

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

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

TEMA:

**SISTEMA INFORMÁTICO DE PEDIDOS Y ENTREGA DE PRODUCTOS
UTILIZANDO ANDROID**

Autor: Daniel Omar Villarreal Caluqui

Director: Ing. Carpio Pineda

Ibarra-Ecuador

Enero-2017

CERTIFICACIÓN

Certifico que la Tesis “**SISTEMA INFORMÁTICO DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID**” realizada para la Empresa TIERRA ZOE de la ciudad de Ibarra provincia de Imbabura, ha sido realizada en su totalidad por el señor: Daniel Omar Villarreal Caluqui, portador de la cédula de identidad número: 1002514188.



.....
Ing. Carpio Pineda
Director de Tesis

CERTIFICACIÓN

Ibarra, 15 de enero del 2017


Señores.
UNIVERSIDAD TÉCNICA DEL NORTE
Presente. –

De nuestras consideraciones. -

Siendo auspiciantes del proyecto de tesis del egresado Daniel Omar Villarreal Caluqui con CI: 1002514188 quien desarrolló su trabajo con el tema “**SISTEMA INFORMÁTICO DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID**”, me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado(a) Daniel Omar Villarreal Caluqui. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El egresado Daniel Omar Villarreal Caluqui, puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Dr. Alfonso Carrillo
GERENTE
EMPRESA DE DISTRIBUCIÓN TIERRA ZOE

Tierra Zoe
ALMACÉN PECUARIO
RUC 1002879888001
Av. Fray V. Galindo y Mariano Acosta
TEL 2600 029 IBARRA-ECUADOR

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



Yo, Daniel Omar Villarreal Caluqui, con cédula de identidad Nro. 100251418-8, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, Art. 4,5 y 6, en calidad de autor del trabajo de grado denominado: “SISTEMA INFORMÁTICO DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID”, que ha sido desarrollado para optar por el título de: Ingeniero en Sistemas Computacionales, quedando la Universidad facultada para ejercer los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago la entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

A handwritten signature in blue ink, appearing to read "Daniel Omar Villarreal Caluqui", is written over a blue scribble.

Firma: -----

Nombre: Daniel Omar Villarreal Caluqui

Cedula: 1002514188

Fecha: Enero del 2017



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE
LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

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

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD	1002514188
APELLIDOS Y NOMBRES	VILLARREAL CALUQUI DANIEL OMAR
DIRECCIÓN	AVENIDA ELOY ALFARO Y MARTIN ZUMETA 1-71
EMAIL	daniel_omar00@hotmail.com
TELÉFONO FIJO	062624244
TELÉFONO MÓVIL	0994539658
DATOS DE LA OBRA	
TÍTULO	SISTEMA INFORMÁTICO DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID
AUTOR	VILLARREAL CALUQUI DANIEL OMAR
FECHA	15 DE ENERO DEL 2017
PROGRAMA	PREGRADO
TÍTULO POR EL QUE OPTA	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	ING. CARPIO PINEDA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Daniel Omar Villarreal Caluqui, con cédula de identidad Nro. 1002514188, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

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

Ibarra, a los 15 días del mes de enero de 2017

EL AUTOR:

A handwritten signature in blue ink, appearing to read 'Daniel Villarreal Caluqui', written over a faint, illegible stamp or background.

.....
DANIEL OMAR VILLARREAL CALUQUI
1002514188

DEDICATORIA

Este trabajo de tesis de grado está dedicado a **DIOS**, por darme la vida a través de mis queridos **PADRES** quienes con mucho cariño, amor, ánimos y ejemplo han hecho de mí una persona con valores morales y éticos para poder desenvolverme como: **PADRE Y PROFESIONAL**.

A mis **HIJOS, Omar Alejandro y Daniel Alexander** que son el motivo y la razón que me ha llevado a seguir adelante día a día, para alcanzar mis ideales de superación, ellos fueron quienes, en los momentos más difíciles de mi vida, me dieron su amor y fuerza para alcanzar esos sueños anhelados.

Y por último quisiera dejar sentada la enseñanza al mundo, que cuando se quiere alcanzar algo en la vida, no hay tiempo ni obstáculo que impida alcanzar nuestras **METAS**.

Daniel Villarreal C.

AGRADECIMIENTO

A Dios por iluminar mi camino, darme fuerzas para vencer las adversidades de la vida y guiarme por el sendero del bien.

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento. Papá y Mamá

A mis hijos, que es el motivo y la razón que me ha llevado a seguir adelante.

A mi esposa Paulina Carrillo, quien en los momentos más difíciles me brindó su amor y apoyo para seguir adelante.

A mi director de tesis Ingeniero: Carpio Pineda, quien durante todo este tiempo me supo guiar en el desarrollo de este trabajo.

A mis docentes mi infinito agradecimiento ya que con sus enseñanzas enrumbaron mi camino para ser una persona con ética y responsabilidad preparada para los retos de la vida, a todos y cada uno de ellos mi agradecimiento sincero.

A mi compañero y amigo Ing. Manuel Cepeda y a todas aquellas personas que en mayor o menor medida han ayudado a que este trabajo se desarrolle.

Daniel Villarreal C.

ÍNDICE DE CONTENIDO

RESUMEN	xvi
ABSTRACT	xviii
CAPÍTULO I: INTRODUCCIÓN	1
1.1 Problema.....	1
1.1.1 Antecedentes	1
1.1.2 Situación Actual.....	2
1.1.3 Planteamiento del Problema	2
1.1.4 Prospectiva.....	2
1.2 Objetivos	3
1.2.1 Objetivo General.....	3
1.2.2 Objetivos Específicos.....	3
1.3 Justificación.....	3
1.4 Alcance.....	4
1.5 Tabla inicial del proyecto: Histórico Inicial.....	4
CAPÍTULO II: MARCO TEÓRICO	5
2 Marco Teórico.	5
2.1 Android.....	5
2.1.1 Versiones.....	6
2.1.2 Características importantes.	9
2.1.3 Ventajas.....	11
2.1.4 Desventajas	11
2.1.5 Arquitectura	11
2.1.5.1 Aplicaciones	12
2.1.5.2 Framework de Aplicaciones	13
2.1.5.3 Librerías.....	13
2.1.5.4 Tiempo de ejecución de Android	14
2.1.5.5 Núcleo Linux.....	14
2.1.5.6 Herramientas de desarrollo	14
2.2 Servidor de Aplicaciones	17
2.3 Framework JSF-Arquitectura.....	17
2.3.1 ¿Qué es Jsf?.....	17
2.3.2 Componentes.....	18
2.3.3 Navigation rules	19
2.3.4 Render Kits	19

2.3.5	EIS Tier	20
2.3.6	Integration Tier	20
2.3.7	Business Logic Tier	21
2.3.8	Presentation Tier	21
2.4	Sistema Gestor de Base de Datos Postgres	22
2.4.1	Historia.....	22
2.4.2	Características	22
2.4.3	Funciones	23
2.4.4	Ventajas.....	23
2.5	Lenguaje de Programación JSF.....	23
2.6	Patrón de Diseño - MVC (Modelo, Vista, Controlador).....	24
2.6.1	Descripción del patrón	25
CAPÍTULO III: DESCRIPCIÓN, PROCEDIMIENTO Y FUNCIONAMIENTO.....		27
3	Descripción, Procedimiento y Funcionamiento del Sistema.....	27
3.1	Visión General.....	27
3.1.1	Requerimientos funcionales.....	27
3.1.2	Requerimientos no funcionales.....	29
3.2	Definición, procesos y subprocesos a implementar	30
3.2.1	Procesos y subprocesos a implementar.....	30
3.2.1.1	Ambiente físico	30
3.2.1.2	Interfaces	30
3.2.1.3	Usuarios y factores humanos.....	31
3.2.1.4	Funcionalidad	31
3.2.1.5	Documentación.....	31
3.2.1.6	Datos	31
3.2.1.7	Seguridad	32
3.2.1.8	Aseguramiento de la calidad.....	32
3.3	Funcionamiento del sistema.....	32
3.3.1	Modelo de negocio.....	33
3.3.2	Identificación de roles en el entorno del negocio:	33
3.4	Funcionamiento de procesos	35
3.4.1	Proceso 1: Funcionamiento de la aplicación.....	35
3.4.2	Proceso 2: Cliente Realiza Pedidos.....	36
3.4.3	Proceso 3: Operador/Administrador	36
3.4.4	Proceso 4: Vendedor	37

