

##### **➤ Configuración del Ambiente**

Es esta tarea es donde el entorno técnico y físico para el proyecto se deben establecer. Esta tarea puede implicar también a los desarrolladores de software, especialmente en relación con el entorno técnico que puede ser parte de la tarea de formación.

##### **➤ Formación**

Esta es una tarea en la que el equipo de desarrollo de software es capacitado de acuerdo a las necesidades específicas. Esto puede incluir la capacitación en temas de



procesos de desarrollo de software, la obtención de datos para el seguimiento del proyecto, así como cuestiones técnicas relacionadas con el propio desarrollo de software, tales como nuevas herramientas y métodos.

### ➤ **Establecimiento de la Comunicación con el Cliente**

Esta es una tarea muy importante en Mobile-D. La gestión del proyecto debe establecer y acordar las formas de comunicación con el grupo de clientes a lo largo del proyecto. Diferentes proyectos y entornos requieren un enfoque diferente para la comunicación, que en el contexto del desarrollo ágil de software requerirá la comunicación firme y constante con el cliente. Se necesitará al Cliente en ciertas tareas a lo largo Mobile-D, como las pruebas de aceptación, no dependiendo de si el cliente está en el lugar de fuera de las instalaciones. Sin embargo, mientras que un proyecto puede tener un cliente en el lugar, disponible todo el tiempo, otro proyecto puede necesitar otras maneras de comunicarse diariamente con los clientes, sobre todo si el cliente está fuera de las instalaciones. El tema principal es el de aceptar el nivel y los medios para la comunicación para permitir un desarrollo de software eficiente, producto final válido y cliente satisfecho.

### **b) Día de planificación**

El propósito del modelo de planificación inicial es obtener una comprensión global del producto a desarrollar, para preparar y perfeccionar los planes para las próximas fases del proyecto y preparar planes de comprobación y resolución de todos los aspectos fundamentales del desarrollo.

Las tareas a elaborar en el día de planificación o planificación inicial vienen dadas en el siguiente orden:

### ➤ **Análisis inicial de requerimientos**

El propósito del análisis inicial de requisitos es priorizar y analizar cuidadosamente los requisitos seleccionados para la implementación durante la fase de inicialización. Durante esta tarea se garantiza que los requisitos seleccionados son afines con la línea de la arquitectura planificada y proporcionan una buena base para propósitos de prueba. Así, la funcionalidad a implementar durante el día de ensayos no se selecciona en base a la prioridad definida por el cliente, sino en función de su importancia relativa, por ejemplo, la estructura arquitectónica del producto o algunas funciones núcleo central.

### ➤ **Planificación de la línea Arquitectura**

La planificación inicial de la línea de Arquitectura tiene como propósito analizar aspectos críticos en la Arquitectura, de tal manera que permita obtener un crecimiento arquitectónico sistemático al momento de implementar los requisitos seleccionados por el cliente, durante las siguientes fases del proyecto.

### ➤ **Planificación de Iteración**

El objetivo de la planificación de la iteración es generar el calendario y el contenido de la iteración a ejecutar. El contenido se define en términos de casos de clientes y tareas para la producción del código de trabajo, mediante la aplicación de los requisitos funcionales seleccionados en el análisis de requisitos inicial y otras tareas para comprobar y resolver problemas críticos de desarrollo solo, sin producir ningún código de trabajo.

### ➤ **Generación de pruebas de Aceptación**

El propósito de las pruebas de aceptación es verificar que los requisitos que el cliente ha fijado para el software, se aplican correctamente. Pruebas de aceptación normalmente se generan únicamente para las tareas que producen código.

### ➤ **Revisión de Pruebas de Aceptación**

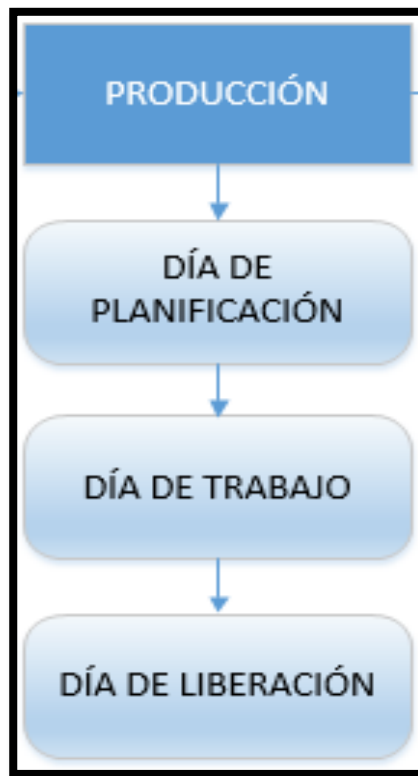
El objetivo de esta tarea es difundir las Pruebas de Aceptación y requisitos del sistema a todo el equipo y permitir que los miembros del equipo comenten sobre las Pruebas de Aceptación para mejorar su calidad.

### c) **Día de trabajo**

El propósito de esta etapa es probar y configurar el entorno de desarrollo técnico y asegurarse de que todo está listo para implementar el desarrollo del producto de software. Además, el propósito es implementar alguna funcionalidad del núcleo del sistema (por ejemplo, la comunicación cliente-servidor) o solucionar algún problema crítico del desarrollo sin producir ningún código de trabajo. Además nuevas investigaciones tecnológicas son posibles en esta etapa. Si el desarrollador decide implementar algunas funciones en este punto, no tiene por qué ser la funcionalidad de más alta prioridad definida por el cliente sino las que han sido seleccionadas en base a su importancia relativa, por ejemplo, la estructura arquitectónica del producto.

El día de trabajo forma la pre-fase para los días reales de desarrollo.

### 4.2.3. Producción



*Fuente: Propia*

Figura 48. Etapas de la Fase de Producción

#### 4.2.3.1. Motivación

El propósito de la fase de Producción es implementar las funcionalidades requeridas en el producto mediante la aplicación del ciclo de desarrollo iterativo e incremental.

#### 4.2.3.2. Objetivos

Los objetivos de la etapa de producción son:

- Implementar las funcionalidades del producto, priorizadas por el cliente.
- Centrarse en el desarrollo de las funcionalidades fundamentales del producto de manera que permita su crecimiento utilizando múltiples ciclos de mejoras.

#### **4.2.3.3. Etapas y tareas**

Las etapas en las que se centra la fase de Producción están dadas de la siguiente manera:

##### **a) Día de planificación**

El propósito en el día de planificación es seleccionar y planificar el contenido de trabajo para la iteración.

Al participar activamente en las actividades de planificación, el cliente se asegura de que los requisitos a los que proporcionó mayor valor para el negocio se identifican y comprenden correctamente.

En el día de planificación se realizan diferentes tareas:

##### **➤ Taller de Post-iteración**

El propósito del taller post-iteración es mejorar de forma iterativa en el proceso de desarrollo de software para ajustarse mejor a las necesidades del equipo actual del proyecto. Esto incluye la identificación de las fortalezas y debilidades en el proceso, así como la generación de acciones de mejora para la siguiente iteración.

##### **➤ Análisis de Requerimientos**

El propósito del análisis de requisitos es priorizar cuidadosamente y analizar los requisitos seleccionados para cada iteración. Durante esta tarea se asegura que se identifican los requisitos que proporcionan mayor valor para el negocio, y se entienden correctamente esos requisitos.

➤ **Planificación de la iteración**

El objetivo de la planificación de la iteración es generar el calendario y el contenido de la iteración para ejecutar. El contenido se define en términos de órdenes de trabajo para el equipo.

➤ **Generación de prueba de Aceptación**

El propósito de las pruebas de aceptación es verificar que los requisitos del cliente se han aplicado correctamente. Las pruebas de aceptación se generan durante el día de Planificación. Más tarde, las pruebas se ejecutan con el cliente y se documentan temas de defectos y mejoras.

➤ **Opinión de Pruebas de Aceptación**

El objetivo de esta tarea es difundir el resultado de las Pruebas de Aceptación del sistema a todo el equipo y permitir que los miembros del equipo comenten sobre estos para mejorar su calidad.

**b) Día de trabajo**

El propósito de esta etapa es la implementación de la funcionalidad del sistema previsto durante el día de planificación. El equipo de desarrollo se centra en la funcionalidad con mayor prioridad definida por el cliente. El día de trabajo se utiliza en las fases de Producción, Estabilización y Pruebas del Sistema. Una iteración puede contener 1-n de días de trabajo. Estas jornadas forman los días reales de desarrollo de la iteración.

En el día de trabajo se realizan las siguientes tareas:

➤ **Wrap-up**

Es una tarea interactiva para comunicar el progreso y los problemas encontrados, se realiza normalmente como la primera y/o la última actividad de la jornada de trabajo.

➤ **Desarrollo basado en pruebas**

En esta tarea las pruebas unitarias se escriben antes de que el código del programa. Se desarrolla a continuación el código del programa para trabajar con las pruebas ya escritas. El propósito es dar la confianza de que los desarrolladores de código orientan el diseño del código con una clara estructura y es más fácilmente comprobable.

➤ **Programación en pares**

Es un estilo de programación de código con dos desarrolladores juntos. Conductor es la persona que maneja el teclado y el ratón durante una sesión de programación. Navegante se sienta delante del ordenador al lado del conductor. El propósito de la programación en parejas es mejorar la comunicación, difundir el conocimiento dentro del equipo y asegurar la calidad del código.

➤ **Integración Continua**

El propósito esta tarea es integrar continuamente nuevo código con el código existente en un repositorio de código. Mediante la integración continua se evita integraciones masivas que conllevan mayor esfuerzo y tiempo.

➤ **Refactorización**

Es el proceso de mejorar la estructura interna de los programas informáticos existentes sin modificar su comportamiento externo. Con pequeñas mejoras al código, la refactorización garantiza que el software es más modificable, extensible, y legible.

➤ **Informar a los clientes**

El objetivo de esta tarea es la de proporcionar una visión honesta de los avances para el cliente, y darle la posibilidad de dar su opinión sobre las características implementadas y guiar el desarrollo.

c) **Día de lanzamiento**

El propósito de la etapa es verificar y validar la funcionalidad implementada. Normalmente, el día del lanzamiento culmina en una liberación real, pero la liberación puede ser no oficial para que el producto sea evaluado únicamente por los interesados en el proyecto.

Las tareas individuales de un día de lanzamiento son:

➤ **Integración de Sistemas**

En el caso de proyecto multi-equipo, el propósito de esta tarea es integrar subsistemas generados en equipos separados, en un solo producto.

➤ **Prueba de Pre-liberación**

El objetivo de esta tarea es asegurarse de que el software que se produce está listo para las pruebas de aceptación y liberación.



La prueba se lleva a cabo dentro del equipo mediante la ejecución de todas las pruebas de aceptación escritas.

➤ **Pruebas de Aceptación**

El propósito de las pruebas de aceptación es para verificar que los requisitos del cliente se cumplen correctamente. Las pruebas de aceptación en el día de lanzamiento se ejecutan con el cliente y si se encuentran problemas (es decir, defectos y mejoras) se documentan.

➤ **Ceremonias de lanzamiento**

Son las tareas finales antes de hacer una versión del software. En la práctica, las ceremonias de liberación consisten la auditoría de liberación y la creación de la línea de base. Estas actividades se llevan a cabo para confirmar que todo se ha hecho bien en la iteración actual y la base para una nueva iteración está asegurado.

#### 4.2.4. Estabilización



*Fuente: Propia*

Figura 49. Etapas de la Fase de Estabilización

##### 4.2.4.1. Motivación

El propósito de la fase de Estabilización es garantizar la calidad de la implementación del proyecto.

##### 4.2.4.2. Objetivos

Los objetivos en la fase de estabilización son:

- Finalizar la implementación del producto.
- Mejorar y garantizar la calidad del producto.
- Finalizar la documentación del producto.

#### **4.2.4.3. Etapas y tareas**

##### **a) Día de planificación**

El objetivo del día de planificación de días en la fase de Estabilización es definir el contenido (es decir, tareas) para la implementación de las características restantes del producto y para mejorar la calidad externa e interna del producto (refactoring).

Las tareas que se desprenden en esta etapa son las mismas que se consideran en las etapas del día de planificación en las fases de Producción, Estabilización y Pruebas del Sistema.

##### **b) Día de trabajo**

El objetivo del día de trabajo de la fase de Estabilización es finalizar la implementación del producto, así como mejorar y garantizar la calidad del producto.

Al igual que en la etapa del día de planificación, Mobile-D propone las mismas tareas en las etapas del día de trabajo para las fases de Producción, Estabilización y Pruebas del Sistema.

##### **c) Documentación**

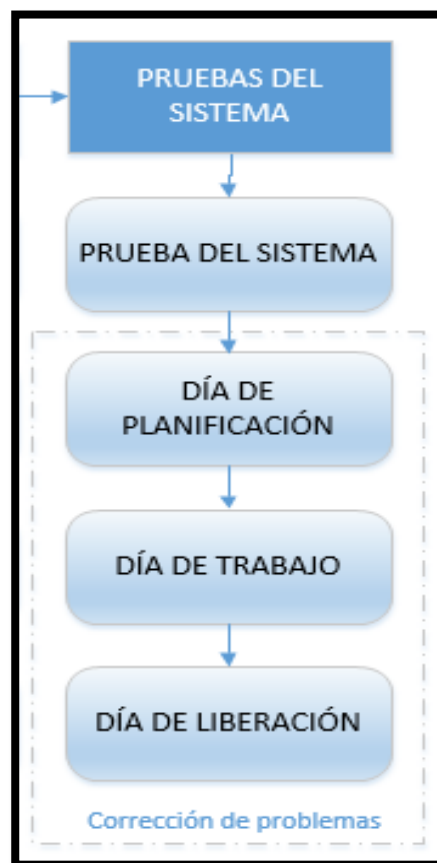
El propósito de esta tarea es producir documentación. Un Software sin documentación es un desastre. El código fuente no es el medio ideal para comunicar la razón fundamental, Estructura e interfaces de un sistema. La documentación será producida por los participantes del proyecto y no por el equipo de desarrollo.

#### d) Día de Liberación o Lanzamiento

El propósito de la etapa de día de lanzamiento es para verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. El día del lanzamiento culmina en la versión final de todo el software.

Las tareas en esta etapa se definen de la misma manera que los días de lanzamiento en las fases de Producción y Pruebas del Sistema.

#### 4.2.5. Pruebas del sistema



*Fuente: Propia*

Figura 50. Etapas de la Fase de Pruebas del sistema

##### 4.2.5.1. Motivación

El propósito de las pruebas del sistema es ver si el sistema producido implementa correctamente la funcionalidad definida por el cliente, proporcionar al equipo del

proyecto una retroalimentación de la funcionalidad de los sistemas y corregir los defectos encontrados.

#### **4.2.5.2. Objetivos**

Los objetivos que se persigue en esta etapa son:

- Probar el sistema, basado en la documentación producida en el proyecto
- Proporcionar información de defectos encontrados
- Deje que el equipo de proyecto planifique una solución para los defectos encontrados
- Corregir los defectos
- Producir un sistema libre de errores como sea posible

#### **4.2.5.3. Etapas y tareas**

##### **a) Prueba del sistema**

Es una etapa en la que el sistema se prueba como se describe en el modelo de la tarea de prueba del sistema. Los defectos encontrados se documentan con el fin de la crear una iteración para corregirlos.

##### **b) Corrección**

La etapa de Corrección es una variación de las iteraciones normales; Sin embargo ninguna nueva funcionalidad se implementa y la escala de tiempo puede ser notablemente más corto. La entrada para esta iteración son los defectos encontrados en la fase de prueba del sistema.

La iteración en la etapa de Corrección involucra una etapa de día de planificación, día de trabajo y día de liberación, semejantes y con las mismas tareas que las mencionadas en las fases de Producción y Estabilización.

### **4.3. ELEMENTOS DE LA METODOLOGÍA**

Existen 9 elementos principales involucrados en las diferentes prácticas en el transcurso del ciclo de desarrollo:

1. Ajuste y enfoque de fases: los proyectos se llevan a cabo en iteraciones donde cada una comienza con un día de planificación.
2. Línea de arquitectura: este enfoque es utilizado junto con los patrones de arquitectura y modelado ágil.
3. Desarrollo basado en pruebas: el enfoque de pruebas-primero es utilizado junto con casos de prueba automatizadas.
4. Integración continua: las prácticas de Software Configuration Manager (SCM) se aplican a través de múltiples medios.
5. Programación en pares: la codificación, pruebas y refactorización se llevan a cabo en pares.
6. Métricas: pocas métricas se recogen con rigurosidad y se utilizan con fines de mejorar la retroalimentación y el proceso de desarrollo.
7. Mejoras en el proceso de software ágil: talleres de post-iteración son utilizados para mejorar continuamente el proceso de desarrollo.
8. Cliente extremo: el cliente participa en las jornadas de planificación y liberación.
9. Enfoque centrado en el usuario: se hace hincapié en la identificación y el cumplimiento de necesidades del usuario final.

Estos elementos son prácticas ya establecidas en metodologías ágiles, con la inclusión de la línea de arquitectura, que se usa para capturar el conocimiento de una organización de soluciones arquitectónicas, tanto de fuentes internas y externas, y usar estas soluciones cuando sea necesario.

#### **4.4. JUSTIFICACIÓN**

Mobile-D al combinar los beneficios de varias metodologías como EX, Crystal y RUP proporciona varias razones para ser la metodología seleccionada en el desarrollo del proyecto.

- Es una metodología ágil con ciclos de desarrollo cortos y para equipos pequeños.
- Está diseñada tanto para desarrollo de aplicaciones móviles como para aplicaciones tradicionales.
- Permite detectar y resolver tempranamente problemas técnicos.
- Baja densidad de defectos en la liberación de los productos.
- Desarrollo basado en pruebas que es una de las mejores maneras de asegurar la calidad del producto y buenos diseños.
- Se enfoca en la satisfacción del usuario final, permitiendo mejorar el producto al realizar iteraciones cortas.
- Define tareas a realizar en cada fase de una manera bien detallada.





## CAPÍTULO V



### DESARROLLO DEL APLICATIVO

---

- Exploración
  - Inicialización
  - Producción
  - Estabilización
  - Pruebas del Sistema
-

## **5.1. EXPLORACIÓN**

### **5.1.1. Establecimiento de los grupos de interés**

Se ha definido como grupos de interés a los restaurantes y lugares de preparación de alimentos que ofrecen a su clientela una amplia variedad de platos a la carta y que deseen implementar el uso de tecnología para mejorar sus servicios.

Por otro lado los posibles clientes de estos lugares, que necesitan conocer los diferentes menús con los que cuentan estos establecimientos.

### **5.1.2. Definición del alcance**

#### **5.1.2.1. Colección de requisitos iniciales**

En base a la observación de varios establecimientos de preparación de alimentos y la conversación con los administradores de dos establecimientos interesados en la aplicación (Caribou Bar & Grill y Aromas Café), se han determinado una lista de requisitos con los que debería contar la solución propuesta.

