



SEDE CUENCA

CARRERA: INGENIERIA DE SISTEMAS

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO DE SISTEMAS

***“DESARROLLO DE UNA APLICACIÓN, PARA DISPOSITIVOS
MÓVILES QUE PERMITA ADMINISTRAR PEDIDOS Y
CONTROLAR RUTAS DE LOS VENEDORES, APLICADA A LA
EMPRESA: “ALMACENES JUAN ELJURI CÍA. LTDA.”
DIVISIÓN PERFUMERÍA”***

AUTOR:

JOSE RICARDO CAJILIMA ALVARADO

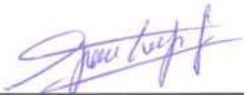
DIRECTORA:

ING. PAOLA INGAVELEZ

Cuenca, Marzo del 2015

Declaratoria de Responsabilidad

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad del autor. Autorizo a la Universidad Politécnica Salesiana el uso de la misma para fines académicos.



José Cajilima Alvarado

AUTOR

Cuenca, Marzo de 2015

Ing. Paola Ingavelez

CERTIFICA

Haber dirigido y revisado cada uno de los capítulos del presente informe realizado por JOSE RICARDO CAJILIMA ALVARADO, así como el cumplimiento y desarrollo de la parte práctica; en base a ello y cumpliendo con todos los requisitos necesarios, autorizo la presentación de la misma.


Ing. Paola Ingavelez
DIRECTORA

Cuenca, Marzo de 2015

DEDICATORIA

A mi *Madre* que está en el cielo, Sabiendo que le hubiese gustado estar presente físicamente y verme cumpliendo una de mis metas, porque espiritualmente siempre ha estado a mi lado. Este Trabajo está dedicado a su memoria.

Este trabajo también va dedicado a las mujeres que más amo: A mi esposa “*Diana*” y a mis hijas: “*Valentina*” y “*Amalia*” quienes son el motor de mi vida, y por las que lucho cada día para salir adelante y verles siempre sonreír.

Por ti *Diana*, gracias por estar a mi lado todo este tiempo, por darme esta gran alegría de poder formar mi propia familia a tu lado y por apoyarme y estar conmigo en las buenas y en las malas y por compartir esta hermosa experiencia de ser “padres” de nuestras hermosas niñas.

AGRADECIMIENTO

Comenzaré agradeciendo a *Dios* por permitirme estar de pie ante las adversidades y dificultades que se me han presentado y por cuidar siempre de mis seres queridos.

Por otro lado quiero agradecer enormemente a mi familia en especial a mis hermanos *Wilson, Edgar y Diego* quienes se encuentran lejos y siempre me han apoyado moralmente como económicamente para poder seguir con mis estudios. También agradecer a todos mis profesores que me supieron ayudar y guiarme durante todos mis estudios.

Finalmente agradecer a esas personas que fueron apareciendo y del alguna forma supieron ganarse mi confianza. Gracias *Karina Matute* por tu apoyo, gracias por ayudarme a diferenciar a un verdadero “amigo” ya que tú eres una de ellas y Gracias *Andres Sampedro* pese a no ser compañero de aula y conocerte hace poco tiempo, me echaste una mano por lo que te estoy eternamente agradecido.

INDICE GENERAL

Contenido

Declaratoria de Responsabilidad.....	I
CERTIFICA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
INDICE GENERAL.....	V
INDICE DE ILUSTRACIONES.....	VIII
INDICE DE TABLAS	IX
INTRODUCCION	X
Capítulo I Situación Actual del Proyecto.....	1
1.1 Tema del Proyecto	2
1.2 Objetivos del Proyecto.....	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos.....	2
1.3 Justificación del Proyecto	2
1.4 Alcance del Proyecto	3
1.5 Empresa.....	5
1.5.1 Antecedentes.....	5
1.5.2 Misión	5
1.5.3 Visión.....	6
1.6 Procesos Actuales en la Gestión de Pedidos	6
Capítulo II Marco Teórico	8
2.1 Tecnología Móvil.....	8
2.1.1 Sistemas Operativos para móviles.	10
2.1.2 Android	11
2.1.2.1 Arquitectura de Android	14
2.1.2.2 Estructura o componentes de una aplicación Android.....	17
2.1.2.3 SDK de Android.....	18
2.1.2.4 Google Maps en Android.....	19
2.1.2.5 SQLITE.....	20
2.1.2.6 DALVIK	21
2.1.2.7 Breve descripción de las Versiones de Android	21

2.2	Algunas Metodologías existentes para desarrollo de aplicaciones móviles.....	23
2.2.1	Metodología RUP	23
2.2.1.1	Estructura del RUP.....	23
2.2.1.2	Ciclo de Vida	24
2.2.1.3	Disciplinas de un proceso RUP.....	25
2.2.2	Metodología XP (Extreme Programming).....	26
2.2.2.1	Proceso XP.....	28
2.2.2.2	Fases de desarrollo de la Programación Extrema	29
2.3	Análisis comparativo	31
2.4	Descripción de la metodología seleccionada.	33
Capítulo III Determinación, Instalación y Configuración de las Herramientas de Desarrollo.....		34
3.1	Plataformas de desarrollo.....	35
3.1.1	Android SDK	35
3.2	Arquitectura de la aplicación	36
3.2.1	Web Service (Servicio Web).....	37
3.2.1.1	Ventajas de los Servicios Web:.....	38
3.2.1.2	JSON (JavaScript Object Notation)	38
3.2.2	Internet Information Services (IIS)	39
3.3	Selección del entorno de desarrollo	39
3.3.1	Eclipse.....	41
3.3.2	Arquitectura de Eclipse	42
3.3.3	Plugin ADT para Eclipse	43
3.4	Especificaciones e instalación de la tecnología de desarrollo.....	43
3.4.1	Instalación y configuración de JDK.....	44
3.4.2	Instalación y configuración del SDK de Android	45
3.4.3	Instalación y configuración de Eclipse	47
3.4.4	Instalar el plugin de Android para Eclipse	49
3.4.5	Creación de un nuevo proyecto Android	51
3.4.6	Creación del Android Virtual Device (AVD)	55
3.4.7	Ejecución de aplicaciones Android usando un dispositivo físico	56
Capítulo IV Desarrollo de la Aplicación para la Administración de Pedidos y Control de Rutas de los Vendedores.		58

4.1	Fase de Planificación	59
4.1.1	Descripción de los Interesados (Stakeholders)	59
4.1.2	Requerimientos Funcionales del proyecto.	61
4.1.3	Requerimientos No Funcionales de la Aplicación.	62
4.1.4	Restricciones	64
4.2	Fase de Diseño	66
4.2.1	Diagrama de Casos de Uso	67
4.2.1.1	Caso de Uso General, Aplicación de Pedidos.	67
4.2.1.2	Caso de Uso General, Página Web.	71
4.2.2	Diagrama de Clases.....	73
4.2.3	Diagramas de Secuencia	76
4.2.3.1	Diagrama de Secuencia, Inicio de Sesión.	76
4.2.3.2	Diagrama de Secuencia, Enviar Pedido	77
4.2.3.3	Diagrama de Secuencias, Interacción con la Intranet	78
4.2.4	Diagrama Arquitectónico.....	79
4.2.5	Diagrama de Navegación.....	80
4.2.6	Diseño de los bocetos de las pantallas del sistema (Prototipos)	81
4.2.6.1	Boceto de la Pantalla de Inicio de Sesión.	82
4.2.6.2	Boceto de la Pantalla del Menú.....	82
4.2.6.3	Boceto de la Pantalla del Catálogo.....	83
4.2.6.4	Boceto de la Pantalla de Pedido	83
4.2.6.5	Boceto de la Pantalla de Ayuda	85
4.3	Fase de Codificación.....	86
4.3.1	Requerimientos de desarrollo.....	86
4.3.2	Desarrollo del módulo de Gestión de Pedidos.	87
4.3.2.1	Descripción La Clase Java MainActivity.....	87
4.3.2.2	Descripción de la Clase Java Login.	87
4.3.2.3	Archivo XML Activity_Menu.xml (Menú Principal).....	87
4.3.3	Desarrollo del módulo de Control de Rutas de los vendedores.	88
4.3.3.1	Obtener una API Key de Google Maps.....	88
4.3.3.2	Agregar un Mapa de Google Maps en ASP.net	88
4.4	Fase de Pruebas.....	89
4.4.1	Elaboración y Ejecución del plan de Pruebas	89

4.4.2	Análisis de resultados.....	91
4.5	Documentación.....	92
4.5.1	Manual de la Aplicación de Pedidos.....	92
4.5.2	Manual de Uso de la Intranet, Modulo Revisión de Rutas.....	92
CONCLUSIONES		93
RECOMENDACIONES		94
BIBLIOGRAFÍA		96
ANEXOS		99
Anexo N° 1.	Fragmento de código clase MainActivity.java	100
Anexo N° 2.	Fragmento código Clase Login.java	101
Anexo N° 3.	Descripción código del archivo Activity_Menu.xml	103
Anexo N° 4.	Manual de la Aplicación de Pedidos.....	106
Anexo N° 5.	Manual de Uso de la Intranet, Control de Rutas de Vendedores.	114

INDICE DE ILUSTRACIONES

<i>Ilustración 1.</i>	<i>Esquema de la aplicación</i>	<i>3</i>
<i>Ilustración 2.</i>	<i>Proceso Actual en la gestión de Pedidos A. Juan Eljuri Div. Perfumería.</i>	<i>6</i>
<i>Ilustración 3.</i>	<i>Logotipo Android.....</i>	<i>12</i>
<i>Ilustración 4.</i>	<i>Arquitectura de Android.....</i>	<i>15</i>
<i>Ilustración 5.</i>	<i>Página de descarga de SDK Android.....</i>	<i>19</i>
<i>Ilustración 6.</i>	<i>Instalación Google Play Services.....</i>	<i>20</i>
<i>Ilustración 7:</i>	<i>Esfuerzo en actividades del RUP</i>	<i>25</i>
<i>Ilustración 8:</i>	<i>Flujo de características de la Programación Extrema.</i>	<i>28</i>
<i>Ilustración 9:</i>	<i>Arquitectura de la Aplicación</i>	<i>36</i>
<i>Ilustración 10:</i>	<i>Arquitectura de la Intranet.....</i>	<i>37</i>
<i>Ilustración 11:</i>	<i>Sitio Web de Oracle para descargar JDK.....</i>	<i>45</i>
<i>Ilustración 12:</i>	<i>Características de la Instalación de JDK.....</i>	<i>45</i>
<i>Ilustración 13:</i>	<i>Sitio Web para descarga del SDK Android.....</i>	<i>46</i>
<i>Ilustración 14:</i>	<i>Instalación de APIs del SDK de Android</i>	<i>47</i>
<i>Ilustración 15:</i>	<i>Descarga de Eclipse Juno del sitio web oficial.....</i>	<i>48</i>
<i>Ilustración 16:</i>	<i>Crear Dispositivo Virtual en Eclipse.....</i>	<i>49</i>
<i>Ilustración 17:</i>	<i>Pasos para agregar plugin Android en Eclipse</i>	<i>50</i>
<i>Ilustración 18:</i>	<i>Ventana Agregar Repositorio.....</i>	<i>50</i>

<i>Ilustración 19: Selección para instalación de paquetes para Android.....</i>	<i>51</i>
<i>Ilustración 20: Creación de un Nuevo Proyecto Android en Eclipse</i>	<i>52</i>
<i>Ilustración 21: Parámetros de un Proyecto Android en Eclipse</i>	<i>53</i>
<i>Ilustración 22: Estructura Proyecto Android en Eclipse.....</i>	<i>54</i>
<i>Ilustración 23: Abrir Android Virtual Manager AVD.....</i>	<i>55</i>
<i>Ilustración 24: Creación y configuración del Android Virtual Manager AVD</i>	<i>56</i>
<i>Ilustración 25: Ejecución de proyecto en Android Virtual Manager AVD.....</i>	<i>56</i>
<i>Ilustración 26: Opciones del Desarrollador en dispositivo móvil.....</i>	<i>57</i>
<i>Ilustración 27: Dispositivo Android instalado en el AVD Manager.....</i>	<i>58</i>
<i>Ilustración 28: Caso de Uso General la Aplicación de Pedidos.....</i>	<i>68</i>
<i>Ilustración 29: Caso de Uso 2, Crear Pedido.....</i>	<i>69</i>
<i>Ilustración 30: Caso de Uso Intranet de la Empresa.....</i>	<i>72</i>
<i>Ilustración 31. Diagrama de Clases de la Aplicación de Pedidos.....</i>	<i>75</i>
<i>Ilustración 32. Diagrama de Secuencia, Inicio de Sesión.....</i>	<i>76</i>
<i>Ilustración 33. Diagrama de Secuencia, Enviar Pedido.....</i>	<i>77</i>
<i>Ilustración 34. Diagrama de Secuencia, Interacción con la Intranet.....</i>	<i>78</i>
<i>Ilustración 35: Diagrama Arquitectónico.....</i>	<i>79</i>
<i>Ilustración 36. Diagrama de Navegación</i>	<i>80</i>
<i>Ilustración 37: Página Oficial NinjaMock para Prototipos.....</i>	<i>81</i>
<i>Ilustración 38: Prototipo Pantalla de Inicio de Sesión.....</i>	<i>82</i>
<i>Ilustración 39. Prototipo Pantalla Menú Aplicación.....</i>	<i>82</i>
<i>Ilustración 40. Prototipo Pantalla Catálogo</i>	<i>83</i>
<i>Ilustración 41. Prototipo Pantalla de Pedido</i>	<i>84</i>
<i>Ilustración 42. Prototipo, Pantalla Buscar por Código.....</i>	<i>84</i>
<i>Ilustración 43. Prototipo Pantalla Ingresar Pedido</i>	<i>85</i>
<i>Ilustración 44. Prototipo Pantalla Detalle Pedido.....</i>	<i>85</i>
<i>Ilustración 45. Prototipo, Pantalla Ayuda.....</i>	<i>86</i>
<i>Ilustración 46. Pruebas Realizadas en una Samsung Galaxy Tab 3 T211.....</i>	<i>91</i>

INDICE DE TABLAS

<i>Tabla 1. Descripción de las Versiones de Android.....</i>	<i>22</i>
<i>Tabla 2. Análisis comparativo entre las Metodologías RUP y XP.....</i>	<i>32</i>
<i>Tabla 3. Herramientas utilizadas para el desarrollo de la aplicación.....</i>	<i>40</i>

Tabla 4. <i>Matriz de Stakeholders del Proyecto</i>	60
Tabla 5. <i>Resumen de las Tareas de los involucrados</i>	61
Tabla 6. <i>Descripción de Tablets con los requisitos para la Aplicación</i>	66
Tabla 7. <i>Actores de la aplicación de Pedidos</i>	67
Tabla 8. <i>Especificación Caso de Uso Número 1, Ingresar a la Aplicación</i>	69
Tabla 9. <i>Especificación Caso de Uso Número 2, Crear Pedido</i>	70
Tabla 10. <i>Especificación Caso de Uso Número 3, Consultar Catalogo</i>	71
Tabla 11. <i>Especificación Caso de Uso Número 5, Revisar Ruta Vendedores</i>	73
Tabla 12. <i>Historia de usuario: Autenticación correcta</i>	90
Tabla 13. <i>Análisis de Resultados de La Primera Prueba Funcional</i>	92
Tabla 14. <i>Fragmento de código clase MainActivity.java</i>	100
Tabla 15. <i>Primera parte de Código de la Clase Loguin.java</i>	102
Tabla 16. <i>Segunda parte de Código de la Clase Loguin.java</i>	103
Tabla 17. <i>Descripción código del archivo Activity_Menu.xml</i>	105

INTRODUCCION

Actualmente vivimos en la era tecnológica específicamente con la aparición de dispositivos móviles como *Tablets*¹, y especialmente con los *Smartphone*², además la disminución de costos de fabricación ante la gran demanda de estos dispositivos ha permitido que estén al alcance de la mayoría de los usuarios, con ello las aplicaciones móviles también han sufrido un auge en cuanto a sus ofertas hacia sus usuarios dándoles la libertad de tener una gran variedad de aplicaciones de todo ámbito y permitiendo el acceso a la información en cualquier momento y desde cualquier lugar del mundo.

Hoy en día un gran número de compañías, sobre todo las del sector de ventas se ven impulsadas a tener o desarrollar su propia aplicación móvil para sacar ventaja ante la competencia y así llegar a muchos otros mercados y agilizar sus procesos. Los usuarios necesitan obtener información de su empresa en tiempo real y en cualquier parte ya sea para consultas de stock y precios de productos, leer y responder un correo electrónico, o conversar directamente con algún cliente o proveedor y no depender de un computador con acceso a internet para obtener esta información.

Teniendo en cuenta el crecimiento que está teniendo el mercado de aplicaciones móviles, el proyecto se enfoca en desarrollar una aplicación móvil específicamente para entorno *Android*³, sin duda, esto nos obliga, como profesionales del sector tecnológico, a conocer los retos y posibilidades de este entorno, para lo cual se realizará un estudio del mismo así como de las herramientas para su desarrollo para luego mediante la selección de la metodología más adecuada, empezar el desarrollo de la aplicación cumpliendo con cada una de sus fases.

¹ *Tablet*: Tableta (español) es una computadora portátil de mayor tamaño que un teléfono inteligente, integrada en una pantalla táctil con la que se interactúa primariamente con los dedos, sin necesidad de teclado físico ni ratón. **Tomado de:** *Wikipedia*.

² *Smartphone*: Teléfono inteligente es un tipo teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora. **Tomado de:** *Wikipedia*.

³ *Android*: Es un sistema operativo basado en el kernel o núcleo de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas. **Tomado de:** *Wikipedia*.

Capítulo I

Situación Actual del Proyecto

“En este Capítulo vamos a ver los Objetivos del proyecto, su Justificación y el Alcance que tendrá, además se realizará una breve revisión de la Empresa en donde la implementaremos (“Almacenes Juan Eljuri”) y finalmente veremos una síntesis de los procesos actuales en la gestión de pedidos en el área de Perfumería.”

1.1 Tema del Proyecto

“Desarrollo de una aplicación, para dispositivos móviles que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: “*Almacenes Juan Eljuri Cía. Ltda.*” “División Perfumería.”

1.2 Objetivos del Proyecto

1.2.1 Objetivo General

Desarrollar una aplicación para dispositivos móviles plataforma Android que permita automatizar la gestión pedidos y controlar rutas de los agentes vendedores de Almacenes Juan Eljuri Cía. Ltda. , división Perfumería.

1.2.2 Objetivos Específicos

- Contar con una aplicación móvil que permita reducir tiempos de operación hasta su facturación, en las tareas de administración de pedidos.
- Proporcionar información actualizada de stock y precios de los productos a los agentes vendedores.
- Eliminar tareas manuales del personal encargado de las ventas.
- Facilitar al Supervisor de ventas, el control de rutas que cumplen los agentes vendedores diariamente mediante la visualización de los puntos de referencia en un mapa que estará integrado a la intranet de la empresa.

1.3 Justificación del Proyecto

Con la aparición de los teléfonos inteligentes, *Tablets*¹ y el masivo uso que se hace de ellos, además con el auge de las aplicaciones móviles principalmente las de *código abierto*. Se aprovechará que la empresa hará la entrega de *Tablets* (con sistema operativo Android) a sus agentes vendedores del área de Perfumería para que realicen su trabajo.

Ante lo mencionado anteriormente surge la necesidad de realizar una aplicación para dispositivos móviles con sistema operativo Android, para Almacenes Juan Eljuri, específicamente la división de Perfumería que permitirá a sus Agentes Vendedores:

- Administrar los pedidos de una forma más rápida, sencilla y eficiente posible.

Y les permitirá a los Supervisores de Ventas:

- Tener un control de sus Agentes Vendedores, posicionando su ubicación en cualquier punto del país y así poder verificar sus recorridos diarios mediante.

Principalmente el presente proyecto busca automatizar el proceso de las ventas reduciendo los tiempos de operación, sin tener que recurrir a hojas de papel como se lo viene realizando, de esta manera pretende dar una mejor atención al cliente en pro del incremento de las ventas, además permitirá a los empleados del área de perfumería: centrarse en las tareas productivas, eliminando tareas manuales y reducir los tiempos de operación.

1.4 Alcance del Proyecto

En el proyecto se busca automatizar procesos en la gestión de pedidos que se vienen realizando de forma manual para lo cual se desarrolla una aplicación para dispositivos móviles con sistema operativo Android desde la versión 4.0 en adelante, la misma que interactuará con un Servicio Web (WebService⁴) que estará publicada en internet mediante (IIS⁵) el Servicio Web que utilizaremos es el JSON el mismo que permitirá a nuestra aplicación intercambiar datos con otras aplicaciones o servicios como con la base de datos, desde cualquier lugar del mundo gracias al internet. Ver *Ilustración 1*.

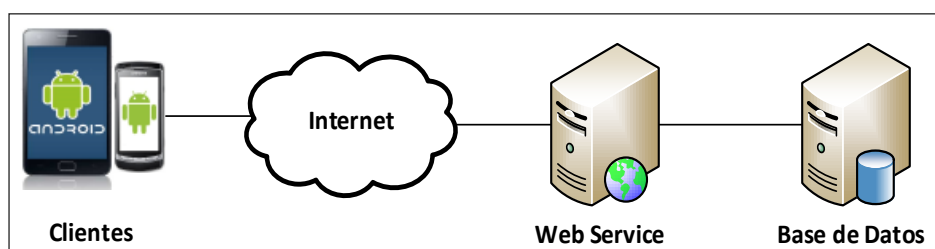


Ilustración 1. Esquema de la aplicación (Elaboración Propia)

⁴ "**WebService**: "Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma." Tomado de *Wikipedia*."

⁵ "**Internet Information Services o IIS7**: "Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows, Este servicio convierte a una PC en un servidor web para Internet o una intranet." Tomado de *Wikipedia*."

Cabe aclarar que el uso de la aplicación se limitará solo para personal de la empresa más concretamente los Agentes Vendedores que les permitirá:

- Ingresar con su respectivo usuario y clave.
- Tener un menú amigable para el vendedor donde pueda: consultar directamente los productos en stock con sus respectivos precios (Catálogo).
- Crear y enviar pedidos al sistema.
- Modificar los pedidos antes de enviarlos.
- Almacenar pedidos en el dispositivo móvil para posterior sincronización.

La segunda parte del proyecto es implementar una Web (agregar a la Intranet) donde el Supervisor de Ventas pueda controlar el recorrido diario de los agentes vendedores, mediante un mapa (Google Maps) donde podamos hacer búsquedas por fechas y por determinado agente vendedor. Esta web será acoplada a la intranet ya existente en la empresa y permitirá al supervisor de ventas:

- Visualizar en un mapa, el recorrido diario de un determinado agente vendedor mediante puntos de referencia en alguna fecha determinada.

Los usuarios serán creados directamente por el administrador de la intranet de la empresa y posteriormente por el administrador de la aplicación.

Con la aplicación desarrollada se ha determinado que su desarrollo se limitará hasta que el pedido se encuentre en el proceso de Facturación, ya que para esto el pedido tiene que pasar por Revisiones por lo que podrán modificar el pedido desde la intranet (Supervisor de Ventas puede agregar Material Publicitario) y lo más importante que será revisado por el Jefe de Cartera para su aprobación (según cartera cliente) y posteriormente pasa por otros procesos de Logística que se encargan del despacho hasta que los productos lleguen a su destino final.

1.5 Empresa

1.5.1 Antecedentes

“Los orígenes de Almacenes “Juan Eljuri Cía. Ltda.” se remontan a la llegada de Gabriel Eljuri, un inmigrante libanés, a principios del siglo anterior (14 de diciembre de 1925). Sus hijos José y Juan Eljuri Chica conformaron una sociedad e instalaron una tienda en Cuenca que se vendía en el almacén llamado “*La Palestina*”, cerca del parque Calderón en la ciudad de Cuenca. En 1940 disuelven la sociedad, y Juan Eljuri se queda con el almacén al que le da su nombre. Dos años después se casa con Olga Antón, quien aporta con su fuerza y carácter para hacer crecer el negocio.

Entre 1943 y 1952 el matrimonio Eljuri Antón, tuvo cinco hijos: Juan, Jorge, Henry, Olga y Gladys, quienes desde pequeños ayudaron en los almacenes y con el tiempo aportaron con grandes ideas. En la actualidad el Grupo Eljuri es fabricante, importador y exportador, de productos de marcas reconocidos a nivel mundial, que van desde chocolates, maquillajes, perfumes, muebles, cerámicas, electrodomésticos, licores, motos, generadores, neumáticos y vehículos entre otros productos” [1]

Actualmente Almacenes Juan Eljuri cuenta con sucursales en las principales ciudades del país, siendo la de la ciudad de Cuenca su matriz principal, ubicada en la Av. Gil Ramírez Dávalos y Armenillas, su división Perfumería se encuentra en el edificio matriz y se caracteriza por la comercialización de productos de las mejores marcas a nivel nacional e internacional de perfumes, cosméticos y artículos de aseo personal; tales como: Carolina Herrera, Hugo Boss, Cartier, Givenchy, REVLON, entre otras marcas.

1.5.2 Misión

Empresa comercial dedicada a satisfacer las necesidades de la sociedad a través de la representación y venta de productos de calidad y renombre mundial en cada una de sus unidades de negocio, mediante una gestión profesional,

honesto y dinámica; contribuyendo al progreso de los clientes, colaboradores, sus familias y accionistas. [2]

1.5.3 Visión

Ser la empresa líder en la comercialización de productos para el hogar, con sólidos recursos tecnológicos, económicos y humanos contando con un personal altamente capacitado en la distribución de productos y servicios de calidad, trabajando con criterio y brindando seguridad a sus socios, empleados y sus familias. [2]

1.6 Procesos Actuales en la Gestión de Pedidos

Para mayor entendimiento de los procesos que se realizan actualmente en la gestión de pedidos en Almacenes Juan Eljuri división Perfumería, se lo detalla mediante el siguiente gráfico: Ver *Ilustración 2*.

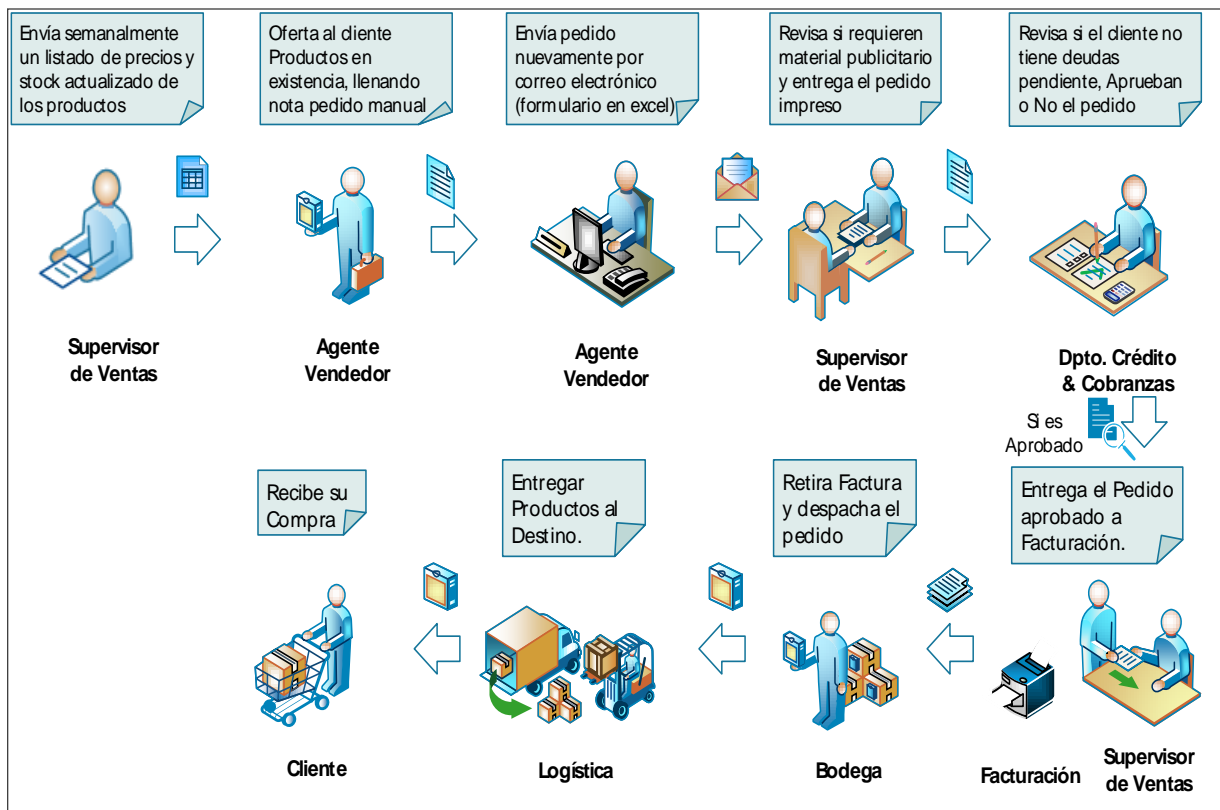


Ilustración 2. Proceso Actual en la gestión de Pedidos Almacenes Juan Eljuri Div. Perfumería. (Elaboración Propia)

Desde que un agente vendedor ya sea de Quito, Guayaquil o el Austro, realiza un pedido, hasta que el mismo pueda ser aprobado por los distintos departamentos, y luego sea facturado, este proceso se demora alrededor de 2, 3 o más días hasta que el cliente reciba su pedido lo que causa que el proceso actual no sea ágil y los pedidos se demoren mucho más de lo que deberían, además el supervisor de ventas tampoco tiene un control en cuanto a las rutas y los clientes que tienen que visitar diariamente cada agente vendedor y no hay forma de saber si los mismos cumplen con lo designado por la empresa. *Tomado de [3]*

Capítulo II

Marco Teórico

“El presente Capítulo nos servirá para poner en claro algunos conceptos teóricos y prácticos, que son indispensables para el desarrollo de la aplicación, así como un estudio de dos metodologías para su correcto desarrollo, de las cuales escogeremos a la más adecuada para nuestro tipo de proyecto.”