3.4.5	Proceso 5: Administrador	37
CAPÍTULO IV: INTEGRACIÓN DE LA APLICACIÓN		39
4	Integración de la Aplicación: Metodología RUP	39
4.1	Diseño y Arquitectura del Sistema.....	39
4.1.1	Selección del entorno de desarrollo: Aplicación WEB.....	39
4.1.2	Selección del entorno de desarrollo: Aplicación Móvil.....	40
4.1.3	Modelo contextual inicial del proyecto.....	41
4.1.4	Arquitectura del sistema	42
4.2	Construcción del Sistema: Fase de Inicio	44
4.2.1	Plan de desarrollo.....	44
4.2.1.1	Propósito.....	44
4.2.1.2	Alcance	44
4.2.2	Vista General.	45
4.2.2.1	Propósito, objetivos y alcance	45
4.2.2.2	Suposiciones y Restricciones.....	46
4.2.2.3	Entregables del proyecto	46
4.2.2.4	Evolución del plan de desarrollo de software.....	48
4.2.3	Organización Esquema del Proyecto	48
4.2.3.1	Copartícipes en el Proyecto	48
4.2.3.2	Interfaces externas	48
4.2.3.3	Roles y responsabilidades	49
4.2.4	Gestión del Esquema del Proyecto	49
4.2.4.1	Plan del proyecto	49
4.2.4.2	Plan de las fases	49
4.2.4.3	Objetivos de las iteraciones	51
4.2.4.4	Calendario	52
4.2.5	Seguimiento y control del desarrollo del proyecto	55
4.2.5.1	Gestión de requerimientos del proyecto.....	55
4.2.5.2	Control de plazos	55
4.2.5.3	Control de calidad	55
4.2.5.4	Gestión de riesgos.....	55
4.2.5.5	Gestión de configuración	55
4.2.6	Lista de riesgos	56
4.3	Fase de Elaboración	57
4.3.1	Especificación caso de uso: Ingreso Datos Empresa	58

4.3.2	Especificación caso de uso: Ingreso de Usuarios	59
4.3.3	Especificación caso de uso: Ingreso de Catálogos Detalles-Provincia.....	60
4.3.4	Especificación caso de uso: Ingreso de Catálogos Detalles-Categoría Producto. ..	61
4.3.5	Especificación caso de uso: Ingreso de Catálogos Detalles-Bodega.	62
4.3.6	Especificación caso de uso: Ingreso de Catálogos Detalles-Unidad de Medida. ...	63
4.3.7	Especificación caso de uso: Ingreso de Catálogos Detalles-Identificación.	64
4.3.8	Especificación caso de uso: Ingreso de Productos.....	65
4.3.9	Especificación caso de uso: Ingreso de Clientes.....	66
4.3.10	Especificación caso de uso: Ingreso de Productos en Bodega.....	67
4.3.11	Especificación caso de uso: Visualización de Lista de Pedidos.	68
4.3.12	Especificación caso de uso: Inventario-Bodega.	69
4.3.13	Especificación caso de uso: Asignar Ruta.	70
4.3.14	Especificación caso de uso: Ver Ruta.	71
4.3.15	Especificación caso de uso: Reporte-Pedidos.....	72
4.3.16	Especificación caso de uso: Cliente-Crear Pedido.....	73
4.3.17	Especificación casos de uso: Cliente-Visualizar Pedidos Ingresados.....	74
4.3.18	Especificación caso de uso: Vendedor-Visualiza y Entrega Pedidos	75
4.3.19	Especificación caso de uso: Vendedor-Crear Cliente.....	76
4.4	Fase de construcción	77
4.4.1	Diagrama de Clases.....	77
4.4.2	Diagrama Entidad Relación	78
4.5	Fase de Implementación.....	80
4.5.1	Especificación de caso de prueba: Ingreso Datos Empresa	80
4.5.2	Especificación de caso de prueba: Ingreso Datos Usuarios.....	81
4.5.3	Especificación de caso de prueba: Catálogos Detalles: Provincia.....	82
4.5.4	Especificación de caso de prueba: Catálogos Detalles: Categoría Producto	83
4.5.5	Especificación caso de prueba: Catálogos Detalles: Bodega.....	84
4.5.6	Especificación caso de prueba: Catálogos Detalles: Unidad de Medida	85
4.5.7	Especificación caso de prueba: Catálogos Detalles: Identificación.....	86
4.5.8	Especificación caso de prueba: Ingreso de Productos.	87
4.5.9	Especificación caso de prueba: Ingreso de Clientes.	88
4.5.10	Especificación caso de prueba: Ingreso de Productos en Bodega.	89
4.5.11	Especificación caso de prueba: Visualización de Lista de Pedidos.....	90
4.5.12	Especificación caso de prueba: Inventario Bodega.	91

4.5.13	Especificación caso de prueba: Asignar Ruta.	92
4.5.14	Especificación caso de prueba: Ver Rutas.	93
4.5.15	Especificación caso de prueba: App-Móvil-Cliente-Crear Pedido.	94
4.5.16	Especificación caso de prueba: App-Móvil-Cliente-Ver Pedidos Ingresados.	95
4.5.17	Especificación caso de prueba: App-Móvil-Vendedor-Ver Pedido Asignado.	96
4.5.18	Especificación caso de prueba: App-Móvil-Vendedor-Entregar Pedido.	97
4.5.19	Especificación caso de prueba: App-Móvil-Vendedor-Ingresa Clientes.	98
CAPÍTULO V		99
5	CONCLUSIONES Y RECOMENDACIONES.	99
5.1	CONCLUSIONES	99
5.2	RECOMENDACIONES	101
GLOSARIO		102
BIBLIOGRAFÍA		104
ANEXOS		106
ANEXO A. DICCIONARIO DE DATOS		106
ANEXO B. MANUAL DE USUARIO		112
ANEXO C. MANUAL TÉCNICO		131