- Creación de usuarios
- Determinación del menú ofrecido
- Configuración de ambientes y mesas
- Determinación de los lugares de preparación
- Toma de comandas a cada mesa
- Verificación de la preparación de las comandas solicitadas.
- Resumen de atenciones diarias.
- Resumen de productos disponibles

Las especificaciones de estos requisitos serán adjuntados en la sección de anexos de este documento.

### **5.1.2.2. Planificación del proyecto inicial**

En base a los requisitos iniciales se ha podido identificar los diferentes módulos de acuerdo a cada aplicación del sistema, listados a continuación:

#### ➤ **Aplicación Web**

##### ○ **Módulo de Seguridad**

- Configuración de Usuarios
- Configuración de Roles
- Permisos de roles

##### ○ **Módulo de Administración**

- Configuración de lugares de preparación (Cocinas, Bares)
- Categorías de productos
- Configuración de Productos
- Configuración de Ambientes
- Configuración de Mesas
- Asignación de Productos a Ambientes
- Asignación de meseros a Ambientes
- Asignación de usuarios a Lugares de Preparación

##### ○ **Comandas**

- Ingreso de Comandas

- Entrega de comandas
  - Cobro de Comandas
  - **Reportes**
    - Atenciones realizadas
    - Productos disponibles
- **Aplicación Móvil**
- **Módulo de Meseros**
    - Mesas disponibles
    - Toma de comandas
    - Entrega de comandas
  - **Módulo de Cocina y Bares**
    - Preparación de comandas

### 5.1.2.3. Definición de Usuarios

Tipo de Usuario	Tareas	Módulo
Administrador	<ul style="list-style-type: none"> <li>▪ Administrar usuarios y permisos</li> <li>▪ Administrar ambientes</li> <li>▪ Administrar productos</li> <li>▪ Establecer meseros y ambientes</li> <li>▪ Administrar mesas</li> </ul>	Aplicación Web Administración
Meseros	<ul style="list-style-type: none"> <li>▪ Ingreso de comandas por mesas</li> <li>▪ Entrega de comandas realizadas</li> </ul>	Aplicación Móvil de Meseros
Cocineros y bartender	<ul style="list-style-type: none"> <li>▪ Preparación de comandas solicitadas</li> </ul>	Aplicación Móvil de Cocina y bar

Fuente: Propia y entrevistas con administradores  
 Tabla 14. Definición de usuarios

### 5.1.3. Establecimiento del proyecto

#### 5.1.3.1. Selección del entorno

En el establecimiento del entorno de trabajo se definen las herramientas a utilizar en el desarrollo de la aplicación.

Para el desarrollo de la aplicación web se utilizará el framework ScriptCase en su versión 8 que es una herramienta de desarrollo rápido de aplicaciones que agiliza el proceso de generación de páginas web con el lenguaje de programación PHP.

Se define también la utilización de JSON como medio de comunicación entre la aplicación móvil y web, siendo los servicios Web escritos en lenguaje PHP.

Para el desarrollo de las aplicaciones móviles se establece la utilización de la herramienta Appcelerator Studio, un framework que codifica en lenguaje JavaScript y compila la aplicación a lenguaje nativo de las plataformas Android y Apple.

#### 5.1.3.2. Asignación del personal

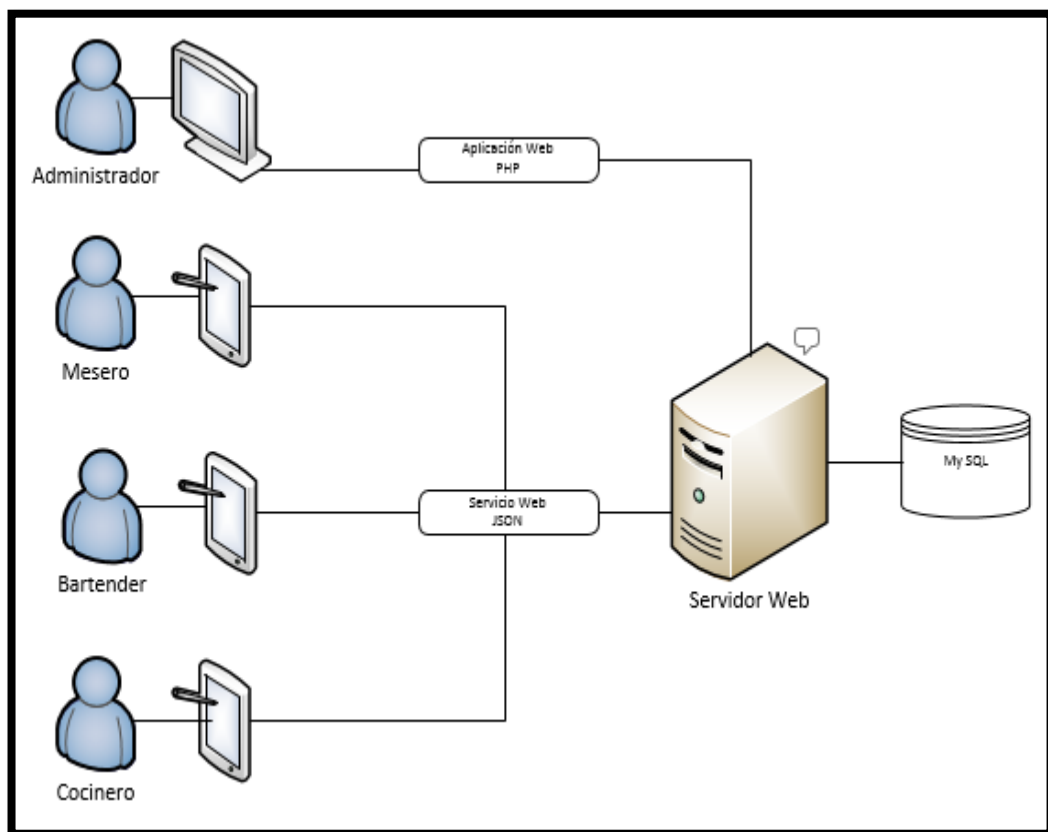
Esta tarea corresponde a la asignación del equipo de desarrollo para el proyecto.

Equipo Técnico	Cargo	Módulos Asignados
Iván González	Desarrollador	<ul style="list-style-type: none"><li>▪ Administración</li><li>▪ Servicios web</li><li>▪ Aplicación móvil de meseros</li><li>▪ Aplicación móvil de cocina</li></ul>

Fuente: Propia

Tabla 15. Asignación de personal

### 5.1.3.3. Definición de la Arquitectura de la Aplicación



*Fuente: Propia*  
*Figura 51. Arquitectura de la Aplicación*

El diseño general del sistema está formado por tres componentes principales: un web service, una aplicación web y una aplicación móvil que accede al web service a través de internet. El servidor donde se alojara el web service también contiene la base de datos y la aplicación web.

## **5.2. INICIALIZACIÓN**

### **5.2.1. Configuración del proyecto**

En esta etapa se realiza la configuración del ambiente de trabajo y la formación del equipo de desarrollo en las herramientas utilizadas.

#### **5.2.1.1. Configuración del Ambiente**

En la configuración del ambiente se han realizado las actividades de instalación y preparación de las herramientas utilizadas para el desarrollo de la aplicación.

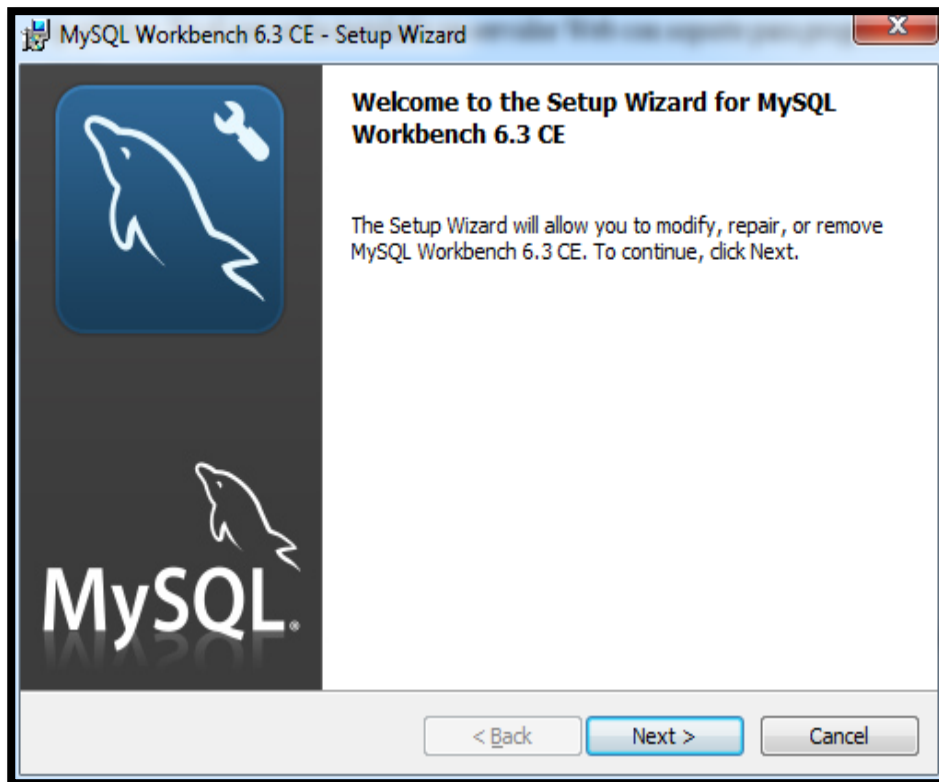
##### **a) Configuración del Servidor**

Gracias a la empresa ByPro's Sistemas Inc. se cuenta con acceso a un Servidor con una dirección ip pública. En el que se ha instalado un servidor Web con soporte para programación con lenguaje PHP.

##### **b) Preparación de MySQL**

En el mismo Servidor se ha instalado y configurado el motor de base de Datos MySQL, dicho instalador se ha obtenido del sitio web oficial de la herramienta <https://www.mysql.com/products/workbench/>.

Para el proceso de instalación basta con ejecutar el instalador, seleccionar los componentes necesarios y la ubicación en donde se desea instalar la base de datos.



*Fuente: Propia*  
*Figura 52. Instalación de MySql*

### **c) Preparación de ScriptCase 8.0**

ScriptCase es un framework de pago para desarrollo rápido de aplicaciones web, que se instala dentro del servidor web y al que se puede acceder desde cualquier navegador de internet. Gracias a la empresa ByPro's Sistemas se cuenta con una licencia de desarrollo para esta plataforma.



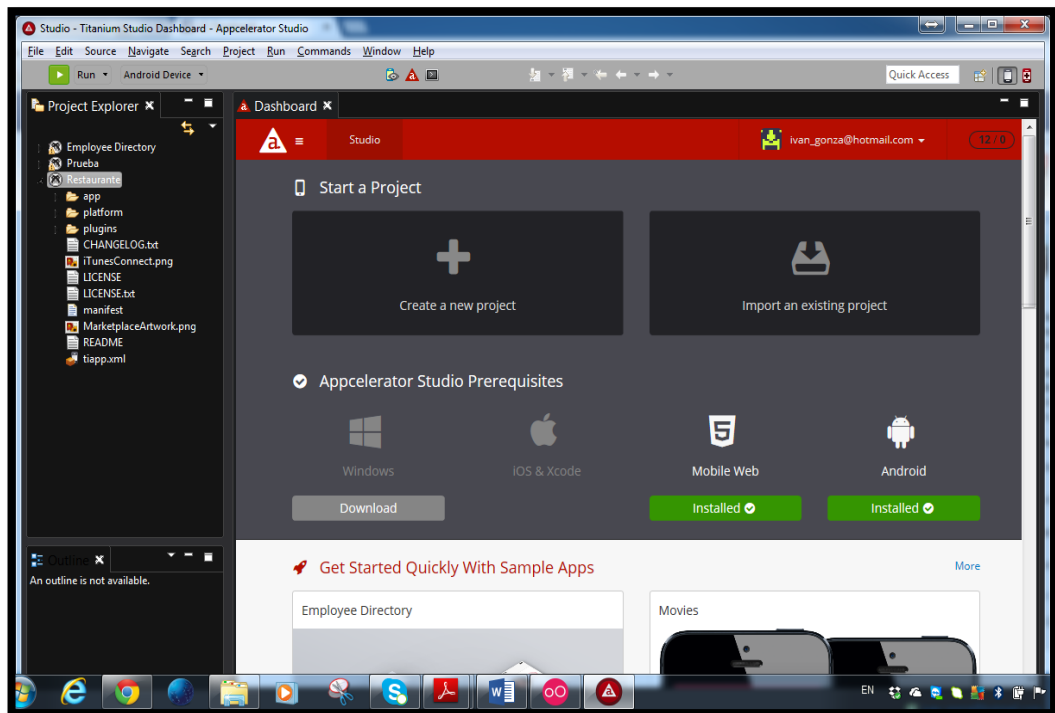


*Fuente: Propia*  
**Figura 53. Página de Inicio de ScriptCase**

#### **d) Instalación de Appcelerator**

Para el uso adecuado de la herramienta Appcelerator se ha realizado todos los pasos descritos en el capítulo III de este documento, tomando en cuenta los requisitos tanto de hardware como de software que se necesita para un correcto funcionamiento.

En un principio se utilizará esta herramienta configurada para crear una aplicación Android ya que se la instaló en un sistema operativo Windows.

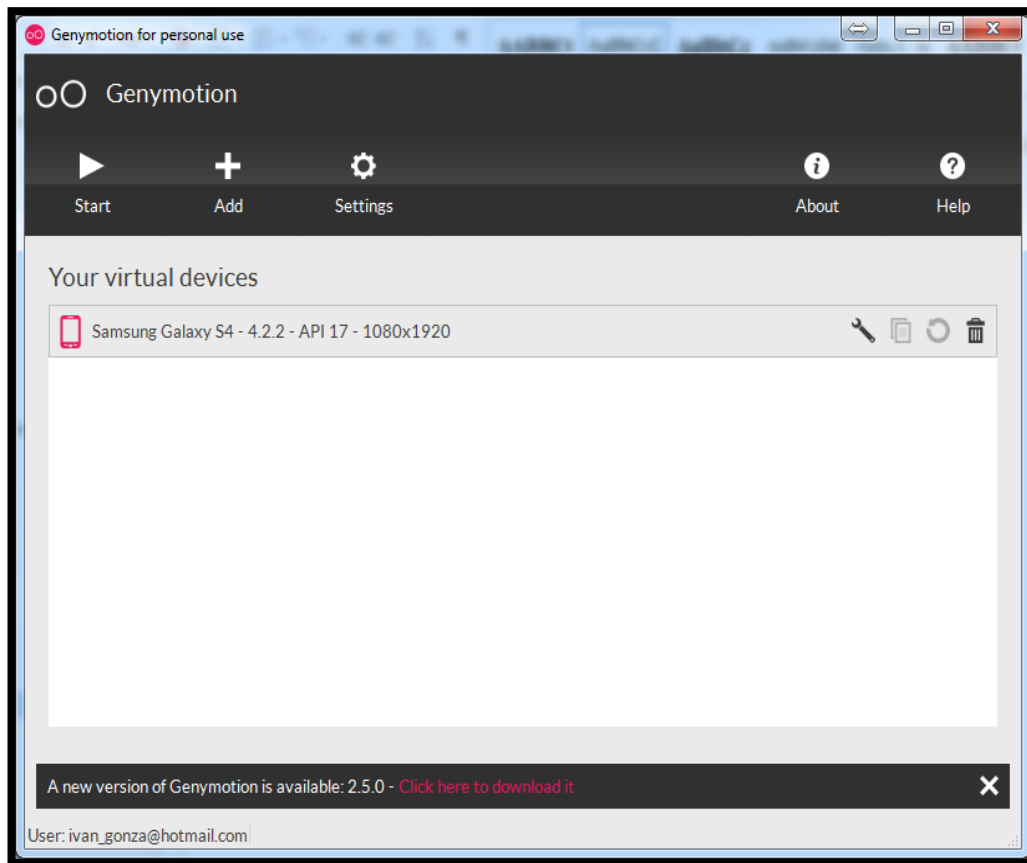


*Fuente: Propia*  
*Figura 54. Entorno de trabajo Appcelerator*

### e) Instalación de Genymotion

Para realizar pruebas de funcionamiento de la aplicación móvil se requiere de un emulador del sistema Android, por lo que es necesario instalar Genymotion, que es una herramienta en la que se puede crear emuladores de las diferentes versiones del sistema operativo Android.

Esta herramienta cuenta con una versión personal que se puede descargar del sitio web <https://www.genymotion.com/#/>.



Fuente: Propia  
 Figura 55. Pantalla Principal de Genymotion

**f) Elementos preparados**

Herramienta	Elementos preparados	Observaciones
MySQL	Base de datos	Repositorio de datos para la aplicación
ScriptCase	Proyecto Web	Entorno para la creación del módulo de Administración Creación de Servicios Web
Appcelerator	Proyecto de aplicación móvil	Entorno para la creación de módulos de Meseros y Cocina
Genymotion	Emulador Android Samsung Galaxy S4.	Entorno para pruebas de funcionamiento de aplicaciones móviles

Fuente: Propia  
 Tabla 16. Elementos preparados para el desarrollo de la Solución.

### 5.2.1.2. Formación

En esta tarea se entrena al equipo de trabajo de acuerdo a las necesidades específicas del proyecto. Incluye preparación en el uso de las herramientas, procesos de desarrollo, metodología, colección de datos, problemas técnicos relacionados con el uso de las herramientas. Se ha establecido un tiempo prudente para la formación adecuada desde el planteamiento del proyecto.

### 5.2.2. Planificación inicial

#### 5.2.2.1. Análisis Inicial de requerimientos

Con el fin de cumplir con un desarrollo fluido, se ha realizado un análisis de prioridades de los requisitos funcionales encontrados.