2.1 Tecnología Móvil

La tecnología móvil está directamente ligada a la comunicación o telefonía móvil y es a la que nos vamos a referir para nuestro estudio.

“Las comunicaciones móviles sin duda alguna han experimentado un enorme crecimiento desarrollándose diversas tecnologías y sistemas para dar servicios de comunicación inalámbrica. En el Ecuador el servicio móvil celular inicia a finales de 1993 con la entrada en el mercado de CONECCEL S.A. (Porta Celular, luego CLARO) y OTECEL S.A. (al inicio Bellsouth y actualmente denominada Movistar), manteniéndose el dominio de estas 2 empresas hasta el año 2003 cuando entró en operación una tercera operadora TELECSA (al inicio Alegro actualmente CNT E.P.)” *Tomado de [4]*

En el pasado las empresas de telecomunicaciones brindaban un solo servicio: telefonía, audio y video por suscripción, servicios portadores y servicios de valor agregado. En la actualidad los servicios han convergido de tal manera que un mismo proveedor de servicios dentro de una misma infraestructura de telecomunicaciones, puede brindar múltiples servicios. Esta convergencia tecnológica de un dispositivo electrónico que proporciona funcionalidades de telefonía y ofrece asistencias iguales a las de una computadora, hace posible el concepto de Smartphone (teléfonos inteligentes) con los que se puede realizar compras, enviar y recibir correos electrónicos, escuchar música, ver videos, acceder a redes sociales, además de hablar por teléfono, es posible con sólo un dispositivo. La gran aceptación del mercado hacia estos aparatos ha permitido que las empresas líderes en tecnología vean atractiva esta plaza, direccionando su trabajo al desarrollo de aplicaciones personales y empresariales que satisfagan al usuario. [4]

Definitivamente la posibilidad de acceso a internet (internet móvil) es el factor que más ha incidido para que los Smartphone logren tener el nivel de penetración alcanzado en el mercado, la sensación de conectividad que ofrecen las redes sociales así como el acceso a mensajería instantánea hace que los usuarios se vean tentados por estos dispositivos. Por otra parte las numerosas aplicaciones disponibles para teléfonos inteligentes le dan un valor agregado para quienes buscan entretenimiento en dispositivos móviles.

Todas estas aplicaciones se ejecutan dentro de un *ecosistema*⁶ existiendo varios factores que lo afectan como son: la infraestructura de la aplicación, el sistema operativo, los métodos de entrada de información, los propios usuarios, los canales de distribución de la aplicación, etc. En el caso de las aplicaciones móviles, el ecosistema es aún más heterogéneo que en el resto de desarrollos. Pueden ejecutarse en diferentes tipos de dispositivo, ya sea en un móvil antiguo o bien en un actualizado ya sea un Smartphone o una Tablet los cuales se encuentran diseñados bajo una plataforma informática y dependiendo de la marca, en diferentes sistemas operativos, por lo tanto para conseguir un desarrollo satisfactorio es ideal conocerlos más profundamente. [5]

2.1.1 Sistemas Operativos para móviles.

Detallamos algunos conceptos tomados (ver bibliografía) de un sistema operativo móvil:

“Un sistema operativo (SO) móvil controla un dispositivo móvil al igual que los PC utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.” [6]

“El Sistema Operativo (SO) móvil de un teléfono o tableta realiza la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez, lo ejecuta con mayor inteligencia. Una de las cualidades más atractivas de un sistema operativo móvil es la rapidez con la que en general se desempeña.” [7]

⁶ *Ecosistema móvil*: “Se incluyen las operadoras de telecomunicaciones, los fabricantes de hardware y todos los elementos de software que intervienen en la ejecución de la aplicación.” Tomado de [5]

“A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el segundo trimestre de 2014 era el siguiente:

- **Android** “84,7% (En países como España las diferencias son más significativas, donde Android tiene el 90,8% de la cuota de mercado” [8])
- **iOS** 11,7%
- **Windows Phone** 2,5%
- **BlackBerry OS** 0,5%
- *Otros* 0,6%
- **Firefox OS** Disponible
- **Ubuntu Touch** (que está en desarrollo)

Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en solo dos años (2009 a comienzos de 2011) ha pasado a ser el SO móvil más utilizado.” [9]

2.1.2 Android

“Android es una solución completa de software de código libre (*GNU Linux*⁷) para teléfonos y dispositivos móviles. Es un paquete que engloban un sistema operativo, un "Runtime"⁸ de ejecución basado en *Java*⁹, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Android se distribuye bajo una licencia libre que permite la integración con soluciones de código propietario”.



ANDROID

Ilustración 3. Logotipo Android

“Android surge como resultado de la “*Open Handset Alliance*” [10] un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil y un compromiso de comercializar dispositivos móviles con este sistema operativo. El desarrollo viene avalado principalmente por Google (tras la compra de Android Inc. en 2005)” [11]

“Como se había descrito anteriormente, Android está basado en Linux, disponiendo de un *Kernel*¹⁰ en este sistema y utilizando una máquina virtual sobre este *Kernel* que es la responsable de convertir el código escrito en Java de

⁷ **GNU Linux:** “Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL.” *Tomado de Wikipedia*

⁸ **Runtime:** “Colección de funciones de utilidad las cuales soporta un programa mientras se está ejecutando, trabajando con el Sistema Operativo para suministrar facilidades tales como funciones matemáticas, entrada y salida.” *Tomado de Wikipedia*

⁹ **Java:** “Lenguaje de programación concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible, el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.” *Tomado de Wikipedia*

¹⁰ **Kernel o Núcleo:** Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

las aplicaciones a código capaz de comprender el *Kernel*. Las aplicaciones para Android se escriben y desarrollan en Java aunque con unas *APIS*¹¹ propias por lo que programas realizados en Java para *PC* y demás plataformas no son compatibles con Android”.

“La cualidad más grande de este sistema operativo es que es de código abierto y se distribuye bajo dos tipos de licencias, una que abarca todo el código del Kernel y que es *GNU GPL v2*¹², Google también por supuesto tiene otra licencia para el resto de componentes del sistema que se licencia bajo *APACHE v2*¹³” [12]

Vamos a detallar sus principales características:

- Alta calidad de gráficos y sonido: gran variedad de formatos soportados
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual *Dalvik*, máquina virtual diseñada para uso en dispositivos móviles.
- Soporte para hardware adicional (cámaras de vídeo, pantallas táctiles, GPS, acelerómetros, etc.).
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE).
- Plataforma realmente abierta al ser basada en Linux y de código libre. Se puede usar y adecuar el sistema sin pagar algún tipo de licencia.

¹¹ *APIs*: Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas

¹² *GNU GPL v2*: Implica que su código se debe poner al alcance de todos y que todos podremos hacer con este código lo que nos parezca oportuno, modificarlo, ampliarlo, recortarlo, pero siempre estaremos en la obligación de volver a licenciarlo con la misma licencia.

¹³ *APACHE v2*: Implica que este código se pueda distribuir para ser modificado y usado a antojo del que lo utilice, pero a diferencia de *GNU*, las modificaciones y el código resultante no es obligatorio el licenciarlo bajo las mismas condiciones en las que se encontraba. *Todos Tomados de Wikipedia*

- Portabilidad asegurada: Al desarrollar las aplicaciones en Java, y gracias al concepto de máquina virtual, las aplicaciones podrán ser ejecutadas en gran variedad de dispositivos tanto actuales como futuros.
- Las interfaces se hacen en formato *xml*, lo que permite el uso de una misma interfaz en dispositivos de distintos tamaños de pantallas.
- Gran cantidad de servicios incorporados: Reconocimiento y síntesis de voz, localización basada en GPS, potentes bases de datos, etc.
- Nivel de seguridad: Los programas se encuentran separados unos de otros. Cada aplicación dispone distintos tipos de permisos que limitan su rango de actuación.

2.1.2.1 Arquitectura de Android

Para empezar con el desarrollo de aplicaciones en Android es importante conocer cómo está estructurado este sistema operativo. La arquitectura de Android está formado por varios niveles o capas lo que facilita el desarrollo de aplicaciones ya que permite trabajar con las capas inferiores por medio de las librerías evitando programar a bajo nivel y lograr que los componentes de hardware del dispositivo móvil interactúen con la aplicación.

“Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como pila.” [13] Para mejor entendimiento ver *Ilustración 4*.



Ilustración 4. Arquitectura de Android [11]

Se explicará las capas más relevantes de abajo hacia arriba.

Kernel de Linux. “El núcleo actúa entre el hardware y el resto de las capas de la arquitectura. El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores. Para cada elemento de hardware existe un controlador o driver dentro del kernel que permite utilizarlo desde el software.”

Librerías. “Estas normalmente están hechas por el fabricante, quien también se encarga de instalarlas en el dispositivo antes de ponerlo a la venta. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez.”

Entorno de ejecución. “Como podemos apreciar (Ver *Ilustración 4*), el entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí encontramos las librerías con las funcionalidades habituales de Java así como otras específicas de Android.

El componente principal del entorno de ejecución de Android es la máquina virtual *Dalvik*”

Framework de Aplicaciones. “Es una estructura de aplicaciones, formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones.

Siguiendo el diagrama encontramos:

- *Activity Manager.* Se encarga de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.
- *Windows Manager.* Se encarga de organizar lo que se mostrará en pantalla.
- *Content Provider.* Esta librería crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
- *Views.* En Android, las vistas los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de *Google Maps*¹⁴.
- *Package Manager.* Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes.

¹⁴ **Google Maps:** Es un servidor de aplicaciones de mapas en la web que pertenece a *Google*. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones.

- *Location Manager*. Permite determinar la posición geográfica del dispositivo Android mediante *GPS*¹⁵ o redes disponibles y trabajar con mapas.
- *Sensor Manager*. Nos permite manipular los elementos de hardware del dispositivo móvil como el acelerómetro, giroscopio, brújula, etc.
- *Cámara*: Con esta librería podemos hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.
- *Multimedia*. Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.”

Aplicaciones. En la última capa se incluyen todas las aplicaciones del dispositivo ya sea con interfaz gráfica o no, las propias del dispositivo y las administradas (programadas en Java), así como las que el usuario ha instalado por su cuenta.

Android nos proporciona un entorno robusto para que podamos programar aplicaciones para alguna funcionalidad. Todo dentro de Android es accesible y podemos contar siempre con las aplicaciones de nuestro dispositivo móvil para la optimización de cualquier tarea de programación. *Tomado de [13]*

2.1.2.2 Estructura o componentes de una aplicación Android

“Las aplicaciones en Android están basadas en componentes y son los elementos básicos con los que se construyen un proyecto. Una aplicación Android será una combinación de uno o más de estos componentes y se deben declarar en el fichero *AndroidManifest.xml*¹⁶ donde se definen todos los

¹⁵ **GPS:** Sistema Posicionamiento Global (en español) está constituido por 24 satélites y utiliza la triangulación para determinar en todo el globo la posición con una precisión de más o menos metros. *Tomado de Wikipedia.*

¹⁶ **AndroidManifest.xml:** Este fichero es un documento xml en el que se declaran los elementos de la aplicación, así como sus restricciones, permisos, procesos, acceso a datos e interacciones con elementos de otras aplicaciones. *Tomado de [13]*

componentes de la aplicación así como los permisos que requiere, o los recursos y librerías que utiliza” [13]

Activity: Son el componente visual de una aplicación, es decir forman la interfaz y se podría decir que son las pantallas o ventanas que ve el usuario.

Service: Son componentes sin interfaz gráfica, se ejecutan en segundo plano y realizan múltiples acciones.

Content Provider: Es la forma que tiene Android de compartir datos entre aplicaciones, compartiendo datos sin la necesidad de dar de detalles sobre su almacenamiento.

Broadcast Reciver: Este componente se encarga de detectar y reaccionar a los eventos generales como pueden ser: batería baja, entrada de llamada, es decir es el encargado de recibir las llamadas que son enviados a cualquier aplicación que esté “escuchando”.

2.1.2.3 SDK de Android

Como se pudimos ver anteriormente, Android es una plataforma de software libre, por lo que cuenta con Kit de desarrollo de Software o *SDK*¹⁷ disponible para todo desarrollador o para alguien que desee experimentar que incluye: el conjunto completo del *API* soportado por este sistema para crear, probar y depurar aplicaciones para Android. Podemos descargarlo de la página web de Android: <http://developer.android.com/intl/es/sdk/index.html> [14] Ver Ilustracion.5.

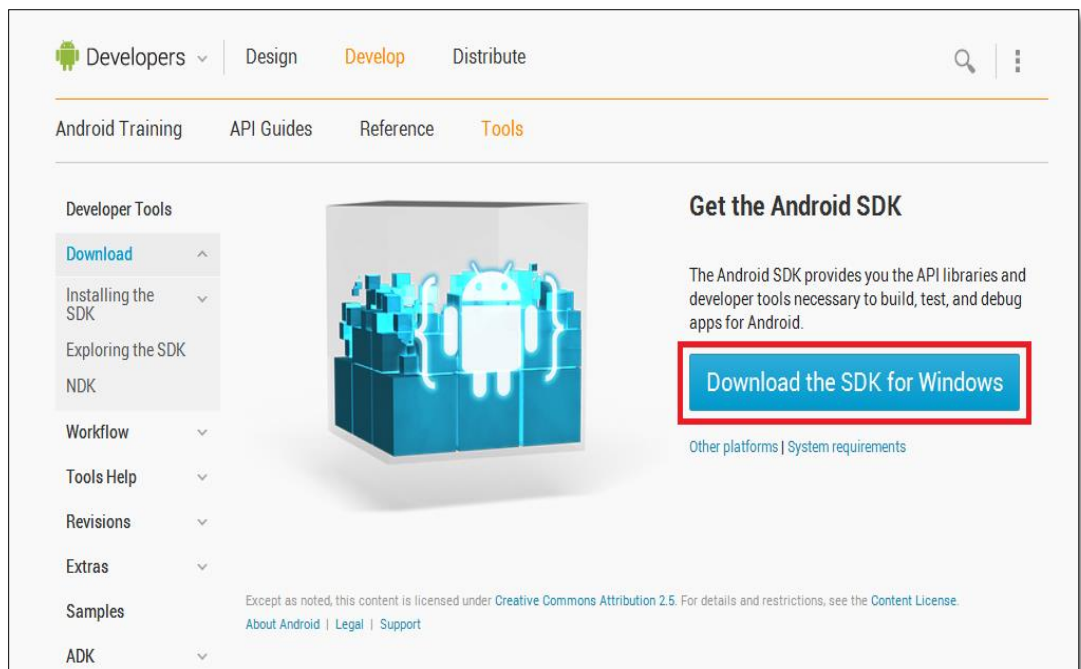


Ilustración 5. Página de descarga de SDK Android

Una vez descargado el *SDK*, es necesario ejecutarlo o descomprimirlo según la opción descargada, la instalación y configuración la veremos en el Siguiete Capítulo.

2.1.2.4 Google Maps en Android

¹⁷ **SDK** (Software Development Kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto. *Tomado de Wikipedia.*

“Google Maps nos proporciona un servicio de cartografía online que podremos utilizar en nuestras aplicaciones Android.

A diferencia de Android, *Google Maps* no es un software libre, por lo que está limitado a una serie de condiciones de servicio. Podemos usarlo de forma gratuita siempre que nuestra aplicación no solicite más de 15.000 codificaciones geográficas al día. Podemos incluir propaganda en los mapas.”
[15] Los pasos para la inclusión de mapas en nuestra aplicación Android son:

- **Descargar y configurar los *Google Play Services*:** Viene con el SDK de Android por lo que la descargamos con la ayuda del SDK Manager. Ver *Ilustración 6*.
- **Obtener una API key:** Es una llave que estará asociada exclusivamente a nuestra aplicación por medio de los certificados que usa Android
- **Definir las especificaciones en el *Application Manifest*:** Que se lo hace colocando una línea de código en el archivo *AndroidManifest.xml*
- **Añadir un mapa a la aplicación:** Se lo realiza añadiendo más líneas de código en los archivos *.xml* como son: el *main* y el *mainActivity*.

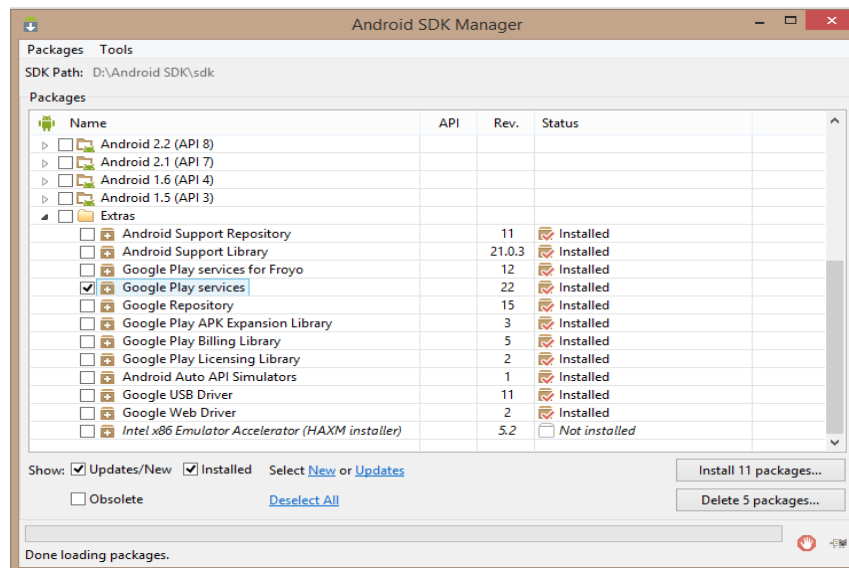


Ilustración 6. Instalación Google Play Services.

2.1.2.5 SQLITE

“*SQLite* es un sistema gestor de base de datos relacional (*RDBMS*¹⁸). Lo que hace único a *SQLite* es que se considera una solución embebida. La mayoría de los sistemas de gestión de bases de datos como *Oracle*, *MySQL*, y *SQL Server* son procesos de servidor autónomos que se ejecutan independientemente.” *SQLite* es en realidad una librería que está enlazada dentro de las aplicaciones. Todas las operaciones de base de datos se manejan dentro de la aplicación mediante llamadas y funciones contenidas en la librería *SQLite*, hace mucho más fácil tratar grandes bases de datos en comparación con otras soluciones más convencionales de base de datos.” [16]

SQLite se basa en el Lenguaje Estructurado de Consultas (*SQL*¹⁹); el mismo lenguaje que utilizan la mayoría de *RDBMS*. Es una gran ventaja si se conoce *SQL* ya que la mayoría de las sentencias se formulan de la misma forma. Android tiene integrado en el propio sistema una *API* completa que nos permite manejar la Base de Datos en *SQLite*. [16]

2.1.2.6 DALVIK

“Dalvik es el nombre de la máquina virtual que utiliza (DalvikVM), la cual está basada en registro, diseñada y escrita por "Dan Bornstein" y algunos otros profesionales de Google. En ella podemos encontrar una gran diferencia con respecto a la máquina virtual Java (*JVM*). *Dalvik VM* es un intérprete que sólo ejecuta archivos con formato *.dex* (*Dalvik executable*). Este formato está optimizado para el almacenamiento eficiente de la memoria, lo cual consigue delegando en el kernel la gestión de hilos (*multithreading*), de memoria y de procesos.” Tomado de [15]

2.1.2.7 Breve descripción de las Versiones de Android

¹⁸ **RDBMS**: Un sistema de Gestión de Bases de Datos Relacionales es aquel que sigue el modelo relacional. Tomado de Wikipedia

¹⁹ **SQL** (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Tomado de Wikipedia

"El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial de prueba, luego fue lanzada Android 1.0 el siguiente año, y ha experimentado en numerosas ocasiones, actualizaciones a su sistema operativo desde su lanzamiento original. Estas actualizaciones corrigen fallos de programa y agregan nuevas funcionalidades. Desde abril de 2009, Las lanzamiento se realiza en orden alfabético (A, B, C, D, etc.)." [17]. A continuación veremos un breve resumen de las versiones de Android: *Tabla 1:*













Versión	Nombre de la Versión	Fecha de lanzamiento de la distribución	Nivel API²⁰	Logotipo
1.0	Apple Pie	22 de octubre de 2008	1	
1.1	Banana Bread	9 de Febrero de 2009	2	
1.5	CupCake	30 de abril de 2009	3	
1.6	Donut	15 de septiembre de 2009	4	
2.0	Eclair	26 de octubre de 2009	5	
2.2	Froyo	20 de mayo de 2010	8	
2.3	Gingerbread	9 de febrero de 2011	10	
3.0	Honeycomb	22 de febrero de 2011	11	
4.0	Ice Cream Sandwich	16 de diciembre de 2011	15	
4.1	Jelly Bean	9 de julio de 2012	16	
4.4	Kit Kat	31 de octubre de 2013	19	
5.0	Lollipop	3 de noviembre de 2014	21	

Tabla 1. Descripción de las Versiones de Android

²⁰ **Nivel API:** "Es un valor entero que identifica de forma unívoca el framework ofrecido por la versión de una plataforma Android." Tomado de Wikipedia

2.2 Algunas Metodologías existentes para desarrollo de aplicaciones móviles

Las metodologías de desarrollo de software se refieren a una estructura de soporte definida mediante el cual un proyecto de software puede ser organizado y desarrollado y están orientadas a estructurar, planear y controlar el proceso de desarrollo de un sistema de información.

Cabe destacar que todas las metodologías tienen puntos a favor como en contra, por lo que es muy importante conocerlas y ver cuál es la que brinda mejor soporte al desarrollo de una aplicación móvil, por lo que vamos a analizar dos de ellas para luego usar la más adecuada para nuestro proyecto.

2.2.1 Metodología RUP

“El Proceso Racional Unificado (*Rational Unified Process* en inglés, RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de *IBM*. Junto con el Lenguaje Unificado de Modelado *UML*, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos, ya que se utilizan en diferentes tipos de software, áreas de aplicación, niveles de competencia, tamaños de proyecto y se adaptan fácilmente a las necesidades de cada organización.” Tomado de [20]

2.2.1.1 Estructura del RUP

RUP es una metodología en la que se describe quién, cómo, qué, en qué tiempo y qué actividades se van a desarrollar en el proyecto, el quién representa los distintos roles que puede desempeñar un individuo en la organización, el cómo se refiere a la unidad de trabajo que se asigna a un trabajador y el qué es la segmento de información utilizada por un proceso. Para el desarrollo satisfactorio de este proceso, es necesario cumplir con ciertas etapas, las mismas que en conjunto determinarán el ciclo de vida y el éxito de la

aplicación. RUP hace uso de cuatro etapas en su metodología descritas de la siguiente manera:

1. Inicio.- En esta fase se obtiene una visión inicial del producto así como su alcance, además se identifican los principales casos de uso.

2. Elaboración.- En esta fase se hace una planificación de las actividades y del equipo de trabajo del proyecto, también se identifican las necesidades y el diseño de la arquitectura.

3. Construcción.- Comprende el desarrollo mismo del producto hasta la entrega al usuario final.

4. Transición.- Esta fase comprende la instalación del producto a los usuarios y la formación de los mismos, en ocasiones suelen surgir nuevos requisitos para el desarrollo.

2.2.1.2 Ciclo de Vida

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semiordenadas. El ciclo de vida organiza las tareas en fases e iteraciones. RUP divide el proceso en cuatro fases:

- Fase de Inicio,
- Fase de Elaboración,
- Fase de Construcción y
- Fase de Transición.

“Dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor empeño en las distintas actividades. Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ambiente del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requisitos,

modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura. En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.” Tomado de [20]

2.2.1.3 Disciplinas de un proceso RUP

“Como se observa en la *Ilustración 10*, claramente podemos distinguir dos tipos de disciplinas, las de proceso y las de soporte. Una disciplina es un conjunto de actividades vinculadas a un área específica del proyecto y *RUP* las utiliza para facilitar la comprensión de todo proyecto de desarrollo. Un flujo de trabajo describe la secuencia en que se realizan las actividades en una disciplina, quienes la realizan y que artefactos se producen.” Tomado de [20]

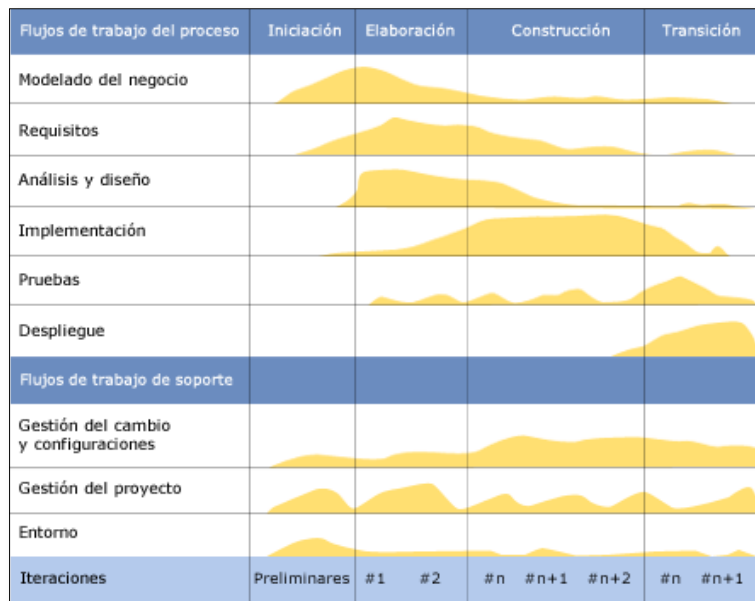


Ilustración 7: Esfuerzo en actividades del RUP (Tomado de [20])

Las disciplinas de proceso son las siguientes:

- **Modelado del Negocio:** Describe la estructura y el modelo de negocio de la organización.
- **Requisitos:** Refleja las necesidades del cliente expresado en casos de uso.

- **Análisis y Diseño:** Describe las diferentes vistas arquitectónicas del proyecto.
 - **Implementación:** Comprende el desarrollo del software, prueba de unidades e integración.
 - **Pruebas:** Son las diferentes métricas de evaluación del proyecto.
 - **Despliegue:** Configuración del sistema que se va a entregar.
- Mientras que las disciplinas de soporte son:
- **Gestión de Configuraciones:** Ayuda en el control de cambios y la integridad de los artefactos del proyecto.
 - **Gestión del Proyecto:** Son las diferentes estrategias de trabajo en un proceso iterativo.
 - **Entorno:** Es la infraestructura necesaria para desarrollar un sistema.

2.2.2 Metodología XP (Extreme Programming)

“La programación extrema o *eXtreme Programming (XP)* es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la *XP* consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.”

“Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. Esta metodología busca eliminar actividades relacionadas con la elaboración de algunos documentos de especificaciones que no tienen relación directa con el resultado final del software. Se basa en los siguientes valores principales:” *Tomado de [21]*

1. “Comunicación: La necesidad de los desarrolladores de intercambiar ideas e información sobre el proyecto ya sea con los directores del proyecto o los usuarios de forma confiable, y fácil. La comunicación debe ser continua y rápida.

2. Sencillez: Cuando se tenga que elegir entre varias opciones, en lo posible elegir soluciones simples, sin que esto signifique aplicar enfoques simplistas; la programación extrema define un diseño simple en el que se realice el menor número posible de clases, métodos y que no tenga código duplicado.

3. Retroalimentación: Debe ser rápida en todos los niveles, principalmente se consigue ejecutando y probando el código, por lo que las entregas tempranas y frecuentes son muy importantes.

4. Valor: Todas las personas que participen en el proyecto deben tener la capacidad de expresar su valoración sobre el proyecto. Deberían ser abiertos y dejar que todos revisen e incluso modificasen su trabajo. Los cambios no deberían ser vistos con terror y los desarrolladores deberían tener el valor de encontrar mejores soluciones y modificar el código siempre que sea necesario y factible.

5. Respeto: Debe manifestarse en diversas formas y situaciones, son la base para una buena relación y cooperación entre todos los componentes del equipo de trabajo.” *Tomado de [21]*

La Programación Extrema se basa en:

- *Desarrollo en iteraciones:* En cada iteración se agregan nuevas funcionalidades, o se corrigen errores generando distintas versiones.
- *Pruebas unitarias continuas:* Estas pruebas están orientadas a comprobar que la aplicación mantenga sus funcionalidades.
- *Programación en parejas:* Se hace esto con el fin de que el código se discuta y revise mientras se desarrolla el programa, basado en que los dos programadores pueden complementarse, generando código de mejor calidad con menos errores.