ÍNDICE DE TABLAS

Tabla 1.1: Histórico Inicial	4
Tabla 2.1: Versiones SDK Android	8
Tabla 2.2: Características de Android.....	10
Tabla 4.1: Roles y Responsabilidades del Proyecto	49
Tabla 4.2: Plan de Fases del Proyecto	50
Tabla 4.3: Plan de Fases: Hitos del Proyecto.....	50
Tabla 4.4: Plan de Fases: Iteraciones del Proyecto.....	52
Tabla 4.5: Roles y Responsabilidades Fase de Inicio.....	53
Tabla 4.6: Roles y responsabilidades Fase de Elaboración	54
Tabla 4.7: Lista de Riesgos.....	56
Tabla 4.8: CU-001-Ingreso Datos Empresa.....	58
Tabla 4.9: CU-002-Ingreso Usuarios.....	59
Tabla 4.10: CU-003-Ingreso de Catálogos Detalles-Provincia.....	60
Tabla 4.11: CU-004-Ingreso de Catálogos Detalles-Categoría Producto.....	61
Tabla 4.12: CU-005-Ingreso de Catálogos Detalles-Categoría Bodega	62
Tabla 4.13: CU-006-Ingreso de Catálogos Detalles-Unidad de Medida	63
Tabla 4.14: CU-007-Ingreso de Catálogos Detalles-Unidad de Medida.....	64
Tabla 4.15: CU-008-Ingreso de Catálogos Detalles-Unidad de Medida	65
Tabla 4.16: CU-009-Ingreso de Clientes	66
Tabla 4.17: CU-010-Ingreso de Productos en Bodega	67
Tabla 4.18: CU-011-Lista de Pedidos.....	68
Tabla 4.19: CU-012-Inventario-Bodega	69
Tabla 4.20: CU-013-Asignar Ruta.....	70
Tabla 4.21: CU-014-Ver Ruta.....	71
Tabla 4.22: CU-015-Reporte-Pedidos	72
Tabla 4.23: CU-016-Cliente-Crear Pedido-App-Móvil.....	73
Tabla 4.24: CU-017-Cliente-Visualiza-Pedidos Ingresados-App-Móvil	74
Tabla 4.25: CU-018-Vendedor-Visualiza y Entrega Pedidos-App-Móvil	75
Tabla 4.26: CU-019-Vendedor-Ingresa Clientes -App-Móvil.....	76
Tabla 6.1: Anexo A. Tabla Rol.....	106
Tabla 6.2: Anexo A. Tabla Usuario	107
Tabla 6.3: Anexo A. Tabla Empresa.....	107
Tabla 6.4: Anexo A. Tabla Cliente	108
Tabla 6.5: Anexo A. Tabla Catálogo	108
Tabla 6.6: Anexo A. Tabla Catalogo Detalle.....	109
Tabla 6.7: Anexo A. Tabla Pedidos	109
Tabla 6.8: Anexo A. Tabla Detalle Pedidos	110
Tabla 6.9: Anexo A. Tabla Producto	110
Tabla 6.10: Anexo A. Tabla Stock Bodega	111

ÍNDICE DE FIGURAS

Figura 2.1: Arquitectura de Android.....	12
Figura 2.2: Framework JSF-Arquitectura	20
Figura 2.3: Patrón de Diseño (MVC).....	24
Figura 3.1: Relaciones de trazabilidad entre los modelos de negocio y de requisitos.....	33
Figura 3.2: Diagrama del negocio-Rol Cliente	34
Figura 3.3: Diagrama del negocio-Rol Administrador-Verifica el Pedido.....	34
Figura 3.4: Diagrama del negocio-Rol Vendedor.....	35
Figura 3.5: Diagrama del negocio/Funcionamiento de la Aplicación	35
Figura 3.6: Diagrama del negocio/Cliente Realiza Pedidos	36
Figura 3.7: Diagrama del negocio/Operador	36
Figura 3.8: Diagrama del negocio/Vendedor.....	37
Figura 3.9: Diagrama del negocio/Administrador	37
Figura 4.1: Modelo conceptual inicial para el caso de uso del negocio Crear Pedido	41
Figura 4.2: Arquitectura del Sistema	43
Figura 4.3: Descripción de la metodología RUP	52
Figura 4.4: CU-001-Ingreso Datos Empresa	58
Figura 4.5: CU-002-Ingreso de Usuarios.....	59
Figura 4.6: CU-003-Ingreso de Catálogos Detalles-Provincia	60
Figura 4.7: CU-004-Ingreso de Catálogos Detalles-Categoría Producto.....	61
Figura 4.8: CU-005-Ingreso de Catálogos Detalles-Categoría Bodega.....	62
Figura 4.9: CU-006-Ingreso de Catálogos Detalles-Unidad de Medida.....	63
Figura 4.10: CU-007-Ingreso de Catálogos Detalles-Identificación	64
Figura 4.11: CU-008-Ingreso de Productos	65
Figura 4.12: CU-009-Ingreso de Clientes	66
Figura 4.13: CU-010-Ingreso de Productos en Bodega	67
Figura 4.14: CU-011-Lista de Pedidos	68
Figura 4.15: CU-012-Inventario-Bodega.....	69
Figura 4.16: CU-013-Ruta-Asignar Ruta.....	70
Figura 4.17: CU-014-Ruta-Ver Ruta	71
Figura 4.18: CU-015-Reporte-Pedidos	72
Figura 4.19: CU-016-Cliente-Crear Pedido-App-Móvil	73
Figura 4.20: CU-017-Cliente-Visualizar-Pedidos Ingresados-App-Móvil.....	74
Figura 4.21: CU-018-Vendedor-Visualiza y Entrega Pedidos-App-Móvil	75
Figura 4.22: CU-019-Vendedor-Ingresar Clientes -App-Móvil	76
Figura 4.23: Diagrama de Clases-SPEP-UA	78
Figura 4.24: Diagrama Entidad Relación-SPEP-UA	79
Figura 6.1: Anexo B. Ingreso Aplicación Web.....	113
Figura 6.2: Anexo B. Ingreso Datos Empresa	114
Figura 6.3: Anexo B. Ingreso Datos Usuario.....	115

Figura 6.4: Anexo B. Ingreso Catálogos Detalles.....	116
Figura 6.5: Anexo B. Ingreso Productos.....	117
Figura 6.6: Anexo B. Ingreso Clientes.....	118
Figura 6.7: Anexo B. Ingreso Productos a Bodega asignada.....	119
Figura 6.8: Anexo B. Revisar Pedidos.....	120
Figura 6.9: Anexo B. Revisar Inventario.....	121
Figura 6.10: Anexo B. Rutas-Asignar Rutas.....	122
Figura 6.11: Anexo B. Rutas-Ver Rutas.....	123
Figura 6.12: Anexo B. Reporte-Pedidos.....	124
Figura 6.13: Anexo B. App. Móvil-Cliente-Crear Pedido.....	127
Figura 6.14: Anexo B. App. Móvil-Vendedor-Visualiza y Entrega Pedido.....	129
Figura 6.15: Anexo B. App. Móvil-Vendedor-Ingresa Cliente.....	130
Figura 6.16: Anexo C. Instalación de Java.....	132
Figura 6.17: Anexo C. Configuración de la Variable de Entorno.....	136
Figura 6.18: Anexo C. Instalación de Postgres.....	139
Figura 6.19: Anexo C. Instalación de Android Studio.....	142
Figura 6.20: Anexo C. Instalación de Paquetes de Android Studio.....	146
Figura 6.21: Anexo C. Estructura de Un Proyecto Android.....	147
Figura 6.22: Anexo C. Instalación de Jboss Developer Studio.....	152
Figura 6.23: Anexo C. Agregar Jdbc para conectarnos a la BDD.....	155
Figura 6.24: Anexo C. Restaurar la Base de Datos.....	156
Figura 6.25: Anexo C. Levantar el Servidor de Aplicaciones Wilffly.....	158
Figura 6.26: Anexo C. Creación del Usuario Consola de Administración.....	159

RESUMEN

El presente trabajo de grado se ha elaborado para que la Distribuidora de alimentos y productos veterinarios “Tierra Zoe” de la ciudad de Ibarra, implemente un sistema de gestión de pedidos y entrega de productos, para mejorar la calidad de los servicios ofertados por la misma, de manera que pueda mejorar la atención y el servicio al cliente por medio de sus renovados procesos. El sistema informático es un conjunto de metodologías, procedimientos y lineamientos metodológicos que se debe seguir detalladamente para poder garantizar su funcionalidad y después de ello implementarlo en la empresa. Además es una guía para crear aplicaciones móviles utilizando Android, tecnología que está presente en los dispositivos móviles en la actualidad, también brinda una visión general de una aplicación desarrollada con Android conjuntamente con la herramienta Java Server Faces (JSF2) que es una tecnología y Framework (Marco de Trabajo) para aplicaciones Web basadas en el patrón MVC (Modelo-Vista-Controlador) que simplifica el desarrollo de interfaces de usuario en aplicaciones utilizando el lenguaje de programación Java EE, además se describen algunas de las herramientas disponibles presentes en el S.O. Android y finalmente se particulariza conceptos básicos para el aprendizaje del sistema operativo. El trabajo se ha estructurado de la siguiente manera: una introducción, un marco teórico investigativo sobre las herramientas que se utilizarán en la realización del proyecto, la integración e implementación del sistema, el documento presenta cinco capítulos.