Módulo de Seguridad		
Requerimiento	Opciones	Pre requisitos
Definición Grupos	<ul style="list-style-type: none"><li>• Crear grupo</li><li>• Editar grupo</li><li>• Eliminar grupo</li></ul>	<ul style="list-style-type: none"><li>• Se debe contar con la empresa creada y un usuario principal.</li></ul>
Definición de Usuarios	<ul style="list-style-type: none"><li>• Crear usuario</li><li>• Editar usuario</li><li>• Eliminar usuario</li></ul>	<ul style="list-style-type: none"><li>• Se debe contar con la empresa creada y un usuario principal.</li><li>• Definición de Grupos</li></ul>
Asignación de permisos a Grupos	<ul style="list-style-type: none"><li>• Asignación de permisos de</li><li>• Modificación de permisos</li></ul>	<ul style="list-style-type: none"><li>• Definición de Grupos</li><li>• Lista de permisos</li></ul>
Cambio de contraseña	<ul style="list-style-type: none"><li>• Cambiar contraseña</li></ul>	<ul style="list-style-type: none"><li>• Definición de Usuarios</li></ul>
Módulo de Administración		
Requerimiento	Opciones	Pre requisitos
Lugares de preparación	<ul style="list-style-type: none"><li>• Crear lugar de preparación</li><li>• Editar lugar de preparación</li><li>• Eliminar lugar de preparación</li></ul>	<ul style="list-style-type: none"><li>• Estados de lugares</li></ul>

Categorías de productos	<ul style="list-style-type: none"> <li>• Crear categoría</li> <li>• Editar categoría</li> <li>• Eliminar categoría</li> </ul>	
Productos	<ul style="list-style-type: none"> <li>• Crear producto</li> <li>• Editar producto</li> <li>• Eliminar producto</li> </ul>	<ul style="list-style-type: none"> <li>• Categoría de productos</li> </ul>
Ambientes	<ul style="list-style-type: none"> <li>• Crear ambiente</li> <li>• Editar ambiente</li> <li>• Eliminar ambiente</li> </ul>	<ul style="list-style-type: none"> <li>• Estado de ambientes</li> </ul>
Asignación de Productos a Ambientes	<ul style="list-style-type: none"> <li>• Asignar producto</li> <li>• Editar asignaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Lugar de preparación</li> <li>• Productos</li> <li>• Ambientes</li> </ul>
Mesas	<ul style="list-style-type: none"> <li>• Crear mesa</li> <li>• Editar mesa</li> <li>• Eliminar mesa</li> </ul>	<ul style="list-style-type: none"> <li>• Ambientes</li> </ul>
Asignación de Meseros a Ambientes	<ul style="list-style-type: none"> <li>• Asignar mesero</li> <li>• Editar asignación</li> </ul>	<ul style="list-style-type: none"> <li>• Usuarios</li> <li>• Ambientes</li> </ul>
Asignación de usuarios a Lugares de Preparación	<ul style="list-style-type: none"> <li>• Asignar Lugar de Preparación</li> <li>• Editar Asignación</li> <li>• Eliminar Asignación</li> </ul>	<ul style="list-style-type: none"> <li>• Usuarios</li> <li>• Lugares de Preparación</li> </ul>
Finalizar comanda	<ul style="list-style-type: none"> <li>• Eliminar Productos</li> <li>• Cobrar</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicación móvil de mesero</li> <li>• Aplicación móvil de cocina</li> </ul>
Resumen de comandas	<ul style="list-style-type: none"> <li>• Listado de Comandas realizadas</li> </ul>	<ul style="list-style-type: none"> <li>• Finalizar Comanda</li> <li>• Aplicación móvil de mesero</li> <li>• Aplicación móvil de cocina</li> </ul>

Servicios Web		
Requerimiento	Opciones	Pre requisitos
Lista de Empresas	<ul style="list-style-type: none"> <li>• Lista las empresas registradas para utilizar la aplicación</li> </ul>	
Inicio de sesión	<ul style="list-style-type: none"> <li>• Verificación de usuarios</li> </ul>	<ul style="list-style-type: none"> <li>• Usuarios</li> </ul>
Lista de Mesas	<ul style="list-style-type: none"> <li>• Listar mesas y productos de acuerdo a usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Usuarios</li> <li>• Productos</li> <li>• Ambientes</li> </ul>

Lista de Categorías de Productos	<ul style="list-style-type: none"> <li>Listar las categorías de los productos</li> </ul>	
Lista de productos	<ul style="list-style-type: none"> <li>Lista de productos por categoría</li> </ul>	<ul style="list-style-type: none"> <li>Categorías</li> </ul>
Agregar Comandas	<ul style="list-style-type: none"> <li>Ingresar comandas para preparar</li> </ul>	
Listar Comandas	<ul style="list-style-type: none"> <li>Listar las comandas por preparar</li> </ul>	<ul style="list-style-type: none"> <li>Agregar comanda</li> </ul>
Modificar Comanda	<ul style="list-style-type: none"> <li>Agregar o quitar productos</li> </ul>	<ul style="list-style-type: none"> <li>Agregar comanda</li> </ul>
Cambiar Comanda	<ul style="list-style-type: none"> <li>Cambiar estado de comanda</li> </ul>	<ul style="list-style-type: none"> <li>Listar comanda</li> </ul>

### Aplicación Móvil para meseros

Requerimiento	Opciones	Pre requisitos
Iniciar sesión	<ul style="list-style-type: none"> <li>Iniciar sesión</li> </ul>	<ul style="list-style-type: none"> <li>Servicio web de inicio de sesión</li> </ul>
Ingreso de Comanda	<ul style="list-style-type: none"> <li>Ingreso de comandas</li> </ul>	<ul style="list-style-type: none"> <li>Módulo de Administración</li> <li>Servicio web agregar comandas</li> </ul>
Modificar comandas	<ul style="list-style-type: none"> <li>Agregar más productos a comanda</li> </ul>	<ul style="list-style-type: none"> <li>Módulo de Administración</li> <li>Servicio web modificar comandas</li> </ul>
Entregar Comanda	<ul style="list-style-type: none"> <li>Entregar las comandas ya preparadas</li> </ul>	<ul style="list-style-type: none"> <li>Modificar Comanda</li> </ul>

### Aplicación Móvil para Cocinas y Bares

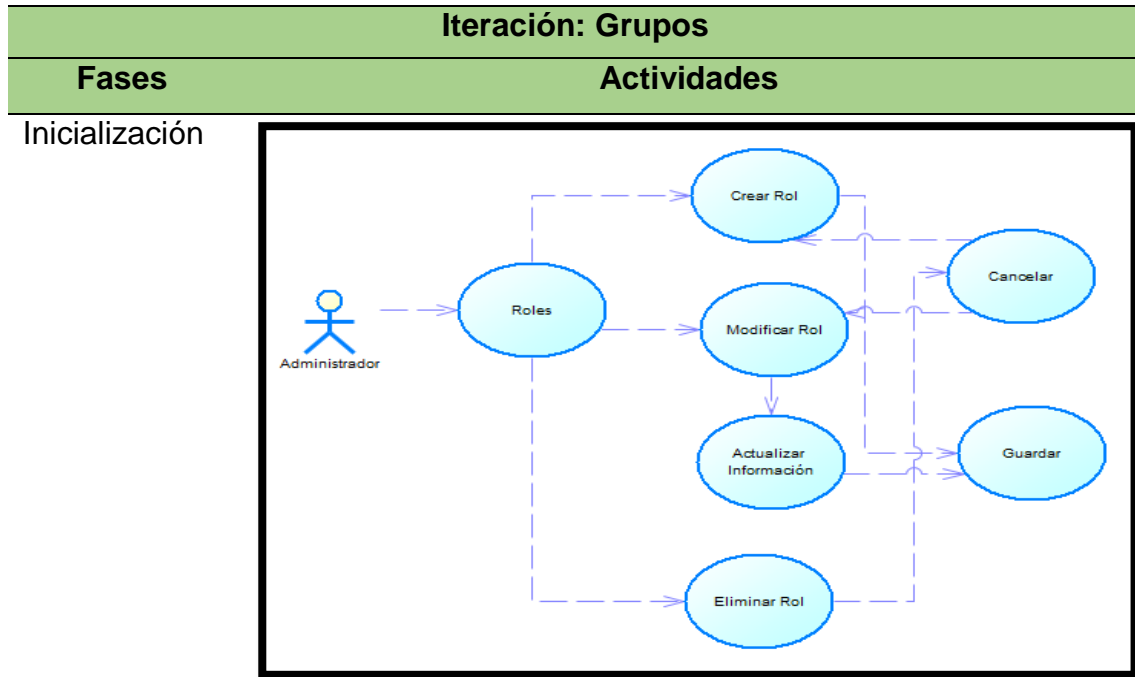
Requerimiento	Opciones	Pre requisitos
Iniciar sesión	<ul style="list-style-type: none"> <li>Iniciar sesión</li> </ul>	<ul style="list-style-type: none"> <li>Servicio web de inicio de sesión</li> </ul>
Listar Comandas	<ul style="list-style-type: none"> <li>Listar comandas por preparar</li> </ul>	<ul style="list-style-type: none"> <li>Servicio Web Listar Comandas</li> </ul>
Cambiar Comanda	<ul style="list-style-type: none"> <li>Cambiar estado de comanda a preparado</li> </ul>	<ul style="list-style-type: none"> <li>Servicio web Cambiar Comanda</li> </ul>

Fuente: Propia

Tabla 17. Análisis de Requisitos

### 5.2.2.2. Planificación de Iteraciones

De acuerdo al análisis inicial de requerimientos se ha plantado el siguiente plan de iteraciones para las fases del desarrollo de la aplicación.



Fuente: Propia

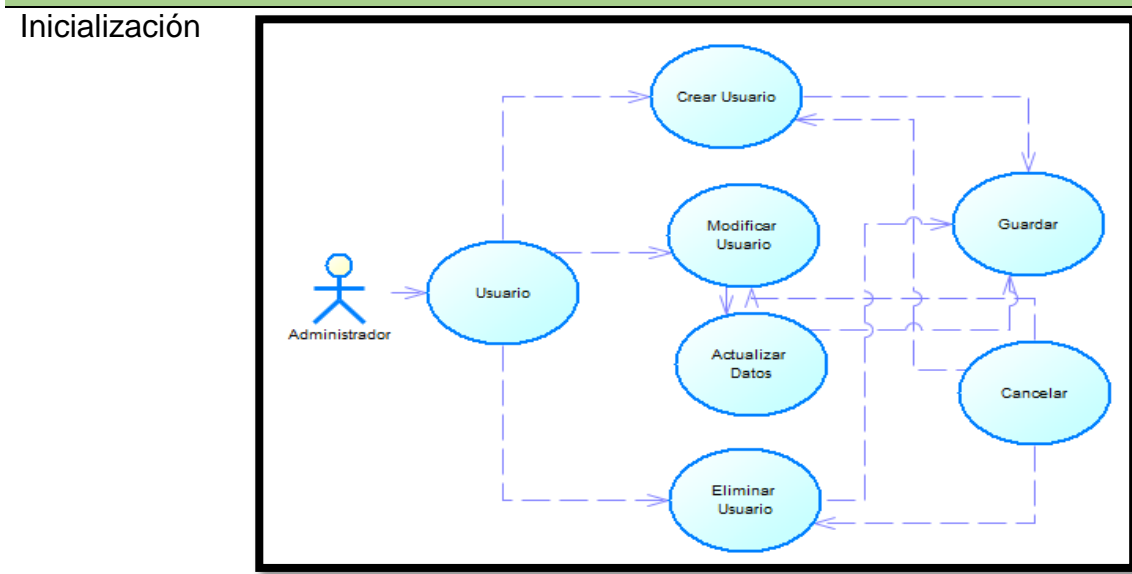
Figura 56. Caso de Uso Roles o Grupos

Establecimientos del requisito de Grupos

Producción	Implementación del requisito de manejo de Grupos Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Grupos. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados.

**Iteración: Usuarios**

Fases	Actividades
-------	-------------



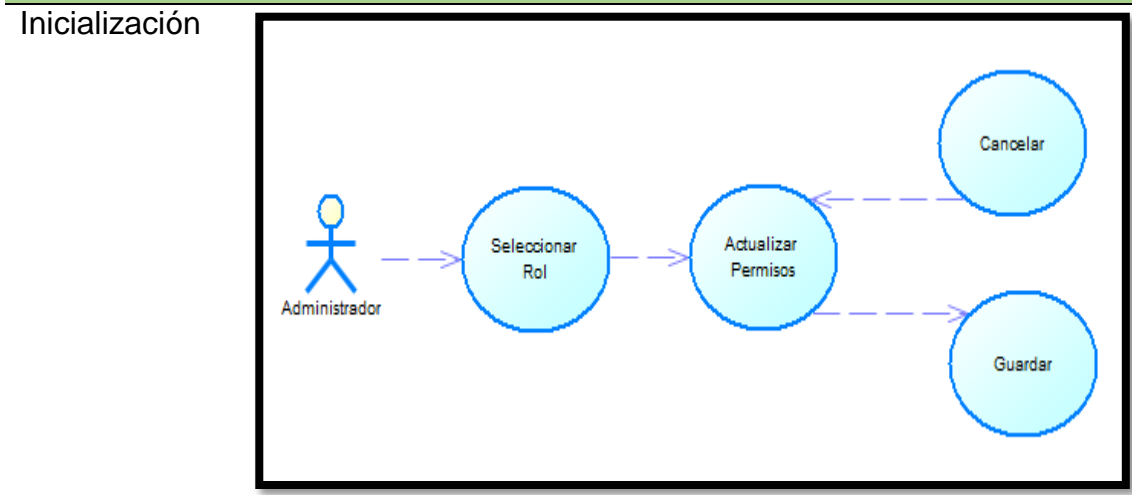
*Fuente: Propia*  
*Figura 57. Caso de Uso Usuarios*

Establecimientos del requisito de Usuarios

Producción	Implementación del requisito de manejo de Usuarios. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Usuarios. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Permisos**

Fases	Actividades
-------	-------------



*Fuente: Propia*  
*Figura 58. Caso de uso Permisos de Grupos*  
Establecimientos del requisito de permisos de grupos.

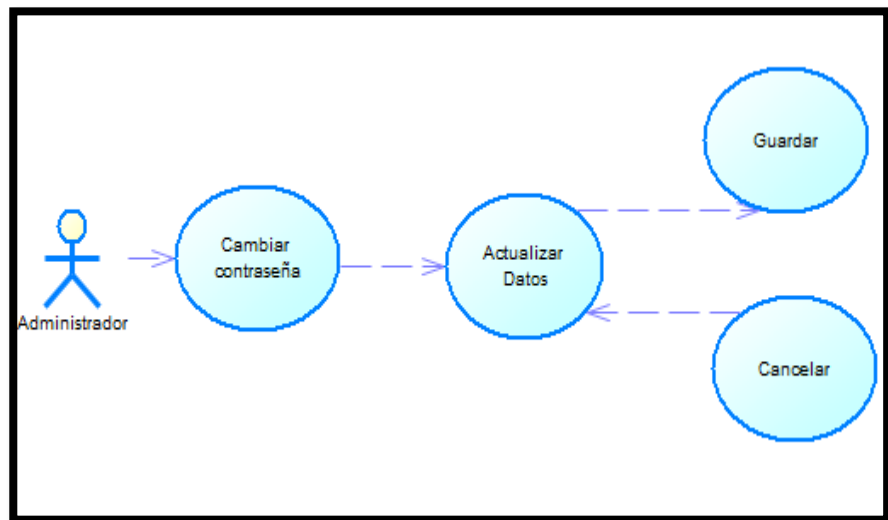


Producción	Implementación del requisito de manejo de permisos de grupos. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de permisos de grupos. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Cambio Contraseña**

Fases	Actividades
-------	-------------

Inicialización



*Fuente: Propia*

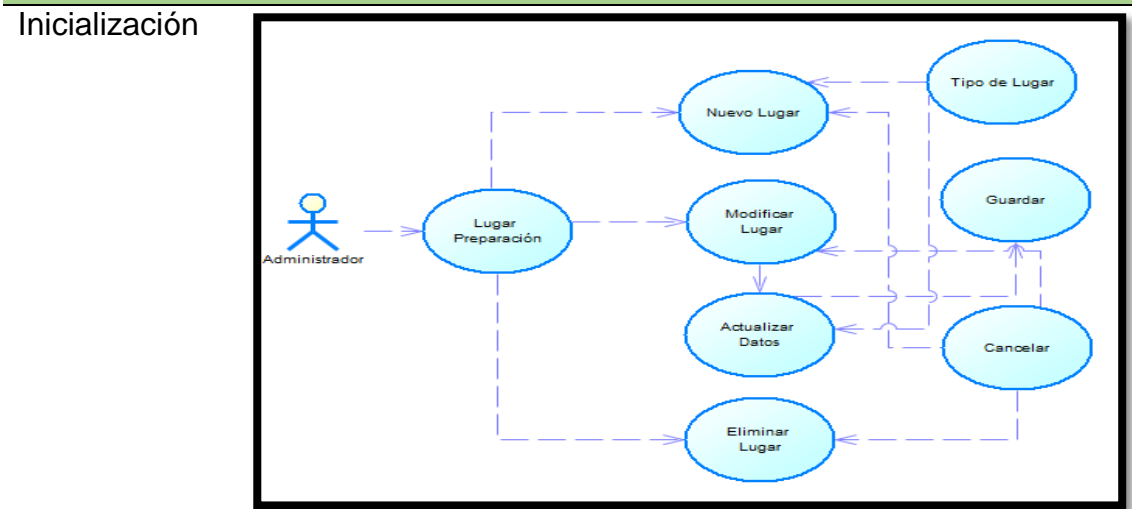
*Figura 59. Caso de uso Cambio de Contraseña*

Establecimientos del requisito de Cambio de contraseña.