- *Interacción entre los desarrolladores y el usuario:* Se minimizara el esfuerzo de ambas partes, pues se podrá tener una mejor comprensión de los problemas o necesidades de los usuarios y las soluciones que puedan brindar los desarrolladores.
- *Refactorización del código:* Busca hacer el código más fácil y mantenible, pero debe garantizar su correcto funcionamiento manteniendo las pruebas unitarias.
- *Propiedad del código compartida:* Busca que cualquier integrante del proyecto pueda colaborar modificando código hecho por otro. La existencia de errores se comprueba mediante las pruebas.
- *Simplicidad del diseño:* Los diseños simples pero funcionales permiten que posteriores funcionalidades se puedan agregar de manera fácil y rápida.

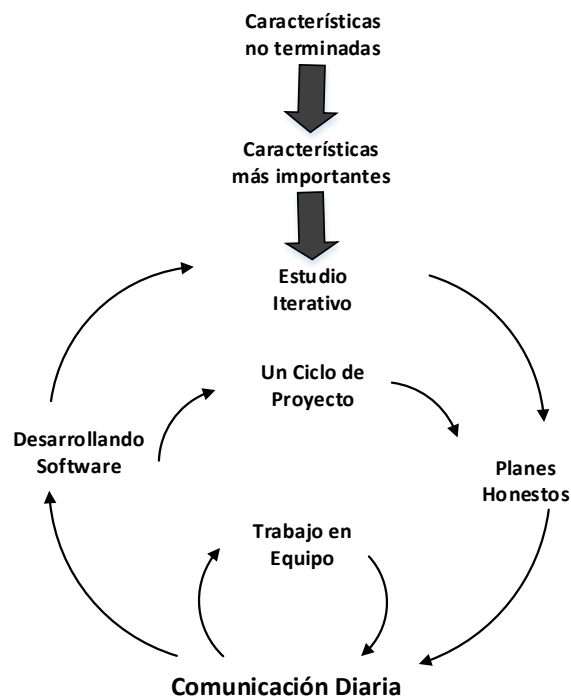


Ilustración 8: Flujo de características de la Programación Extrema. (Fuente [21])

2.2.2.1 Proceso XP

“Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste en los siguientes pasos:” Tomado de [22]

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

2.2.2.2 Fases de desarrollo de la Programación Extrema

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. [22]

“Fase I: Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II Planificación de la Entrega: En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

Fase III Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Fase IV: Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

Fase V: Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo”. Tomado de [22]

2.3 Análisis comparativo

El análisis a realizarse es una comparación de sus principales características, además de sus ventajas y desventajas de algunos puntos fundamentales tanto de la Metodología RUP como de la Extreme Programming, además de cuando usarlas. Este cuadro comparativo nos ayudará a visualizar de una mejor manera cual es la metodología que se adapta a nuestro proyecto de desarrollo. Ver **Tabla 2**.

De acuerdo al cuadro comparativo podemos ver que la *Metodología XP* es la que adecua a nuestro tipo de proyecto ya que provee al programador un sinnúmero de ventajas que detallaremos en el siguiente punto.

Met.	Características	Ventajas	Desventajas	Cuando Usar
R.U.P.	Desarrollo iterativo en etapas	Utiliza las mejores prácticas de desarrollo para lograr la eficiencia en la producción de software.	No se toma en cuenta a respecto a la gestión y acuerdo de suministros	Cuando existe una comunicación entre los equipos de desarrollo del proyecto
	Programación por equipos	Permite definir de manera clara: Quien de hacer las cosas, Que debe hacerse, Cuando y Como.	La gestión y la evaluación de la calidad no se realizan de forma muy detallada en la fase de transición.	Cuando existen proyectos de desarrollo complejos
EXTREME PROGRAMMING	Desarrollo incremental y en Iteraciones.	Se adapta muy bien a los requisitos cambiantes.	Dificultad para llegar a saber el costo y el tiempo que tomará el proyecto	Cuando los requerimiento cambian constantemente (clientes indecisos)
	Programación por parejas	Disminuye la tasa de errores		Cuando se tiene un equipo de desarrollo motivado y responsable
	El Usuario es parte del equipo de desarrollo	Permite superar con mayor facilidad circunstancias imprevistas en el desarrollo de software.		
	Simplicidad del Código	Garantizar que el software funcione como el usuario lo requiere	Generalmente se aplica solo para proyectos pequeños, no muy eficiente en proyectos de gran tamaño.	Cuando se cuenta con usuarios que entienden y participan activamente en el proyecto
	Entregas rápidas y frecuentes	Permite una mejor y más eficiente comunicación entre todos los integrantes del proyecto		
	Pruebas unitarias Continuas	Calidad de software en el menor tiempo		

Tabla 2. Análisis comparativo entre las Metodologías RUP y XP (Elaboración Propia)

2.4 Descripción de la metodología seleccionada.

Cabe mencionar que no existe una metodología universal para aplicarla con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada a todo el contexto del proyecto como son: recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.

Sin embargo la Programación Extrema proporciona una serie de ventajas para la realización del proyecto ya que ayudan al ahorro de tiempo y recursos.

- Permite la entrega de productos usables con mayor rapidez lo que permite la entrega de versiones continuas al usuario.
- Presenta un código más simple y fácil de entender que permite la reducción del número de errores en su desarrollo.
- Capacidad de respuesta ante cambios permitiendo la adaptación a nuevos requisitos de la empresa u organización.

Por estas y muchas otras ventajas es la Metodología que más se adecua a nuestro proyecto de desarrollo ya que pertenece a las “Metodologías Ágiles de Desarrollo” y como su nombre lo indica nos agiliza su desarrollo.

Capítulo III

Determinación, Instalación y Configuración de las Herramientas de Desarrollo

“En este Capítulo se analizará la plataforma de desarrollo, seleccionando el entorno de programación para finalmente indicar la instalación y configuración de cada una de las herramientas necesarias para el desarrollo de una aplicación y detallando los pasos a seguir.”

3.1 Plataformas de desarrollo

Para el desarrollo de aplicaciones para Android hay varias opciones disponibles ya sea Herramientas de Desarrollo Oficiales y Herramientas de Terceros, pero generalmente se hace con el lenguaje de programación similar a *Java* y el conjunto de herramientas de desarrollo *SDK* (Software Development Kit) que es una herramienta oficial, y cual la utilizaremos en nuestro proyecto para la parte de la aplicación.

3.1.1 Android SDK

“El *SDK* (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en *QEMU*²¹, documentación, ejemplos de código y tutoriales. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento *ADT* (Android Development Tools plugin). El *IDE IntelliJ IDEA*²² soporta completamente del desarrollo en Android en origen, y el IDE *NetBeans*²³ también lo hace utilizando un plugin (complemento, extensión o programa adicional).” Tomado de [23]

“Las Actualizaciones del *SDK* están coordinadas con el desarrollo general de Android. El *SDK* soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.” Tomado de [23]

²¹ **QEMU:** Es un emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped). Tomado de Wikipedia.

²² **IntelliJ IDEA:** Editor de código fuente diseñado específicamente para desarrollar para Android. Tomado de Wikipedia.

²³ **NetBeans:** Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso.

3.2 Arquitectura de la aplicación

El proyecto en su totalidad a desarrollar se divide en dos partes. La primera parte es la aplicación para Android y para su funcionamiento necesitamos de un dispositivo móvil Android, con la aplicación instalada y con acceso a internet por lo que se podrá consultar desde cualquier parte del mundo, esta se conecta al Servicio Web (WebService específicamente JSON) que estará publicada en internet mediante (IIS7), que a su vez se comunica con la base de datos de la empresa (DB2) la que almacena la información de los pedidos e intercambian información, para mayor detalle, *Ver Ilustración12.*

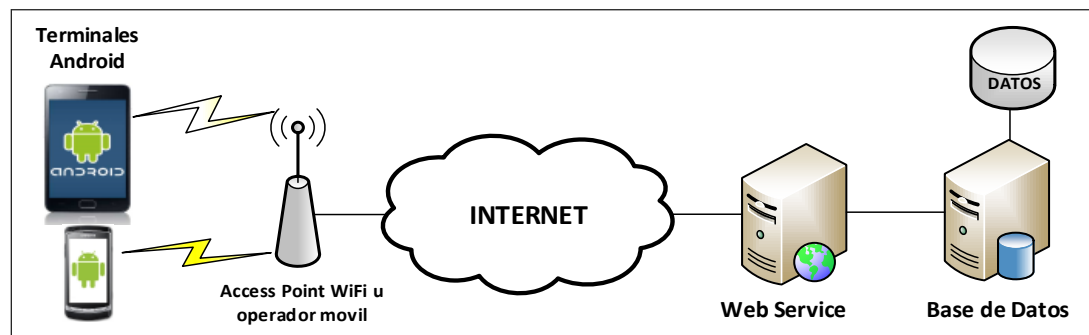


Ilustración 9: Arquitectura de la Aplicación (Elaboración Propia)

La segunda parte del proyecto consta de un módulo Web agregada a la intranet existente de la empresa, podemos mencionar que la intranet de la empresa está publicada en un servidor Web Apache que mediante peticiones *PHP*, interactúa con las peticiones del usuario (Paginas *HTML*) ya que esta publicado en Internet, el usuario usa un navegador (Internet Explorer) para consultar información ya sea localmente por la red interna o por su página web. *Ver Ilustración 13.*

La intranet permite que otros usuarios (Supervisor y Jefe de Cartera) puedan revisar y aprobar o desaprobar los pedidos, además de controlar el recorrido de los agentes vendedores. La Web o módulo desarrollado interactúa directamente con la Base de Datos de la empresa por lo tanto afectará directamente a los datos que son consultados por la aplicación.

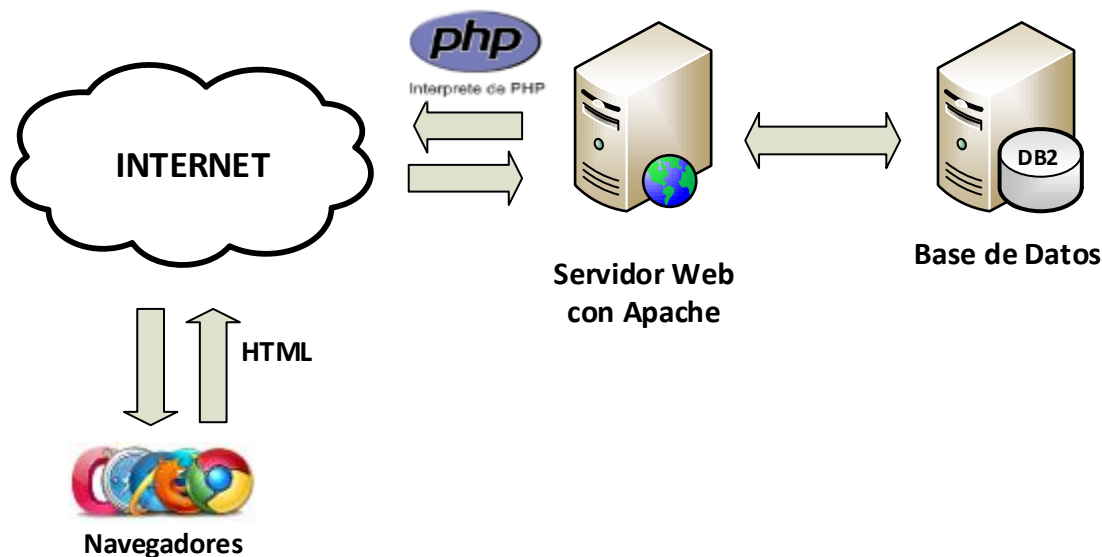


Ilustración 10: Arquitectura de la Intranet (Elaboración Propia)

3.2.1 Web Service (Servicio Web)

Un servicio web (Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones *OASIS*²⁴ y *W3C*²⁵ son los comités responsables de la arquitectura y reglamentación de los servicios Web. Es una máquina que atiende las peticiones de los clientes web y les envía los recursos solicitados. [24]

²⁴ **OASIS:** (Organization for the Advancement of Structured Information Standards) es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web

²⁵ **W3C:** Proviene de World Wide Web Consortium es un consorcio internacional que produce recomendaciones para la World Wide Web. Tomado de Wikipedia,

3.2.1.1 Ventajas de los Servicios Web:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web promueven los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y sistemas o programas de diferentes compañías ubicadas en diferentes lugares del mundo puedan ser combinados fácilmente para proveer servicios integrados.

Ahora entre más datos se estén intercambiando entre las aplicaciones, los mensajes crecerán más y el tiempo de entrega se irá incrementando. Esta situación ha hecho que haya surgido un nuevo formato para el intercambio de datos con el fin de hacer más ligeros los mensajes de intercambio. Este nuevo formato es JSON.

3.2.1.2 JSON (JavaScript Object Notation)

“JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) 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*²⁶. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML. También es frecuente el uso de JSON y XML en la misma aplicación.” Tomado de [25]. Así es que para su configuración se debe descargar 2 librerías que nos ayudará a simplificar el código necesario para consumir un web service *SOAP*²⁷ con formato JSON:

²⁶ **XML:** (lenguaje de marcas extensible), es un lenguaje de marcas desarrollado por el World Wide Web Consortium utilizado para almacenar datos en forma legible, permite definir la gramática de lenguajes específicos. Tomado de Wikipedia

²⁷ **SOAP:** Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML

- “**Gson** es una biblioteca Java que se puede utilizar para convertir objetos Java en su representación JSON. También se puede utilizar para convertir una cadena JSON a un equivalente de objetos Java.
- **ksoap2-android**: es una biblioteca ligera y eficiente para web services basados en el protocolo *SOAP*.” Tomado de [25]

3.2.2 Internet Information Services (IIS)

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP. Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de *ASP* y *ASP.NET*²⁸. También pueden ser incluidos los de otros fabricantes, como *PHP*²⁹. [26]

3.3 Selección del entorno de desarrollo

Para el desarrollo de la aplicación que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: “Almacenes Juan Eljuri Cía. Ltda.” División Perfumería.”, se usaran herramientas oficiales para el desarrollo de aplicaciones Android, y el entorno de desarrollo integrado, llamado también *IDE* que usaremos en nuestro proyecto será Eclipse que se detalla en el siguiente punto.

²⁸ **ASP.NET**: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML

²⁹ **PHP**: Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. *Tomados de Wikipedia.*

En la siguiente tabla se resume cada una de las herramientas utilizadas en el proyecto así como el enlace de descarga. Ver **Tabla 3**.

Aplicación Android		
Herramienta	Descripción	Enlace Descarga o Instalación
Eclipse Juno	Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar	http://www.eclipse.org/downloads/
Java Platform (JDK) 8	Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java.	http://www.oracle.com/technetwork/es/java/javase/downloads/index.htm
ADT(Android Development Tools) Plugin	(ADT) es un plugin para el IDE de Eclipse que está diseñado para darle un ambiente de gran alcance.	Se lo hace desde Eclipse: agregando repositorio: https://dl-ssl.google.com/android/eclipse/
Android SDK(Software Development Kit)	En el SDK podemos encontrar todas las librerías necesarias para poder trabajar con Android. Además de un Emulador.	http://developer.android.com/intl/es/sdk/index.html
Página Web integrada a la Intranet		
Visual Studio .NET	Crear aplicaciones, sitios y aplicaciones web como ASP.net, como servicios web en cualquier entorno que soporte la plataforma .NET	https://www.visualstudio.com/es-es/downloads (Versión Prueba 30 Días)
API Google Maps para .NET	El API de Google Maps es un servicio gratuito que te permite insertar Google Maps en tus aplicaciones o en tus páginas web	Se necesita tener una clave para usar el API de Google Maps se puede obtener: https://developers.google.com/console/help/?hl=es#generatingdevkeys
IIS (Internet Information Server 7)	Servidor web, conjunto de servicios para el S.O. M.Windows, convierte a una PC en un servidor web para Internet.	Viene integrado en M. Windows, Hay activar características y escoger Internet Information Services en Panel de Control

Tabla 3. Herramientas utilizadas para el desarrollo de la aplicación.

3.3.1 Eclipse

“Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "*Aplicaciones de Cliente Enriquecido*"³⁰, opuesto a las aplicaciones "*Cliente-liviano*"³¹ basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse.” Tomado de [27]

Ventajas de Eclipse

- Tiene una interfaz gráfica muy personalizable y avanzada para el desarrollo de aplicaciones ya que dispone de *Frameworks (entornos)* muy completos
- “El *SDK* de Eclipse incluye las herramientas de desarrollo de Java, con un *IDE* con un compilador de *Java* interno.”
- Permite escribir cualquier extensión deseada en el ambiente, es decir permiten a terceros crear complementos que interactúan con la aplicación principal y con la posibilidad de añadir control de versiones.
- Realiza la compilación del código en tiempo real pudiendo así mostrar los errores en el código instantáneamente y siendo más fácil para el programador poder corregirlos.

³⁰ **RCP:** “Son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.”

³¹ **Cliente Liviano:** “Es una computadora cliente o un software de cliente en una arquitectura de red cliente-servidor que depende primariamente del servidor central para las tareas de procesamiento.” Tomados de Wikipedia.

3.3.2 Arquitectura de Eclipse

“La base para Eclipse es la Plataforma de cliente enriquecido (Rich Client Platform RCP). Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes

Los widgets de Eclipse están implementados por una herramienta de widget para Java llamada Standard Widget Toolkit, a diferencia de la mayoría de las aplicaciones Java, que usan las opciones estándar *Abstract Window Toolkit* (AWT) o Swing. La interfaz de usuario de Eclipse también tiene una capa *GUI*³² intermedia llamada JFace, la cual simplifica la construcción de aplicaciones basadas en *SWT*³³.

El entorno de desarrollo integrado (*IDE*) de Eclipse emplea módulos (en inglés plugin) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software.”
Tomado de [27]

³² **GUI:** Interfaz gráfica de usuario es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

³³ **SWT:** (Standard Widget Toolkit) es un conjunto de componentes para construir interfaces gráficas en Java, desarrollados por el proyecto Eclipse.

3.3.3 Plugin ADT para Eclipse

Eclipse es un entorno de software multilinguaje de IDE y un sistema de *plugins*³⁴ extensibles. Está escrito principalmente en Java y se puede utilizar para desarrollar aplicaciones en Java y, por medio de diversos plugins, otros lenguajes de programación.

El *ADT* de Android es un plugin para el *IDE* de Eclipse que está diseñado para darle un ambiente de gran alcance e integrado en la construcción de aplicaciones Android. *ADT* amplía las capacidades de Eclipse para que pueda configurar rápidamente nuevos proyectos de Android, crear una interfaz de usuario de la aplicación, añadir componentes basados en el Android *Framework API*, depurar sus aplicaciones utilizando las herramientas de *SDK* de Android, e incluso exportar un *.Apk*³⁵ firmado (o sin firmar) con el fin de distribuir una aplicación.

El desarrollo en Eclipse con *ADT* es altamente recomendable y es la manera más rápida para empezar. Con la configuración guiada de proyecto que ofrece, así como la integración de herramientas, editores de *XML* personalizados, y el panel de salida de la señal de depuración, *ADT* le da un impulso increíble en el desarrollo de aplicaciones de Android.

3.4 Especificaciones e instalación de la tecnología de desarrollo.

³⁴ **Plugin:** Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. *Tomados de Wikipedia*

³⁵ **Apk:** (Application Package File) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android. Tomado de Wikipedia.

Para desarrollar una aplicación Android, lo primero que necesitamos es definir bien un entorno de desarrollo. Por lo que vamos a explicar detalladamente los pasos necesarios para construir nuestro entorno de desarrollo de aplicaciones Android.

El entorno se puede implementar tanto en Windows como en Linux. Los requisitos para cada sistema operativo se pueden consultar en la web de Android, aunque el proceso es prácticamente idéntico en otros sistemas operativos. El proceso consta de los siguientes pasos:

- Instalar y configurar *JDK*: Kit de desarrollo de Java.
- Instalar y configurar el *SDK* de Android: Kit de desarrollo de aplicaciones Android.
- Instalar y configurar Eclipse: Herramientas para desarrollar software.
- Instalar el plugin *ADT* para Eclipse: para poder desarrollar aplicaciones Android utilizando Eclipse.
- Agregar APIs y componentes al SDK: descarga del resto de componentes necesarios para desarrollar aplicaciones.
- Crear y Configurar un Dispositivo virtual Android o AVD (Android Virtual Device) para poder ejecutar la aplicación.

3.4.1 Instalación y configuración de JDK

Las librerías del Java JDK (Java Development Kit) son indispensables para ejecutar el emulador de Android y algunas herramientas de depuración. Para instalar el entorno de desarrollo JDK 8 en Windows, debemos seguir los siguientes pasos:

1. Descargar el JDK del sitio web de Oracle

<http://www.oracle.com/technetwork/es/java/javase/downloads/index.htm>. En esta página encontramos las diversas versiones, en nuestro caso utilizaremos la última versión que es la 8u31, para Windows 64bits. Ver *Ilustración 14*.

- Una vez descargado el JDK procedemos a instalarlo en nuestro computador, Escogemos las características a instalar y seleccionamos siguiente. Ver *Ilustración 15*.

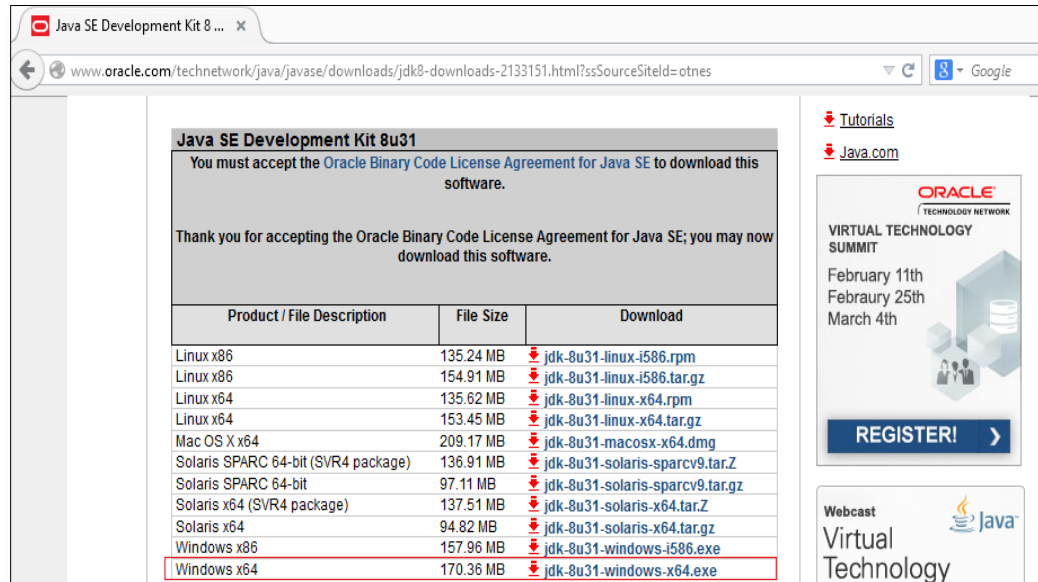


Ilustración 11: Sitio Web de Oracle para descargar JDK

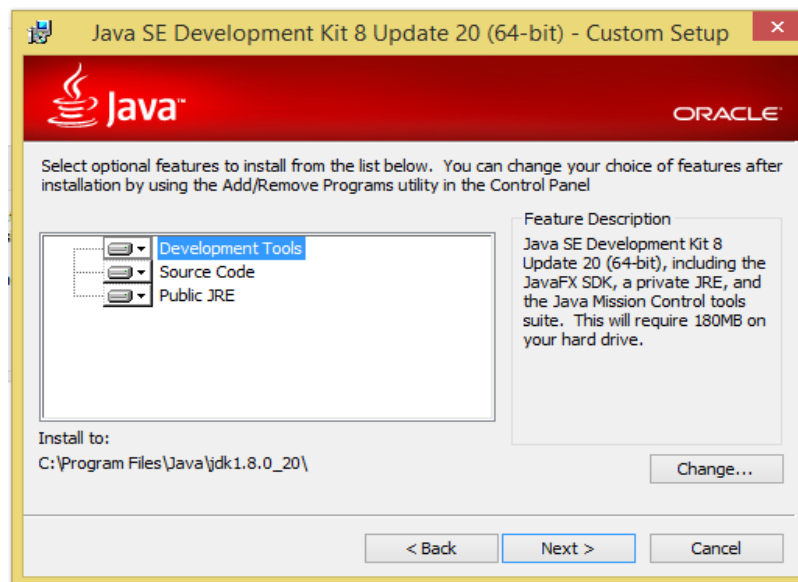


Ilustración 12: Características de la Instalación de JDK

3.4.2 Instalación y configuración del SDK de Android

El Android SDK Es un paquete bastante completo e incluye emuladores de sistema operativo y aplicaciones especializadas para controlar y monitorear

todos los elementos de una aplicación móvil. Para su instalación vamos a seguir los siguientes pasos:

1. Descargar el paquete el cual se lo hace desde el sitio oficial de desarrolladores de Google en esta dirección: <http://developer.android.com/sdk/index.htm> Ver **Ilustración 16**.
2. Una vez descargado el archivo comprimido, procedemos a descomprimir el archivo en el disco duro del equipo, en donde tendremos el SDK Manager.
3. Posteriormente ejecutamos el SDK Manager para instalar paquetes adicionales del sistema operativo Android, como las versiones o (APIs), para realizar las pruebas en las diferentes versiones, Además en Extras seleccionamos algunos complementos o librerías que vamos a necesitar, damos clic en instalar paquetes. Ver **Ilustración 17**.

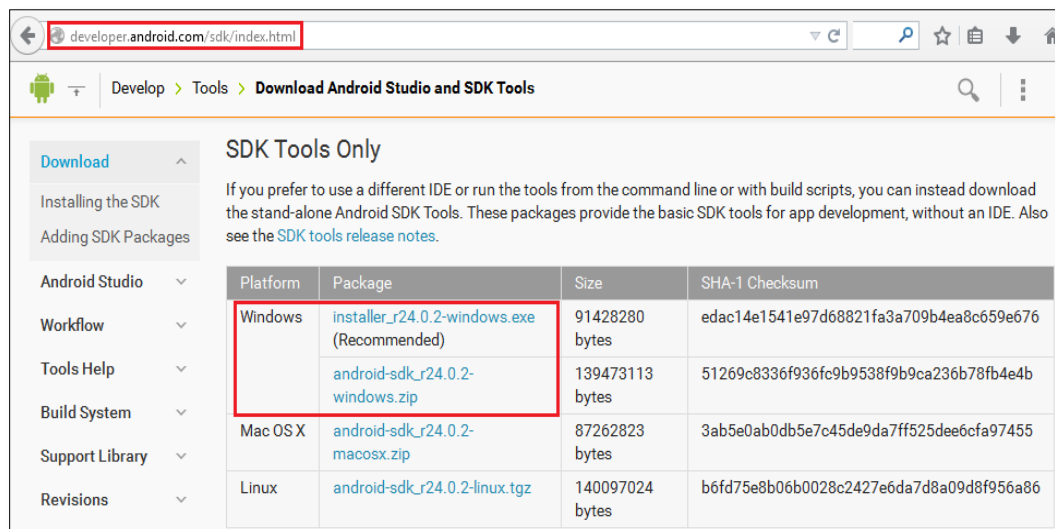
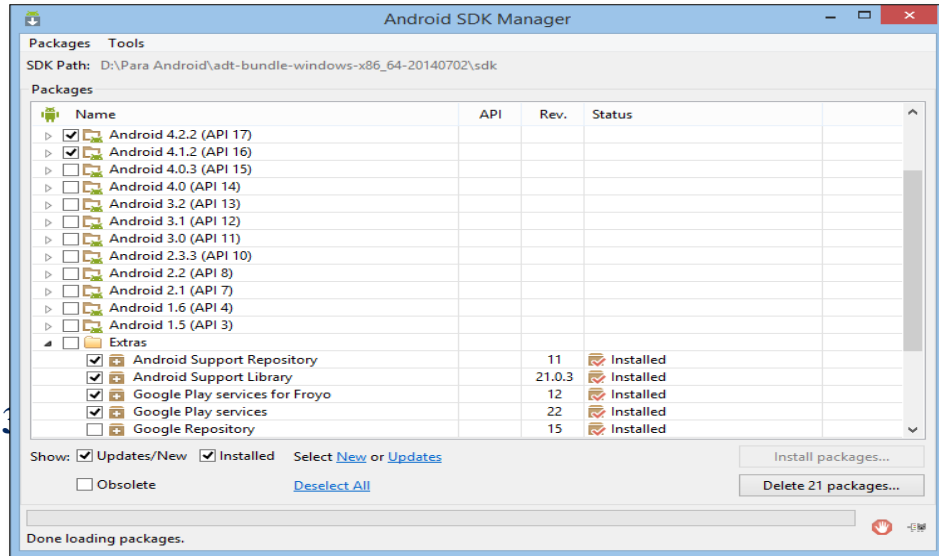


Ilustración 13: Sitio Web para descarga del SDK Android

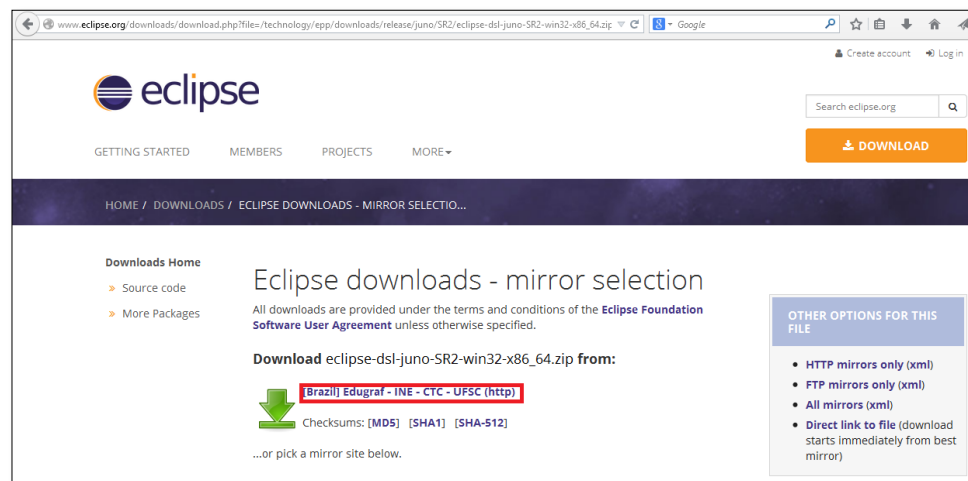
Ilustración 14:
Instalación de APIs del SDK de Android



Instalación y configuración de Eclipse

Vamos a detallar la instalación y configuración del *IDE* Eclipse, para lo cual seguiremos los siguientes pasos:

1. El primer paso es descargar Eclipse Juno SR2 Packages como entorno de desarrollo, el cual se descarga del sitio web de Eclipse: http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/SR2/eclipse-dsl-juno-SR2-win32-x86_64.zip, dentro de este sitio escogemos el servidor de



V

er Ilustración 18.