- En el capítulo uno, se realiza una breve introducción sobre el por qué de la realización del proyecto y sus ventajas.
- En el capítulo dos, se detallan los conceptos técnicos sobre las herramientas que se utilizarán en el desarrollo de la aplicación.
- En el capítulo tres, se detalla el procedimiento y funcionamiento del sistema, definiendo los procesos y subprocesos a implementar.
- En el capítulo cuatro, se integra la aplicación detallando el uso de la metodología RUP (Proceso Unificado Racional) como conjunto de metodologías adaptables al contexto y necesidades de cada organización. Además, se detalla la lista de riesgos existentes relacionados con el proyecto y su nivel de impacto.
- Por último, en el capítulo cinco se detallan las conclusiones, recomendaciones y los respectivos anexos a los que se ha llegado, al realizar la investigación y el proyecto.

ABSTRACT

The present work of degree has been elaborated so that the distributor of foods and veterinary products “Zoe Land” of the city of Ibarra, implement a system of ordering and delivery of products, to improve the quality of the services offered by it, so that it can improve the attention and the customer service through its renewed processes. The computer system is a set of methodologies, procedures and methodological guidelines that must be followed in detail in order to guarantee its functionality and then implement it in the company. It is also a guide to creating mobile applications using Android, technology that is present in mobile devices today, also provides an overview of an application developed with Android in conjunction with the tool Java Server Faces (JSF2) which is a Technology and Framework for Pattern-Based Web Applications MVC (Model-View-Controller) which simplifies the development of user interfaces in applications using the Java programming language EE, In addition, some of the tools available OS Android and finally particularizes basic concepts for learning the operating system. The work has been structured as follows: an introduction, an investigative theoretical framework on the tools to be used in the implementation of the project, the integration and implementation of the system, the document presents five chapters.

- In chapter one, a brief introduction on why the project is done and its advantages.
- In chapter two, the technical concepts of the tools that were used to develop the application are detailed.
- In chapter three, the method and system performance is detailed, defining processes and threads to implement.
- In chapter four, the application detailing the use of RUP (Rational Unified Process) as set adaptable to context and needs of each organization integrates methodologies. In addition, the list of existing risks related to the project and its level of impact is detailed.
- Finally in chapter five conclusions, recommendations and the respective annexes to which it has been, to perform research and detailed project.

CAPÍTULO I: INTRODUCCIÓN

La sociedad en la que vivimos no podría ser concebida si no fuera gracias a un elemento que ha unificado las comunicaciones, llevando el entretenimiento a niveles no imaginados, un elemento que ha facilitado nuestras vidas, ha hecho más productivas las industrias y más exequible el conocimiento. Nos estamos refiriendo a la tecnología. El presente trabajo es una breve recopilación y descripción de los beneficios que la tecnología aporta a la sociedad moderna, lo que facilitará en medida la construcción y desarrollo del aplicativo, siguiendo lineamientos técnicos sobre las herramientas que se utilizarán con una arquitectura MVC¹, además brindará una visión general de desarrollo de aplicaciones en Android, describiendo algunas de sus funcionalidades y conceptos básicos del sistema operativo.

1.1 Problema

1.1.1 Antecedentes

La automatización de procesos es un cambio prioritario y necesario dentro de las empresas gracias a los beneficios que brinda el uso de las tecnologías de la información y comunicación, lo que se ha vuelto un tema de interés por parte de las entidades que prácticamente no hacen uso correcto de esos conocimientos disponibles hoy en día, ya que con el surgimiento de tecnologías móviles en el cual proporciona todas las interfaces necesarias para desarrollar aplicaciones con la facilidad de acceder a las funcionalidades del teléfono de forma dinámica y sencilla, permitiéndonos interactuar con la aplicación desde cualquier dispositivo móvil con Sistema Operativo Android.

La Empresa Tierra-Zoe es una distribuidora de alimentos para animales tales como equinos, bovinos, mascotas, aves y peces, que entrega sus productos en toda la provincia de Imbabura. En esta empresa la entrega de los productos consiste en la distribución detallada y minuciosa en respuesta a los pedidos de sus clientes, siendo así una combinación compleja de actividades, procesos, imaginación e ingenio. A pesar de que la empresa distribuye sus productos de alta calidad no cuenta con un sistema informático de ninguna naturaleza que ayude a optimizar los recursos y obtener información real de su desempeño.

¹ MVC: Arquitectura Modelo Vista Controlador

Por lo que se ha visto la necesidad de implementar una aplicación que solucione y automatice todos estos procesos de forma eficaz y eficiente, utilizando tecnologías gratuitas y disponibles en nuestro medio.

1.1.2 Situación Actual

Al no tener un aplicativo que permita el proceso de recopilación, organización y gestión de pedidos de los clientes la empresa lleva sus registros en herramientas de office y de forma manual en papel, lo que dificulta cumplir con las rutas de distribución en pedidos de los clientes, ya que todo este proceso se lo lleva manualmente, para luego dirigirse a la empresa y realizar la respectiva gestión de dicho pedido, perdiendo así tiempo y dificultando en gran escala un desempeño ágil y acertado en esta actividad, estando expuesta a errores, pérdida o inconsistencia de la misma y confusión al momento de requerir dicha información.

1.1.3 Planteamiento del Problema

En la empresa el registro de información correspondiente a los pedidos y entrega de los productos se realiza de forma manual y poco eficiente, generando así datos erróneos y pérdida de los mismos lo que ocasiona desorganización y retrasos en el cumplimiento de sus funciones.

1.1.4 Prospectiva

Una de las principales consecuencias de continuar con esta modalidad de pedidos en la empresa es no tener datos verdaderos de ésta actividad, como también se daría lugar a la exposición a diversos errores al momento de analizar la información generada, limitándose así a un desempeño ineficiente causando molestias a sus clientes ya que pudieran existir retrasos y desorden en las entregas, por lo que la aplicación será de gran ayuda para el manejo de información no solo en la actualidad sino también a futuro ya que las herramientas que se utilizarán para desarrollar la aplicación están acordes con las tecnologías actuales, lo que permitirá a la aplicación estar adaptable a cambios futuros.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema informático web de entrega y pedidos para la Empresa Pecuaria “Tierra Zoe”, basado en el uso de herramientas de software libre utilizando Android para la automatización de los procesos.

1.2.2 Objetivos Específicos

- 1 Realizar un estudio de la tecnología Android, como S.O. Móvil, basado en Linux, para utilizarla en el desarrollo del Sistema Informático de Entrega y Pedidos de Productos Comerciales.
- 2 Analizar el proceso de pedidos en la empresa.
- 3 Realizar el diseño de la arquitectura y funcionamiento de la solución planteada.
- 4 Desarrollar el sistema.
- 5 Integrar el sistema en la Empresa y actualizar los datos en el servidor para tener información en tiempo real.
- 6 Elaborar la documentación Técnica

1.3 Justificación

En la actualidad y hoy en día el uso de la tecnología ha sido vital en el desarrollo de muchos procesos, lo cual se ha vuelto un tema primordial y casi prioritario, ya que aporta en gran manera en el avance y mejoramiento de las mismas, por medio de las funcionalidades que brindan se logra obtener mejor tiempo de respuesta en el servicio al cliente, como también en los beneficios propios de la empresa.

Se justifica el presente proyecto en vista a la necesidad que tiene la Empresa “Tierra Zoe” de actualizar y automatizar los procesos internos, específicamente el proceso de pedidos, con la finalidad de obtener la información organizada, eficiencia en la emisión de facturas, gestión de pedidos y un control total de las ventas que se realizan dentro de la misma lo que se traduce en brindar un mejor servicio a todos sus clientes.