Producción	Implementación del requisito de manejo de Cambio de contraseña. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Cambio de contraseña. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Lugares de Preparación**

Fases	Actividades
-------	-------------



*Fuente: Propia*

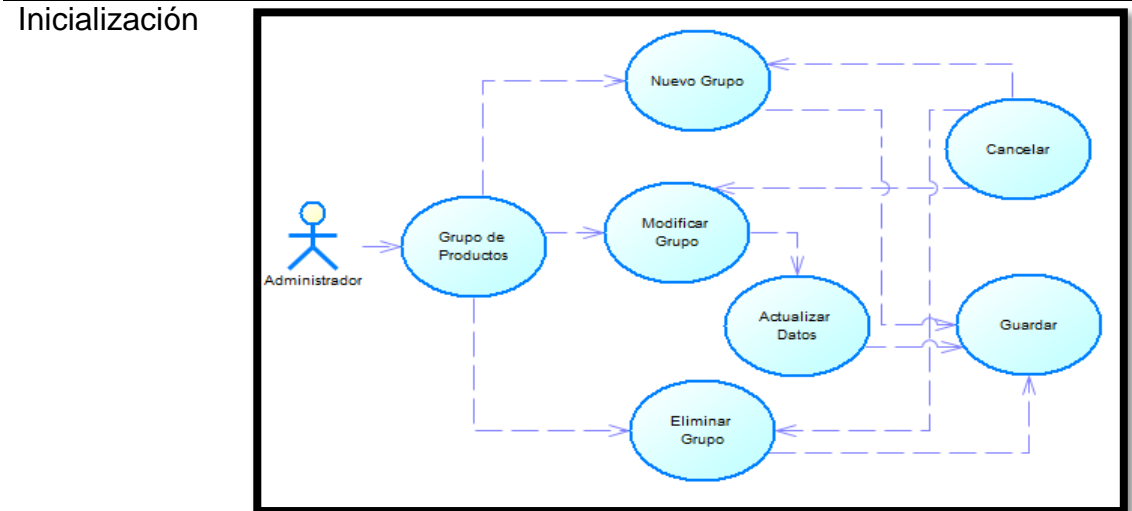
*Figura 60. Caso de uso Lugares de Preparación*

Establecimientos del requisito de Lugares de Preparación

Producción	Implementación del requisito de manejo de lugares de preparación. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de lugares de preparación. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Categoría Producto**

Fases	Actividades
-------	-------------



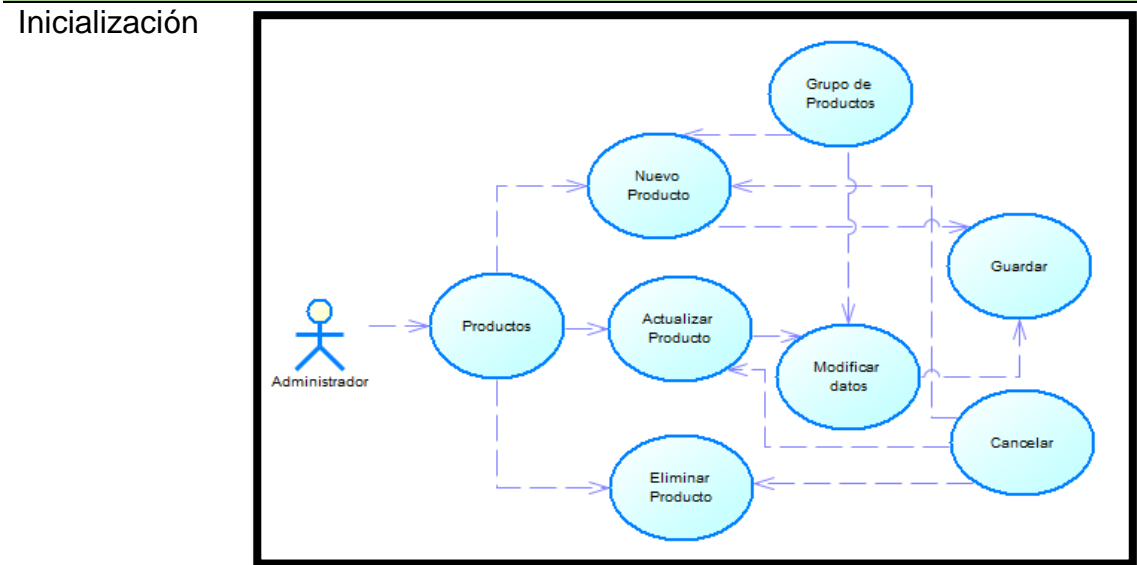
*Fuente: Propia*

*Figura 61. Caso de uso Categoría de Productos*

	Establecimientos del requisito de categorías de productos.
Producción	Implementación del requisito de manejo de categorías de productos. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de categorías de productos. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Productos**

Fases	Actividades
-------	-------------

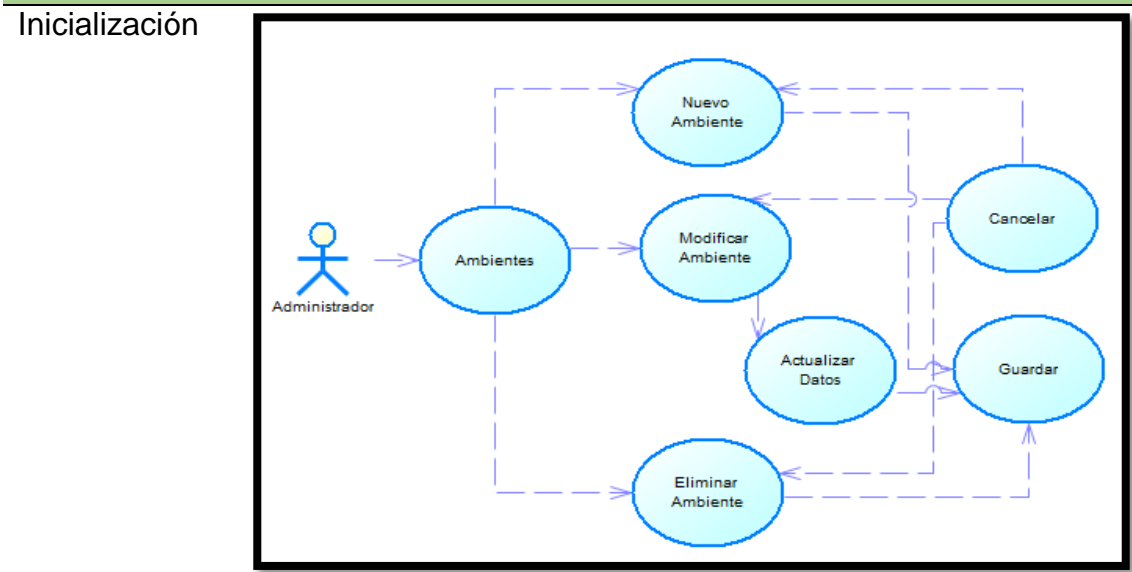


*Fuente: Propia*  
*Figura 62. Caso de uso Productos.*

	Establecimientos del requisito de Productos.
Producción	Implementación del requisito de manejo de Productos. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Productos. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Ambientes**

Fases	Actividades
-------	-------------

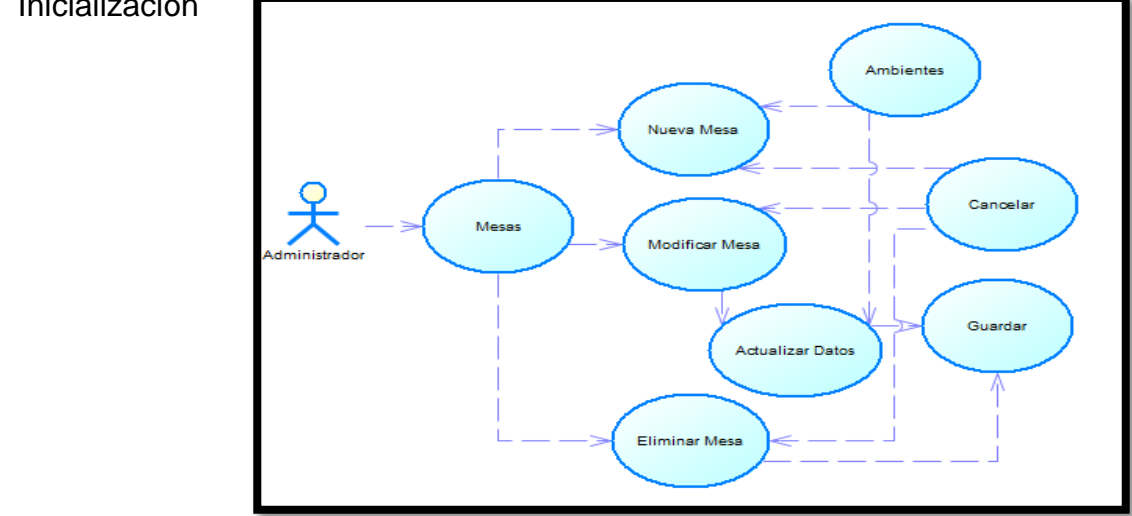


*Fuente: Propia*  
*Figura 63. Caso de Uso Ambientes*

	Establecimientos del requisito de Ambientes.
Producción	Implementación del requisito de manejo de Ambientes. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Ambientes. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Mesas**

Fases	Actividades
-------	-------------



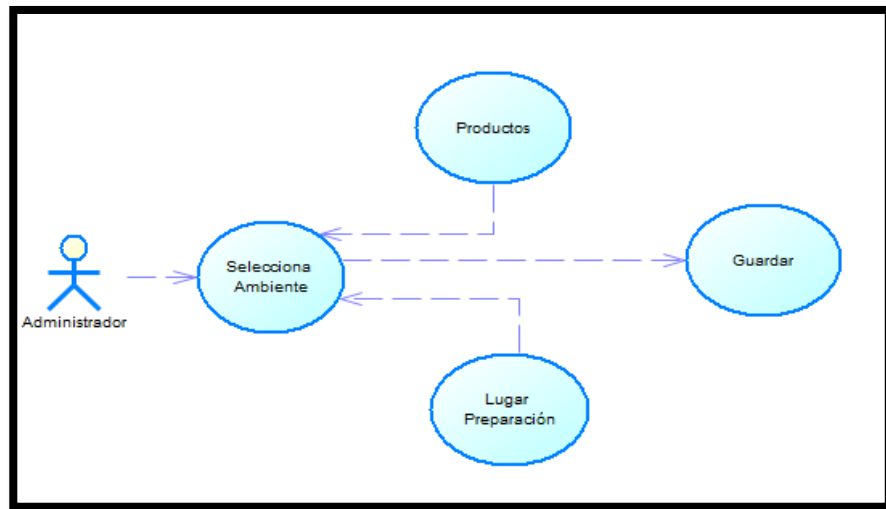
*Fuente: Propia*  
*Figura 64. Caso de Uso Mesas*

	Establecimientos del requisito de Mesas.
Producción	Implementación del requisito de manejo de Mesas. Refinamiento de interfaces. Generacion de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Mesas. Refinamientos de interfaces. Ejecucion de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Asignación Productos a Ambientes**

Fases	Actividades
-------	-------------

Inicialización



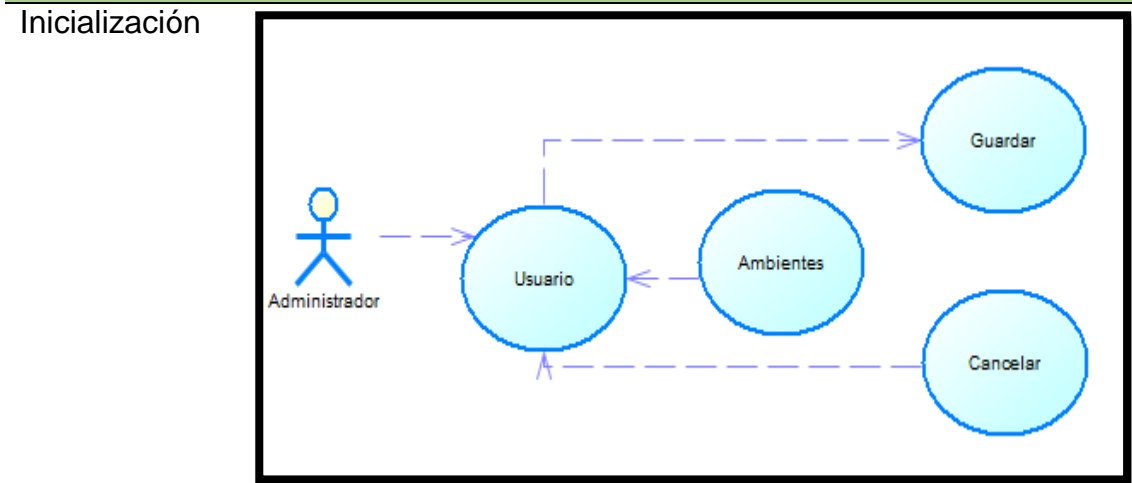
*Fuente: Propia*

*Figura 65. Caso de Uso Asignación productos a ambientes*  
Establecimientos del requisito de Asignación de productos a Ambientes.

Producción	Implementación del requisito de manejo de Asignación de productos a Ambientes. Refinamiento de interfaces. Generacion de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Asignación de productos a Ambientes. Refinamientos de interfaces. Ejecucion de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Asignación de usuarios a Ambientes**

Fases	Actividades
-------	-------------



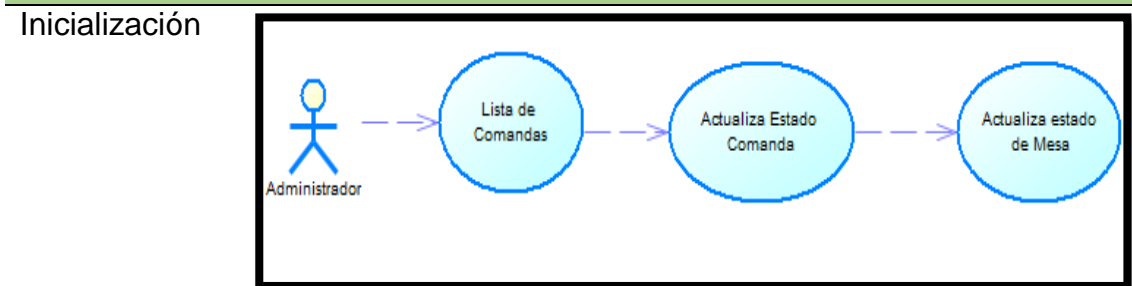
*Fuente: Propia*

*Figura 66. Caso de Uso Asignación de usuarios a Ambientes*  
 Establecimientos del requisito de Asignación de usuarios a Ambientes.

Producción	Implementación del requisito de manejo de Asignación de usuarios a Ambientes. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Asignación de usuarios a Ambientes. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Finalizar Comanda**

Fases	Actividades
-------	-------------



*Fuente: Propia*

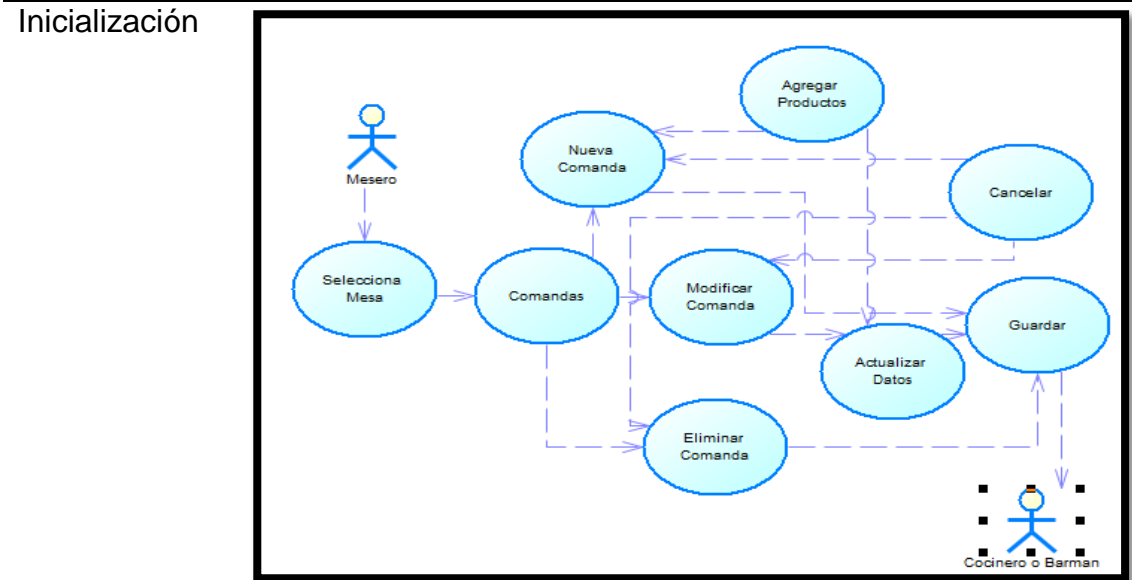
*Figura 67. Caso de Uso Finalizar Comanda*  
 Establecimientos del requisito de Finalizar comanda.

Producción	Implementación del requisito de manejo de Finalizar comanda. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Finalizar comanda.

	Refinamientos de interfaces. Ejecucion de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Recepción de Comanda**

Fases	Actividades
-------	-------------



*Fuente: Propia*

*Figura 68. Caso de Uso Recepción de Comanda*

Establecimientos del requisito de Recepción de comanda.

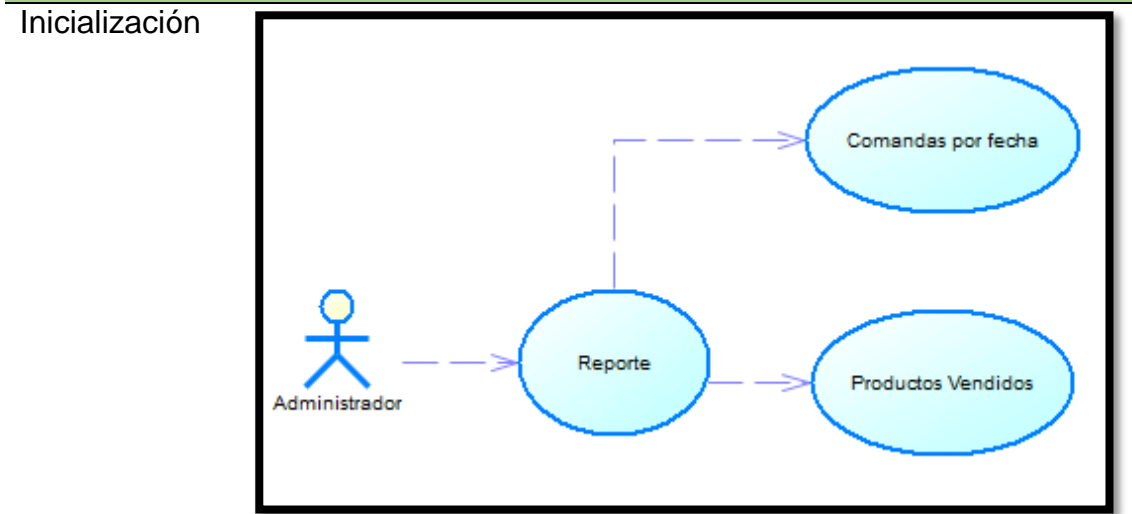
Producción	Implementación del requisito de manejo de Recepción de comanda. Refinamiento de interfaces. Generacion de pruebas de aceptación.
------------	--

Estabilización	Refactorización del requisito de manejo de Recepción de comanda. Refinamientos de interfaces. Ejecucion de pruebas de aceptación.
----------------	---

Pruebas	Evaluación de las pruebas y análisis de resultados
---------	--

**Iteración: Reportes**

Fases	Actividades
-------	-------------



*Fuente: Propia*

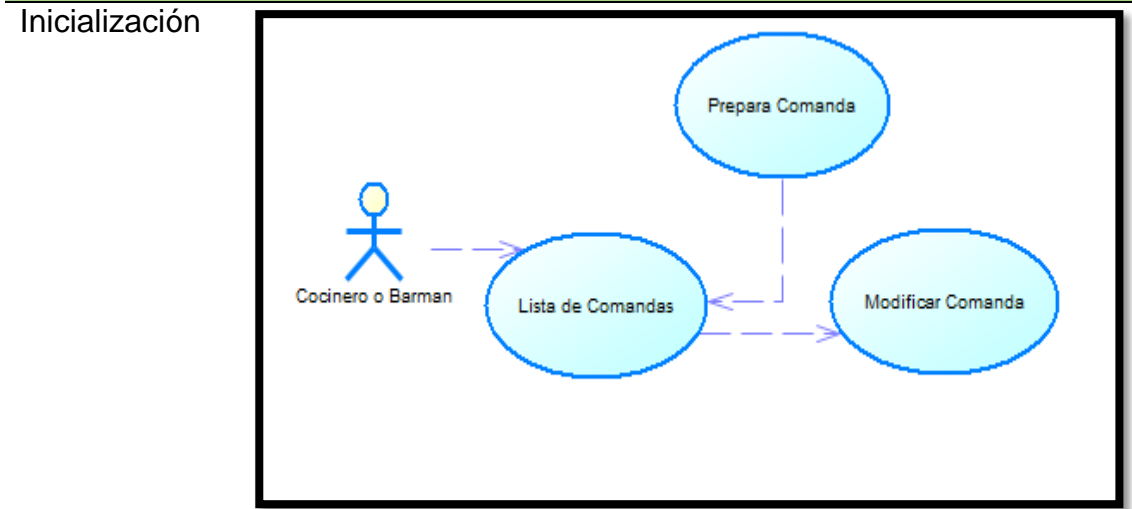
*Figura 69. Caso de Uso Reportes*

Establecimientos del requisito de Reportes.