Ilustración 15: Descarga de Eclipse Juno del sitio web oficial.

1. Cuando ya tengamos descargado debemos descomprimir el fichero en un nuevo directorio, damos click en el ejecutable de Eclipse, la primera vez que se ejecuta la aplicación nos pide escoger nuestro workspace (espacio de trabajo) es decir la carpeta donde se almacenarán todas las aplicaciones desarrolladas.
2. Finalmente se procederá a crear una máquina virtual de Android, para esto nos dirigimos a la pestaña *Windows* (1) y seleccionamos *Android Virtual Device Manager* (2), aquí podemos crear los dispositivos virtuales para la ejecución de las aplicaciones en un entorno virtual, Para lo cual seleccionamos en la opción *Create* (3). Ver *Ilustración 19*.

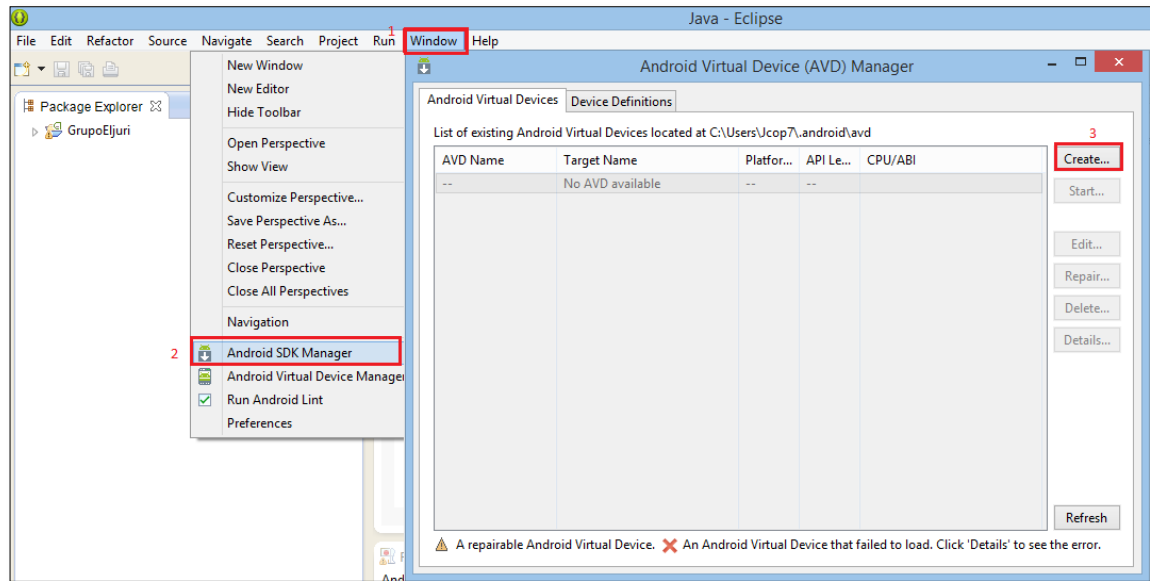


Ilustración 16: Crear Dispositivo Virtual en Eclipse.

3.4.4 Instalar el plugin de Android para Eclipse

Para instalar el plugin específico de Android para la plataforma de desarrollo Eclipse Juno nos valemos de la herramienta ADT (Android Development Tools) la cual nos facilita el desarrollo de proyectos basados en Android. Vamos a detallar los pasos a seguir:

1. Seleccionamos *Help>Install New Software* Ver *Ilustración 20*.

2. El siguiente paso es agregar repositorio, en nombre colocamos cualquier nombre (Android) y en la localización ingresamos el siguiente repositorio web.
`https://dl-ssl.google.com/android/eclipse/` Ver *Ilustración 21.*

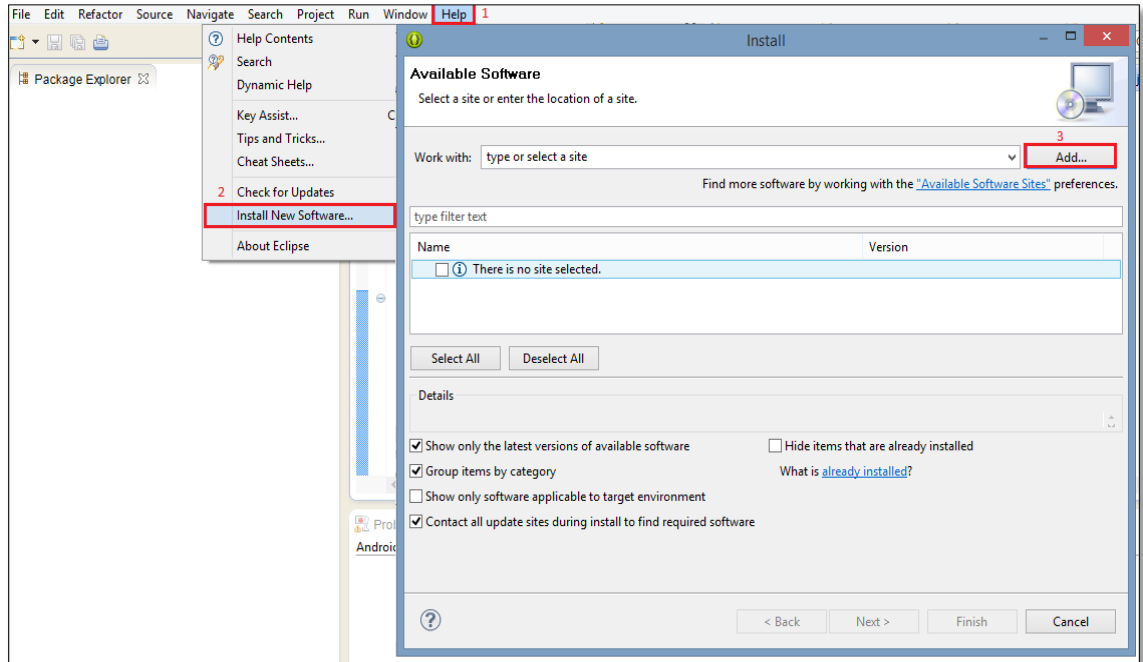


Ilustración 17: Pasos para agregar plugin Android en Eclipse

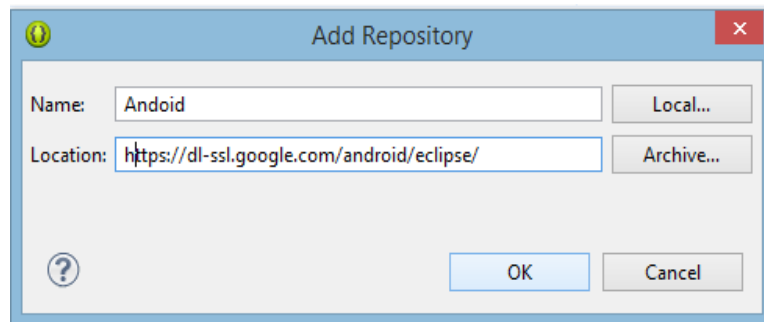


Ilustración 18: Ventana Agregar Repositorio

1. Posteriormente seleccionamos la opción Development Tools y seleccionamos siguiente, ahora esperamos que instalen los paquetes necesarios para el desarrollo de software en Android. Ver *Ilustración 22.*

- Finalmente cuando se termine la instalación nos queda cerrar Eclipse y volver a abrirlo para que se realicen los cambios.

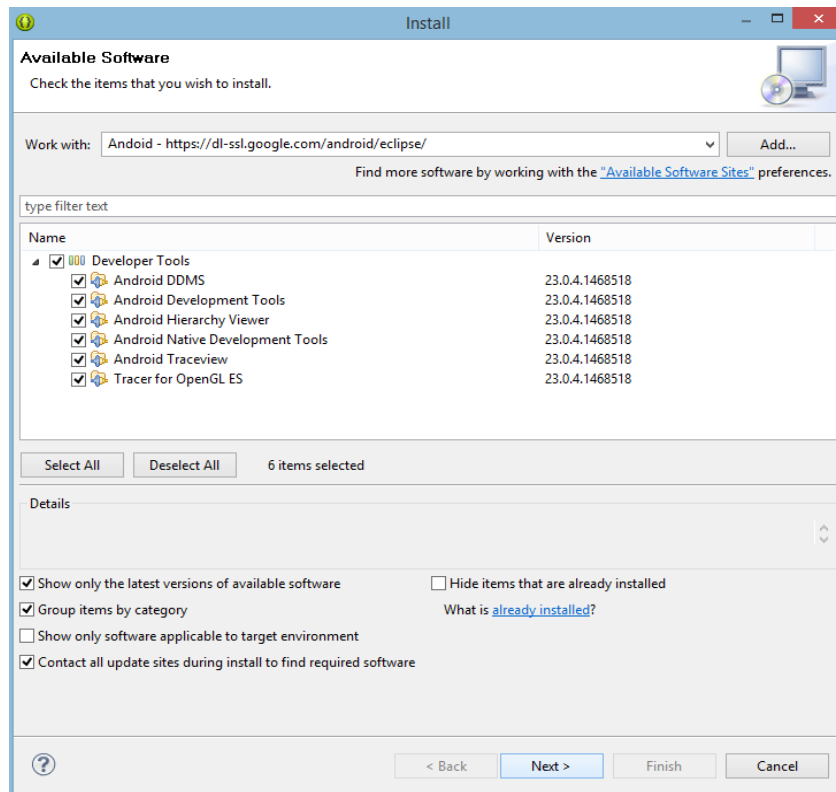


Ilustración 19: Selección para instalación de paquetes para Android.

3.4.5 Creación de un nuevo proyecto Android

Este apartado nos servirá de mucho ya que aprenderemos mediante un pequeño ejemplo, como crear un nuevo proyecto de Android para lo cual seguiremos los siguientes pasos:

- Seleccionar en el Menú *File > New > Project*, y en el menú resultante desplegar la opción Android y elegir Android Project y damos clic en siguiente. *Ilustración 23*

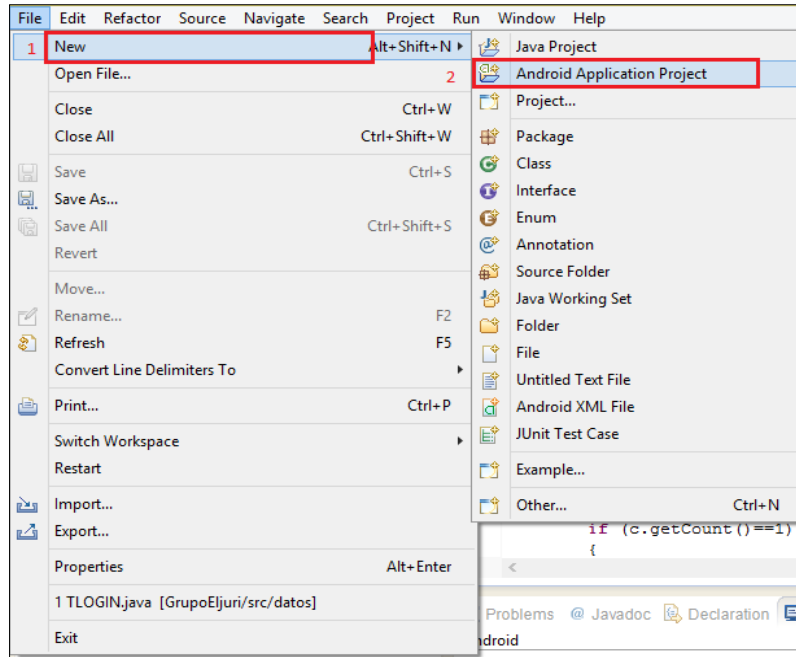


Ilustración 20: Creación de un Nuevo Proyecto Android en Eclipse

2. Procedemos a la configuración del nuevo proyecto, el cual tiene los siguientes parámetros: *Ver Ilustración 24.*
 - a. **Applicationname:** El nombre de la aplicación que se va a desarrollar, constituye el nombre visible para el usuario del dispositivo móvil.
 - b. **Project name:** Es el nombre del proyecto, en la práctica será el nombre que reciba la carpeta donde se guardará todo lo relativo al presente proyecto dentro del workspace.
 - c. **Packagename:** El nombre del paquete bajo el cual será desarrollado todo el código.
 - d. **Minimum Required SDK:** Nos permite escoger con que versión mínima de Android va a correr la aplicación. Las otras opciones de igual manera nos piden las Versiones de Android (*API*) que se va a configurar el proyecto.

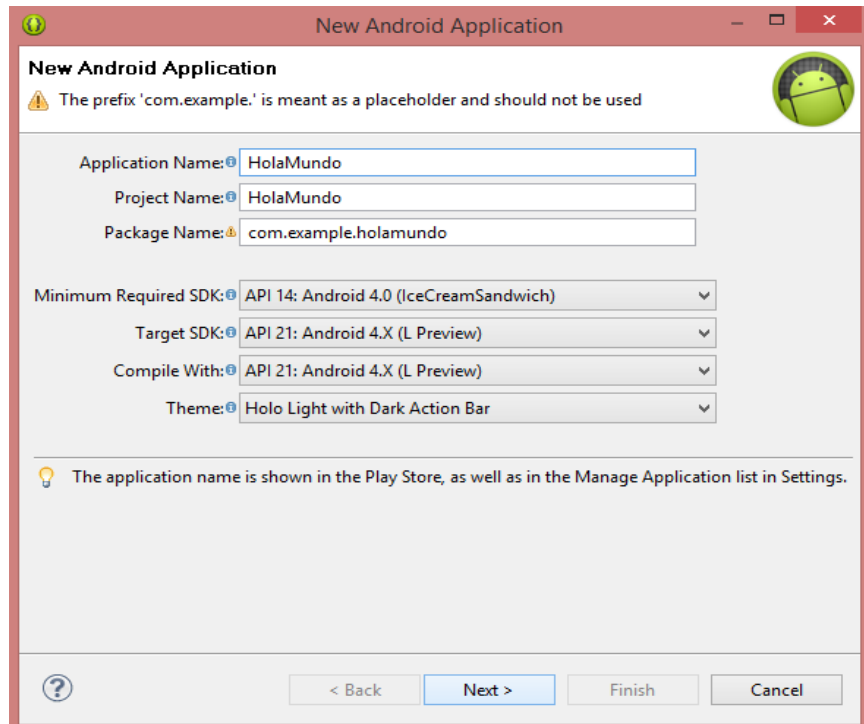


Ilustración 21: Parámetros de un Proyecto Android en Eclipse

Aunque hemos puesto los mismos nombres en los 3 sitios no siempre tiene por qué coincidir ya que el nombre de la aplicación es lo que se verá en el dispositivo móvil, las actividades son los controladores de Android, el nombre del paquete es algo interno y el proyecto es lo que se verá en Eclipse.

3. Seleccionamos Finalizar tendremos nuestro proyecto creado por lo vamos a ver la estructura de nuestro proyecto:

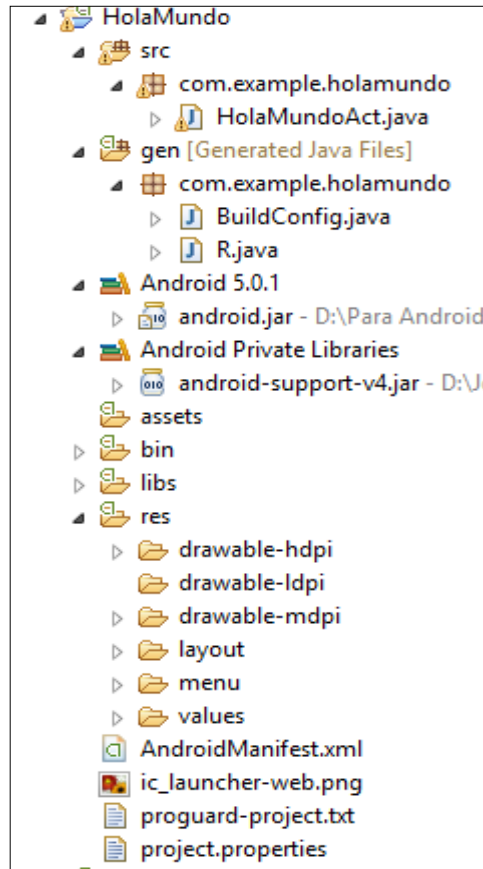
a. Assets: En este directorio se pone cualquier tipo de fichero externo que sea necesario por la aplicación que se esté ejecutando. Por ejemplo: HTML para el navegador sin necesidad de conexión, tipografías, `.jars`³⁶ de terceros. Es un repositorio de archivos.

³⁶ **JAR:** “Es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java. Los archivos JAR están comprimidos con el formato ZIP y cambiada su extensión a `.jar`.” *Tomado de Wikipedia.*

b. Bin: Los binarios compilados de la aplicación.

c. Res: Los recursos, formada por 3 directorios: – Drawable: ficheros de imágenes. – Layout: Ficheros de diseño de las interfaz de usuario. Ficheros xml que describen las interfaces. – Values: definición de variables. Ficheros xml en cada uno de ellos

constantes, se agrupa por temática, es decir: definición de colores, definición de texto, etc.



d. Gen: Se almacenan los archivos auto-generados por el compilador de Android. Contiene el archivo “R.java”, que es un índice a todos los recursos definidos en el proyecto.

e. Src: Aquí va todo el código Java de la aplicación.

f. .project, .classpath: ficheros necesarios para abrir el proyecto con Eclipse.

Ilustración 22: Estructura Proyecto Android en Eclipse

g. AndroidManifest.xml: es un archivo requerido para cada aplicación, describe los valores globales de su paquete, incluida la aplicación, componentes (actividades, servicios, etc.), las clases para cada uno de los componentes, qué tipo de datos puede manejar cada uno, y donde puede ser lanzado.

4. Para ejecutar la aplicación debemos dar clic derecho sobre el proyecto y seleccionamos *Run As > AndroidApplication*, la cual se podrá observar en el Android Virtual Device (ADV) creado o mediante un dispositivo real.

3.4.6 Creación del Android Virtual Device (AVD)

Para poder ejecutar o probar nuestro proyecto en Android tenemos que crear un dispositivo virtual Android o AVD (Android Virtual Device) Detallamos los pasos para su creación:

1. Abrimos el menú y seleccionamos: *Windows > Android Virtual Device Manager*.

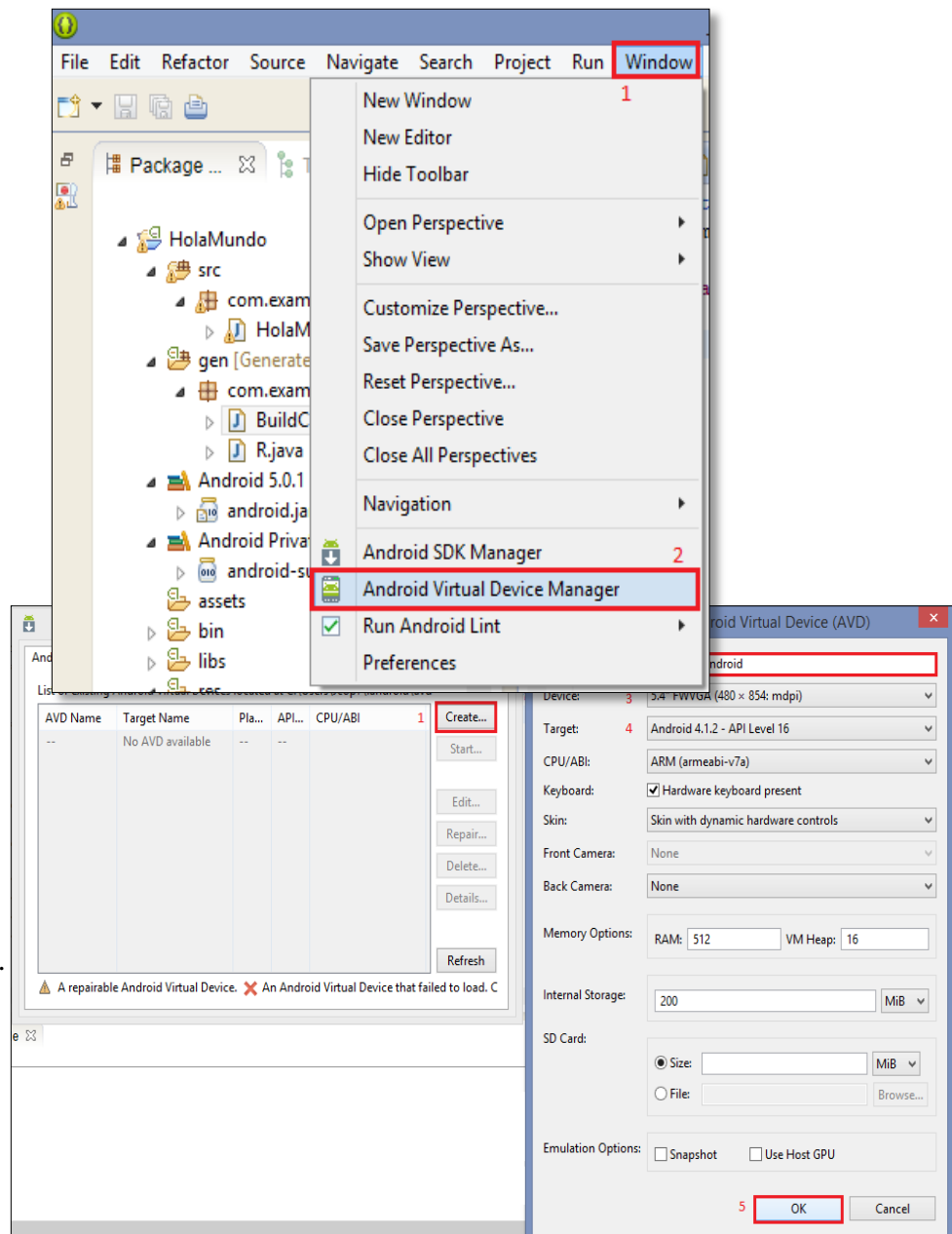


Ilustración 23: Abrir Android Virtual Manager AVD

2.

leccionamos Crear: y se nos abre otra ventana en la cual configuramos el dispositivo virtual, Le asignamos un nombre y configuraremos sus características como son: *Dispositivo*, *Versión del sistema en cual va correr (Target)*, *CPU*, *Memoria*, etc. Configuramos según nuestros requerimientos o de la aplicación. Y finalmente presionamos *OK*.

Ilustración 24: Creación y configuración del Android Virtual Manager AVD

Finalmente al ejecutar nuestra aplicación y tendremos:

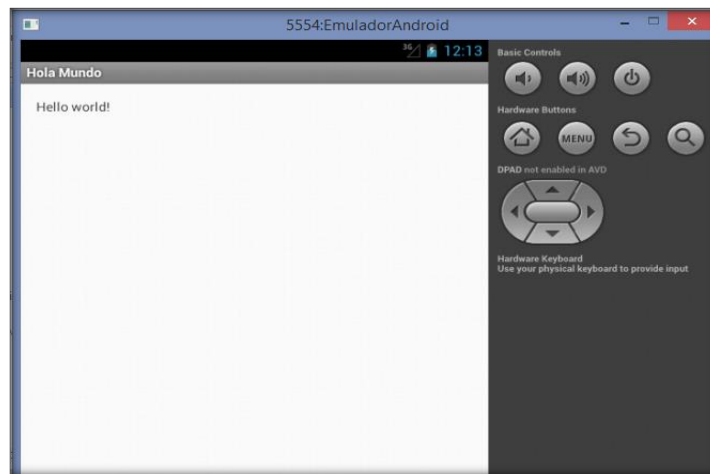


Ilustración 25: Ejecución de proyecto en Android Virtual Manager AVD

3.4.7 Ejecución de aplicaciones Android usando un dispositivo físico

Otra posibilidad de ejecutar y probar nuestra aplicación Android es en un Dispositivo físico Android ya sea una Tablet o un Smarthphone, con esto tenemos muchas ventajas que al emular en un AVD (Android Virtual Device) ventajas tales como:

- Contar con otras funcionalidades que un emulador no puede realizar tales como *acelerómetro*, *brújula* o *GPS*, entre otros.
- Ahorro de procesamiento ya que el computador ya no ejecuta ningún ADV, por lo que se evalúan las verdaderas prestaciones de la aplicación.
- Contar con aplicaciones como el Market o *PlayStore* la cual no está presente en el emulador.

- La presencia de más *Codecs*³⁷ para la reproducción de ciertos tipos de formato.

La aplicación a desarrollarse será emulada directamente en un dispositivo real por todas estas y otras más funcionalidades principalmente porque nuestra aplicación requiere del uso del *GPS*, para guardar la posición del Agente vendedor al momento de visitar al cliente por lo que vamos a ver los pasos para hacerlo:

1. Para ello, habilitamos el modo depuración en el dispositivo móvil (Ajustes > Aplicaciones > Desarrollo > Depuración USB) y lo conectamos al ordenador mediante USB. Ver *Ilustración 29*.
2. Eclipse lo reconocerá como un dispositivo en el que poder emular la aplicación y aparecerá en la lista de dispositivos cuando pulsemos *Run*. Aparece una ventana para con el dispositivo físico donde se ejecutará la aplicación sin consumir recursos del computador. Ver *Ilustración 30*. Se podrá también elegir emular en un dispositivo virtual.

Ilustración 26: Opciones del Desarrollador en dispositivo móvil.



³⁷ **Códec:** “Es la abreviatura de codificador-decodificador. Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos (stream) o una señal.” *Tomado de Wikipedia*

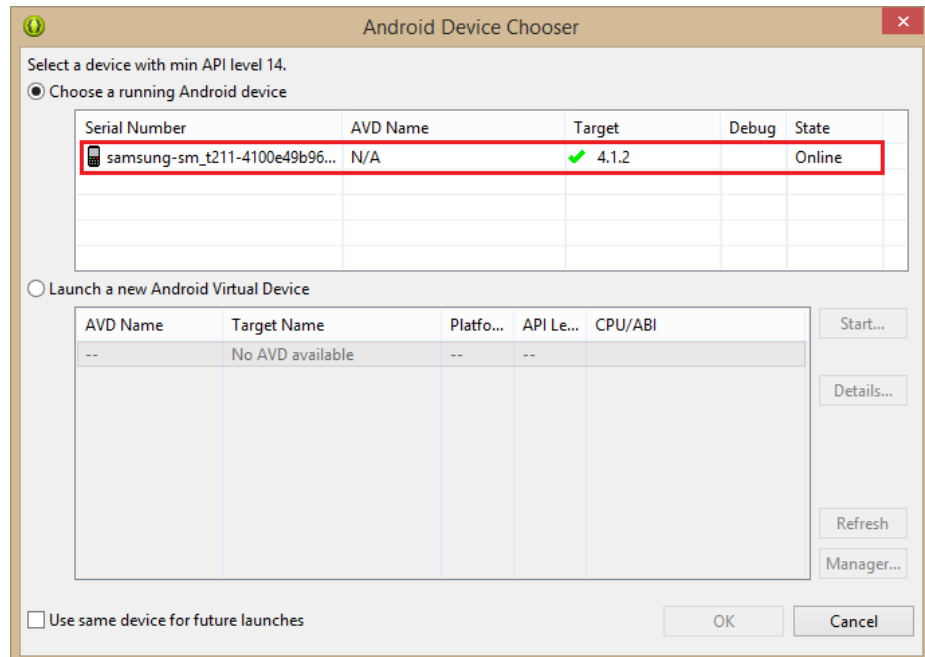


Ilustración 27: Dispositivo Android instalado en el AVD Manager.

Capítulo IV

Desarrollo de la Aplicación para la Administración de Pedidos y Control de Rutas de los Vendedores.