1.4 Alcance

El sistema beneficiará de forma directa a la empresa “Tierra Zoe”, en especial al personal encargado del proceso de pedidos logrando así mejor desempeño del trabajo, como también la mejora a gran escala en el servicio al cliente, por medio del registro automático de toda la información pertinente como también la disponibilidad total de la misma alcanzando eficiencia y rapidez.

1.5 Tabla inicial del proyecto: Histórico Inicial

Date	Versión	Descripción	Autor
Mayo 2014	1.0	Caso de desarrollo SISTEMA DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID	DANIEL VILLARREAL

Tabla 1.1: Histórico Inicial
Fuente: (Propia)

Logo de La Distribuidora



Figura 1.1: Logo Distribuidora Tierra Zoe
Fuente: (Propia)

CAPÍTULO II: MARCO TEÓRICO



2 Marco Teórico.

2.1 Android

Robledo (2011) señala que “Android es un sistema operativo, diseñado principalmente para dispositivos móviles como, teléfonos inteligentes, tablets o tabléfonos, también para relojes inteligentes, televisores y automóviles”. (p.3)

En la actualidad dan soporte y desarrollan quienes integran la Open Handset Alliance². (Orozco, 2011)

Android es un S.O. abierto, multipropósito, se accede a las funcionalidades mediante aplicaciones, estas aplicaciones pueden ser instaladas o desinstaladas a voluntad, además desarrollarlas por terceras personas a través de herramientas suministradas por Google, y por medio de los lenguajes de programación Java y C. (Orozco, 2011).

El código fuente de Android está disponible bajo diversas licencias de software libre y código abierto, Google liberó la mayoría del código bajo la licencia Apache³. Esto permite que un desarrollador no solo pueda modificar su código sino también mejorarlo. (Orozco, 2011).

² Open Handset Alliance: Es una alianza de 84 compañías que desarrollan aplicaciones para dispositivos móviles.

³ Apache: Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.).

Android depende de un Linux versión 2.6 para los esquemas base del sistema como la seguridad, procesos de memoria, gestión de procesos, stack⁴ de red y tipo de drivers. (Orozco, 2011).

Otras características que presenta Android es un navegador web integrado basado en el motor Webkit⁵, soporte para grafico 2D y 3D, soporte multimedia para audio, video e imágenes en varios formatos, conectividad Bluetooth, EDGE, 3G, WIFI, entre otros. (Orozco, 2011).

Cabe resaltar, que la libertad del código Android ha logrado que se implante en dispositivos electrónicos, desde móviles hasta computadoras portátiles, notebooks, microondas, lavadoras, marcos digitales, navegadores GPS⁶, relojes e incluso en navegadores de abordo de coches. (Orozco, 2011).

2.1.1 Versiones

El historial de versiones inició en noviembre de 2007. La versión comercial, Android 1.0, fue lanzada en 2008. Desarrollado por Google y la Open Handset Alliance. Desde abril de 2009, versiones desarrolladas con nombres en clave como: Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop y la más reciente es Marshmallow 6.0 la cual fue anunciada en el 2015.

A continuación, se detallan las versiones y los niveles de API⁷ ver en la siguiente tabla.

⁴ Stack: Pila

⁵ Webkit: Equipo Web

⁶ GPS: Sistema de Posicionamiento Global.

⁷ API: Interfaz de Programación de Aplicaciones.

Versiones	Fecha de lanzamiento	Niveles de API
Versión Beta	5 de Noviembre del 2007	
Apple pie v1.0 (Tarta de manzana)	23 de Septiembre del 2008	Nivel de API 1
Banana Bread v1.1 (Pan de plátano)	9 de Febrero del 2009	Nivel de API 2
Cupcake v1.5 (Panqué)	15 de Septiembre de 2009	Nivel de API 3
Donut v1.6 (Rosquilla)	15 de Septiembre de 2009	Nivel de API 4
Eclair v2.0 (Pastel Francés)	25 de Octubre del 2009	Nivel de API 5
Eclair v2.0.1	3 de Diciembre del 2009	
Eclair v2.1	12 de Enero del 2010	Nivel de API 7
Froyo v2.2 (Yogurt helado)	20 de Mayo del 2010	Nivel de API 8
Froyo v2.2.1	18 de Enero del 2011	
Froyo v2.2.2	22 de Enero del 2011	
Froyo v2.2.3	21 de Noviembre del 2011	
Gingerbread v2.3.0/v2.3.1 (Pan de jengibre)	6 de Diciembre del 2010	Nivel de API 9
Gingerbread v2.3.3	9 de Febrero del 2011	
Gingerbread v2.3.4	28 de Abril del 2011	
Gingerbread v2.3.5	25 de Julio del 2011	
Gingerbread v2.3.6	2 de Septiembre del 2011	
Gingerbread v2.3.7	21 de Septiembre del 2011	
Honeycomb v3.0	22 de Febrero del 2012	Nivel de API 11

(Panal de miel)		
Honeycomb v3.1	10 de Mayo del 2011	Nivel de API 12
Honeycomb v3.2	15 de Julio del 2011	Nivel de API 13
Honeycomb v3.2.1	20 de Septiembre del 2011	
Honeycomb v3.2.2	30 de agosto del 2011	
Honeycomb v3.2.3	1 de Diciembre del 2011	
Honeycomb v3.2.4	28 de Febrero del 2012	
Ice Cream v4.0.0/ v4.0.1 (Sandwich de helado)	19 de Octubre del 2012	Nivel de API 14
Ice Cream v4.0.2	29 de Noviembre del 2012	
Ice Cream v4.0.3	16 de Diciembre del 2011	Nivel de API 15
Ice Cream v4.0.4	8 de Noviembre del 2012	
Jelly Bean v4.1 (Gominola)	24 de Julio del 2013	Nivel de API 16
Jelly Bean v4.2		Nivel de API 17
Jelly Bean v4.3		Nivel de API 18
KitKat (v4.4) (Barquillo de Chocolate)	31 de Octubre del 2013	Nivel de API 19
KitKat (v4.4.2)		
lollipop v5.0 (Paleta)	15 de Noviembre del 2014	Nivel de API 21
Marshmallow (Malvavisco)	Octubre 2015	Nivel de API 23

Tabla 2.1: Versiones SDK Android

Fuente: (Tomás, 2015)

2.1.2 Características importantes.

Características y especificaciones actuales:

Diseño de dispositivo	El escenario es adaptable a pantallas de mayor resolución.
Almacenamiento	Android usa como base de datos <u>SQLite</u> .
Conectividad	Soporta una variedad de tipos de conexiones como: Bluetooth, GSM/EDGE, WiFi, LTE, WiMAX, etc.
Mensajería	Cloud to Device Messaging Framework ⁸ (C2DM).
Navegador web	Basado en el motor de renderizado de código abierto WebKit, ajustado con el motor JavaScript V8 de Chrome.
Soporte de Java	Se compila en la Máquina Virtual Dalvik, específicamente para Android y mejorada para dispositivos móviles.
Soporte multimedia	Soporta varios formatos como: JPEG, BMP, MP3, WAV, PNG, MIDI, GIF entre otros.
Soporte para Streaming⁹	HTML5, etc.
Soporte para hardware adicional	Soporta: pantallas táctiles, GPS, cámaras de video, de fotos, etc.

⁸ Cloud to Device Messaging Framework: Nube de dispositivo, marco de Mensajería

⁹ Streaming: Transmisión

Entorno de desarrollo	Utiliza Herramientas de Desarrollo proporcionadas por Android.
Google Play	Repertorio de entretenimiento para centralizar las aplicaciones y contenidos para dispositivos con S.O. Android.
Multi-táctil	Soporte para pantallas multi-táctil.
Bluetooth	Soporte para Bluetooth.
Video llamada	Soporta video llamada a través de Google Talk ¹⁰ .
Multitarea	Aplicaciones que se ejecutan al mismo tiempo compartiendo procesadores o que no trabajan reciben ciclos de reloj.
Características basadas en voz	La consulta o búsqueda en Google a través de la voz del usuario.
Tethering	Soporta tethering ¹¹ , esto hace que el teléfono pueda ser utilizado como punto de entrada o salida alámbrico o inalámbrico.