Producción	Implementación del requisito de manejo de Reportes. Refinamiento de interfaces. Generación de pruebas de aceptación.
Estabilización	Refactorización del requisito de manejo de Reportes. Refinamientos de interfaces. Ejecución de pruebas de aceptación.
Pruebas	Evaluación de las pruebas y análisis de resultados

**Iteración: Preparación Comanda**

Fases	Actividades
-------	-------------



*Fuente: Propia*

*Figura 70. Caso de Uso Preparación Comanda*

Establecimientos del requisito de Preparación Comanda.



Producción	Implementación del requisito de manejo Comanda. Refinamiento de interfaces. Generacion de pruebas de aceptación.	Preparación
Estabilización	Refactorización del requisito de manejo de Comanda. Refinamientos de interfaces. Ejecucion de pruebas de aceptación.	Preparación
Pruebas	Evaluación de las pruebas y análisis de resultados	
<b>Iteración: Servicios Web</b>		
<b>Fases</b>		
<b>Actividades</b>		
Inicialización	Establecimientos del requisito de Servicios Web.	
Producción	Implementación del requisito de manejo Servicios Web. Refinamiento de interfaces. Generacion de pruebas de aceptación.	
Estabilización	Refactorización del requisito de manejo de Servicios Web. Refinamientos de interfaces. Ejecucion de pruebas de aceptación.	
Pruebas	Evaluación de las pruebas y análisis de resultados	

Fuente: Propia

Tabla 18. Planificación de Iteraciones

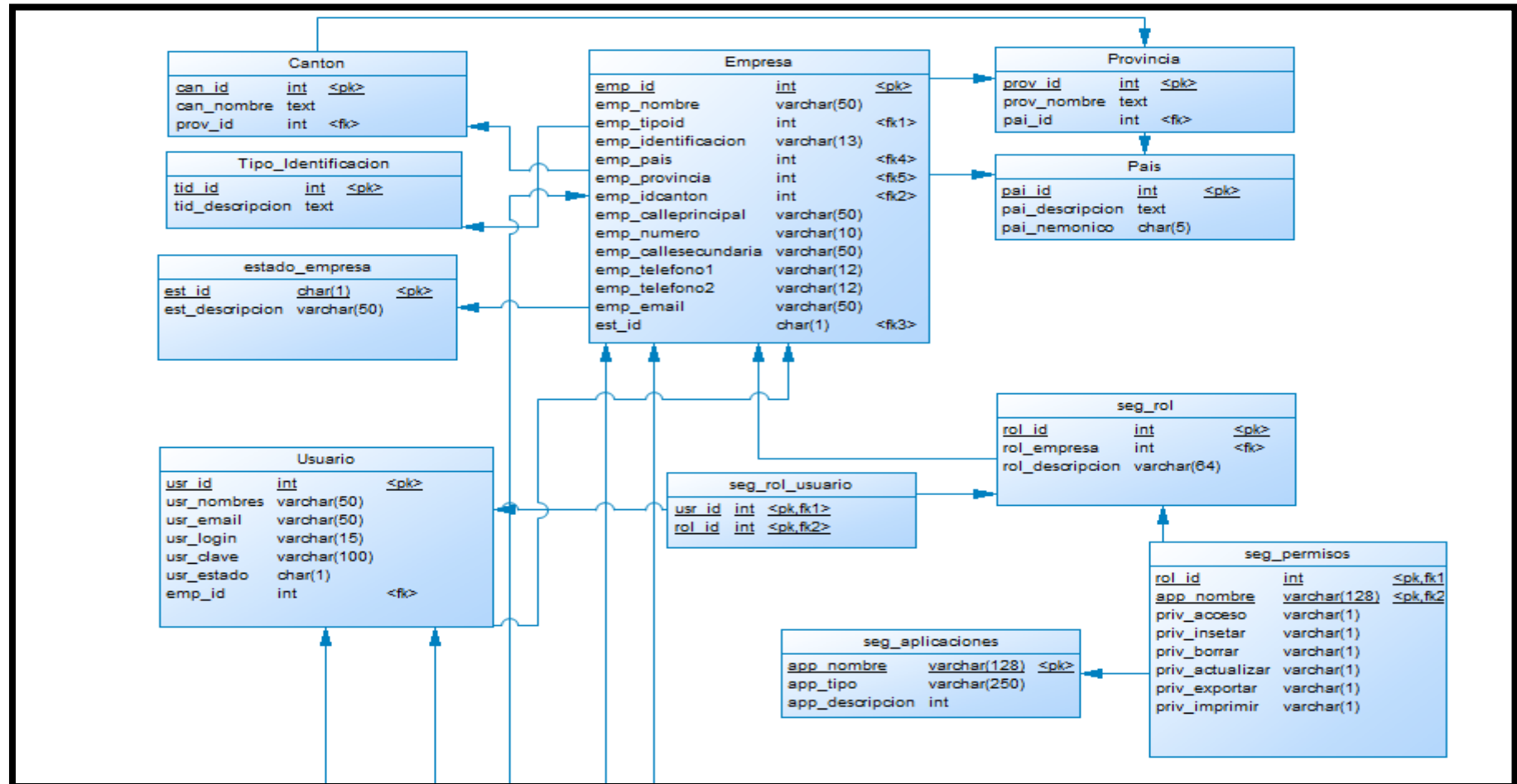
### 5.3. PRODUCCIÓN

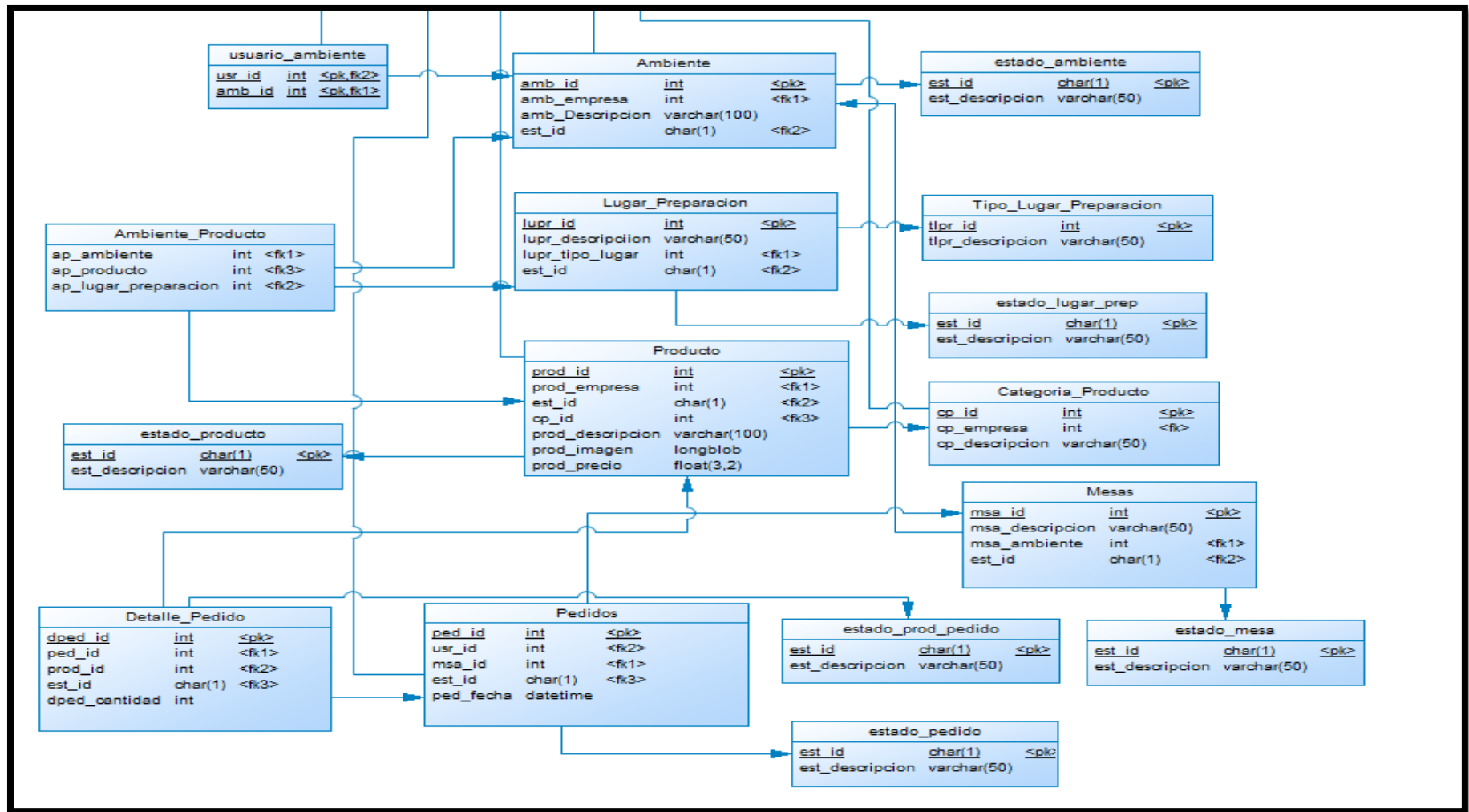
En la etapa de producción se realiza el desarrollo del sistema en base a las iteraciones planteadas en la etapa de inicialización.

Como primera tarea está el diseño de la base de datos relacional que se utilizara para el almacenamiento de los datos.



### 5.3.1. Diagrama Relacional de Base de datos

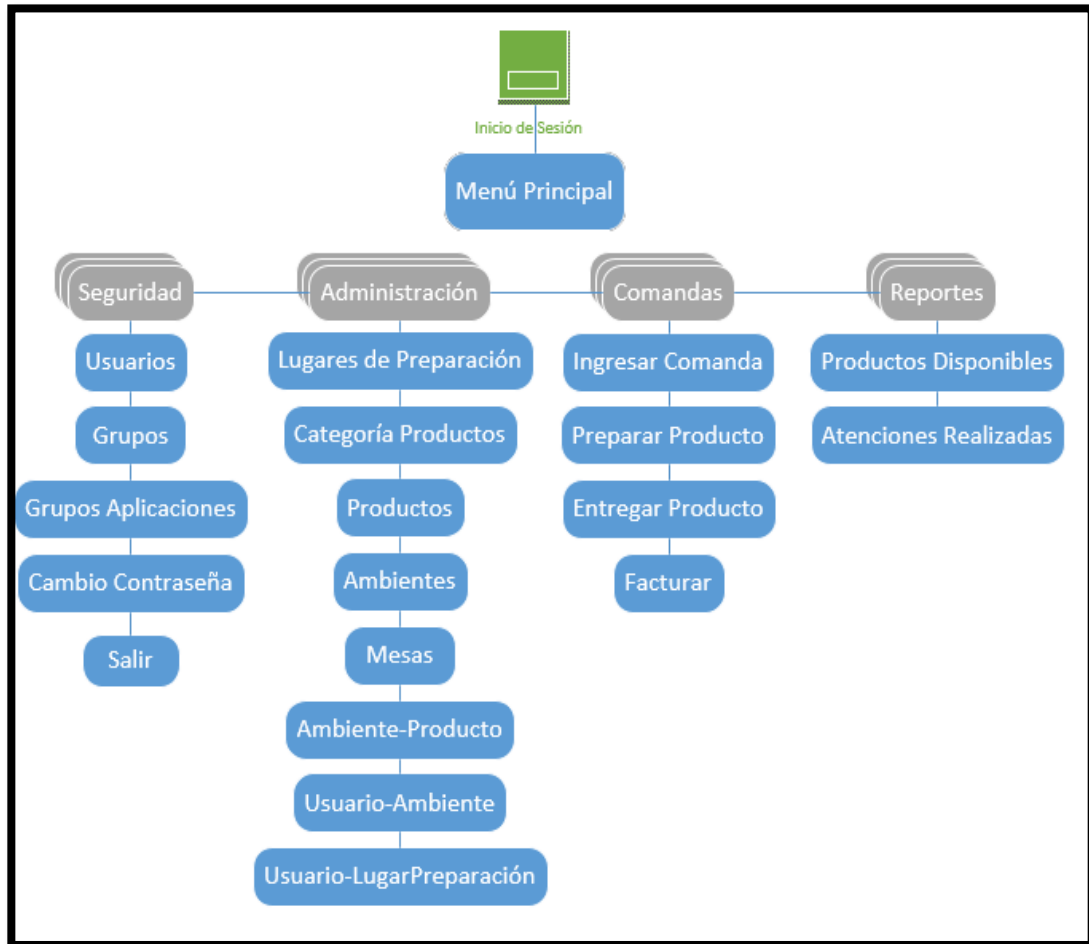




Fuente: Propia  
 Figura 71. Diagrama Relacional de Base de Datos

## 5.3.2. Producción de Iteraciones

### 5.3.2.1. Diseño Menú Principal



*Fuente: Propia*  
*Figura 72. Diseño Menú Principal*

### 5.3.2.2. Pantalla de inicio

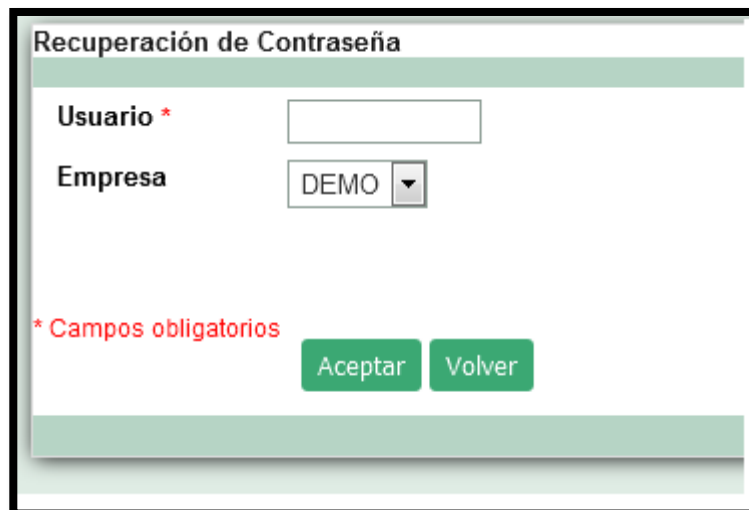


The screenshot shows the login page for 'Restaurante'. At the top left, the word 'Inicio' is displayed. The main header features the 'Restaurante' logo in a green script font, accompanied by three images of food dishes. Below the header, there are two dropdown menus for language selection: 'Español (Ecuador)' and 'Sc8\_Ceropegia'. The login form includes fields for 'Empresa \*' (with a 'DEMO' dropdown), 'Usuario \*' (containing 'admin'), and 'Contraseña \*' (with masked characters). A green 'Login' button is located at the bottom right. A red asterisk indicates that the fields are mandatory. A link for 'Recordar contraseña' is also present.

Fuente: Propia  
Figura 73. Pantalla de Inicio

En la pantalla de inicio los usuarios seleccionan la empresa en la que trabajan y podrán iniciar sesión con las credenciales asignadas.

En caso de no recordarlas puede hacer uso de la opción de Recordar contraseña.



The screenshot shows the 'Recuperación de Contraseña' (Password Recovery) page. It features a title bar at the top. Below it, there are two input fields: 'Usuario \*' (with an empty text box) and 'Empresa' (with a 'DEMO' dropdown menu). A red asterisk indicates that the fields are mandatory. At the bottom, there are two green buttons: 'Aceptar' and 'Volver'.

Fuente: Propia  
Figura 74. Pantalla de Recuperación de Contraseña

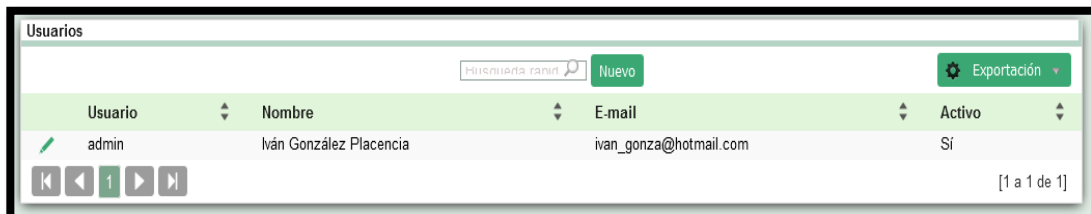
### 5.3.2.3. Menú principal



*Fuente: Propia*  
Figura 75. Pantalla de Menú Principal

Una vez que el usuario inicia sesión con sus credenciales, se le presenta el menú principal con las opciones de acuerdo al rol asignado.

### 5.3.2.4. Manejo de usuarios



*Fuente: Propia*  
Figura 76. Pantalla de Manejo de Usuarios

Desde la pantalla de Usuarios se tiene acceso a la creación de nuevos usuarios o la actualización de los usuarios ya existentes.

**Nuevo Usuario**

[Guardar](#)

Usuario \*

Contraseña \*

Confirmar contraseña \*

Nombre

E-mail \*

Estado  Sí  No

\* Campos obligatorios

Fuente: Propia  
 Figura 77. Pantalla de Nuevo Usuario

**Actualizar Usuario**

[Volver](#)

USR ID 0

Nombre \*

E-mail \*  [Email](#)

Activo  Sí  No

GRUPO

CAJERO  
 COCINA  
 MESERO

\* Campos obligatorios

[Actualizar](#) [Borrar](#)

Fuente: Propia  
 Figura 78. Pantalla de Actualizar Usuario

### 5.3.2.5. Manejo de Grupos

**Lista de Grupos**

[Nuevo](#)  [Exportación](#)

	Id	Descripción
	1	ADMINISTRADOR
	2	MESERO
	3	COCINA
	4	CAJERO

[1 a 4 de 4]

Fuente: Propia  
 Figura 79. Pantalla de manejo de Grupos



La pantalla de manejo de Grupos tiene las opciones de crear un nuevo Grupo o editar y eliminar un Grupo ya existente.

En la opción Nuevo se despliega una pantalla con el campo del Nombre del nuevo Grupo.



The screenshot shows a web form titled "Nuevo Grupos". It features a single text input field for entering the name of the new group. Below the input field are two green buttons: "Guardar" (Save) and "Salir" (Exit).

*Fuente: Propia*  
Figura 80. Pantalla de Nuevo Grupo

En la opción de editar se abre una pantalla con el campo que se desea actualizar, en esta pantalla también tiene opción para eliminar o crear un nuevo Grupo.



The screenshot shows a web form titled "Actualizar Grupos". It features a text input field containing the text "ADMINISTRADOR". Below the input field are four green buttons: "Nuevo" (New), "Actualizar" (Update), "Borrar" (Delete), and "Salir" (Exit).