“En este Capítulo se tratará cada una de las Fases para el desarrollo de un aplicación móvil según la metodología Extreme Programming (XP) como son: La Fase de

Planificación en donde se analiza involucrados y requerimientos de software. La fase de Diseño donde está toda la parte de diagramación, La Fase de Codificación que es concretamente la fase de programación y por último la Fase de Pruebas donde se realiza las pruebas respectivas con su debida documentación”

4.1 Fase de Planificación

4.1.1 Descripción de los Interesados (Stakeholders)

En la siguiente matriz se puede apreciar cuáles son los *stakeholders*³⁸ o involucrados en el proyecto de desarrollo de la aplicación, para dispositivos móviles Android para administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa: “Almacenes Juan Eljuri Cía. Ltda.” División Perfumería.

NOMBRE	CARGO	INSTUTICION	RELEVANCIA PROYECTO (1-5)
Eco. Lucia Alvear	Gerente Administrativo	AJE ³⁹	5
Ing. Rene	Ingeniero de Sistemas	AJE	5

³⁸ **Stakeholders:** Involucrados o interesados, son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa. *Tomado de Wikipedia*

³⁹ **AJE:** Acrónimo de Almacenes Juan Eljuri.

Escobar	- Desarrollo		
Sra. Ma. Eugenia Chacho	Supervisor de Ventas	AJE- Perfumería	3
Sr. Manuel Lazo	Jefe Dto. Crédito y Cobranza	AJE- Crédito y Cobranza	3
Sra. Ines Idrovo	Facturadora	AJE- Perfumería	2
Sr. Juan Carlos Ordoñez	Agente Vendedor	AJE- Perfumería	3
Ing. Paola Ingavelez	Tutor Tesis	UPS	3

Tabla 4. Matriz de Stakeholders del Proyecto

Todos cumplen un papel fundamental dentro de la aplicación a desarrollar pero unos tienen más relevancia que otros, Ver **Tabla 4**. Para un mejor entendimiento de las funciones vamos a elaborar otra tabla con las tareas o funciones de cada involucrado. Ver **Tabla 5**

Cargo o Dto.	Descripción	Responsabilidad
Gerencia	Persona encargada de la parte Administrativa de la empresa	<ul style="list-style-type: none"> • Autorizar contratación personal. • Tomar decisiones administrativas de la empresa
Sistemas	Persona encargada de los Sistemas existentes de la empresa y de los requerimientos de Software	<ul style="list-style-type: none"> • Administración Base de datos • Desarrollar requerimientos de Software
Supervisor de Ventas	Persona encargada de revisar los Pedidos y de Controlar y a los diferentes Agentes Vendedores	<ul style="list-style-type: none"> • Revisar cada uno de los pedidos para entrega material publicitario • Controlar el trabajo de los agentes vendedores
Crédito y Cobranza	Encargado de la Cartera de los clientes	<ul style="list-style-type: none"> • Revisar cartera de algún cliente específico. • Aprobar o desaprobar pedidos perfumería.
Facturación	Encargada de facturar las diferentes órdenes de pedido,	<ul style="list-style-type: none"> • Ingresar manualmente los pedidos al sistema. • Facturar el pedido ingresado.
Tutoría	Persona que dirige el proyecto	<ul style="list-style-type: none"> • Verificar el correcto desarrollo del proyecto.

		<ul style="list-style-type: none"> • Revisar avances proyecto
Ventas	Persona que recepta los pedidos de los clientes	<ul style="list-style-type: none"> • Recorrer rutas asignadas y visitar a los diferentes clientes • Receptar los pedidos de clientes

Tabla 5. Resumen de las Tareas de los involucrados.

4.1.2 Requerimientos Funcionales del proyecto.

La aplicación estará compuesta de 4 partes o módulos en las cuales vamos a detallar cada una de las opciones con las que contará:

1. Pedido

- Seleccionar Cliente
- Añadir Artículo por:
 - Catalogo
 - Código
- Guardar posición en GPS
- Detallar Pedido:
 - Seleccionar Dirección Entrega
 - Adelanto (En caso de haberlo)
 - Forma de Pago
 - Descripción
- Visualizar resumen del Pedido

2. Catalogo

- Visualizar Tipo
- Visualizar Marca
- Visualizar detalle del Artículo seleccionado.

3. Modificar Pedido

- Selección del Pedido:
 - Modificar Dirección Entrega

- Modificar Adelanto (En caso de haberlo)
- Modificar Forma de Pago
- Modificar Descripción
- Visualizar resumen del Pedido

4. Ayuda

- Visualizar ayuda (Texto HTML)

En cuanto a la Intranet que usara el supervisor de Ventas constará de las siguientes opciones:

1. Inicio Sesión:

2. Visualizar en Mapa puntos de referencia

- Ver Detalle (Movimientos)
- Ver Mapa (Visualizar Recorrido)

Para un mejor entendimiento podemos revisar más adelante la interacción de los Usuarios con los diferentes módulos en los Casos de Uso.

4.1.3 Requerimientos No Funcionales de la Aplicación.

Rendimiento de la Aplicación:

- La aplicación ofrecerá respuesta (datos) al usuario en tiempo real.
- El tiempo de respuesta promedio de la aplicación no debe superar los 10 segundos.

Seguridad

- Requisito de conexión. (Debe verificar que sea un usuario creado para la aplicación por lo que consulta directamente en la Base de Datos para acceder a la aplicación.). Este caso aplica tanto para la Aplicación como para la Intranet.

Disponibilidad

- La aplicación estará disponible el 100% del tiempo, ya que se trata de una aplicación nativa que se instalará en el dispositivo móvil.
- En cuanto a la Intranet esta también estará siempre disponible ya sea localmente en la empresa o fuera de ella ya que esta publicada al internet para acceder mediante la página web de la empresa: <http://www.grupoeljuri.com/>
- La aplicación dependerá de una conexión a internet o plan de datos **solo** para el envío de información. En caso de no tener conexión a internet la misma se guardara en la aplicación (localmente) y se reenviará una vez que tenga conexión a internet
- La aplicación por ahora no será cargada al *Google Play Store*⁴⁰ de Android ya que al ser una aplicación para una empresa privada se limitará a que solo el personal encargado lo instale en las respectivas Tablets de los usuarios de la aplicación.

Mantenibilidad

- El sistema estará en constante mantenimiento ya que se podría agregar nuevas funcionalidades o realizar modificaciones o correcciones.

Portabilidad

⁴⁰ **Google Play Store:** “Es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google. Ésta plataforma permite a los usuarios navegar y descargar aplicaciones.”
Tomado de Wikipedia.

- Compatibilidad con plataformas: En el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0. ya sea en una Tablet o un Smarthphone.
- La intranet es compatible con todos los navegadores pero se recomienda el uso de Internet Explores desde la Versión 8. Para una mejor visualización, también se recomienda actualizar a la última versión del navegador.

Operabilidad

- La aplicación solo podrá ser operada por un empleado de Almacenes Juan Eljuri y que tenga su debido usuario y contraseña, de igual forma para el ingreso a la Intranet.

4.1.4 Restricciones

El dispositivo móvil en donde se ejecutará la aplicación para su correcto funcionamiento deberá tener los siguientes requisitos mínimos, debido a que necesitamos velocidad de procesamiento al manejar una gran cantidad de información, además necesitaremos almacenamiento para la base de datos local donde se almacenara la información descargada.

- Procesador: 1 GHz Dual Core
- Memoria RAM: 1 GB.
- Almacenamiento: 30 MB disponibles.
- Pantalla de 7" pulgadas en adelante.
- Sistema operativo. Android 4.0 o superior

- GPS con soporte A-GPS⁴¹

Se ha considerado la versión de *Android 4.0 Ice Cream Sandwich* o superior ya que la aplicación será programada con el *API 15* como mínima, correspondiente a esta versión.

Para poder enviar los pedidos y posteriormente se grabe en la base de datos en el dispositivo móvil se necesita de una conexión a Internet ya sea:

- Wifi.
- Plan de Datos móviles (3G⁴²)

Si por alguna razón no se tendría conexión a internet, la información se graba localmente (en el dispositivo) y se enviara una vez que se tenga conexión.

Para la Parte de la *Intranet* y los requisitos mínimos serian:

- Computador Dual Core o superior.
- Memoria 1GB RAM, 120 GB Disco Duro o superior
- Navegador Internet Explorer 8 o superior
- Conexión a la intranet de la empresa (red interna).

Se detallará algunas marcas y modelos de Tablets más reconocidas (*Ver Tabla 6*), las cuales cumplen los requisitos mínimos y en las cuales la aplicación desarrollada, no tendría problemas en la ejecución ya se por problemas de procesamiento o de visualización, además detallaremos algunas características y un precio de mercado aproximado a la fecha (*Marzo 2015*).

⁴¹ **A-GPS:** o “GPS asistido ayudan al acceso a un Servidor de Asistencia en línea (modo "on-line") o fuera de línea (modo "off-line") es decir con conexión o sin conexión a internet.”
Tomado de Wikipedia

⁴² **3G:** “Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz como datos. En algunos dispositivos requieren de una tarjeta SIM para su uso.”
Tomado de Wikipedia.

Marca/Modelo	Procesador	Memoria	Versión Android	Precio Aprox. ⁴³
Titán 7080ME 7"	ARM Dual Cortex 1.3 GHz	1GB	4.2 Jelly Bean	\$175
Quasad Quad Core	Quad Core 1.2 GHz ARM	1GB	4.4 Kitkat	\$150
Ultratech X7 3G 7"	Dual Core 1.2ghz	1GB	4.2 Jelly Bean	\$180
Samsung GALAXY Tab 2 10.1	Cortex-A9 Dual Core 1GHz	1GB	4.0 Ice Cream Sandwich	\$350
Samsung Galaxy Tab 3 T211	Dual Core 1.2 GHz	1GB	4.1.2 Jelly Bean	\$360
Samsung Galaxy Tab 4 Sm-t231	Quad-Core 1.2 GHz	1.5GB	4.4 Kitkat	\$400

Tabla 6. Descripción de Tablets con los requisitos para la Aplicación. Tomado de [28]

Cabe recalcar que todas las restricciones descritas y analizadas anteriormente es solo una sugerencia para el óptimo funcionamiento de la aplicación es decir que sea rápida y tenga una buena visualización, y no quita o restringe que la aplicación pueda funcionar en dispositivos de menor gama o incluso en Smartphone que tienen la pantalla más pequeña que una Tablet.

4.2 Fase de Diseño

Se elaborarán diseños breves que sirven de referencia para la implementación. Otra práctica fundamental de la metodología de Programación Extrema (*XP*) es utilizar diseños tan simples como sea posible. El principio es utilizar el diseño más sencillo que consiga que todo funcione evitando diseñar características extra y que tomaran demasiado tiempo.

Para esta fase vamos a emplear el modelado *UML* ya que se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. *UML* ofrece varios diagramas en los cuales modelar sistemas, los desarrolladores lo eligen puesto que es un lenguaje que ayuda a discutir los

⁴³ *Precio Aproximado*: “Los precios fueron en base de las Tablets que cumplen el requisito mínimo para la aplicación y las más reconocidas en el mercado.” *Tomado de Mercado Libre.*

problemas y soluciones implicadas en la construcción del sistema y principalmente porque es el lenguaje de modelado de software más conocido y utilizado.

En nuestro proyecto hemos optado en la realización de los Diagramas más relevantes como son:

- Diagramas de Casos de Uso.
- Diagramas de Clases
- Diagramas de Secuencia.

A parte de los modelados *UML* vamos a complementar nuestro diseño con modelos como el **Arquitectónico** con el objetivo de ofrecer una visión simplificada del sistema, de forma que una persona pueda mirar el diagrama y entender de una pasada lo que se quiere conseguir o desarrollar. Y el de **Navegación** el cual nos ayudará en la comprensión del orden de presentación de las pantallas de nuestra aplicación con los contenidos y los vínculos que existen en cada una de ellas.

4.2.1 Diagrama de Casos de Uso

Identificación de los Actores.

Actores de la Aplicación de Pedidos	
Agente Vendedor	Usar la aplicación, Realizar pedidos
Supervisor Ventas	Ingresar Intranet empresa, Revisión pedidos ingresados, asignar material publicitario
Jefe Cartera	Ingresar Intranet empresa, Aprobar o Rechazar pedidos de acuerdo a cartera de cliente.

Tabla 7. Actores de la aplicación de Pedidos

4.2.1.1 Caso de Uso General, Aplicación de Pedidos.

Vamos a ver como el Agente Vendedor interactúa en el escenario de la aplicación de pedidos por lo consiguiente aparecerán muchos casos de uso, que luego vamos a irlos especificando uno a uno:

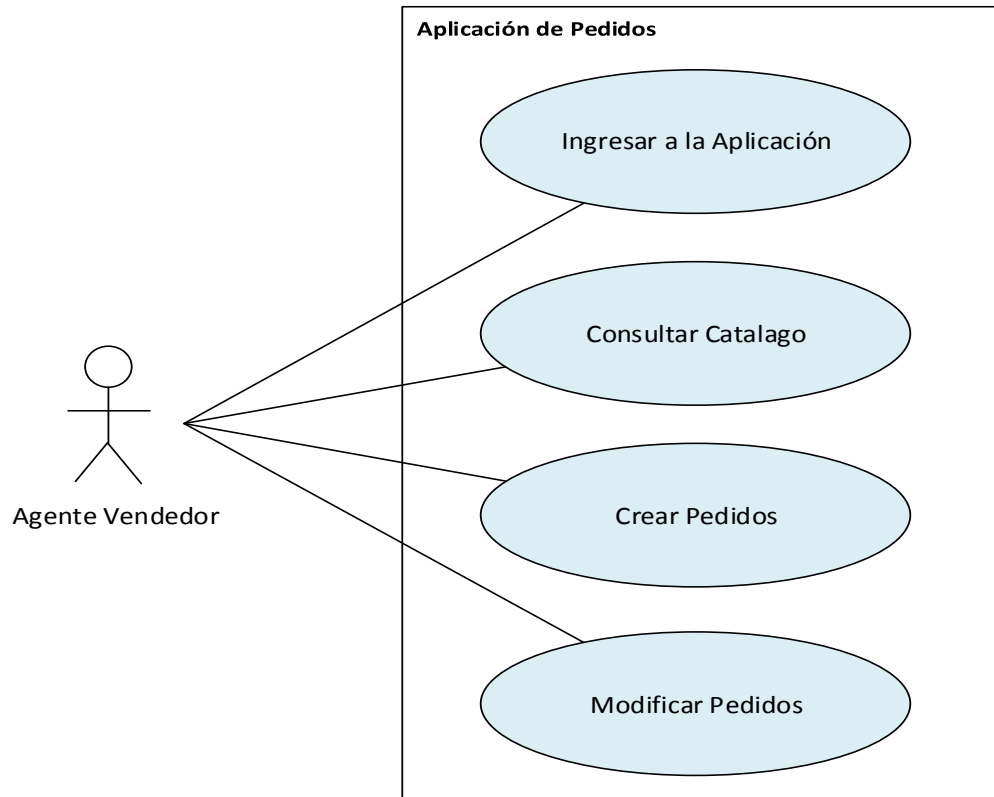


Ilustración 28: Caso de Uso General la Aplicación de Pedidos

Especificación de Casos de Uso:

CASO DE USO NUMERO 1:		
Nombre	Ingresar a la Aplicación.	
Descripción	Identificar al usuario para el uso de la Aplicación.	
Actores	Agente Vendedor	
Precondición	<ul style="list-style-type: none"> • Conexión a la base de datos satisfactoria • Tener un usuario registrado por el administrador • Que la aplicación se haya ejecutado. 	
Flujo del Sistema	Paso	Acción
	1	La aplicación despliega ventana de Inicio de Sesión.

	2	El agente Vendedor ingresa su respectivo usuario y clave.
	3	La aplicación valida los datos ingresados en la base de datos.
	4	La aplicación muestra la interfaz gráfica para el usuario.
Post Condición:		<ul style="list-style-type: none"> ○ Si el agente vendedor digita mal su clave varias veces el usuario ingresado se bloquea.

Tabla 8. Especificación Caso de Uso Número 1, Ingresar a la Aplicación.

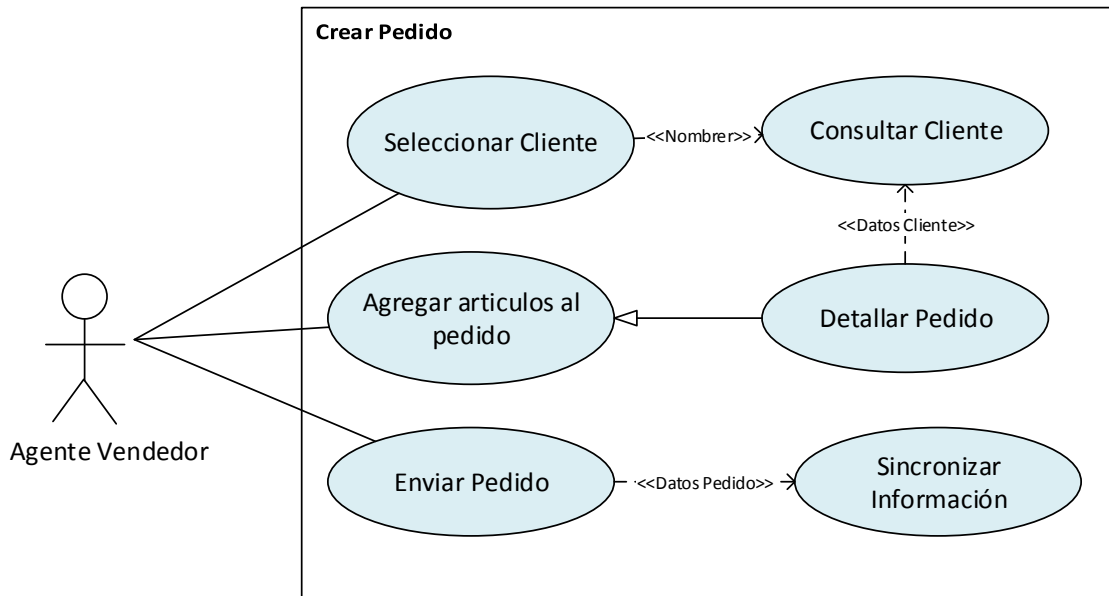


Ilustración 29: Caso de Uso 2, Crear Pedido

CASO DE USO NUMERO 2:																							
Nombre	Crear Pedido																						
Descripción	Crear pedido en la aplicación																						
Actores	Agente Vendedor																						
Precondición	<ul style="list-style-type: none"> • Haberse conectado a una red Wifi o de datos • Iniciado sesión correctamente • Haber sincronizado información correctamente. 																						
Flujo del Sistema	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El agente vendedor elige la opción Pedido en el Menú.</td> </tr> <tr> <td>2</td> <td>La aplicación le pide que seleccione Cliente.</td> </tr> <tr> <td>3</td> <td>El agente vendedor selecciona el cliente</td> </tr> <tr> <td>4</td> <td>La aplicación pide que seleccione los Artículos ya sea por Catalogo o ingrese por código.</td> </tr> <tr> <td>5</td> <td>El agente vendedor selecciona los artículos que desea comprar el cliente.</td> </tr> <tr> <td>6</td> <td>La aplicación guarda la posición en GPS automáticamente (Transparente usuario)</td> </tr> <tr> <td>7</td> <td>La aplicación pide que seleccione Dirección de entrega, Adelanto (opcional) y Forma de Pago y alguna descripción</td> </tr> <tr> <td>8</td> <td>El agente vendedor selecciona o modifica Dirección de Entrega, así como el adelanto y la forma de pago</td> </tr> <tr> <td>9</td> <td>La aplicación muestra el Detalle del pedido generado</td> </tr> <tr> <td>10</td> <td>El agente vendedor graba el Pedido Realizado.</td> </tr> </tbody> </table>	Paso	Acción	1	El agente vendedor elige la opción Pedido en el Menú.	2	La aplicación le pide que seleccione Cliente.	3	El agente vendedor selecciona el cliente	4	La aplicación pide que seleccione los Artículos ya sea por Catalogo o ingrese por código.	5	El agente vendedor selecciona los artículos que desea comprar el cliente.	6	La aplicación guarda la posición en GPS automáticamente (Transparente usuario)	7	La aplicación pide que seleccione Dirección de entrega, Adelanto (opcional) y Forma de Pago y alguna descripción	8	El agente vendedor selecciona o modifica Dirección de Entrega, así como el adelanto y la forma de pago	9	La aplicación muestra el Detalle del pedido generado	10	El agente vendedor graba el Pedido Realizado.
	Paso	Acción																					
	1	El agente vendedor elige la opción Pedido en el Menú.																					
	2	La aplicación le pide que seleccione Cliente.																					
	3	El agente vendedor selecciona el cliente																					
	4	La aplicación pide que seleccione los Artículos ya sea por Catalogo o ingrese por código.																					
	5	El agente vendedor selecciona los artículos que desea comprar el cliente.																					
	6	La aplicación guarda la posición en GPS automáticamente (Transparente usuario)																					
	7	La aplicación pide que seleccione Dirección de entrega, Adelanto (opcional) y Forma de Pago y alguna descripción																					
	8	El agente vendedor selecciona o modifica Dirección de Entrega, así como el adelanto y la forma de pago																					
9	La aplicación muestra el Detalle del pedido generado																						
10	El agente vendedor graba el Pedido Realizado.																						
Post Condición:	<ul style="list-style-type: none"> ○ Si la aplicación detecta conexión a internet envía el Pedido caso contrario lo almacena localmente. 																						

Tabla 9. Especificación Caso de Uso Número 2, Crear Pedido

Especificación de Casos de Uso Catalogo.

CASO DE USO NUMERO 3:		
Nombre	Consultar Catalogo	
Descripción	Mostrar Artículos disponibles debidamente organizados.	
Actores	Agente Vendedor	
Precondición	<ul style="list-style-type: none">• Haberse conectado a una red Wifi o de datos• Iniciado sesión correctamente• Haber sincronizado información correctamente.	
Flujo del Sistema	Paso	Acción
	1	El agente vendedor elige la opción Catálogo en el Menú.
	2	La aplicación muestra los artículos organizado por Tipo, Marca
	3	El agente vendedor visualiza los artículos existentes con sus características.
Post Condición:	<ul style="list-style-type: none">○ El agente vendedor puede ver el Catalogo existente o puede regresar al Menú principal.	

Tabla 10. Especificación Caso de Uso Número 3, Consultar Catalogo.

El Caso de Uso número 4 de *Modificar Pedido* Viene siendo casi lo mismo que el de *Crear Pedido* con la única diferencia que el Pedido ya está creado y para poder Modificarlo lo seleccionamos previamente, por lo que vamos a ver otros casos de uso.

4.2.1.2 Caso de Uso General, Página Web.

Dentro del escenario de la Intranet de la empresa vamos a analizar los Casos de Uso más importantes y el que se implementó que viene a ser el Caso de Uso: Revisar Ruta de Vendedores.

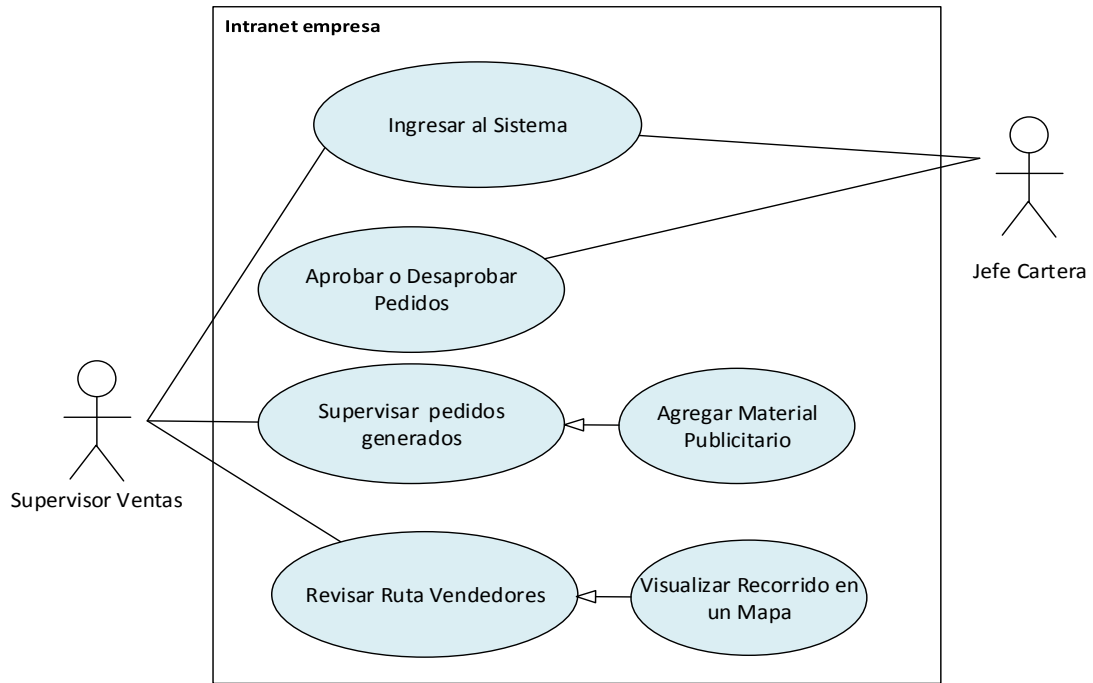


Ilustración 30: Caso de Uso Intranet de la Empresa

CASO DE USO NUMERO 5:		
Nombre	Revisar Ruta de Vendedores	
Descripción	Supervisar el recorrido de cada Agente vendedor mediante un mapa.	
Actores	Supervisor Ventas	
Precondición	<ul style="list-style-type: none"> • Disponer de un computador que tenga algún navegado instalado. • Haber ingresado con respectivo usuario y clave • Estar conectado a la red interna de la empresa o externamente (internet). 	
Flujo del Sistema	Paso	Acción
	1	El actor ingresa a la dirección de la intranet en el navegador
	2	La página Web muestra la ventana para incido de Sesión
	3	El actor ingresa su debido usuario y contraseña
	4	La página Web muestra las operaciones permitidas al actor (Ver recorrido)
	5	El actor selecciona la operación que desea realizar en la Web.

	6	La página Web muestra diversos filtros para dibujar en el mapa.
	7	El actor ingresa criterios de búsqueda en el mapa.
	8	La página Web, muestra recorrido en mapa según criterios ingresados, con información detallada
	9	El actor puede realizar más búsquedas.
Post Condición:		<ul style="list-style-type: none"> ○ El supervisor de ventas puede realizar otras operaciones en la página web como revisar pedido,

Tabla 11. Especificación Caso de Uso Número 5, Revisar Ruta Vendedores

4.2.2 Diagrama de Clases

El diagrama que se mostrara a continuación (*Ver Ilustración 34*) está directamente relacionado con las clases que se utilizaron en la aplicación móvil,

ya que para poder construir o desarrollar la misma fue necesario crear nuestras propias tablas en la Base de Datos de nuestro proyecto Android (*SQLite*), Es importante recalcar que algunas tablas y clases son las mismas que existen en la base de datos principal de donde extraemos los datos (DB2), algunas otras son creadas de acuerdo a la necesidad de nuestro proyecto.

El en siguiente Diagrama vamos a ver todas las clases con sus respectivos atributos y como están relacionados unas con otras.