Tabla 2.2: Características de Android
Fuente: (Robledo Sacristán, 2011)

¹⁰ Talk: Llamada

¹¹ Tethering: Anclaje a Red: proceso por el cual un dispositivo móvil actúa como pasarela para acceso a la red.

2.1.3 Ventajas



- Código abierto: Android fue liberado bajo la licencia Apache. (Robledo Sacristán, 2011).
- La libertad de código permite ajustar el Sistema Operativo a otros dispositivos. Se encuentra Instalado en Tablets, GPS, relojes, etc. (Robledo Sacristán, 2011).
- Se evita consumo de batería ya que al no utilizar las aplicaciones esta las deja en modo suspensión o las cierra si no están en actividad. (Robledo Sacristán, 2011)

2.1.4 Desventajas

- Al tener varias aplicaciones abiertas genera consumo de batería y como no todas las aplicaciones el Sistema Operativo las cierra hay que hacer uso de otro programa para que las cierre, sin embargo, existen aplicaciones para este fin. (Robledo Sacristán, 2011).
- La batería se agota muy rápido. Utilización de las aplicaciones de manera moderada. (Robledo Sacristán, 2011).
- Poco intuitivo: Por ejemplo, es complicado configurar el dispositivo móvil y esto es por la interfaz de Android. Por lo que se utilizan aplicaciones que asisten en tareas como desinstalar otras aplicaciones. (Robledo Sacristán, 2011)

2.1.5 Arquitectura

Android es un escenario para equipos móviles el cual posee una pila de software donde se incluye un S.O., aplicaciones básicas para el usuario y middleware¹².(García Cancela & Ostos Lobo, 2014)

¹² Middleware: Lógica de intercambio de información entre aplicaciones.

A continuación, se muestra de forma global las capas empleadas en la arquitectura de Android.

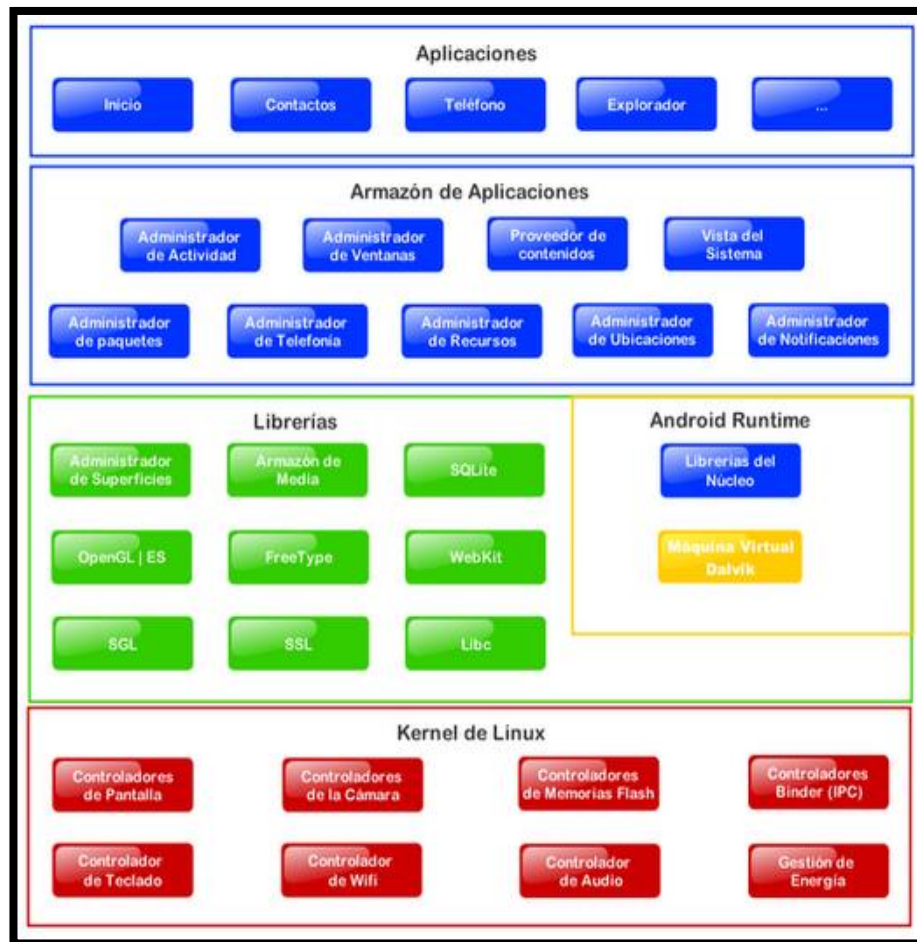


Figura 2.1: Arquitectura de Android
Fuente: (García Cancela & Ostos Lobo, 2014)

A continuación, se realiza una breve descripción de cada capa que compone la arquitectura del sistema operativo Android.

2.1.5.1 Aplicaciones

Incluye las que están por defecto, también las que el usuario agregue ya sean propias o de terceras personas, estas utilizan las API, los servicios y librerías de los anteriores niveles. (García Cancela & Ostos Lobo, 2014)

2.1.5.2 Framework de Aplicaciones

Representa todas las aplicaciones que se desarrolle para el sistema operativo, ya sean las que vienen por defecto, por Google o desarrolladas por terceras empresas, o incluso propias del usuario, utilizan el mismo framework (marco de trabajo) y conjunto de API. (García Cancela & Ostos Lobo, 2014)

Las API más importantes son las siguientes:

- Activity Manager.
- Window Manager.
- Telephone Manager.
- Content Provider.
- View System.
- Location Manager.
- Notification Manager.
- XMPP Service.

2.1.5.3 Librerías

Éstas librerías han sido creadas bajo C/C++ proporcionando a Android la mayor parte de sus capacidades más relevantes. Junto al núcleo basado en Linux, estas constituyen la razón de Android. (García Cancela & Ostos Lobo, 2014)

Entre las librerías más importantes tenemos:

- Librería libc:
- Librería Surface Manager:
- OpenGL/SL y SGL:
- Librería Media Libraries:
- FreeType:
- Librería SSL:
- Librería SQLite:
- Librería WebKit:

2.1.5.4 Tiempo de ejecución de Android

Conforma las Core Libraries¹³, librerías que contienen un conjunto de clases Java. (García Cancela & Ostos Lobo, 2014)

2.1.5.5 Núcleo Linux

Utilizado como capa de abstracción para el hardware en los equipos móviles. Comprende los drivers para que cualquier dispositivo sea utilizado. De manera que un fabricante cuando crea un nuevo componente de hardware para ser utilizado se deberá crear los drivers o librerías necesarios para el funcionamiento, dentro de este kernel¹⁴. (García Cancela & Ostos Lobo, 2014)

2.1.5.6 Herramientas de desarrollo

Android SDK trae consigo variedad de herramientas diseñadas para ayudar en el desarrollo de aplicaciones móviles.

La herramienta más importante es el "Android Emulador" y las "Android Development Tools", pero el SDK incluye también una variedad de otras herramientas para depuración, empaquetado e instalación de aplicaciones en el dispositivo. (Tellez Ariza, 2013)

- Emulador Android

El emulador tiene como finalidad, diseñar y depurar tus aplicaciones en un ambiente igual al que existe en un dispositivo real.