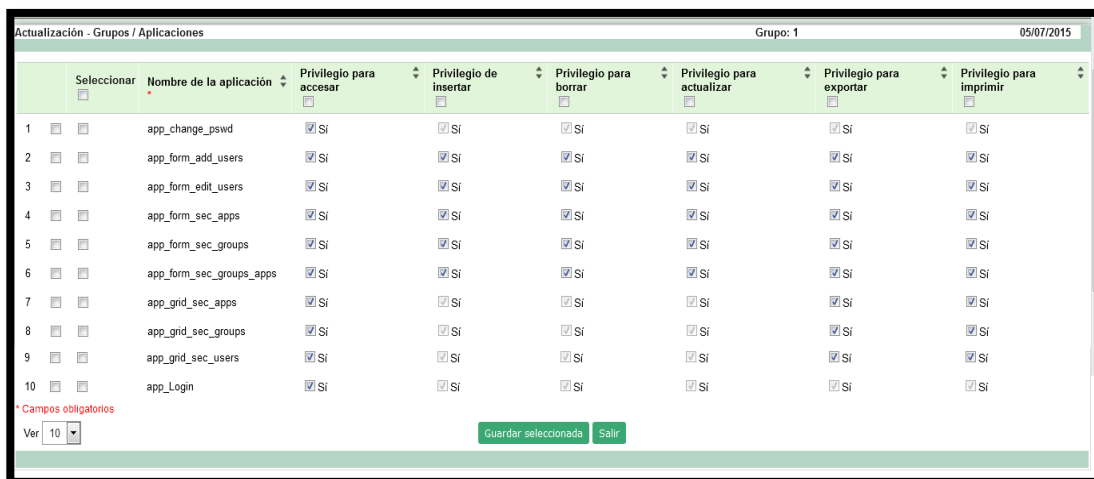
*Fuente: Propia*  
Figura 81. Pantalla de Actualizar Grupo

### 5.3.2.6. Asignación de Permisos a Grupos



Fuente: Propia  
Figura 82. Pantalla de Seguridad de Grupos

Desde la pantalla de Seguridad de Grupos, se selecciona el grupo al que se le va a asignar permisos, en donde se podrá asignar privilegios de acceso, insertar, borrar, actualizar, exportar o imprimir.



Fuente: Propia  
Figura 83. Pantalla de Permisos de Grupos.

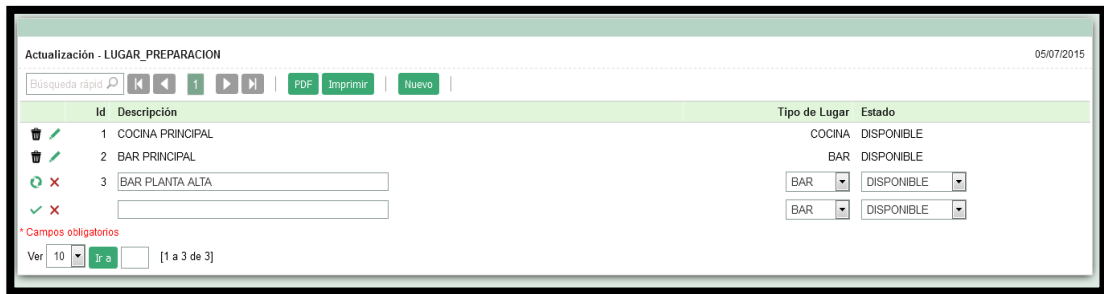
### 5.3.2.7. Cambiar contraseña



Fuente: Propia  
Figura 84. Pantalla de cambio de contraseña

Desde la opción de cambiar contraseña cada usuario puede establecer una nueva contraseña para el ingreso a la aplicación.

### 5.3.2.8. Manejo de Lugares de Preparación

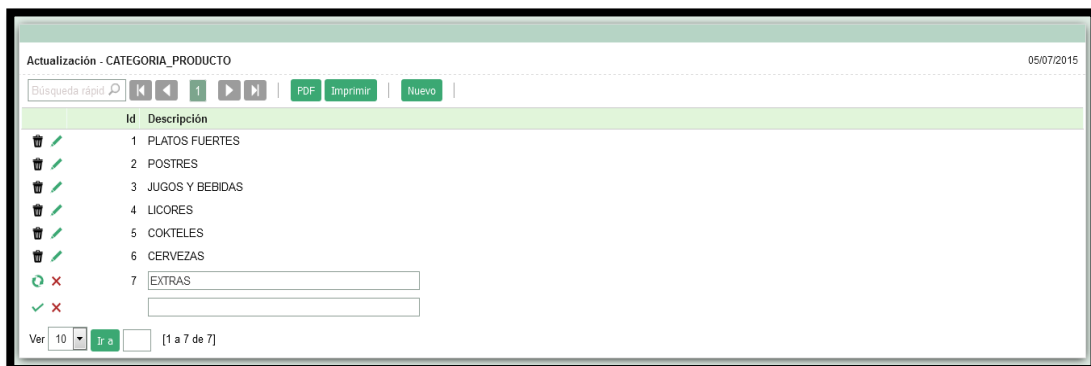


Fuente: Propia  
Figura 85. Pantalla de Lugares de Preparación

La pantalla de Lugares de preparación tiene la opción de Crear un nuevo lugar o editar un Lugar de preparación ya existente. Estos lugares pueden ser de tipo Cocina o Bar de acuerdo a como quiera configurar el administrador de la empresa.

Este tipo sirve para diferenciar a que aplicación móvil enviar la solicitud de elaboración de una comanda.

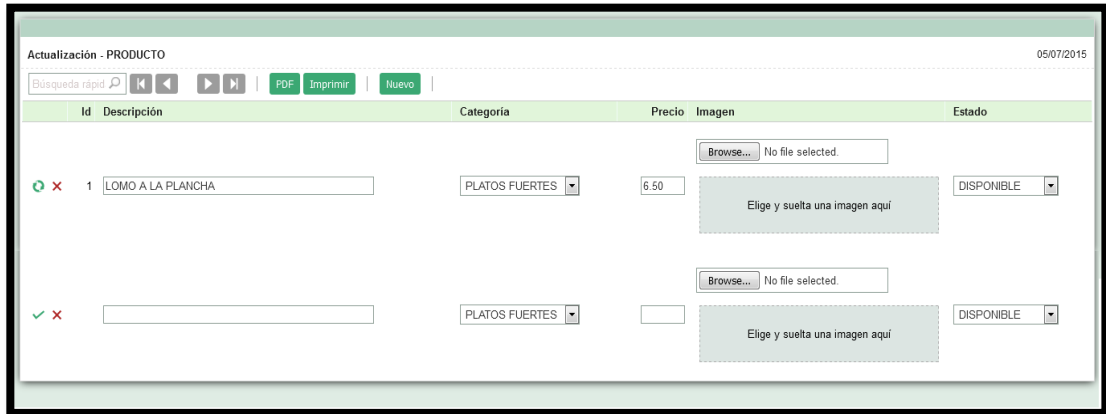
### 5.3.2.9. Manejo de categoría de productos



Fuente: Propia  
Figura 86. Pantalla de Manejo de Categorías de Productos

Esta pantalla cuenta con la opción de crear, editar o eliminar una Categoría de Producto, estas categorías de productos se utilizan posteriormente en la creación de Productos.

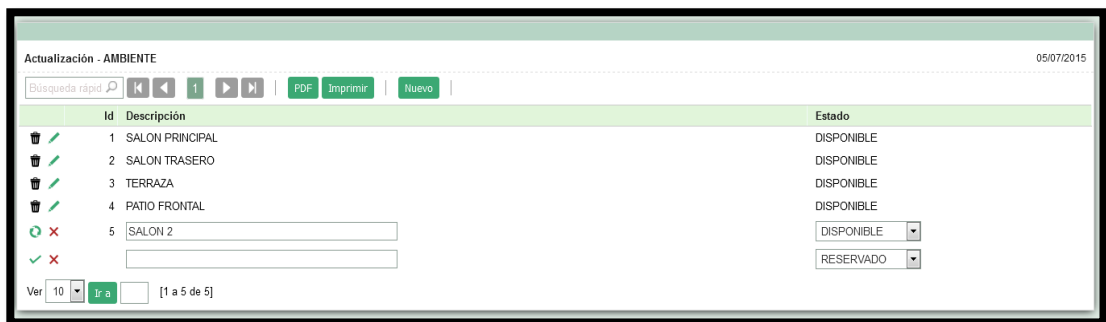
### 5.3.2.10. Manejo de Productos



Fuente: Propia  
Figura 87. Pantalla de manejo de Productos

La opción de manejo de Productos cuenta con las opciones de crear, editar o eliminar productos disponibles para cada empresa, estos productos se los clasifica de acuerdo una categoría.

### 5.3.2.11. Manejo de Ambientes



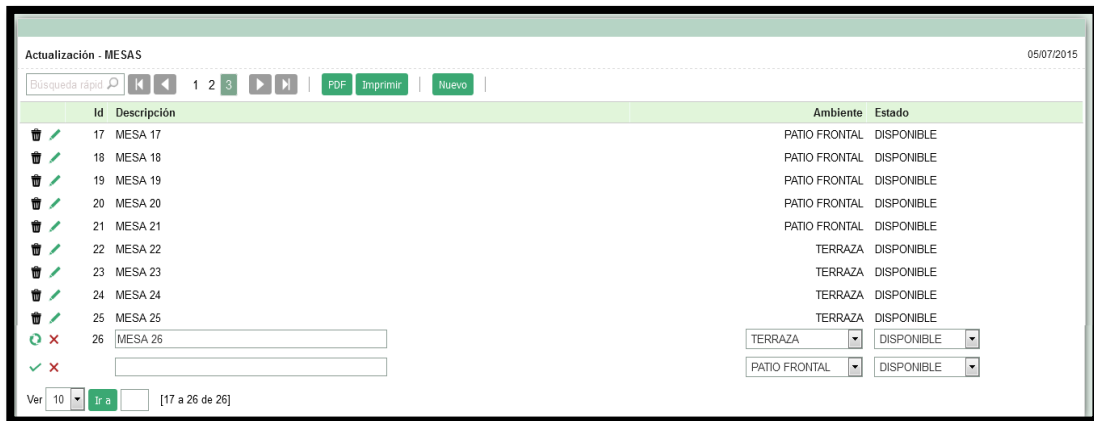
Fuente: Propia  
Figura 88. Pantalla de manejo de Ambientes

Para un adecuado manejo de la empresa, el sistema brinda la opción de dividir su infraestructura en ambientes.

La pantalla de manejo de Ambientes cuenta con las opciones de crear, editar o eliminar Ambientes de acuerdo a la subdivisión que se desea dar a la empresa.

A estos ambientes posteriormente se les asignará Mesas.

### 5.3.2.12. Manejo de Mesas



Fuente: Propia  
Figura 89. Pantalla de Manejo de Mesas

La pantalla de Manejo de Mesas tiene las opciones de crear, editar o eliminar una mesa y configurar a que Ambiente pertenece dicha Mesa.

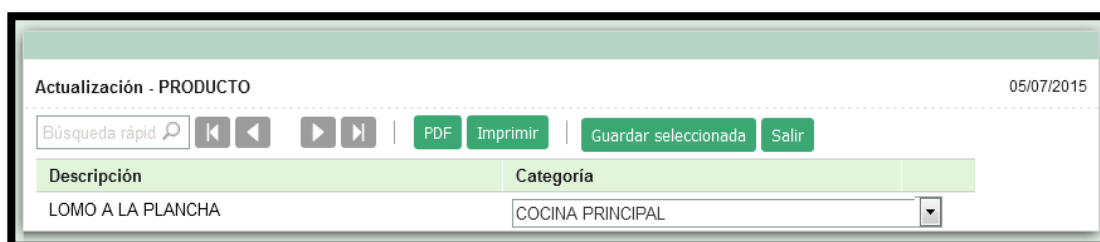
### 5.3.2.13. Asignación de Productos a Ambientes



Fuente: Propia  
Figura 90. Pantalla de Selección de Ambiente

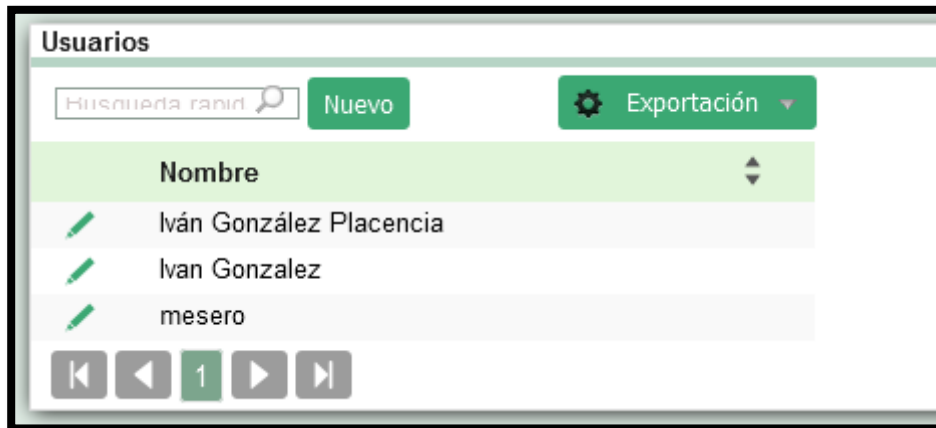
La idea principal de la Asignación de Productos a Ambientes es configurar a que lugares de preparación se enviará la solicitud de productos de una comanda.

En primer lugar se selecciona el ambiente a configurar, se muestra una pantalla con la lista de productos disponibles en la empresa y en esta pantalla se selecciona el lugar de preparación.



Fuente: Propia  
Figura 91. Pantalla de configuración de Lugar de Preparación de Productos.

### 5.3.2.14. Asignación de Usuarios a Ambientes



Fuente: Propia  
Figura 92. Pantalla de Selección de Usuario

La pantalla de Asignación de Usuarios a Ambientes tiene la opción de configurar que ambientes van a ser atendidos por un determinado usuario.

Parte de la selección de un Usuario y se presenta una pantalla en la que puede Agregar uno o más Ambientes de la empresa.



Fuente: Propia  
Figura 93. Pantalla de Asignación de Ambientes a Usuario.

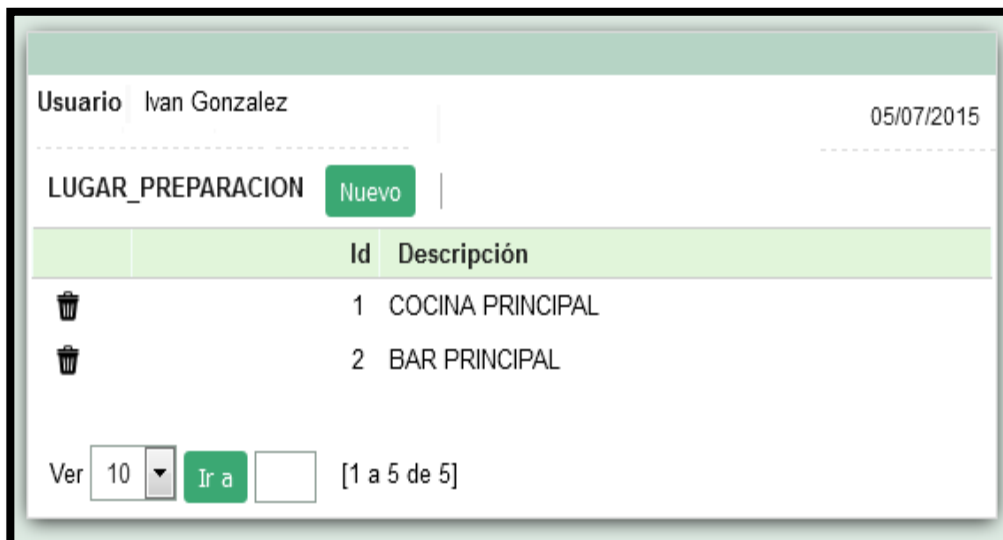
### 5.3.2.15. Asignación de Usuarios a Lugares de Preparación



Fuente: Propia  
Figura 94. Pantalla de Selección de Usuario

La pantalla de Asignación de Usuarios a Lugares de Preparación tiene la opción de configurar en qué Lugares de Preparación va a trabajar un determinado usuario.

Parte de la selección de un Usuario y se presenta una pantalla en la que puede Agregar uno o más Lugares de Preparación de la empresa.



Fuente: Propia  
Figura 95. Pantalla de Asignación de Lugares de Preparación a Usuario



### 5.3.2.16. Manejos de estados de Comandas



Fuente: Propia  
Figura 96. Pantalla de Manejo de Comandas

En la pantalla de manejo de estado de Comandas se muestra una lista de Comandas en proceso de preparación o que han sido entregadas, para que el administrador continúe con el proceso de cobro y cambie el estado a PAGADO.

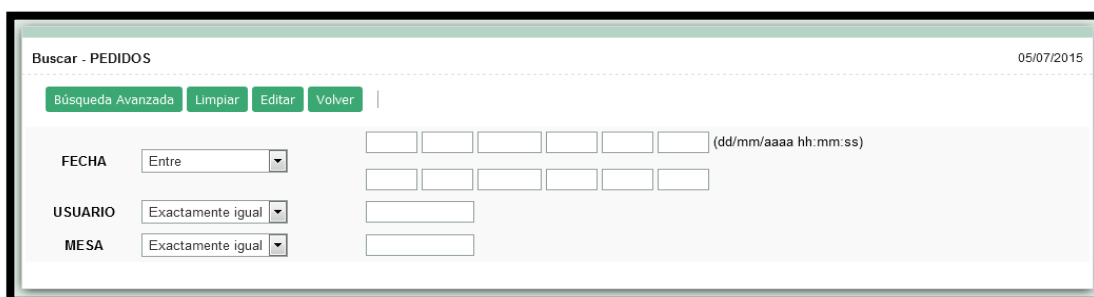
### 5.3.2.17. Reportes de Comandas



Fuente: Propia  
Figura 97. Pantalla de Reporte de Comandas.

En la pantalla de Reporte de Comandas se muestra un listado de todas las comandas que ya han sido Pagadas.

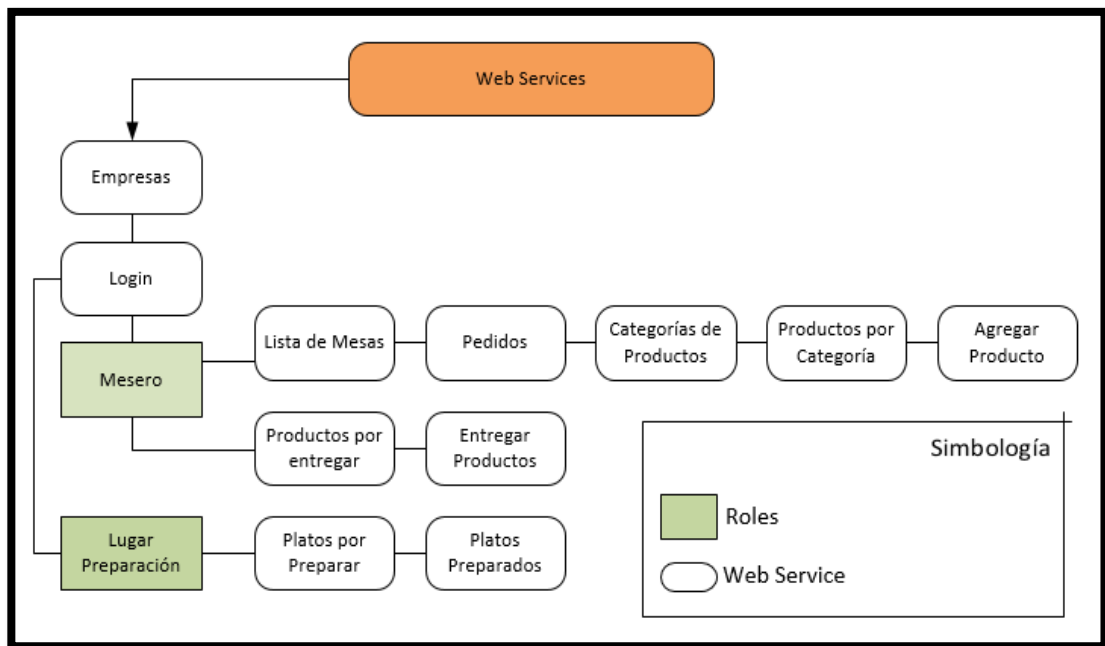
Estos reportes cuentan con filtros para mejorar la calidad de los reportes.