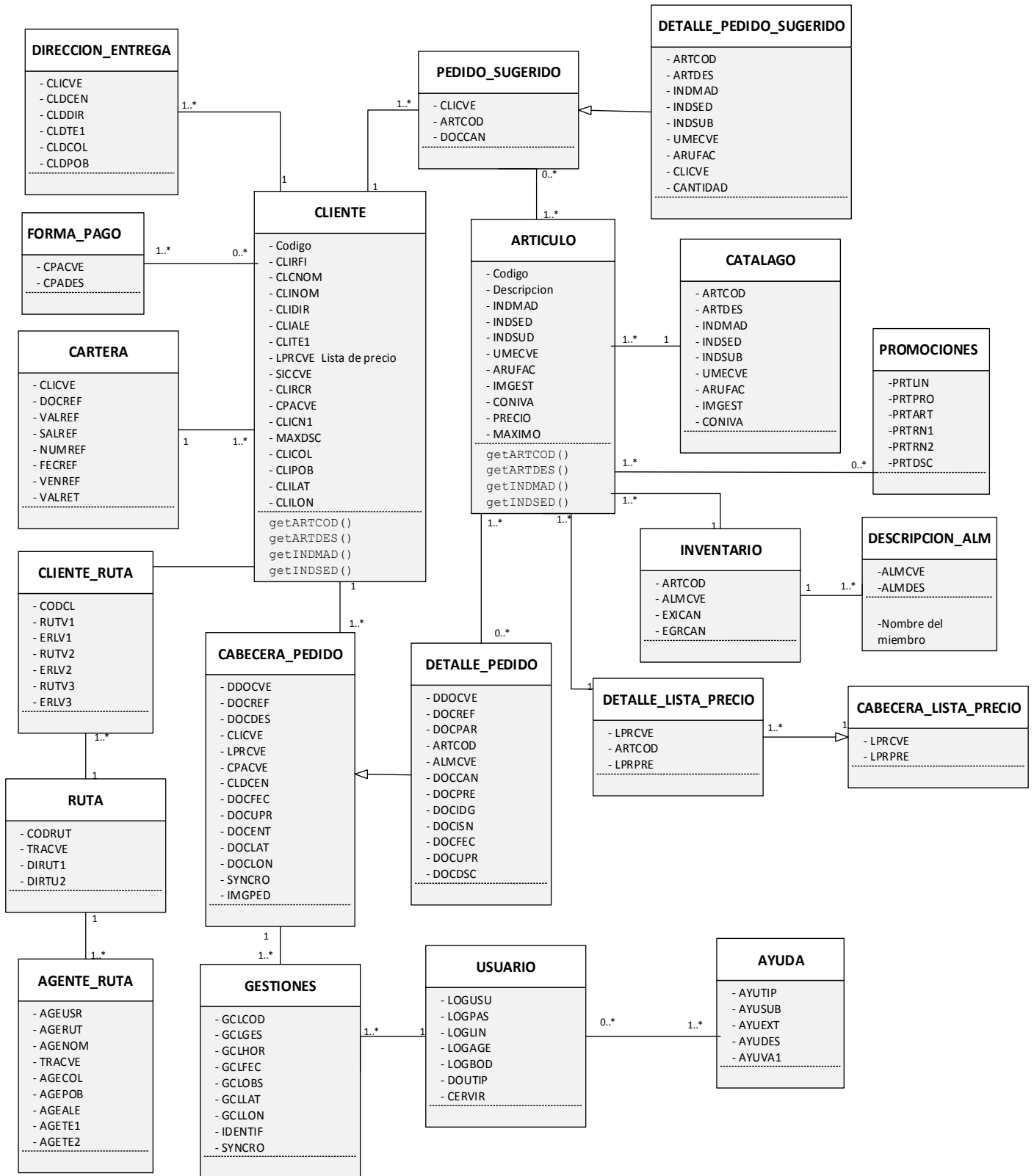


Ilustración 31. Diagrama de Clases de la Aplicación de Pedidos

4.2.3 Diagramas de Secuencia

4.2.3.1 Diagrama de Secuencia, Inicio de Sesión.

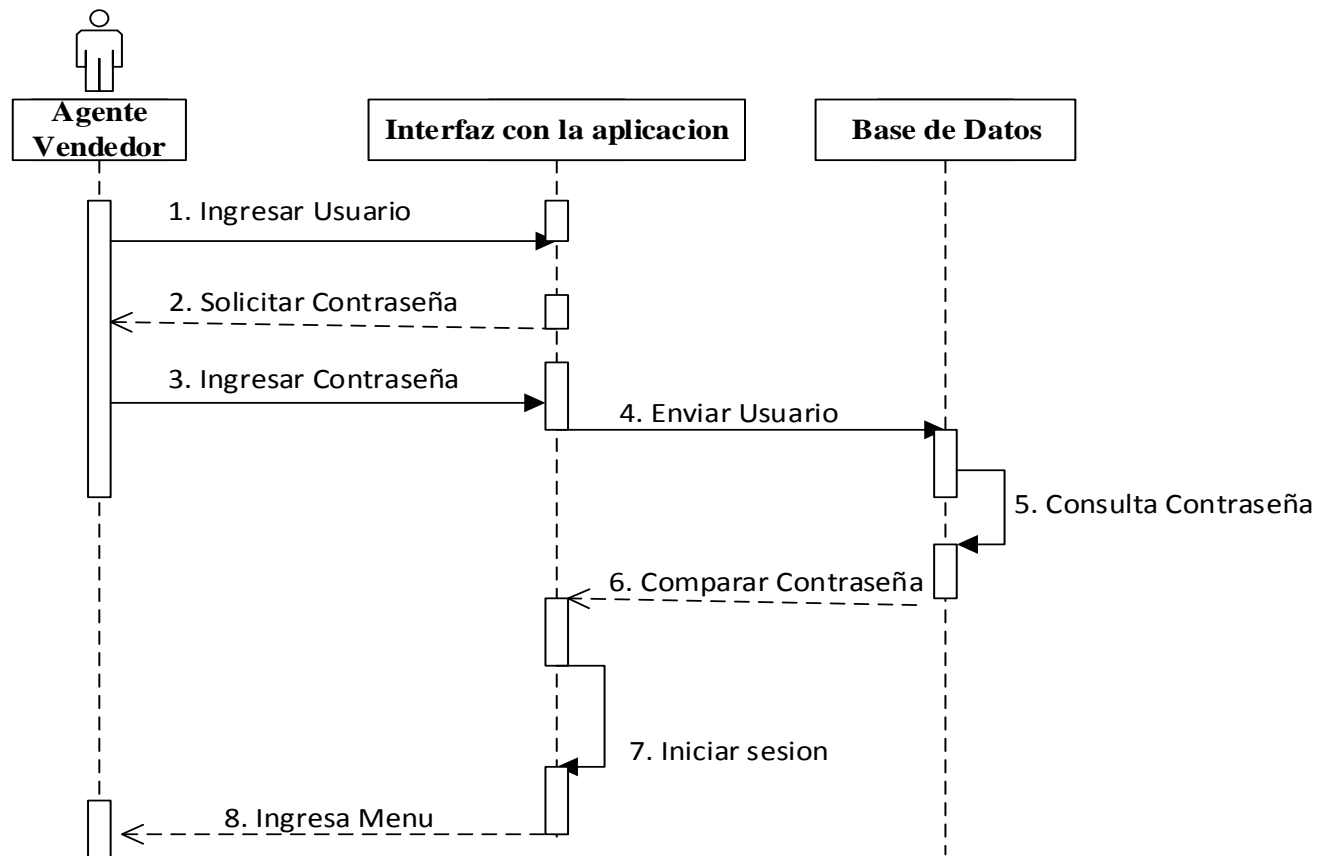


Ilustración 32. Diagrama de Secuencia, Inicio de Sesión

4.2.3.2 Diagrama de Secuencia, Enviar Pedido

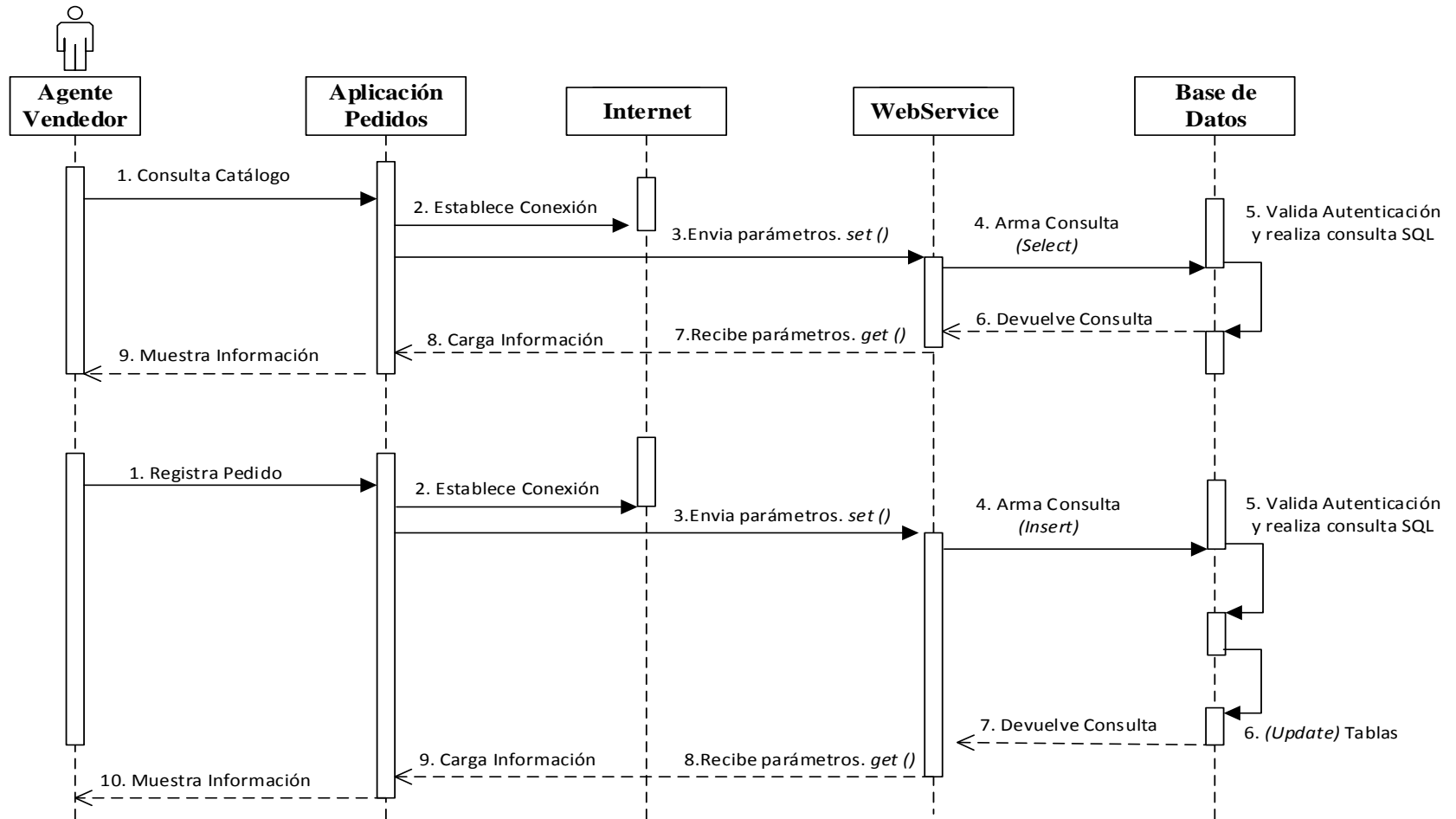


Ilustración 33. Diagrama de Secuencia, Enviar Pedido

4.2.3.3 Diagrama de Secuencias, Interacción con la Intranet

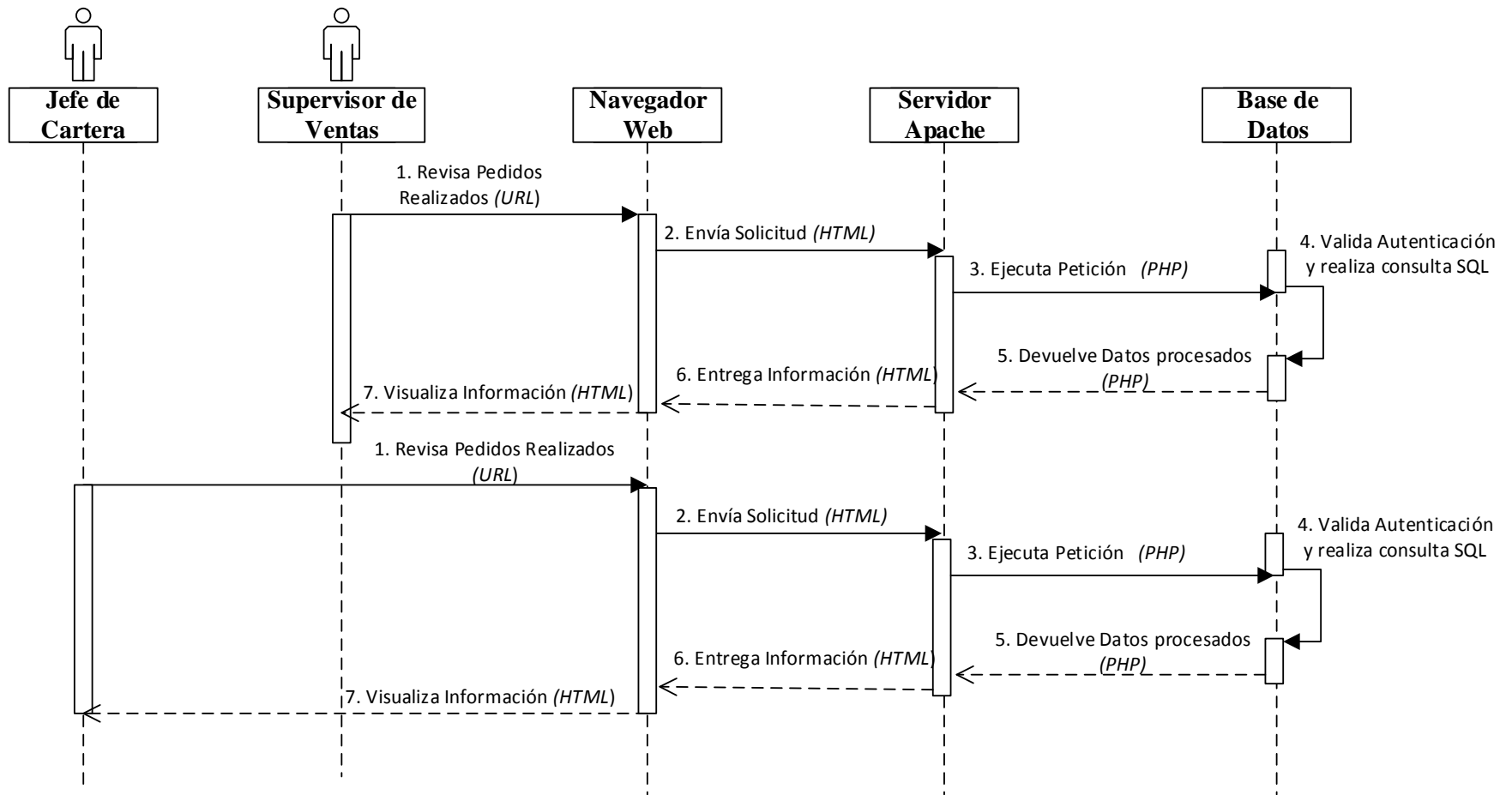


Ilustración 34. Diagrama de Secuencia, Interacción con la Intranet.

4.2.4 Diagrama Arquitectónico.

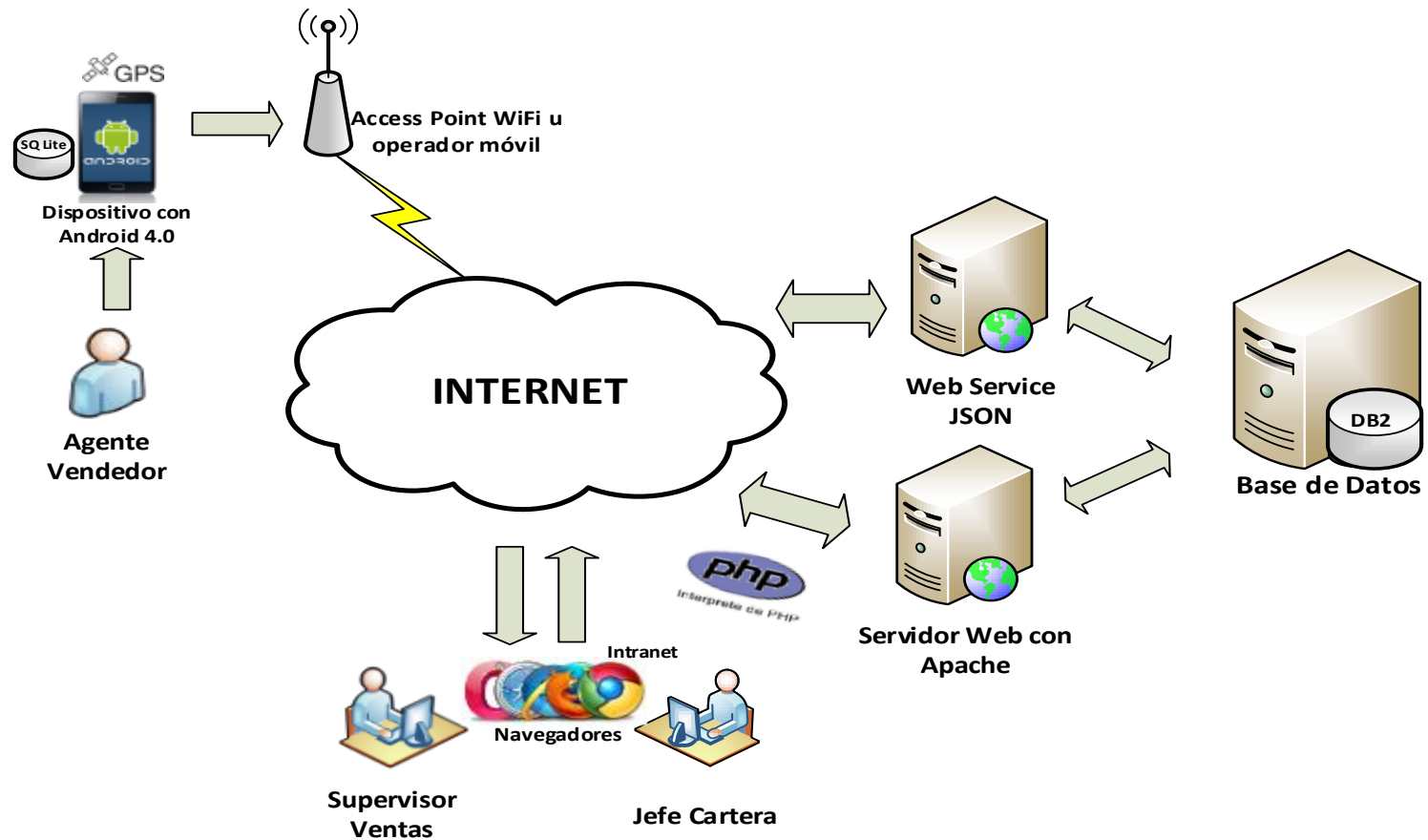


Ilustración 35: Diagrama Arquitectónico (Elaboración Propia)

4.2.5 Diagrama de Navegación.

El Diagrama de Navegación indica o muestra el orden de relación de las pantallas o componentes del software, y sirve para la comprensión del orden de presentación de las pantallas o *Activities*⁴⁴ con los contenidos de la aplicación. La siguiente imagen muestra de una manera gráfica la relación entre las diferentes pantallas.

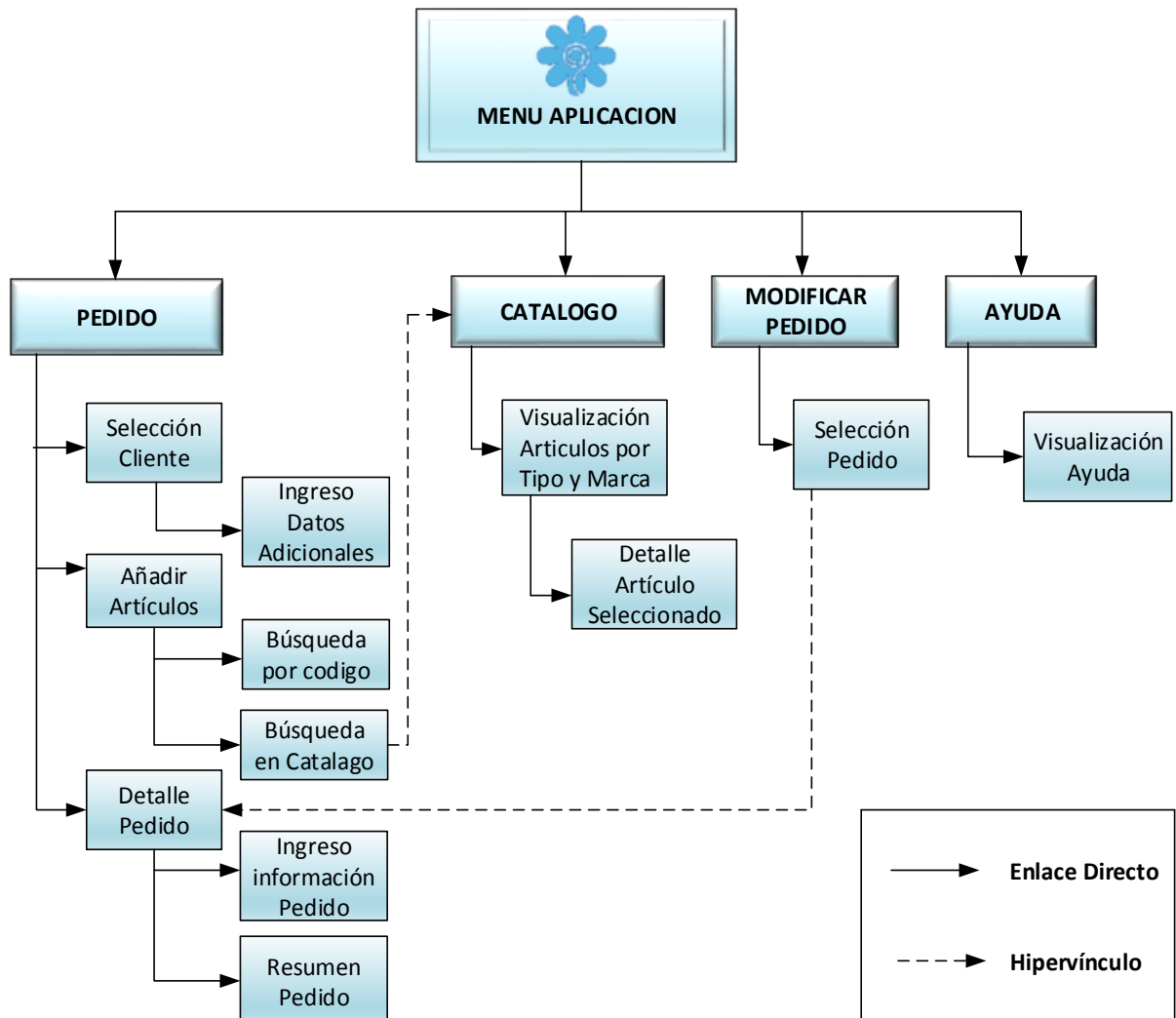


Ilustración 36. Diagrama de Navegación (Elaboración Propia)

⁴⁴ **Activity:** Es una ventana que contiene la interface de usuario de nuestra aplicación, el objetivo principal de una activity es interactuar con el usuario.

4.2.6 Diseño de los bocetos de las pantallas del sistema (Prototipos)

Para la parte de diseño de prototipos de nuestro proyecto cabe recalcar que la realizaremos gracias a la ayuda de una herramienta Web (Online) llamada *Ninja Mock*. Que la podemos utilizar ingresando a la página: <http://ninjamock.com/> Basta con registrarnos para empezar a utilizarla.

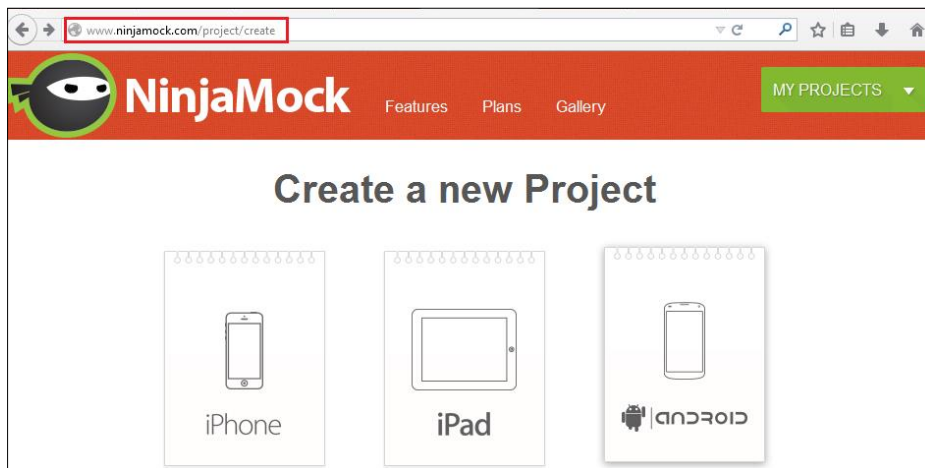


Ilustración 37: Página Oficial NinjaMock para Prototipos.

Al ser una herramienta de nivel intermedio y no muy avanzada sirve para la creación de bocetos para dispositivos móviles con sistema Operativo como: iOS, Android y Windows Phone. Los elementos que se agrega a la aplicación ofrecen un acabado de dibujo a mano alzada. Su principal ventaja es que es gratuita. La interacción de sus prototipos no va más allá de enlaces entre pantallas. Para ideas básicas, para un diseño rápido y para un primer boceto, resulta muy eficiente para lo que buscamos en el proyecto.

4.2.6.1 Boceto de la Pantalla de Inicio de Sesión.

Nuestro primer boceto va ser la pantalla de Logueo o inicio de sesión la cual va ser muy simple, ya que solo requiere de una ventana o Activity.



Ilustración 38: Prototipo Pantalla de Inicio de Sesión.

4.2.6.2 Boceto de la Pantalla del Menú.

Continuando con los Bocetos, Vamos a realizar la pantalla del Menú, esta será muy importante ya que siempre estará visible casi todo el tiempo para el usuario, ya que en el accederemos a las diferentes opciones que tiene nuestra aplicación por lo que tiene que ser muy intuitiva y muy agradable a la vista.



Ilustración 39.

Prototipo Pantalla Menú Aplicación

4.2.6.3 Boceto de la Pantalla del Catálogo

Vamos a ver el Prototipo para la pantalla del Catálogo, este contendrá todos los artículos en Venta por lo que tendrá que contener una especie de tabla donde pueda filtrar los artículos ya sea por Tipo y Marca. Podría ser así más o menos:

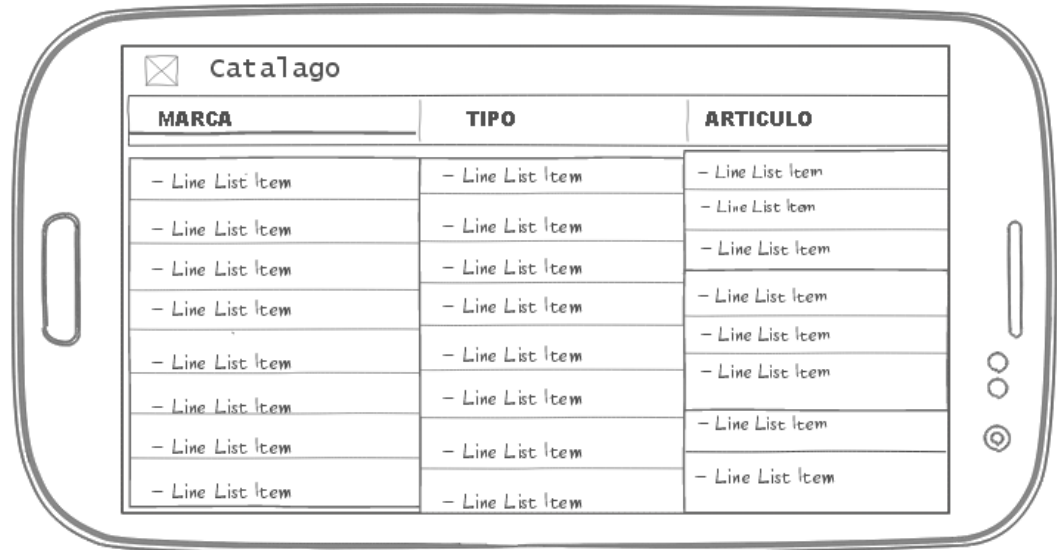


Ilustración 40. Prototipo Pantalla Catálogo

4.2.6.4 Boceto de la Pantalla de Pedido

Para esta opción hay que seguir varios pasos que se detallaran a continuación:

1. Aparecerá una pantalla o *Activity* donde se pida Seleccionar Cliente y posteriormente un Dialogo confirmando su selección. Ver **Ilustración 44**).
2. Luego solicitará que se ingrese los artículos ya sea por Código (Ver **Ilustración 45**) o directamente del Catálogo (Ver **Ilustración 43**).