- Android Development Tools Plugin Para Eclipse Ide

El "ADT plugin¹⁵" añade poderosas extensiones al entorno integrado de Eclipse haciendo que la creación y depuración de las aplicaciones sea fácil y rápida. Si por ejemplo desarrollamos usando Eclipse, el "ADT plugin" da un increíble estímulo para el desarrollo de las aplicaciones. A continuación, se indican algunas de sus características:

¹³ Core Libraries: Bibliotecas de Núcleo.

¹⁴ Kernel: Núcleo de una Fusión.

¹⁵ ADT plugin: Herramientas de desarrollo para Android.

1. Proporciona acceso a otras herramientas de desarrollo, desde el entorno de Eclipse IDE. Por ejemplo, ADT permite el acceso a muchas de las capacidades de la herramienta DDMS¹⁶, como tomar fotografías de la pantalla, administrar el direccionamiento de puertos, fijar puntos de quiebre y examinar la información thread¹⁷ y procesos dentro de Eclipse.
 2. Provee un asistente para la creación de proyectos, en donde ayuda a crear todos los directorios y archivos necesarios para crear un nuevo proyecto.
 3. Automatiza y simplifica el proceso de construcción de una aplicación.
 4. Provee un editor de código Android que ayuda a escribir XML válido para el archivo "AndroidManifest.xml".
- Dalvik Debug Monitor Service- Ddms
La "Dalvik Debug Monitor Service" permite administrar los procesos que corren en una instancia de emulador/dispositivo además de asistir en la depuración de ellos. Se usa para terminar la ejecución de un proceso, seleccionar un proceso para depurar, generar reportes a partir de información de bitácoras, examinar el "heap"¹⁸ y la información de "thread", tomar fotografías de la pantalla y mucho más.
 - Android Debug Bridge-Adb
La herramienta nos permite instalar aplicaciones ".apk"¹⁹ en una instancia de emulador/dispositivo y acceder a una instancia de emulador/dispositivo usando comandos de línea.
 - Herramienta Android Asset Packaging-Aapt
La herramienta permite crear archivos ".apk", los cuales contienen las imágenes binarias del código y recursos de tus aplicaciones.

¹⁶ DDMS: Monitor de depuración de la máquina.

¹⁷ Thread: Hilo de Ejecución.

¹⁸ Heap: Estructura de datos de tipo árbol.

¹⁹ APK: Formato de archivo utilizado para la instalación de software en el sistema operativo Android.

- **Android Interface Description Language- Aidl**
Genera código para una interfaz de interproceso, como la que un servicio podría utilizar.
- **Sqlite3**
Proporciona acceso a los archivos de datos “SQLite”²⁰ creados y usados por las aplicaciones Android.
- **Traceview**
Esta herramienta genera una vista gráfica del análisis de información contenida en bitácoras que puede ser generada desde una aplicación Android.
- **Mksdcard**
Ayuda a crear una imagen de disco que se puede usar con el emulador para simular la presencia de una tarjeta de almacenamiento externa.
- **DX**
La herramienta "dx" convierte los archivos de "bytecode"²¹ estándar (".class") en archivos "Android bytecode" (".dex"²²).
- **Activitycreator**
Es un “script” que se puede utilizar para compilar aplicaciones Android.

²⁰ SQLite: Base de datos que utiliza Android.

²¹ Bytecode: Código Binario.

²² .dex: Se utilizan para inicializar y ejecutar aplicaciones desarrolladas para el sistema operativo móvil Android.

2.2 Servidor de Aplicaciones



En informática, se nombra **servidor de aplicaciones** a un servidor en una red que ejecuta ciertas aplicaciones, proporcionando servicios que soportan la ejecución y disponibilidad de las aplicaciones puestas en producción. Comúnmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Generalmente gestiona la mayor parte o la totalidad de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios: son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. (EcuRed, 2015).

El servidor de aplicaciones Jboss WildFly²³, compone una mejora en tecnologías de servidor de aplicaciones. La cual se seguirá usando como esquema inicial de JBoss Enterprise de Red Hat centrándose en las ventajas del middleware²⁴, centralizándose en la rápida elaboración de la especificación Java EE7, que aporta estandarización, simplicidad e integración para entornos de desarrollo en la nube. (EcuRed, 2015)

2.3 Framework JSF-Arquitectura

2.3.1 ¿Qué es JsF?

Java Server Faces es un framework o ambiente de trabajo, destinado a facilitar la construcción y desarrollo de interfaces para aplicaciones basadas en entornos Web. JSF tiene la función de proveer un grupo muy extenso de herramientas desarrolladas para facilitar la creación de interfaces; herramientas tales como: botones, cajas de texto, tablas de datos, etc. Similar a los objetos empleados para la generación de aplicaciones de escritorio (aplicaciones Swing²⁵ por ejemplo). (Loor, 2014)

²³ WildFly: Mosca Salvaje.

²⁴ Middleware: Lógica de intercambio de información entre aplicaciones.

²⁵ Swing: Biblioteca gráfica.

Uno de los puntos fuertes hacia los cuales se dirige la arquitectura JSF es la generación de interfaces que puedan ser consumidas por diferentes tipos de dispositivos clientes. De esta manera, la misma interface puede verse desde una PC de escritorio, o una Tablet PC o un PDA, sin tener que crear nuevas interfaces ni desarrollar diferentes procesos para cada uno de los ambientes antes mencionados. (Loor, 2014)

Otra de las ventajas para los ambientes de desarrollo que presenta JSF es que muchos de los IDE's actuales utilizados para el desarrollo, incorporan muchos de los controles utilizados por el estándar, de manera que el proceso de la generación de una interfaz de aplicación, puede ser llevado a cabo utilizando ambientes visuales, sin tener que escribir todo el código que se requiere para generar los controles necesarios. (Loor, 2014)

2.3.2 Componentes

Jsf introduce dos nuevos términos al mundo de desarrollo de aplicaciones para JAVA:

1. **Managed Bean**

Un Managed Bean²⁶ es un objeto identificado para el ambiente de la aplicación para el cual se describe:

- Una identificación
- Un alcance (scope) que puede ser: Request, Session, Application, etc.
- Propiedades

2. **Backing Bean**

Un Backing Bean es usualmente un Bean común de java que sirve de soporte para un objeto manejado dentro de la aplicación. Pongamos por ejemplo una página JSP en una aplicación orientada a web:

- La página JSP esta especificada como un ManagedBean para la aplicación, con un identificador que la describa para toda la aplicación en general.
- En este archivo JSP se dibujaban todos los controles necesarios para proveer a la página de funcionalidad.

²⁶ Managed Bean: Gestión de clases con sus respectivos constructores.

- Estas páginas tienen asociado un Backing Bean²⁷ que es un Bean de Java. En este Bean se codifican los comportamientos específicos asociados a cada control del Managed Bean representado por la página JSP.

La ventaja de los Backing Beans es que pueden ser compartidos por un mismo Managed Bean, de manera que para diferentes páginas se pueden agrupar comportamientos comunes en un mismo Bean que se comparte con ambos. (Loor, 2014).

2.3.3 Navigation rules

Cuando los Managed Beans expresados en la aplicación se refieren a páginas JSP, se pueden expresar Navigation Rules²⁸ que definen los flujos de proceso entre ellos. Estos Navigation Rules son expresiones utilizadas en los Managed Beans que representan al flujo de navegación entre una página y otra. Básicamente, los Navigation Rules son expresiones literales (String²⁹) que definen una salida desde una página hasta otra. Estas expresiones se escriben en el archivo de definición de JSF (facesconfig.xml). Este es el mismo archivo donde se definen todos los Managed Beans y demás componentes de la aplicación. (Loor, 2014).

2.3.4 Render Kits

La tecnología JSF no está directamente orientada al desarrollo de aplicaciones dirigidas a ambiente Web. Muy por el contrario, la implementación de Managed Beans como páginas JSP es solo una de sus capacidades. En verdad, los Managed Beans pueden estar expresados como Beans que implementan componentes propios de JSF. De esta manera los componentes pueden luego ser convertidos en una salida de interface mediante diferentes tecnologías.