Fuente: Propia  
Figura 98. Pantalla de filtros para Reportes.

### 5.3.2.18.Elaboración de Servicios Web

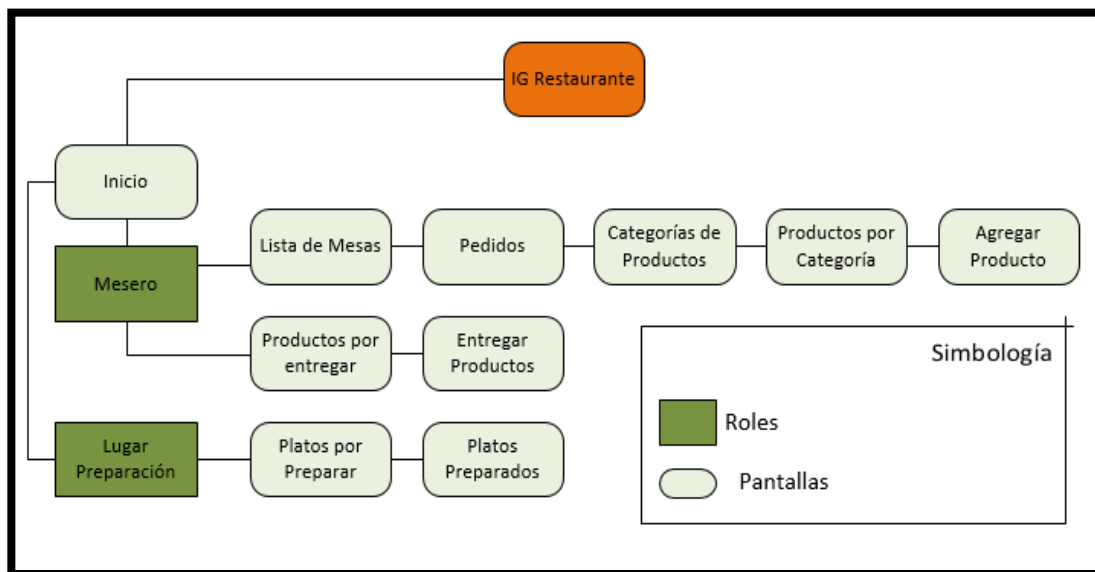
Para la comunicación entre las aplicaciones móviles y el servidor web se han elaborado varios servicios web que se detallan en el siguiente gráfico.



*Fuente: Propia*

*Figura 99. Esquema de Servicios Web de la Aplicación Móvil*

### 5.3.2.19. Diseño de Aplicación Móvil



Fuente: Propia  
Figura 100. Diseño Aplicación de Lugares de Preparación

### 5.3.2.20. Inicio de Sesión



*Fuente: Propia*  
*Figura 101. Diseño de Inicio de Sesión*

En la pantalla de inicio de sesión se selecciona la empresa y se ingresa las credenciales asignadas a cada usuario, si las credenciales ingresadas son válidas se mostrara la siguiente pantalla con las opciones para cada tipo de usuario.

Si el usuario registrado es un Mesero se le presentará una pantalla con las opciones:

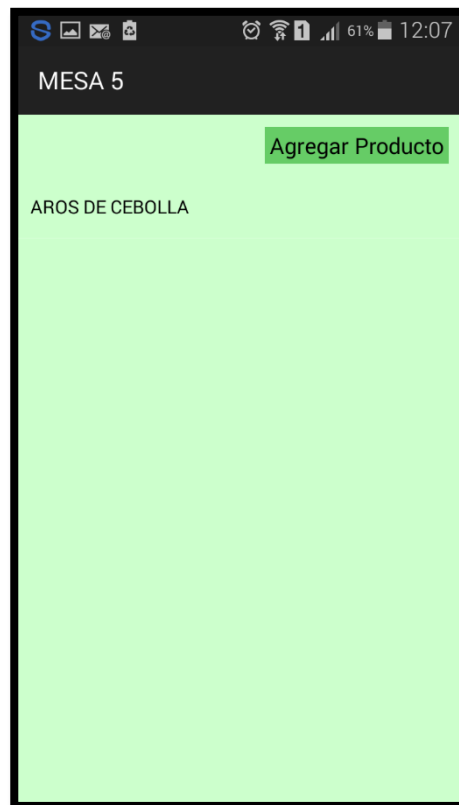
### 5.3.2.21. Lista de mesas

E-restaurante		
MESAS		ENTREGAR
MESA 15	PATIO FRONTAL	Ir
MESA 16	PATIO FRONTAL	Ir
MESA 17	PATIO FRONTAL	Ir
MESA 18	PATIO FRONTAL	Ir
MESA 19	PATIO FRONTAL	Ir
MESA 20	PATIO FRONTAL	Ir
MESA 21	PATIO FRONTAL	Ir
MESA 7	SALON 2	Ir
MESA 8	SALON 2	Ir
MESA 9	SALON 2	Ir

*Fuente: Propia*  
*Figura 102. Lista de Mesas*

Esta pantalla muestra al usuario todas las mesas que le han sido asignadas para poder atenderlas.

### 5.3.2.22. Presentación de la Mesa



*Fuente: Propia*

Figura 103. Presentación del Menú

Cuando selecciona una de las mesas se le mostrará el estado actual de la mesa seleccionada, es decir si la mesa está vacía se mostrará una pantalla en blanco para poder agregar los productos que los clientes indiquen, caso contrario se mostrara la lista de productos ya solicitados y se podrá ingresar nuevos productos.

### 5.3.2.23. Agregar Productos

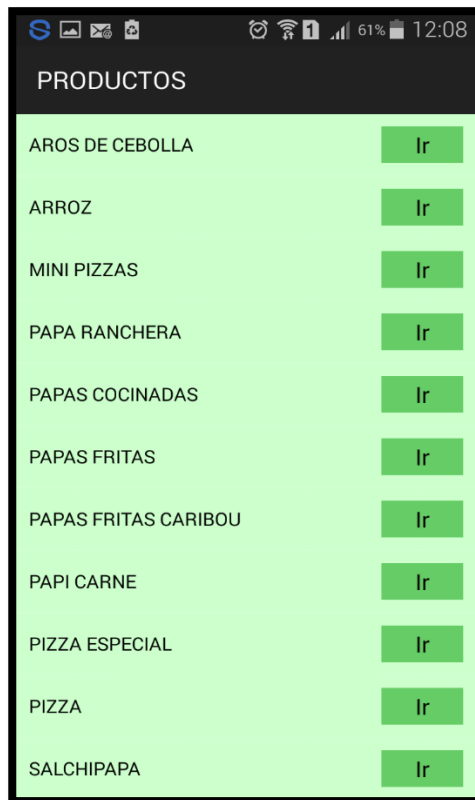


*Fuente: Propia*

*Figura 104. Presentación de Categorías de productos*

Cuando se desea agregar un producto, lo primero que se muestra es una pantalla con las categorías de productos con las que la empresa cuenta.

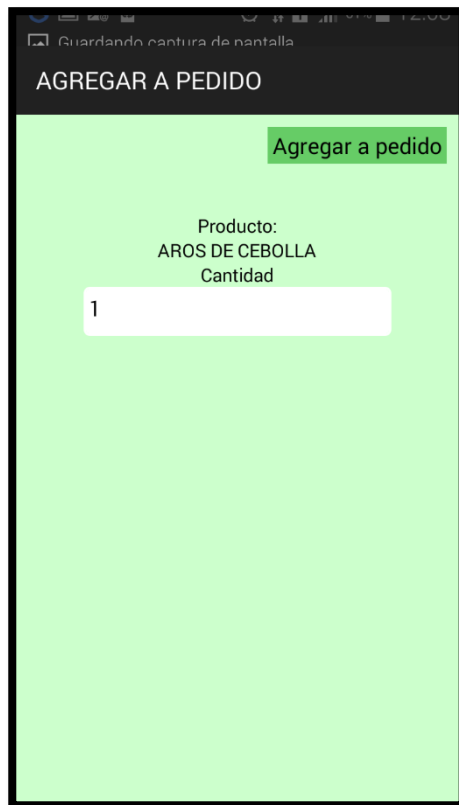
Al seleccionar una de estas categorías aparecerá en la pantalla un listado de todos los productos pertenecientes a esta categoría.



*Fuente: Propia*  
*Figura 105. Presentación de productos*

Cuando selecciona el producto se mostrara una pantalla para poder agregar la cantidad que el cliente requiere.





*Fuente: Propia*

Figura 106. Agregar producto a pedido

Al presionar el botón de *Agregar a Pedido* automáticamente se enviará la orden de preparación a los lugares predefinidos en el módulo de administración.

### 5.3.2.24. Entregar productos



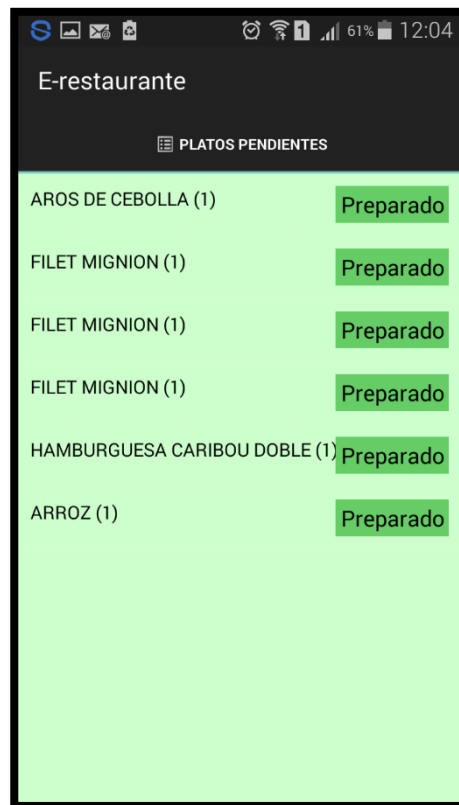
*Fuente: Propia*

Figura 107. Listado de productos por entregar

Cuando el mesero selecciona la pestaña de *Entregar* se le mostrara un listado de todos los productos pendientes por entregar, con la información de la mesa y la cantidad.

El mesero deberá presionar el botón de entregar para que el producto ya no se muestre en la lista y pueda ser cobrado.

### 5.3.2.25. Preparación de Productos



*Fuente: Propia*

*Figura 108. Lista de productos por preparar*

Para los usuarios encargados de la preparación de las comandas, en esta opción se les muestra un listado de las comandas ingresadas por los meseros, este listado se muestra ordenado de acuerdo al primer pedido ingresado.

Una vez que se prepara un producto de la comanda se actualiza su estado a **Preparado**, esto indica al mesero que puede ir a recoger el pedido para su entrega.

## **5.4. ESTABILIZACIÓN**

### **5.4.1. Día de planificación**

El día de planificación en la etapa de estabilización se programa las tareas para la implementación de las funcionalidades desarrolladas.

### **5.4.2. Día de trabajo**

En el día de trabajo se realiza la implementación de las funcionalidades desarrolladas en la etapa de producción.

### **5.4.3. Documentación**

En la etapa de estabilización además se realiza la documentación de la aplicación que se presentara en la sección de anexos del presente documento.

## **5.5. PRUEBAS DEL SISTEMA**

### **5.5.1. Día de Planificación**

Para las pruebas del sistema se realiza una planificación en cuanto a los puntos que se van a probar.

Se establecen los formatos para documentar los resultados de las pruebas.

IG Restaurante						
Plan de Pruebas Unitarias						
Número del Caso de Prueba		Componente	Descripción de lo que se Probará	Prerrequisitos		
CA001						
N°	Descripción	Método	Datos Entrada	Salida Esperada	¿OK?	Observaciones

*Fuente: Propia*  
**Figura 109. Modelo para realización de Pruebas**

Para cada una de las funcionalidades se realiza pruebas en los siguientes aspectos:

- Datos validos
- Valores limite
- Datos inválidos
- Diseño de acuerdo a la documentación
- Enlaces a otras pantallas adecuadas.

Para esto se lleva la documentación adecuada.

### **5.5.2. Día de trabajo**

En el día de trabajo se realizan las pruebas de acuerdo a la planificación realizada.

Se realiza la documentación con las observaciones de errores encontrados a fin de realizar las mejoras necesarias.

### **5.5.3. Correcciones**

Para cada uno de los errores encontrados se realiza una iteración de pruebas y corrección de errores hasta que todos los problemas sean debidamente solventados y obtener un producto de calidad.

**HOJA DE CONTROL**

<b>Organismo</b>	SGM		
<b>Proyecto</b>	IG Restaurante		
<b>Entregable</b>	Plan de Pruebas Unitarias		
<b>Autor</b>	Iván González		
<b>Versión / Edición</b>		<b>Fecha Versión</b>	
<b>Aprobado Por</b>		<b>Fecha Aprobación</b>	
		<b>Nº Total de Páginas</b>	

REGISTRO DE CAMBIOS

Versión	Causa del cambio	Responsable del cambio	Fecha del cambio

Fuente: Propia  
 Figura 110. Hoja de Control de Correcciones.

**5.5.4. Análisis de Resultados Módulo de Administración**

Se presenta los resultados de las pruebas del sistema.

**5.5.4.1. Inicio de Sesión**

<b>Iteración:</b>	<b>Inicio de Sesión</b>	
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	6	60%
<b>Pruebas Reprobadas</b>	4	40%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	4	40%

Fuente: Propia  
 Tabla 19. Resultados de Iteración Inicio de Sesión

#### 5.5.4.2. Definición Grupos

Iteración:	Definición Grupos	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	9	90%
Pruebas Reprobadas	1	10%
Total	10	100%
Pruebas Corregidas	1	10%

Fuente: Propia

Tabla 20. Resultados de Iteración Definición de Grupos

#### 5.5.4.3. Definición de Usuarios

Iteración:	Definición de Usuarios	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	7	70%
Pruebas Reprobadas	3	30%
Total	10	100%
Pruebas Corregidas	3	30%

Fuente: Propia

Tabla 21. Resultados de Iteración Definición de Usuarios

#### 5.5.4.4. Asignación de permisos a Grupos

Iteración:	Asignación de permisos a Grupos	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	3	30%
Pruebas Reprobadas	7	70%
Total	10	100%
Pruebas Corregidas	7	70%

Fuente: Propia

Tabla 22. Resultados de Iteración Asignación de Permisos a Grupos

#### 5.5.4.5. Cambio de contraseña

<b>Iteración: Cambio de contraseña</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	7	70%
<b>Pruebas Reprobadas</b>	3	30%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	3	30%

Fuente: Propia

Tabla 23. Resultados de Iteración cambio de Contraseña

#### 5.5.4.6. Lugares de preparación

<b>Iteración: Lugares de preparación</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	6	60%
<b>Pruebas Reprobadas</b>	4	40%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	4	40%

Fuente: Propia

Tabla 24. Resultados de Iteración Lugares de Preparación

#### 5.5.4.7. Categorías de productos

<b>Iteración: Categorías de productos</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	5	50%
<b>Pruebas Reprobadas</b>	5	50%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	5	50%

Fuente: Propia

Tabla 25. Resultados de Iteración Categoría de Productos



#### 5.5.4.8. Productos

<b>Iteración:</b>	<b>Productos</b>	
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	8	80%
<b>Pruebas Reprobadas</b>	2	20%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	2	20%

Fuente: Propia

Tabla 26. Resultados de Iteración Productos

#### 5.5.4.9. Ambientes

<b>Iteración:</b>	<b>Ambientes</b>	
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	9	90%
<b>Pruebas Reprobadas</b>	1	10%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	1	10%

Fuente: Propia

Tabla 27. Resultados de Iteración Ambientes

#### 5.5.4.10. Asignación de Productos a Ambientes

<b>Iteración:</b>	<b>Asignación de Productos a Ambientes</b>	
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	7	70%
<b>Pruebas Reprobadas</b>	3	30%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	3	30%

Fuente: Propia

Tabla 28. Resultados de Iteración Asignación de Productos a Ambientes

#### 5.5.4.11.Mesas

Iteración:	Mesas	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	8	80%
Pruebas Reprobadas	2	20%
<b>Total</b>	<b>10</b>	<b>100%</b>
Pruebas Corregidas	2	20%

Fuente: Propia

Tabla 29. Resultados de Iteración Mesas

#### 5.5.4.12.Asignación de Meseros a Ambientes

Iteración:	Asignación de Meseros a Ambientes	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	5	50%
Pruebas Reprobadas	5	50%
<b>Total</b>	<b>10</b>	<b>100%</b>
Pruebas Corregidas	5	50%

Fuente: Propia

Tabla 30. Resultados de Iteración Asignación de Meseros a Ambientes

#### 5.5.4.13.Finalizar comanda

Iteración:	Finalizar comanda	
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	10	100%
Pruebas Reprobadas	0	0%
<b>Total</b>	<b>10</b>	<b>100%</b>
Pruebas Corregidas	0	0%

Fuente: Propia

Tabla 31. Resultados de Iteración Finalizar Comanda

#### 5.5.4.14. Resumen de comandas

Iteración: Resumen de comandas		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	8	80%
Pruebas Reprobadas	2	20%
Total	10	100%
Pruebas Corregidas	2	20%

Fuente: Propia

Tabla 32. Resultados de Iteración Resumen de Comandas

#### 5.5.5. Análisis de Resultados Servicios Web

##### 5.5.5.1. Inicio de sesión

Iteración: Inicio de sesión		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	6	60%
Pruebas Reprobadas	4	40%
Total	10	100%
Pruebas Corregidas	4	40%

Fuente: Propia

Tabla 33. Resultados de Iteración Inicio de Sesión

##### 5.5.5.2. Lista de Mesas y productos

Iteración: Lista de Mesas y productos		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	3	30%
Pruebas Reprobadas	7	70%
Total	10	100%
Pruebas Corregidas	7	70%

Fuente: Propia

Tabla 34. Resultados de Iteración Lista de Mesas y Productos

### 5.5.5.3. Agregar Comandas

<b>Iteración: Agregar Comandas</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	5	50%
<b>Pruebas Reprobadas</b>	5	50%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	5	50%

Fuente: Propia

Tabla 35. Resultados de Iteración Agregar Comandas

### 5.5.5.4. Listar Comandas

<b>Iteración: Listar Comandas</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	5	50%
<b>Pruebas Reprobadas</b>	5	50%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	5	50%