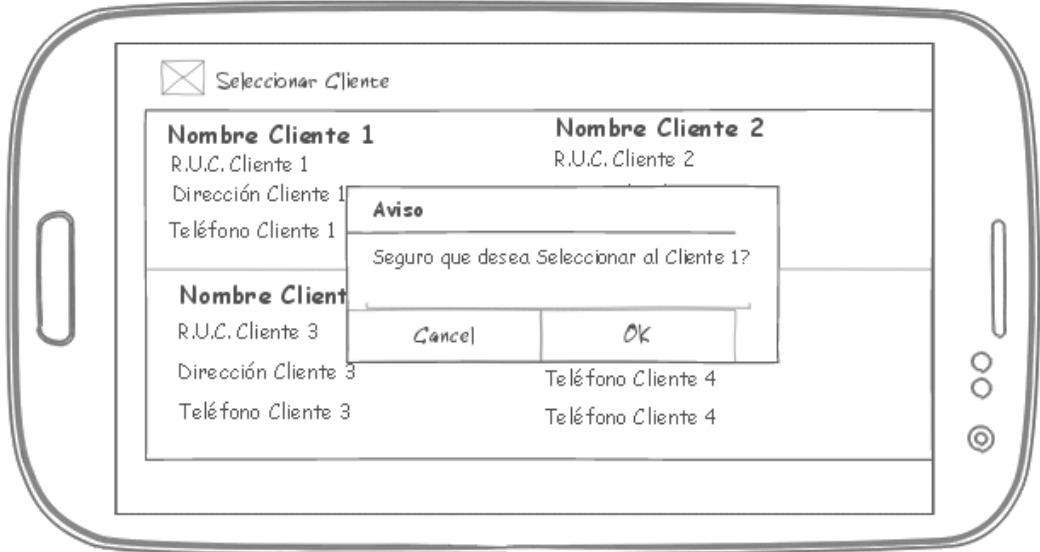


Ilustración 41. Prototipo Pantalla de Pedido

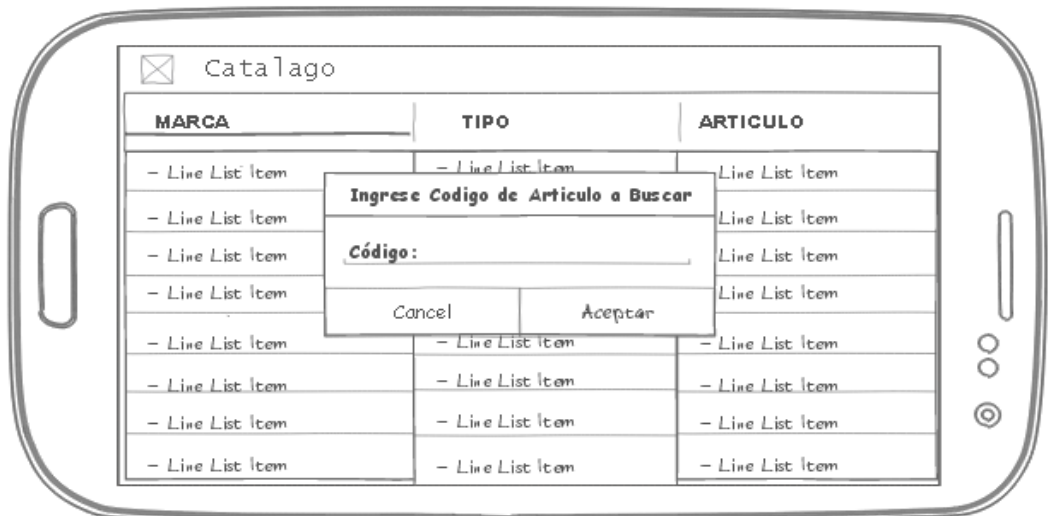


Ilustración 42. Prototipo, Pantalla Buscar por Código

3. Posteriormente nos pedirá Detallar el pedido para lo cual se seleccionará la Dirección de entrega, Adelanto y Forma de Pago. (Ver *Ilustración 46.*)
4. Y finalmente podemos ver el Pedido Generado Detallado antes de enviar a que se grabe. (Ver *Ilustración 47.*)

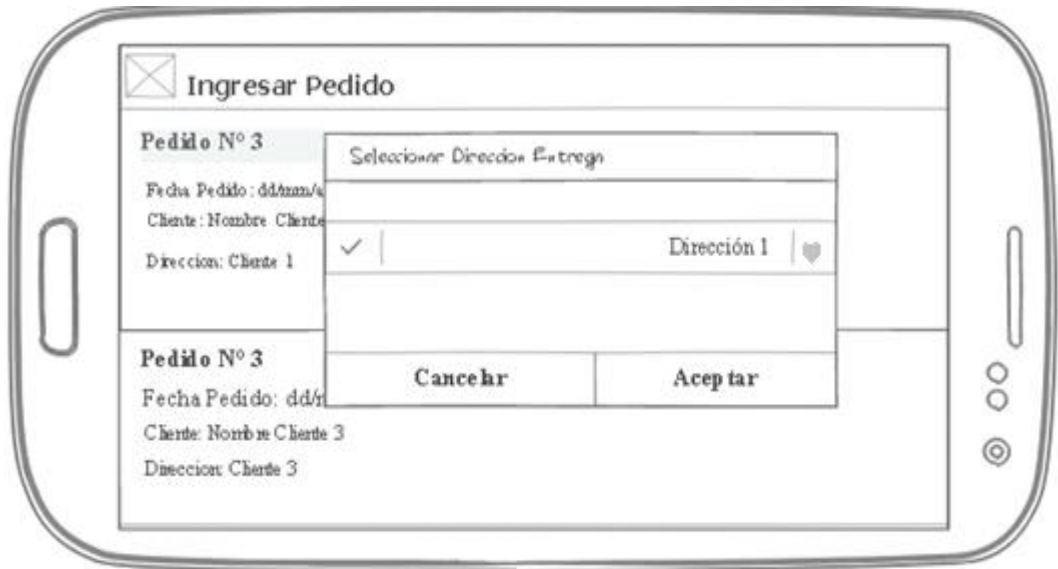


Ilustración 43. Prototipo Pantalla Ingresar Pedido

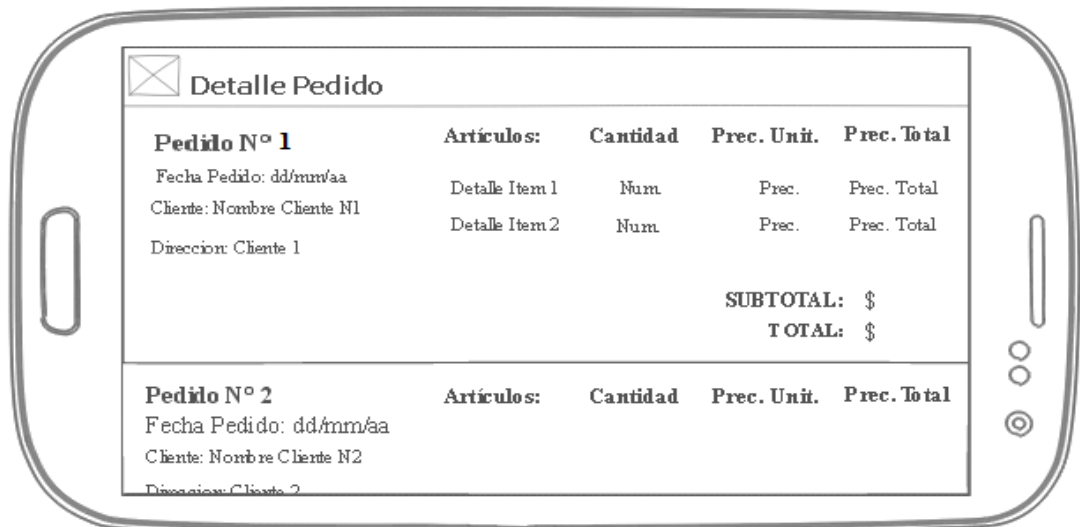


Ilustración 44. Prototipo Pantalla Detalle Pedido.

4.2.6.5 Boceto de la Pantalla de Ayuda

Para esta pantalla simplemente visualizaremos texto acompañado de imágenes (código *HTML*), Con capturas de la Aplicación lo que ayudará al usuario para verificar de qué se trata cada módulo de la aplicación.

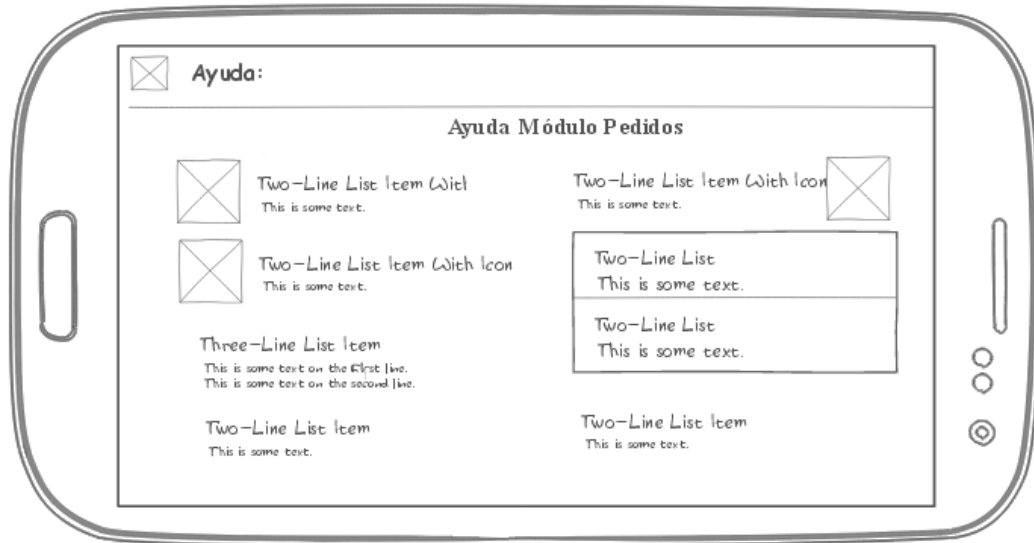


Ilustración 45. Prototipo, Pantalla Ayuda.

4.3 Fase de Codificación

Para esta parte vamos a ver los requerimientos de desarrollo que se necesitan para crear un proyecto y empezar a programar en Android, así como una breve descripción del código de las clases más relevante dentro de la aplicación como son:

- La Clase Java MainActivity (Clase Principal, llamada diferentes objetos)
- La Clase Java Login (Conexión y Autenticación en la Base de Datos mediante el Web Service)
- El Archivo XML Activity_Menu (Crear interfaz para el usuario)

4.3.1 Requerimientos de desarrollo

En el Capítulo 3 (Sección 3.3) (**Cfr. Supra**) Ver Tabla 3, se revisó la selección del entorno de Desarrollo pero a manera de resumen vamos a detallar cada una de las herramientas además de servicios o aplicaciones que se necesita para empezar a desarrollar un proyecto en Android, y para la publicación de la Web.

- Entorno de Desarrollo para Aplicación.
 - JDK Java.
 - Eclipse (IDE)
 - ADT (Plugins)
 - Android SDK (Librerías y Paquetes)
 - SQLite (Base de Datos)
 - ADV (Emulador Android)
 - Webservice (JSON)
 - IIS (Internet Information Server) publicar Webservice

- Entorno de Desarrollo para la Web
 - Visual Studio 2012. (.NET)
 - Apache Tomcat (Servidor PHP)
 - API Google Maps para .NET

4.3.2 Desarrollo del módulo de Gestión de Pedidos.

En este apartado vamos a ver cómo un breve detalle del código básicamente de algunas llamadas o líneas donde sea importante acotar de las clases más relevantes de nuestro proyecto:

4.3.2.1 Descripción La Clase Java MainActivity

Ver **Anexo N° 1.**

4.3.2.2 Descripción de la Clase Java Login.

Ver **Anexo N° 2.**

4.3.2.3 Archivo XML Activity_Menu.xml (Menú Principal)

Ver **Anexo N° 3.**

4.3.3 Desarrollo del módulo de Control de Rutas de los vendedores.

4.3.3.1 Obtener una API Key de Google Maps

En este apartado solo nos limitamos a ver cómo obtener una *API Key* pues lo exige Google para poder utilizar su API y se lo puede realizar desde la siguiente página web:

<https://developers.google.com/maps/signup?csw=1>

Debemos tener en cuenta que esta clave depende del dominio donde vayamos a utilizarla, por lo que por cada Web donde la utilicemos tendremos que obtener la *Key* apropiada.

4.3.3.2 Agregar un Mapa de Google Maps en ASP.net

Como primer paso procedemos a descargarnos el fichero **GMaps.dll** desde el siguiente enlace: <http://googlemaps.subgurim.net/descargar.aspx>.

Este archivo “Google Map” (GMaps) es un control de usuario ASP.NET para con escasas líneas poder agregar mapas de Google o Google Map a nuestras aplicaciones WEB.

Una vez descargado hay que copiar dicho archivo en la carpeta *bin* de nuestro proyecto, pasaremos a crear nuestra página web a la que le adicionaremos el mapa o Google Map. El siguiente código muestra como agregar GMap en ASPX:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div style="width:660px; height:560px;">
      <cc:GMap ID="GMap1" runat="server" />
    </div>
  </form>
</body>
</html>
```

Posteriormente nos queda obtener nuestro API Key de Google el cual ya lo vimos en el apartado anterior (**Cfr. Supra**). Finalmente veremos en donde ingresar el API Key en ASP.net, el modo más sencillo donde ubicar la clave en el archivo *web.config*.

```
...
<appSettings>
<add key="googlemaps.subgurim.net"value="tuAPI_Key" />
</appSettings>
...
```

De este modo, nos toca controlar el mapa según nuestras necesidades, en nuestro caso en el control de rutas.

4.4 Fase de Pruebas

Una vez finalizada la fase de codificación y para la comprobación del correcto funcionamiento de la aplicación en cada uno de sus Menús, se realizaran las pruebas respectivas tomando en cuenta los aspectos siguientes:

- Para la primera Prueba Funcional vamos a elaborar una pequeña síntesis sobre las “historias de usuarios” al momento de utilizar la aplicación para lo cual se ha tomado en cuenta una Autenticación Correcta y una autenticación Incorrecta de usuario al momento de iniciar sesión.
- En segundo lugar vamos a probar la aplicación en una Tablet que cumple los requisitos mínimos de nuestra aplicación. Lo realizaremos con empleados de la División Perfumería de Almacenes Juan Eljuri y con datos reales para ver los tiempos de respuesta.

4.4.1 Elaboración y Ejecución del plan de Pruebas

Esta es una Pruebas Funcional relacionada a las historias de usuario referentes a los Usuarios autenticados.

Historia de usuario: Autenticación correcta de usuarios.	
Descripción:	El usuario una vez que ha ejecutado la aplicación previamente instalada le aparecerá una la Ventana de “Inicio de Sesión”, donde ingresar su Usuario y la Contraseña, internamente se verificará si estos datos son los correctos y posteriormente ingresará a la aplicación.
Condiciones de Ejecución	✓ El usuario deberá estar registrado en el sistema.
Entrada	<ul style="list-style-type: none"> ✓ El usuario ingresará su usuario y la contraseña y presionará el botón Ingresar. ✓ Internamente se verificará si el usuario está registrado e ingresará a realizar sus consultas. ✓ El proceso de autenticación se considera finalizado.
Resultado Esperado	✓ Tras el ingreso de usuario y contraseña, si el procesado ha sido correcto en la base de datos se registrarán datos de su ingreso y el usuario podrá utilizar la aplicación que está cargada de información de ese usuario específico.
Evaluación de la prueba	✓ Prueba satisfactoria

Tabla 12. Historia de usuario: Autenticación correcta

Para la segunda prueba Funcional, se lo realizara en una Tablet (*Ver Ilustración 51*) de las siguientes características:

- Samsung Galaxy Tab 3 T211
- Dual Core 1.2 GHz de 1GB de RAM
- Sistema Operativo *Android 4.1.2 Jelly Bean*

Los usuarios de la aplicación pudieron constatar del funcionamiento de cada una de las funciones programadas lo que permitió verificar el cumplimiento de los requisitos planteados al inicio de este proyecto. El resultado de estas pruebas lo veremos en el apartado 4.4.2

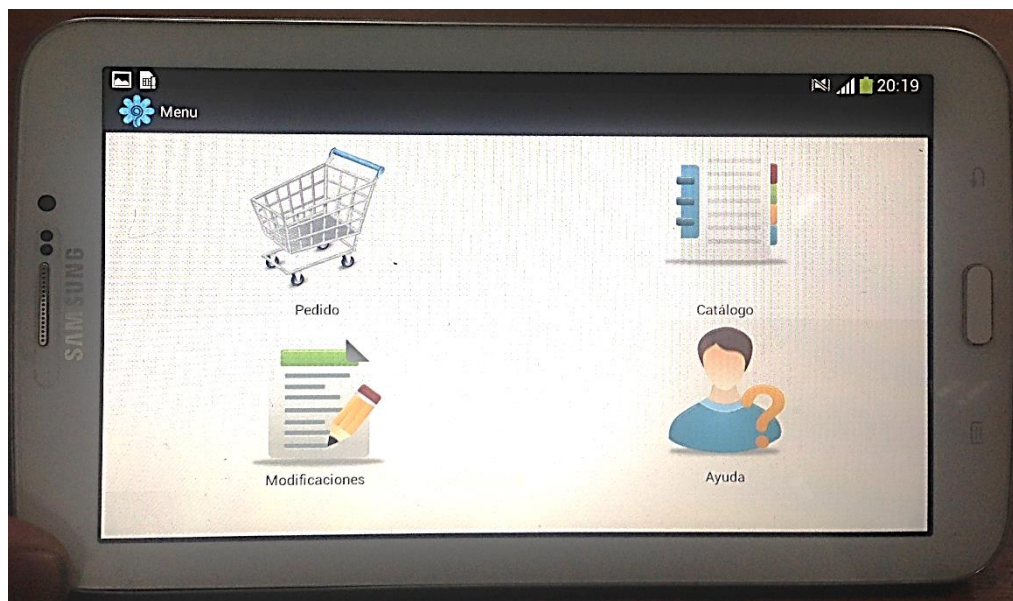


Ilustración 46. Pruebas Realizadas en una Samsung Galaxy Tab 3 T211

4.4.2 Análisis de resultados

Este análisis es en base a la segunda prueba funcional lo detallamos en la *Tabla 13*.

N °	Actividad	Responsable	Observaciones
1.	Inicio de Sesión	Agente Vendedor	El Logueo funciona correctamente, la aplicación se demora un tiempo considerable ya que tiene que cargar toda la información en la base de datos local.
2.	Visualizar Catálogo	Agente Vendedor	La aplicación no se demora ya que los datos se encuentran almacenados localmente, Se sugiere algunos cambios
3.	Crear Pedido	Agente Vendedor	Se observa una fluidez al cargarse pocos datos pero cuando se carga muchos artículos la

			aplicación tiende a demorarse. El usuario manifiesta que los pasos son muy intuitivos a la hora de crear un pedido.
4.	Modificar Pedido	Agente Vendedor	De igual manera los datos se cargan rápidamente y los pasos son los mismos que Crear el pedido, el usuario no tuvo mayores dificultades.
5.	Ayuda	Agente Vendedor	La ayuda es de lectura por lo que no hubo mayores inconvenientes. Se sugieren agregar algunas funcionalidades.

Tabla 13. Análisis de Resultados de La Primera Prueba Funcional

Las pruebas por lo general indican cuando el software está terminado. Si las pruebas funcionan la codificación ha terminado.

4.5 Documentación.

4.5.1 Manual de la Aplicación de Pedidos

Ver **Anexo N° 4**

4.5.2 Manual de Uso de la Intranet, Modulo Revisión de Rutas.

Ver **Anexo N° 5**

CONCLUSIONES

- En lo referente al desarrollo de aplicaciones en Android al ser un sistema operativo libre (open source) es decir nos brinda la libertad a los usuarios de adquirirlo y usarlo, podemos concluir que nos fue mucho más fácil la implementación en este sistema operativo ya que nos permite desarrollar aplicaciones con herramientas gratuitas y potentes como *Eclipse* y el *SDK* de Android, también de alguna manera se nos facilitó ya que Android emplea lenguaje *Java*, con el cual ya estamos familiarizados durante el transcurso de nuestra carrera.
- El uso del sistema operativo Android para nuestra aplicación fue de gran ayuda para el usuario final ya que puede encontrar mucha más variedad de dispositivos en el mercado como *Tablets* y *Smartphones*, con los cuales nuestra aplicación es compatible, dando lugar que el usuario tenga más opciones a la hora de comprar o invertir en tecnología.
- El uso de una Metodología para el desarrollo de la aplicación móvil específicamente la metodología *Extreme Programming* o *XP*, fue fundamental ya que gracias a ella podemos elaborar software de calidad, ya que se fue cumpliendo con cada una de sus etapas permitiendo que podamos mostrar algunas versiones previas al usuario lo que fue de gran ayuda porque se van realizando los correctivos correspondientes para dejar la aplicación totalmente funcional antes de la versión final.
- En base a los Objetivos planteados podemos decir que se fueron cumpliendo con el avance del desarrollo de la aplicación móvil para Android, siendo esta aplicación para el uso de los empleados de la División perfumería de Almacenes Juan Eljuri, específicamente a sus Agentes Vendedores, pudiendo constatar que este usuario ya cuentan con información actualizada en línea de sus productos y no dependen que su Supervisor les haga llegar esta información, eliminando ciertas tareas manuales y la impresión de hojas, y por lo tanto reduciendo el tiempo de entrega del producto al cliente.

RECOMENDACIONES

- Se recomienda el uso de alguna *Metodología Ágil* para el desarrollo de una aplicación móvil, ya que gracias a la metodología podemos ser más disciplinados a la hora de elaborar o construir algún proyecto de software, las metodologías ágiles permiten construir un buen equipo de trabajo y nos permite configurar nuestro propio entorno de desarrollo en base a las necesidades además la planificación es flexible permitiendo responder a cambios repentinos en el proyecto.
- En cuanto al uso del *WebService JSON*, para cargar datos en la aplicación pudimos constatar que con unas decenas de miles de registros este tiende a no responder adecuadamente, esto debido al procesamiento de mucha información que toma de la base de datos, para un óptimo funcionamiento se recomienda tener normalizada la Base de Datos o en su caso filtrar la información más relevante en alguna consulta o Vista en nuestro caso al cargar los ítems o productos tendía a colgarse por lo que se tuvo que filtrar información no necesaria por ejemplo se tuvo que obviar los productos que no tenían Registro Sanitario ya que estos no se podían vender.
- Para la emulación se recomienda usar un dispositivo Real ya sea una Tablet o un Smartphone ya que el *ADV* o *Emulador de Android* carece de algunas herramientas como *GPS*, brújula, etc. y sobre todo evitamos consumir recursos de procesamiento de nuestro computador y poder probar la verdadera funcionalidad de la Aplicación en un entorno real.
- Se recomienda seguir modificando o mejorando esta aplicación para que sea más completa ya que al ser una versión inicial carece de algunas funcionalidades como por ejemplo fotos en los Catálogos, la opción de guardar una firma del cliente que ha sido visitado o disponer de un chat en línea para los vendedores que usen la aplicación, etc. También se podría publicarlo en el *PlayStore* para facilitar que cualquier vendedor pudiera descargarse e instalar la aplicación.

- Por último en lo relacionado a la Institución, se recomienda a la Universidad dar el correcto seguimiento a sus egresados ya que por diversos motivos están fuera por mucho tiempo y no han podido graduarse, ya que muchas de las veces no es culpa del estudiante sino la necesidad de empezar a laborar que luego termina siendo difícil poder concluir la carrera. Se le debería dar más opciones en cuanto a temas o sugerencias de alguna institución, también debería haber más facilidades en los horarios para la apertura de cursos o seminarios para poder mejorar el perfil profesional y estar realmente preparado para el mundo laboral.

BIBLIOGRAFÍA

- [1] Diario EL COMERCIO, «159 Empresas son parte del Grupo Eljuri,» *Politica*, Octubre 2010.
- [2] P. RRHH, Interviewee, *Misión y Vision de Almacenes Juan ELjuri Cia. Ltda.*. [Entrevista]. Noviembre 2014.
- [3] Supervisor Ventas Perfumeria Almacenes Juan Eljuri Cia Ltda., Interviewee, *Procesos actuales en la Gestión de pedidos*. [Entrevista]. Octubre 2014.
- [4] Superintendencia de Telecomunicaciones, «Evolución de la Telefonía Movil en el Ecuador,» *Revista Institucional SUPERTEL*, nº 16, pp. 4, 31, 32, 2012.
- [5] R. R. Vique, *Métodos para el desarrollo de aplicaciones moviles*, Cataluña, 2012.
- [6] Wikipedia, «Sistema operativo móvil,» [En línea]. Available: http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil. [Último acceso: Noviembre 2014].
- [7] P. World, «Sistemas Operativos Móviles: Comunicación en tiempo real,» [En línea]. Available: <http://www.pcworld.com.mx/Articulos/20734.htm>. [Último acceso: Noviembre 2014].
- [8] T. El Pais, «España abraza los Android y desprecia los Windows,» Octubre 2013. [En línea]. Available: http://tecnologia.elpais.com/tecnologia/2013/10/01/actualidad/1380620296_338056.html.
- [9] M. Zona, «Mercado dominado por iOS y Android mientras Windows Phone retrocede,» [En línea]. Available: <http://www.movilzona.es/2014/08/15/el-964-del-mercado-dominado-por-ios-y-android-mientras-windows-phone/>.
- [10] Open Handset Alliance, 2014. [En línea]. Available: <http://www.openhandsetalliance.com>.
- [11] C. J. F. A. W. A. R. P. Blanco Paco, «Metodología de desarrollo ágil para sistemas móviles, Introducción al desarrollo con Android y el iPhone,» 2012. [En línea]. Available: http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf.

- [12] Wikipedia, «Android,» [En línea]. Available: http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil. [Último acceso: 2014].
- [13] Á. J. Vico, «Blog La columna 80, Arquitectura de Android,» 2011. [En línea]. Available: <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>.
- [14] Android, «Developers Android,» [En línea]. Available: <http://developer.android.com/intl/es/sdk/index.html>. [Último acceso: 2014].
- [15] Universidad Carlos III de Madrid, «Software de Comunicaciones, Programación en dispositivos móviles portables,» 2012. [En línea]. Available: <https://sites.google.com/site/swcuc3m/home/android/generalidades/dalvikvm-1>.
- [16] Wikipedia, «SQLite,» 2014. [En línea]. Available: <http://es.wikipedia.org/wiki/SQLite>.
- [17] Wikipedia, «Historial de versiones de Android,» 2014. [En línea]. Available: http://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android.
- [18] Wikipedia, «Lenguaje unificado de modelado,» [En línea]. Available: http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado. [Último acceso: 2014].
- [19] Wikipedia, «Casos de uso,» [En línea]. Available: http://es.wikipedia.org/wiki/Caso_de_uso. [Último acceso: 2014].
- [20] Wikipedia, «Proceso Unificado Racional,» [En línea]. Available: http://es.wikipedia.org/wiki/Proceso_Unificado_de_Racional.
- [21] E. Programming, «Extreme Programming: A gentle introduction,» [En línea]. Available: <http://www.extremeprogramming.org/>. [Último acceso: 01 2015].
- [22] K. B. w. C. Andres, Extreme Programming Explained, 2004.
- [23] Wikipedia, «Desarrollo de programas para Android,» 02 2015. [En línea]. Available: http://es.wikipedia.org/wiki/Desarrollo_de_programas_para_Android.
- [24] Wikipedia, «Servicio Web,» 2014. [En línea]. Available: http://es.wikipedia.org/wiki/Servicio_web.
- [25] Json.org, «Introducción a JSON,» [En línea]. Available: <http://json.org/json-es.html>.
- [26] Wikipedia, «Internet Information Services,» 2015. [En línea]. Available: http://es.wikipedia.org/wiki/Internet_Information_Services.

- [27] Wikipedia, «Eclipse (Software),» 01 2015. [En línea]. Available: http://es.wikipedia.org/wiki/Eclipse_%28software%29.
- [28] M. Libre, «Mercado Libre,» [En línea]. Available: <http://listado.mercadolibre.com.ec/tablets-3g#D%5BA:tablets-3g%5D>. [Último acceso: 03 2015].
- [29] El Androide Libre, «Desarrollando en Android #2 Google Maps API,» [En línea]. Available: <http://www.elandroidelibre.com/2013/10/desarrollando-en-android-2-google-maps-api.html>.
- [30] AndroidZone, «Historia de Android: La Evolución a lo largo de sus versiones,» 2013. [En línea]. Available: <http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/>.

ANEXOS

Anexo N° 1.

Fragmento de código clase MainActivity.java

Fragmento de código clase MainActivity.java

```

public class MainActivity extends Activity {
    private AgregarDatos info;
    private DataBase db;
    private FuncionesVarias fv;
    String Resultado;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //CAMBIA MODO DE THREAD A PERMITIR TODO PARA EL USO EXCLUSIVO DEL WEBSV
        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        //LLAMA AL ACTIVITY PRINCIPAL
        setContentView(R.layout.activity_main);
        //OBTENER LOS CONTROLES PRINCIPALES
        final Button btnLogin = (Button) findViewById(R.id.btnSignIn);
        final EditText etxUsername = (EditText) findViewById(R.id.edtUserName);
        final EditText etxPassword = (EditText) findViewById(R.id.edtPassword);
        //INICIALIZA LA BASE DE DATOS
        fv=new FuncionesVarias();
        db= new DataBase(this,"PerfuemriaDB",null,fv.VersionDB());
        info=new AgregarDatos(db);
        info.AbrirConexion();
        btnLogin.setOnClickListener(new View.OnClickListener() {
            public void onClick(final View v) {
                //CREA UN OBJETO DE CLASE LOGIN
                LOGIN ObjLogin=new LOGIN(v.getContext());
                //CREA UN OBJETO PARA VERIFICAR CONEXION
                CONEXION ObjConexion=new CONEXION();
                if(ObjConexion.estaConectado(v.getContext())==true)
                {
                    //TIENE CONEXION A INTERNET
                    StringEstLogin=ObjLogin.loginOnline(etxUsername.getText().toString().toUpperCase(), etxPassword.getText().toString().toUpperCase(), "", "");
                    final DATOS ObjDatos=new DATOS(v.getContext());
                    final ProgressDialog pd = ProgressDialog.show(v.getContext(), "", "Guardando
                    datos", true)

```

Descripción: Esta clase se encarga de:

- Establecer la comunicación con el *Webservice*.
- Llamar al Activity o Ventana principal.
- Obtener los Controles Principales.
- Inicializar Base de Datos (Borrar el Contenido cuando se ingresa, y cargarla de datos nuevamente).
- Validar la conexión del usuario con la aplicación, Ingresará si es un usuario creado en la base de datos y si es que posee conexión a internet. Caso contrario no podrá ingresar a la aplicación.

Tabla 14. Fragmento de código clase MainActivity.java

Anexo N° 2.

Fragmento código Clase Login.java

Primera Parte código Clase Login.java