Fuente: Propia

Tabla 36. Resultados de Iteración Listar Comandas

### 5.5.5.5. Modificar Comanda

<b>Iteración: Modificar Comanda</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	8	80%
<b>Pruebas Reprobadas</b>	2	20%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	2	20%

Fuente: Propia

Tabla 37. Resultados de Iteración Modificar Comanda

### 5.5.5.6. Cambiar Comanda

Iteración: Cambiar Comanda		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	4	40%
Pruebas Reprobadas	6	60%
Total	10	100%
Pruebas Corregidas	6	60%

Fuente: Propia

Tabla 38. Resultados de Iteración Cambiar Comanda

### 5.5.6. Análisis de Resultados aplicación móvil

#### 5.5.6.1. Iniciar sesión

Iteración: Inicio de sesión		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	7	70%
Pruebas Reprobadas	3	30%
Total	10	100%
Pruebas Corregidas	3	30%

Fuente: Propia

Tabla 39. Resultados de Iteración Iniciar Sesión

#### 5.5.6.2. Ingreso de Comanda

Iteración: Ingreso de Comanda		
	Número de Pruebas	Porcentaje
Pruebas Aceptadas	4	40%
Pruebas Reprobadas	6	60%
Total	10	100%
Pruebas Corregidas	6	60%

Fuente: Propia

Tabla 40. Resultados de Iteración Ingreso de Comanda

### 5.5.6.3. Modificar comandas

<b>Iteración: Modificar comandas</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	8	80%
<b>Pruebas Reprobadas</b>	2	20%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	2	20%

Fuente: Propia

Tabla 41. Resultados de Iteración Modificar Comanda

### 5.5.6.4. Iniciar sesión

<b>Iteración: Iniciar sesión</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	7	70%
<b>Pruebas Reprobadas</b>	3	30%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	3	30%

Fuente: Propia

Tabla 42. Resultados de Iteración Iniciar Sesión

### 5.5.6.5. Listar Comandas

<b>Iteración: Listar Comandas</b>		
	<b>Número de Pruebas</b>	<b>Porcentaje</b>
<b>Pruebas Aceptadas</b>	9	90%
<b>Pruebas Reprobadas</b>	1	10%
<b>Total</b>	10	100%
<b>Pruebas Corregidas</b>	1	10%

Fuente: Propia

Tabla 43. Resultados de Iteración Listar Comandas

#### 5.5.6.6. Cambiar Comanda

<b>Iteración:</b>		<b>Cambiar Comanda</b>	
	<b>Número de Pruebas</b>	<b>Porcentaje</b>	
<b>Pruebas Aceptadas</b>	8	80%	
<b>Pruebas Reprobadas</b>	2	20%	
<b>Total</b>	10	100%	
<b>Pruebas Corregidas</b>	2	20%	

Fuente: Propia

*Tabla 44. Resultados de Iteración Cambiar Comanda*





## CAPÍTULO VI



### CONCLUSIONES Y RECOMENDACIONES

---

- Conclusiones
  - Recomendaciones
  - Impactos
  - Posibles temas para trabajos de grado
-

## 6.1. CONCLUSIONES

Tomando en cuenta los objetivos y alcance planteados en el desarrollo de este proyecto se ha llegado a las siguientes conclusiones:

- El uso de nuevas herramientas de desarrollo de aplicaciones móviles como Appcelerator disminuye significativamente el tiempo y coste de un proyecto, ya que permite generar aplicaciones compatibles para diferentes plataformas a partir de un solo proyecto de desarrollo basado en un único lenguaje de programación.
- El estudio minucioso de las herramientas de desarrollo utilizadas, en el diseño e implementación de nuevas aplicaciones permite aprovechar al máximo las bondades que estas nos brindan.
- La selección de una metodología de desarrollo de software como Mobile-D hace que se agilice la generación de un producto, ya que fue diseñada para proyectos de aplicaciones móviles y se fundamenta en metodologías de desarrollo rápido.
- La utilización de la tecnología informática esta cada día más inmersa en todo tipo de negocios, y las empresas dedicadas a la comercialización de alimentos preparados no son la excepción, por lo que el uso de esta aplicación puede mejorar la calidad del servicio brindado, reduciendo los tiempos de espera para que un cliente sea atendido.
- Utilizar JSON como formato de comunicación entre la aplicación móvil y el servidor web es la mejor opción al momento de trabajar con Titanium Appcelerator ya que los dos están íntimamente relacionados con el lenguaje JavaScript.

## 6.2. RECOMENDACIONES

Una vez concluido este proyecto se ha llegado a las siguientes recomendaciones:

- Profundizar el estudio del framework Appcelerator para explotar todas las ventajas que presenta al momento de desarrollar un proyecto de aplicaciones móviles.
- Utilizar la herramienta Titanium Appcelerator para desarrollar aplicaciones móviles ya que permite desde un solo proceso de desarrollo, generar aplicaciones compatibles con varias plataformas móviles.
- Realizar actualizaciones a la aplicación con los nuevos requerimientos que los usuarios puedan determinar.
- Utilizar la metodología Mobile-D en nuevos proyectos de desarrollo ya que es una metodología de desarrollo ágil que permite generar proyectos escalables y en donde se detectan y solventan errores oportunamente.

## 6.3. IMPACTOS

Tipo de Impactos	Descripción
Económico	Se mejora los procesos de atención, lo que conlleva al ahorro de tiempos y aprovechamientos del talento humano.
Ambiental	Con esta aplicación se reduce las impresiones y el uso de papel al tomar ordenes por parte de los meseros.
Tecnológico	Las instituciones hacen uso de la tecnología para mejorar sus procesos.
Social	Las instituciones ganan prestigio al contar con mejor servicios que la competencia.

Fuente: Propia

Tabla 45. Análisis de Impactos

#### **6.4. POSIBLES TEMAS PARA TRABAJOS DE GRADO**

Como complemento a la aplicación desarrollada se pueden tener algunos nuevos proyectos, como:

- Aplicación móvil para compra de materia prima para elaboración de productos en restaurantes en base a un inventario.
- Aplicación móvil para realizar pedidos a domicilio a un restaurante.
- Sistema de facturación electrónica.

## REFERENCIAS BIBLIOGRÁFICAS

### Páginas Web

1. Androideity. (19 de Septiembre de 2011). <http://androideity.com/>. Obtenido de <http://androideity.com/2011/09/19/appcelerator-titanium-para-crear-aplicaciones-android>
2. Alonso, E. G. (2012). Obtenido de <http://mural.uv.es/egara3/inf2p2/secciones.html#css>
3. Appcelerator. (06 de 2015). <https://docs.appcelerator.com>. Obtenido de <https://docs.appcelerator.com>: <https://docs.appcelerator.com>
4. Appcelerator Wiki. (s.f.). *Appcelerator Wiki*. Obtenido de <https://wiki.appcelerator.org>
5. Artedynamico. (s.f.). <http://www.artedynamico.net>. Obtenido de <http://www.artedynamico.net/sitio/cursos.php?c=205>
6. Blanco, P., Camarero, J., Fumero, A., Wertesky, A., & Rodriguez, P. (2013). *Metodología de desarrollo ágil para sistemas móviles*. Universidad Politécnica de Madrid.
7. Cherry. (02 de 2011). <http://sistemamanejadorbasededatosmbd.blogspot.com>. Recuperado el 03 de 2014, de <http://sistemamanejadorbasededatosmbd.blogspot.com>: <http://sistemamanejadorbasededatosmbd.blogspot.com/2011/02/diferentes-tipos-de-sistemas-de.html>

8. ConceptoWeb. (s.f.). <http://www.conceptoweb.cl/servicios/programacion-web.html>.  
Obtenido de <http://www.conceptoweb.cl/servicios/programacion-web.html>
9. DeIdeaAapp. (5 de 5 de 2014). [deideaaapp.org](http://deideaaapp.org). Obtenido de <http://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>
10. [deideaaapp.org](http://deideaaapp.org). (2015). Obtenido de <https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>
11. Formacion, D. (15 de 04 de 2014). <http://www.deustoformacion.com/>. Obtenido de <http://www.deustoformacion.com/blog-tic/2014/04/15/que-es-y-como-se-hace-una-app/>
12. GONZÁLEZ, D. R. (2013). *EVALUACIÓN DE LA HERRAMIENTA TITANIUM STUDIO PARA EL DESARROLLO DE APLICACIONES MULTIDISPOSITIVO*. Oviedo.
13. Heller, M. (7 de julio de 2014). *InfoWorld*. Obtenido de <http://cioperu.pe/fotoreportaje/16329/10-herramientas-para-la-creacion-de-aplicaciones-moviles/>
14. Heurtel, O. (2011). *PHP 5.3 desarrollar un sitio Web dinámico e Interactivo*. Ediciones ENI.
15. <http://phonegap.com/>. (s.f.). <http://phonegap.com/>. Obtenido de <http://phonegap.com/>: <http://phonegap.com/>
16. Json., O. (2015). <http://json.org/json-es.html>. Obtenido de <http://json.org/json-es.html>: <http://json.org/json-es.html>

17. PHP. (03 de 2014). *php.net*. Obtenido de php.net:  
<http://www.php.net/manual/es/intro-what-is.php>
18. Qode. (6 de 2014). *www.qode.pro*. Obtenido de <http://qode.pro/blog/web-app-vs-app-nativa/>
19. Sánchez Maza, M. Á. (2001). *JavaScript*. INNOVA.
20. Santillán, L. A. (2004). *uoc.edu*. Recuperado el 03 de 2014, de uoc.edu:  
[http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02151.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf)
21. ScriptCase. (2015). <http://www.scriptcase.net/>. Obtenido de <http://www.scriptcase.net/>: <http://www.scriptcase.net/>
22. VVT . (s.f.). *Agile Software Mobile-D*. Obtenido de <http://agile.vtt.fi/mobiled.html>

### **Libros digitales**

1. Brousseau, C. (2013). *Creating Mobile Apps with Appcelerator Titanium*. Packt Publishing, Obtenido de <https://books.google.com.ec/books?id=fSC8AQAQBAJ&lpg=PT548&dq=Appcelerator&hl=es&pg=PT548#v=onepage&q=Appcelerator&f=false>.
2. Arroyo, N. (2011). *Información en el móvil*. Barcelona: UPC. Obtenido de <https://books.google.com.ec/books?id=KtAtAwwAAQBAJ&lpg=PP1&dq=Informaci%C3%B3n%20en%20el%20m%C3%B3vil%20arroyo&hl=es&pg=PP1#v=onepage&q=Informaci%C3%B3n%20en%20el%20m%C3%B3vil%20arroyo&f=false>.

3. Minera, F. (2011). Desarrollador web. Buenos Aires: Fox Andina. Obtenido de <https://books.google.com.ec/books?id=MLjtHxxjITgC&lpg=PA402&dq=Desarrollador%20web&hl=es&pg=PA402#v=onepage&q=Desarrollador%20web&f=false>
  
4. Minera, F. (s.f.). Desarrollo PHP + MySQL. Obtenido de <https://books.google.com.ec/books?id=IMk02EiOrcAC&lpg=PA66&dq=Desarrollo%20PHP%20%2B%20MySQL%20minera&hl=es&pg=PA68#v=onepage&q=Desarrollo%20PHP%20+%20MySQL%20minera&f=false>
  
5. Minera, F. (s.f.). Desarrollo Web Profesional. Obtenido de <https://books.google.com.ec/books?id=dIyfaFtlxrAC&lpg=PA2&dq=Desarrollo%20Web%20Profesional%20minera&hl=es&pg=PA2#v=onepage&q=Desarrollo%20Web%20Profesional%20minera&f=false>.
  
6. Pollentine, B. (2011). Appcelerator Titanium Smartphone App Development Cookbook. Obtenido de <https://books.google.com.ec/books?id=uUPeCS9MnNIC&lpg=PA1&dq=Appcelerator&hl=es&pg=PA1#v=onepage&q=Appcelerator&f=false>.
  
7. Sriparasa, S. S. (2013). JavaScript and JSON Essentials. Packt Publishing Ltd. Obtenido de <https://books.google.com.ec/books?id=MZOkAQAAQBAJ&lpg=PT124&dq=JavaScript%20and%20JSON%20Essentials&hl=es&pg=PT124#v=onepage&q=JavaScript%20and%20JSON%20Essentials&f=false>.



8. Vásquez Roja, J. (2008). Consulta y actualización de base de datos mediante equipos móviles. Medellín. Obtenido de <https://books.google.com.ec/books?id=1oHQcenEP1MC&lpg=PP1&dq=Consulta%20y%20actualizaci%C3%B3n%20de%20base%20de%20datos%20mediante%20equipos%20m%C3%B3viles&hl=es&pg=PP1#v=onepage&q=Consulta%20y%20actualizaci%C3%B3n%20de%20base%20de%20datos%20mediante%20equipos%20m%C3%B3viles&f=false>.
  
9. Zakas, N. C. (2010). High Performance JavaScript. (M. E. Treseler, Ed.) Estados Unidos: O'Reilly Media. Obtenido de <https://books.google.com.ec/books?id=ED6ph4WEIoQC&lpg=PP1&dq=High%20Performance%20JavaScript&hl=es&pg=PP1#v=onepage&q=High%20Performance%20JavaScript&f=false>.
  
10. Bahrenburg, B. (2013). Appcelerator Titanium Business Application Development Cookbook. Packt Publishing Ltd. Obtenido de <https://books.google.com.ec/books?id=igix9GkI8VIC&lpg=PT28&dq=Appcelerator%20Titanium%20Business%20Application%20Development%20Cookbook&hl=es&pg=PT28#v=onepage&q=Appcelerator%20Titanium%20Business%20Application%20Development%20Cookbook&f=false>

## GLOSARIO

### A

**Android:** Sistema Operativo para dispositivos móviles basado en linux, de libre distribución.

**API:** es la abreviatura de Application Programming Interface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como por ejemplo, imprimir un caracter en pantalla, leer el teclado, escribir en un fichero de disco, etc.

**App:** Abreviatura de application. Utilizado para referirse a una aplicación móvil.

**ASP:** (Active Server Pages) Es un tipo de páginas Web que permiten ser personalizadas a medida de las características y necesidades del usuario visitante. Además, también hace referencia a Application Service Provider. Es decir, proveedor de servicios de aplicaciones.

**AVD:** Android Virtual Device o Dispositivo Virtual Android es una aplicación de escritorio que emula las funciones de un equipo Android.

### B

**BaaS:** Backend as a service, es un modelo para proporcionar a los desarrolladores web y de aplicaciones móviles una forma de vincular estas aplicaciones al almacenamiento en nube.

**BlackBerry:** Sistema Operativo desarrollado por BlackBerry para sus productos.

**BLOB:** (Binary Large Objects, objetos binarios grandes) son elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica.

## C

**Comandas:** Término que se utiliza para identificar un pedido en un restaurante.

**Crystal methodologies:** familia de metodologías descritas por Alistair Cockburn y se considera un ejemplo de metodologías ágiles, 67

**CSS:** Es un lenguaje utilizado en la presentación de documentos HTML.

## D

**DBMS:** del inglés Database Manager Sistem, sistema gestor de base de datos.

## E

**ECMA:** es una organización internacional basada en membresías de estándares para la comunicación y la información.

## F

**Framework:** Marco de trabajo para desarrollo de software.

## G

**GPL:** Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU GPL), 22

## H

**HTML:** Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, lenguaje de programación para desarrollo de páginas de internet.

**HTML5:** Es la última evolución de la norma que define HTML.

## I

**IDE:** (Integrated Development Environment - Entorno integrado de desarrollo).

Aplicación compuesta por un conjunto de herramientas útiles para un programador.

**IOs:** Sistema Operativo desarrollado por Apple para sus productos iPhone, iPod, iPad.

**IP:** Dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo de Internet.

**ISO 9126:** es un estándar internacional para la evaluación de la calidad del software.

## J

**JavaScript:** (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**JDK:** Java Development Kit, conjunto de herramientas y librerías para desarrollar en lenguaje Java.

**jQuery Mobile:** Framework optimizado para dispositivos móviles.

**JSON:** Es un acrónimo de JavaScript Object Notation, un formato ligero originalmente concebido para el intercambio de datos en Internet.

**JSP:** Java Servlet Pages, es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML

## M

**Mobile-D:** Metodología de desarrollo de software.

**Multiplataforma:** Aplicación que puede utilizarse en diversos entornos o sistemas operativos.

**MySQL:** Es un sistema de gestión de base de datos relacional de código abierto, basado en lenguaje de consulta estructurado (SQL),

**MVC:** Modelo Vista Controlador, es un patrón de arquitectura de software que separa los datos y la lógica de negocio de la interfaz de usuario en una aplicación.

## N

**Node.js:** es una plataforma que permite programar servidores dedicados en Javascript de forma sencilla y eficaz.

## O

**Open Source:** Es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente.

## P

**PDA:** Asistente digital personal, del inglés Personal Digital Assistant.

**PhoneGap:** Framework para desarrollo de aplicaciones móviles híbridas.

**PHP:** (acrónimo recursivo de PHP; Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

## R

**RAD:** (acrónimo en inglés de Rapid Application Development) desarrollo rápido de aplicaciones.

**REST:** (Representational State Transfer) es un estilo de arquitectura para desarrollar servicios.

**ROI:** son las siglas en inglés de Return On Investment (Retorno de la Inversión) y es un porcentaje que se calcula en función de la inversión y los beneficios obtenidos.

**RUP:** Rational Unified Process (RUP) es un framework iterativo del proceso de desarrollo de software creado por el Rational Software Corporation.

## S

**SD:** de las siglas ("Secure Digital") ó seguridad digital, medio de almacenamiento digital.

**SDK:** Software Developed Kit o Paquete de desarrollo de software.

**Servicios web:** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

**SGBD:** Sistema Gestor de Bases de Datos.

**Sistema Operativo:** Conjunto de órdenes y programas que controlan los procesos básicos de una computadora o dispositivo móvil y permiten el funcionamiento de otros programas.

**Smartphones:** Smarthpones; teléfono movil inteligente.

**SQLite:** es un sistema de gestión de base de datos simplificada.

**Symbian:** Sistema operativo para productos de la Empresa Nokia.

## T

**Titanium Appcelerator:** Framework de desarrollo móvil y web multiplataforma.

## U

**Unicode:** es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de múltiples lenguajes y disciplinas técnicas, además de textos clásicos de lenguas muertas.

**URL:** Como medio de localización (direccionamiento) de los distintos recursos de internet.

## **V**

**VTT:** VTT Research Centre Ltd. es la empresa líder de la investigación y la tecnología de la región nórdica de Europa

## **W**

**W3C:** Consorcio World Wide Web,

**Windows Phone:** Sistema Operativo para equipos móviles desarrollado por Microsoft.

## **X**

**XP:** eXtreme Programming (Programación extrema) es una metodología de desarrollo de la ingeniería de software

## **ANEXOS**

### **SOFTWARE**

- Código fuente sistema de toma de pedidos en restaurantes.
- Appcelerator
- MySql Workbench
- Genymotion

### **MANUALES**

- Requerimientos
- Manual de instalación
- Manual de usuario
- Manual técnico.