```
public class LOGIN extends SQLiteOpenHelper {
    private static final String
    SOAP_ACTION = "http://tempuri.org/Cselect_CATWS";
    private static final String METHOD_NAME = "Cselect_CATWS";
    private static final String NAMESPACE = "http://tempuri.org/";
    private static final String URL =
    "http://200.63.208.40/WebService1.asmx"; //URL Webservice que vamos a
    consumir (Dirección IP Publica)
    public String LINEA;
    public String AGENTE;
    public String USUARIO;
    public LOGIN(Context ctx){
    super(ctx,"LOGIN", null, 1);
        LINEA="";
        AGENTE="";
        String sql;
        sql="SELECT * FROM LOGIN";
        Cursor c =this.getReadableDatabase().rawQuery(sql, null);
        if (c.getCount()==1){
        c.moveToFirst();
        String LOGPASS=c.getString(c.getColumnIndex("LOGPAS"));
            LINEA=c.getString(c.getColumnIndex("LOGLIN"));
            AGENTE=c.getString(c.getColumnIndex("LOGAGE"));
            USUARIO=c.getString(c.getColumnIndex("LOGUSU"));}}

    @Override
    public void onCreate(SQLiteDatabase db) {
    String query ="CREATE TABLE LOGIN (LOGUSU TEXT," +
    "LOGPAS TEXT," +
    "LOGLIN TEXT," +
    "LOGAGE TEXT);";
    db.execSQL(query);}
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
    newVersion) {
        db.execSQL("DROP TABLE IF EXISTS LOGIN");
        onCreate(db);}
    public void Insertar(String pLOGUSU,
        String pLOGPAS,
        String pLOGLIN,
        String pLOGAGE){
        ContentValues valores = new ContentValues();
        valores.put("LOGUSU", pLOGUSU);
        valores.put("LOGPAS", pLOGPAS);
        valores.put("LOGLIN", pLOGLIN);
        valores.put("LOGAGE", pLOGAGE);
        this.getReadableDatabase().insert("LOGIN", null,
valores);}
```

Descripción: Esta clase se encarga básicamente del logueo o inicio de sesión por lo que trabajara directamente con el Webservice que nos devolverá información de la Base de

Datos para lo cual realiza las siguientes tareas:

- Crear cadenas o *String* con *URL* para realizar la conexión con el servicio web y el nombre del método web concreto que vamos a ejecutar
- Crea una tabla temporal (LOGIN) donde almacena las variables o datos del usuario que inicia sesión, para lo cual recibe como parámetros
- Borrar datos de tabla si es que contiene datos al momento del Logueo (Manejo de Versiones en *SQLite*)

Tabla 15. Primera parte de Código de la Clase Login.java

Segunda Parte código Clase Login.java

```
public String manejoLogin(String pLOGUSU,
    String pLOGPAS,
    String pLOGLIN,
    String pLOGAGE) {
    SoapObject Request = new
    SoapObject(NAMESPACE, METHOD_NAME);
        String Resultado = "";
        Request.addProperty("Id", pLOGUSU);
        Request.addProperty("Paswd", pLOGPAS);
        Request.addProperty("Bds", "DB2");
        Request.addProperty("Server", "Dir ip serv.");
        Request.addProperty("MsgErr", Resultado);
        Request.addProperty("PBANSEL", 8);
        soapEnvelope.dotNet = true;
        soapEnvelope.setOutputSoapObject(Request);
    AndroidHttpTransport aht = new AndroidHttpTransport(URL);
        try{this.getWritableDatabase();
        aht.call(SOAP_ACTION, soapEnvelope);
        SoapPrimitive
mensaje=(SoapPrimitive) soapEnvelope.getResponse();
        Resultado=mensaje.toString();
        catch(Exception e){
        Resultado=e.toString();//ERROR DE CONEXION}
        return Resultado;}

public String loginOnline(String pLOGUSU,
    String pLOGPAS,
    String pLOGLIN,
    String pLOGAGE) {String
resultado=manejoLogin(pLOGUSU, pLOGPAS, pLOGLIN, pLOGAGE);
    if (resultado.contains("BYPASS")) {
        resultado="CORRECTO";//GRABA REGISTRO CORRECTO
onUpgrade(this.getReadableDatabase(), 1, 1);
        Insertar(pLOGUSU, pLOGPAS, LINEA, AGENTE);
    }
    else if(resultado.contains("CWBSY0002")) {
        resultado="CLAVE INCORRECTA";
    }
    else if(resultado.contains("CWBSY0011")) {
        resultado="USUARIO DESHABILITADO";//BORRA DATOS}
    else if(resultado.contains("CWBSY0001")) {
        resultado="USUARIO NO EXISTE";
    }
}
```

```

else{resultado=resultado + ", ERROR CIERRE A APLICACION
COMPLETAMENTE Y VUELVA A INTENTARLO, O INTENTELO OFFLINE.";}
return resultado;}
public String loginOffline(String pLOGUSU,
String pLOGPAS, String pLOGLIN, String pLOGAGE){
String resultado="";
String sql;
sql="SELECT * FROM LOGIN WHERE LOGUSU='"+pLOGUSU.toUpperCase()+"'";
Cursor c =this.getReadableDatabase().rawQuery(sql, null);
if (c.getCount()==0){
resultado="USUARIO NO EXISTE, NECESARIO INGRESAR ONLINE";}
if (c.getCount()==1){c.moveToFirst();
String LOGPASS=c.getString(c.getColumnIndex("LOGPAS"));
if(LOGPASS.toUpperCase().equals(pLOGPAS.toUpperCase())){
resultado="CORRECTO";
LINEA=c.getString(c.getColumnIndex("LOGLIN"));
AGENTE=c.getString(c.getColumnIndex("LOGAGE"));}else{
resultado="CLAVE INCORRECTA";}

```

Descripción: En la segunda parte del código se verá el Método manejoLogin que será el encargado de:


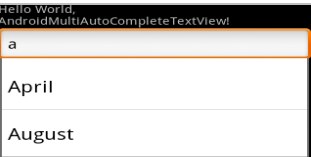

- Realizar las peticiones SOAP al Servicio Web teniendo que asociar los parámetros de entrada mediante el método *addProperty()*. Para lo cual pasamos como parámetros: el ID, nombre de la base de datos, Servidor, Dirección IP, y más datos que necesitemos.
- Controlar Logueo de modo ON Line (con conexión a internet) para lo cual se realizará la llamada del servicio Web con el método *androidHttpTransport.call()* una vez obtenido los datos que me devuelve el WebService, se compara con la ingresada y mostrando los diferentes mensajes de error.
- Manejo del Logueo modo OFF Line (Sin necesidad de conexión a internet) con la única diferencia que la consulta lo realiza de la tabla LOGUIN de la base de datos local para su comparación.

Tabla 16. Segunda parte de Código de la Clase Loguin.java

Anexo N° 3.

Descripción código del archivo Activity_Menu.xml

Descripción código del archivo Activity_Menu.xml

Nombre:	Button
Imagen	
Código xml	<pre><ImageButton android:id="@+id/Pedido" android:layout_width="510dp" android:layout_height="165dp" android:layout_margin="10dp" android:background="#0000" android:onClick="IrPedido" android:scaleType="centerInside" android:src="@drawable/pedido" /></pre>
Nombre:	TextView
Imagen	
Código xml	<pre><TextView android:id="@+id/TxvInicioPedidosPrincipal" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_gravity="center" android:text="Pedido" android:textAppearance="?android:attr/textAppearanceMedium" /></pre>
Nombre:	Button
	
Código xml	<pre><TableRow android:id="@+id/tableRow1" android:layout_width="fill_parent" android:layout_height="fill_parent" android:layout_weight="1" > </TableRow></pre>
Nombre:	List View

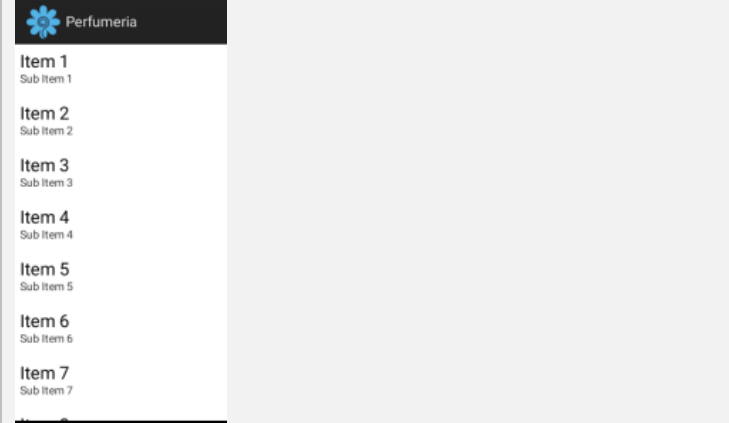
<p>Imagen</p>	
<p>Código xml</p>	<pre data-bbox="574 611 1308 823"><ListView android:id="@+id/lstClientes" android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_centerHorizontal="true" android:layout_centerVertical="true" > </ListView></pre>

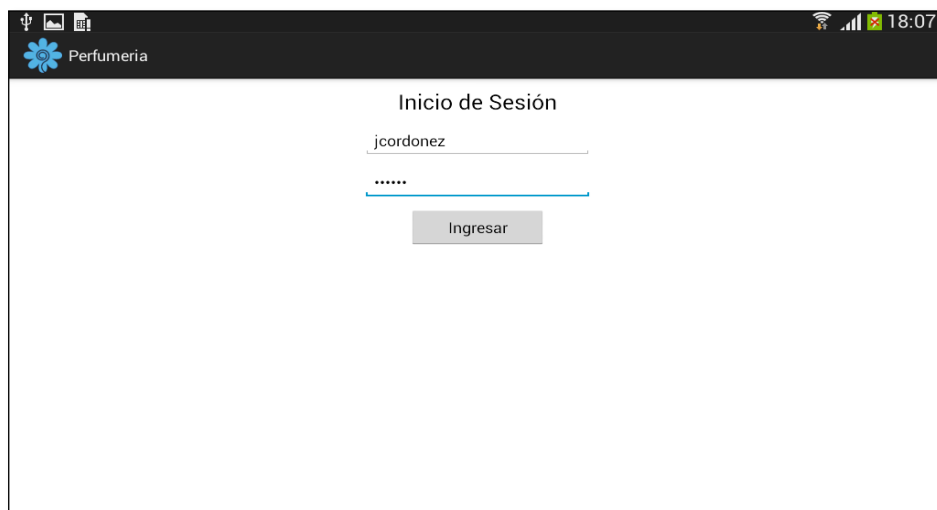
Tabla 17. Descripción código del archivo Activity_Menu.xml

Anexo N° 4.

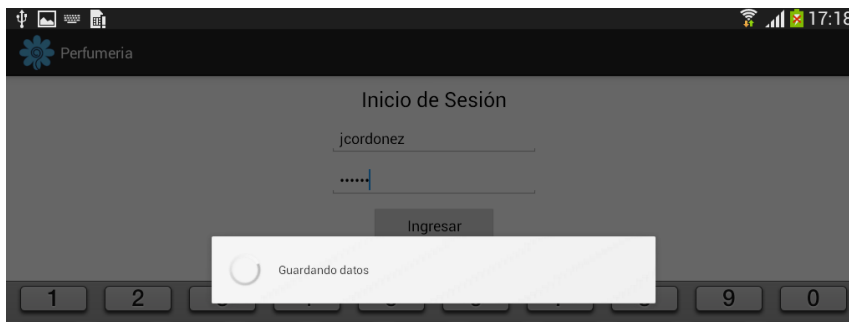
Manual de la Aplicación de Pedidos

Para el Manual de la aplicación se lo realiza con datos reales y va dirigido principalmente para el agente vendedor quien será el que utilizara la apelación.

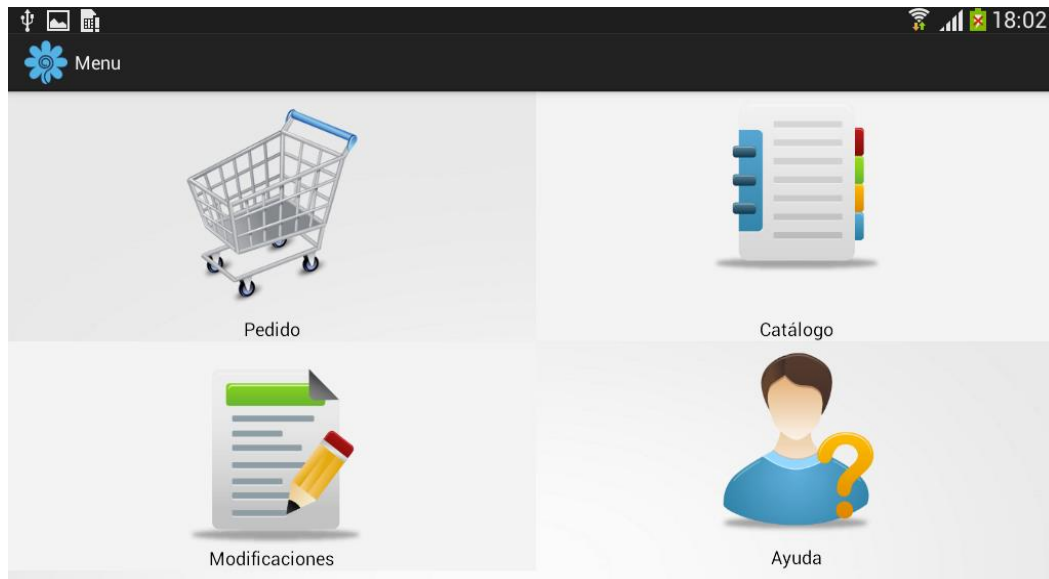
- 1 En la Ventana o Activity de Inicio de Sesión Procedemos a ingresar el respectivo usuario y clave que le será asignado por el administrador de la aplicación:



- 2 Para la opción anterior cabe recalcar que necesitamos una conexión a Internet ya sea por Wifi o por Plan de datos (3G), podemos observar que al ingresar se demora un tiempo “Guardando Datos” es porque está consultando la información y cargando toda la información necesaria.



- 3 Una vez ingresado a la apelación y luego de que se haya cargado los datos nos aparece el menú Principal:



- 4 Vamos a ver cuándo seleccionamos Catálogo, donde podemos filtrar los artículos por tipo y por Marca pudiendo ver el detalle del artículo seleccionado.

TIPO		MARCA	ARTICULOS
CABALLEROS	397 ›	CAFE ›	Codigo :1239690 CHT SUBL EDP 50ML NS 65069775 0 ML PVP : \$140.17 Stock : 525 Unidades CAR./HERRERA Unidades x caja: 1.0
CLASICO OJOS	2 ›	CALVIN KLEIN ›	
COLORFAST - OJOS	1 ›	CAR./HERRERA ›	
CORRECTOR	7 ›	CARON ›	
CUIDADO DEL CABELLO	1 ›	CARTIER ›	Codigo :1130441 EDT 100ML VAPO 65043790 0 ML PVP : \$158.27 Stock : 0 Unidades CAR./HERRERA Unidades x caja: 1.0
CUIDADO PESONAL-BODY 5 CARE	›	CARVEN ›	
DAMAS	914 ›	CHARLES JOURDAN ›	
DESODORANTES	1 ›	CHARLIE ›	Codigo :1237135 EDT 60 ML VAPO 65043793 0 ML PVP : \$122.65 Stock : 49 Unidades CAR./HERRERA Unidades x caja: 1.0
EL GRAN DESEO	2 ›	CLINIQUE ›	
ESTUCHES	25 ›	CONCEPT II ›	
		CONCEPT II NEW ›	


5 Vamos a Ver las opciones para realizar un Pedido al selecciona la opción “Pedido”.

- a) Lo primero que no aparece es el listado de los clientes asignado a ese agente vendedor, Ver detalle del cliente y seleccionarlo si es el caso.



Listado De Clientes	
ALMAC.JUAN ELJURI CIA-LTDA AUT	3561
ALMACENES JUAN ELJURI CIA LTDA AV.GIL RAMIREZ DAVALOS 5-32 Y ARMENILLAS 72862111 Distancia : 0 Mts	
ALMACE.JUAN ELJURI CIA.LTDA AU	356
ALMACENES JUAN ELJURI CIA LTDA Av GIL RAMIREZ DAVALOS 5-32 Y ARMENILLAS 72862111 Distancia : 0 Mts	
ALMACENES LUIS BARAHONA P	22510
BARAHONA PESANTEZ LUIS HUMBERTO MARIANO CUEVA N.10-19 Y GRAN COLOMBIA 2831399 Distancia : 0 Mts	
ALMACENES MANANTHIAL	15515
CONTRERAS LOJANO LUIS BOLIVAR SANGURIMA 8-02 846231 Distancia : 0 Mts	
AREVALO LUZ MARIA	9870

- b) Una vez seleccionado el cliente podemos ver el Detalle y aparece un botón



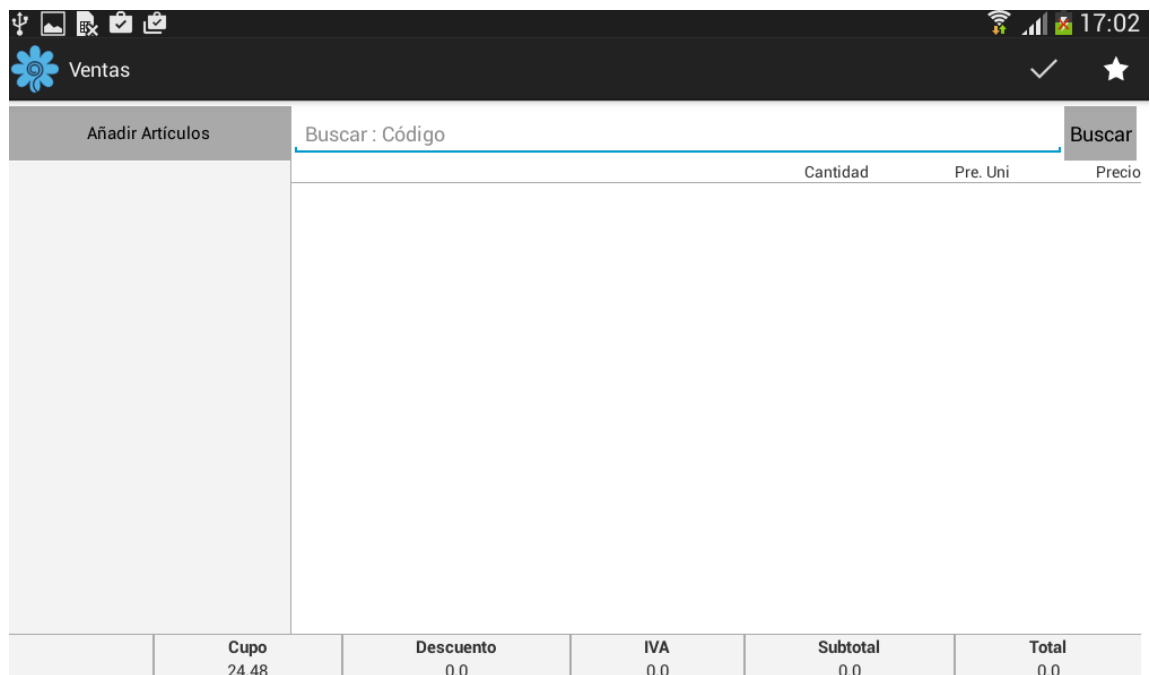
Detalle De Cliente		
DISTRIBUIDORA PEDRO SANTOS	SANTOS GODOY PEDRO LUIS	23853
RUC/CI :0104700463001	EDUARDO ARIAS S/N Y	ROBERTO CRESPO
Crédito		
Cliente tiene Crédito a A 30 Y 60 DÍAS		
Cupo \$ 24.48		
Canal		
Canal 0.0 NOT DEFINE		
SubCanal 5.1 Mayorista		
Max Descuento :	10.0	
Ciudad :	CUENCA	
Inicio Proceso		
Cerrar Proceso		

para “Iniciar proceso”

c) En Gestión Precedemos a selecciona Pedido.



d) Nos Aparece la Pantalla de Ventas donde vamos a registrar loas artículos a manera de una detalle de Factura:



e) Al seleccionar “Añadir artículos” no lleva directamente al Catalogo donde al escoger el artículos deseado nos pide que ingresemos la cantidad del articulo seleccionado y posteriormente confirmamos (pantalla de la derecha)



f) Podemos observar el Detalle de lo que se ingresó con los cálculos respectivos, además a lado izquierdo nos aparece los detalles organizado por tipos, si estamos seguros de que nuestro pedido está bien y no queremos agregar más artículos procedemos a confirmar en el visto para confirmar el pedido.

Añadir Articulos		Buscar : Código		Buscar		
				Cantidad	Pre. Uni	Precio
DAMAS	1 >	(1241362) BORDEAUX	490112 090 0 ML	2.0	6.47	\$12.94
LABIOS	1 >					
UÑAS	1 >					
		Cupo	Descuento	IVA	Subtotal	Total
		24.48	0.0	50.76	423.01	473.77

confirmar el pedido nos aparece una ventana donde podemos ingresar el adelanto en caso de haberlo y la forma de pago.

Pago Predeterminado : A 30 Y 60 DÍAS

\$ 473.77 Total a Facturar EDUARDO A. ▾

Ingrese Adelanto : 0.0

A 30 Y 60 DÍAS

A 30 DÍAS

CONTADO EFECTIVO

A 60 DÍAS

A 45 DIAS

A 30 DIAS

A 30 Y 45 DÍAS

Descripcion :

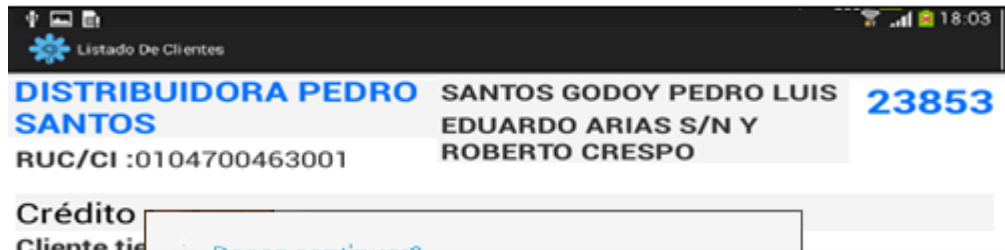
Cancel Ok

- h) Por ultimo vemos el resumen del pedido y si estamos seguros sincronizamos o confirmamos en el visto.

ResumenPedido				✓
DISTRIBUIDORA PEDRO SANTOS			Subtotal : 423.01	
SANTOS GODOY PEDRO LUIS			Descuento : 0.0	
			Iva : 50.76	
			Total : 473.77	
Articulos		Cantidad	Precio	Total
BORDEAUX	490112 090	2	6.47	12.94
EDT 50 ML VAPO	R.1023031	5	80.2	401.0
NUDE PEARL	R.4668	3	19.94	59.82

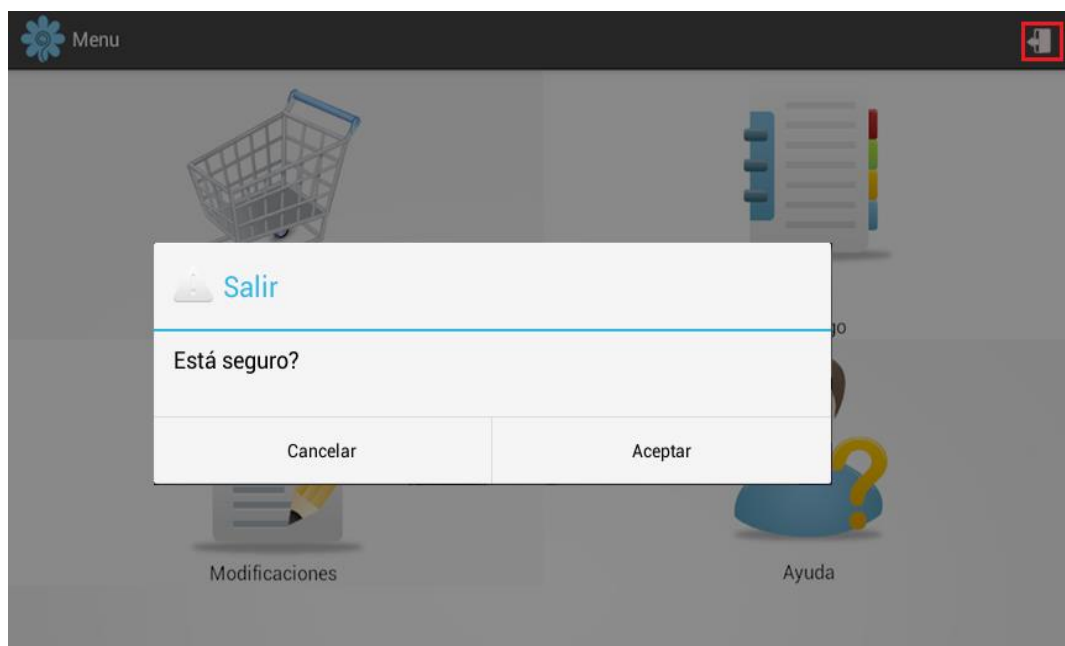
- i) Finalmente si no queremos realizar otro pedido y queremos finalizar el proceso con el cliente seleccionamos “Cerrar proceso” y volvemos al menú principal.

6 E
n
e
l
M
e
n
ú
p
r
i
n
c
i



pal también tenemos el menú ayuda, el cual nos sirve para ver este manual que estamos editando.

- 7 Finalmente para salir de la aplicación presionamos el icono de salir (Puerta con una flecha) y salimos del sistema.



Anexo N° 5. Manual de Uso de la Intranet, Control de Rutas de Vendedores.

El presente manual guiará paso a paso el Ingreso a la Intranet de la empresa, para lo cual del Agente vendedor deberá contar con su respectivo usuario y contraseñas y debe tener acceso de “Gestión de Actividades por Agente” este control lo asigna el Administrador de la Intranet. Vamos a detallar cada uno de los pasos para el Control de Rutas de los Vendedores.

- 1 Ingresamos a la intranet ya sea localmente o externamente cualquier parte con internet. www.grupoeljuri.com. Y procedemos a ingresar usuario y contraseña.



- 2 Damos click en Operaciones y nos debe aparecer habilitado la opción *Gestión de*



Actividades por Agente escogemos esta opción:

- 3 Luego Vemos un menú como lo muestra en la imagen en la cual solo tendremos opción al Botón “Criterios”



- 4 Una vez ingresado los criterios procedemos a seleccionar en el botón “Ejecutar”

5 Según los criterios que hayamos ingresado (rango de fechas) nos aparece una tabla

Almacenes Juan Eljuri >> Sugerencias

Nuevo Editar Grabar Eliminar Imprimir Procesar Criterios Ejecutar |< < > >| Help ? Atrás

Fec. Inicial 2014-03-09 Fec. Final 2015-03-09 Linea SE

9Registro(s) Seleccionado(s)

Nombre	Agente	Fecha
JUAN CARLOS ORDOÑEZ	1307	2014-03-10	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-11	Ver Detalle	Ver Mapa
ADRINA LARA	2493	2014-03-12	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-13	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-12	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-15	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-17	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-14	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-18	Ver Detalle	Ver Mapa

con los detalles y mapas generado por determinado agente vendedor.

6 Al seleccionar detalle nos muestra los movimientos u operaciones realizadas por determinado agente vendedor, organizada en una tabla con las gestiones realizadas con la respectiva fecha y hora.

Almacenes Juan Eljuri ..:Soporte y Desarrollo Patricio Tenezaca >> Sugerencias

Nuevo Editar Grabar Eliminar Imprimir Procesar Criterios Ejecutar |< < > >| Help ? Atrás

Fec. Inicial 2014-03-09 Fec. Final 2015-03-09 Linea SE

SRregistro(s) Seleccionado(s)

DETALLE DIARIO

Nombre: JUAN CARLOS ORDOÑEZ, Cod. Agente: 1307, Agente: ORDO#EZ JUAN CARLOS

Cod. Cliente	Nom. Cliente	Gestión	Fecha	Hora	Observación
709	MARUN RODRIGUEZ MARIA ELENA	CAPACITACION AL CLIENTE	2014-03-10	15:25	COBRANZA
707	ZAMORA GRANIZO ANA LUISA	REVISION DE LOS OBJETIVOS	2014-03-10	16:19	COBRANZA
20574	PAZMIÑO OROZCO AMALIA AUGUSTA	ACTIVACION DE SALES DRIVERS	2014-03-10	17:23	COBRANZA
23547	SORIA AYON RAFAEL HUMBERTO	VISITA ORDINARIA	2014-03-10	18:03	PEDIDO
20500	MONTERO MORLA ELENA GRACIELA	CAPACITACION AL CLIENTE	2014-03-10	18:42	COBRANZA

Cerrar

7 Y finalmente al seleccionar en “Ver Mapa” nos aparece un mapa ubicando cada operación realizada y detallado en el punto anterior, estos puntos están enumerados en orden cronológico.

Nombre	Agente	Fecha
JUAN CARLOS ORDOÑEZ	1307	2014-03-10	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-11	Ver Detalle	Ver Mapa
ADRINA LARA	2493	2014-03-12	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-13	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-12	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-15	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-17	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-14	Ver Detalle	Ver Mapa
JUAN CARLOS ORDOÑEZ	1307	2014-03-18	Ver Detalle	Ver Mapa