²⁷ Backing Bean: Soporte a las clases.

²⁸ Navigation Rules: Reglas de navegación.

²⁹ String: Valor de un tipo de entrada.

En este punto es en donde entran en escena los Render Kits³⁰. Estos componentes están encargados de generar una salida comprensible para el escenario hacia el cual están orientados. (Loor, 2014). Por ejemplo: XML, WML, SVG, entre otros.

Ejemplo de una aplicación utilizando componentes JSF

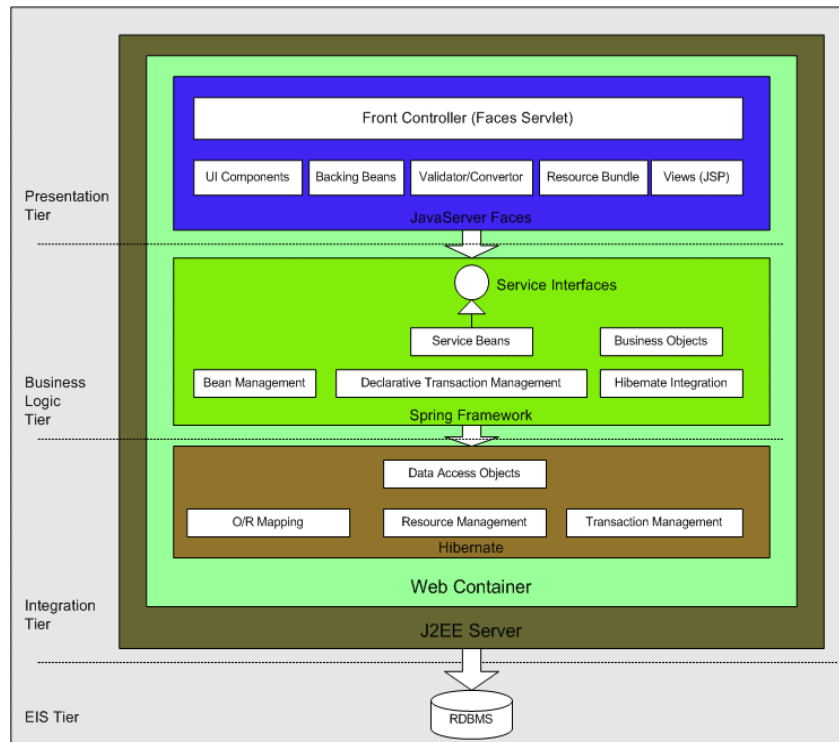


Figura 2.2: Framework JSF-Arquitectura
Fuente: (Loor, 2014)

En el gráfico podemos apreciar como integramos JSF dentro del esquema de una aplicación orientada a ambiente WEB, manteniendo todas las capas típicas de un modelo MVC.

2.3.5 EIS Tier³¹

Es el repositorio de datos que se vaya a utilizar, por ejemplo: una Base de Datos.

2.3.6 Integration Tier

La capa de integración representa los mapeos que se hacen para manejar la persistencia y acceso a datos del repositorio seleccionado, de manera transparente.

³⁰ Render Kits: Componentes personalizados de Java

³¹ Tier: Estandar ANSI/TIA-942 Infraestructura estándar de telecomunicaciones para centros de datos.

Para el caso del ejemplo mostrado se utiliza Hibernate como motor de persistencia, aunque podrían utilizarse otras técnicas como acceso directo por JDBC o EJB 3 Entity Beans³² o TopLink³³ mappings³⁴ o ADF Entity Objects o cualquier otra estrategia similar. (Loor, 2014)

2.3.7 Business Logic Tier

La capa de Lógica de negocios representa la abstracción hecha entre la capa de acceso a datos y las interfaces. En esta capa almacenamos todas las operaciones pertinentes a las actividades propias de la aplicación que estemos desarrollando. En el ejemplo se está utilizando Spring Framework, pero se puede utilizar EJB 3 Session Beans, ADF Application Modules o una combinación de cualquiera de ellos. (Loor, 2014).

2.3.8 Presentation Tier

En esta capa es en donde encontramos la actividad propia de JSF. Podemos apreciar claramente que los Backing Beans de esta capa no están ejecutando ninguna actividad propia del negocio.

Esas actividades las realiza la capa de Business Logic³⁵. Lo que está realizando son solamente las actividades propias de manejo de la interface. Sin embargo, esta arquitectura podría extenderse o reducirse dependiendo de la necesidad presentada por la aplicación. Por ejemplo: En la capa de presentación se podría añadir Struts (Herramienta de soporte) para manejo de procesos inherentes al funcionamiento de la interface que estuvieran fuera de la capacidad prevista para JSF (por ejemplo, conversión de Objetos recibidos desde las capas inferiores). Podría omitirse la capa Business Logic, y manejar los datos recibidos desde la capa de Integración directamente en los Backing Beans y desde ahí a las interfaces. En este caso la aplicación expondría los datos directamente desde la capa de Datos a la interface (Data Driven Application³⁶). (Loor, 2014).

³² Entity Beans: Entidad de las clases.

³³ TopLink: Enlace superior.

³⁴ Mappings: Mapeo.

³⁵ Business Logic: Lógica del negocio.

³⁶ Data Driven Application: Datos de aplicación conducido.

2.4 Sistema Gestor de Base de Datos Postgres

2.4.1 Historia

PostgreSQL³⁷ comienza en 1982 en la Universidad de Berkeley, iniciado y guiado por Michael Stonebraker, fue el inicio en la implementación de un motor de base de datos relacional. En 1985 se retoma el trabajo el cual fue denominado POSTGRES. El proyecto esperaba resolver los problemas con el modelo de base de datos relacional vistos en los años anteriores. Uno de ellos era la imposibilidad del modelo relacional de entender tipos (combinaciones de datos simples que comprendían una única unidad). Conocidos como objetos. Al finalizar el proyecto Andrew Yu y Jolly Chen, comenzaron a trabajar sobre el código y lo que hicieron fue añadir soporte para el lenguaje SQL. (Martinez, 2012)

En 1996 se acoplaron al proyecto, Marc Fournier de Hub.Org Networking Services, Bruce Momjian y Vadim B. Mikheev, quienes trabajaron para estabilizar el código. Para luego cambiar el nombre llamándolo PostgreSQL.(Martinez, 2012)

En 2001, Command Prompt Inc. lanzó Mammoth PostgreSQL, incluyendo PL/Perl, PL/php. (Martinez, 2012)

En agosto de 2007 EnterpriseDB anunció el Postgres Resource³⁸ Center y EnterpriseDB Postgres. Realizando lanzamientos principales al año y lanzamientos menores de reparación de bugs, disponibles bajo la licencia BSD³⁹. (Martinez, 2012).

2.4.2 Características

- Alta concurrencia
- Amplia variedad de tipos nativos
- Claves ajenas
- Disparadores
- Vistas

³⁷ PostgreSQL: PostgreSql Sistema de gestión de base de datos objeto-relacional.

³⁸ Resource: Recursos.

³⁹ BSD: Licencia de software otorgada para los sistemas Berkeley Software Distribution BSD

2.4.3 Funciones

Código que se ejecutan en el servidor escritos en varios lenguajes, desde instrucciones básicas hasta las complejidades de la programación funcional o la orientada a objetos.

Lenguajes que se pueden usar:

- PL/PgSQL (Propio), C, C++, Java PL/Java web, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme.
- Lenguaje para aplicaciones estadísticas R por medio de PL/R

2.4.4 Ventajas

Ideal para desarrollo en tecnologías Web, administración sencilla, su sintaxis SQL es estándar y fácil de entender, multiplataforma, replicación de datos, soporte empresarial entre otras. (Carlos Silva, 2011)

2.5 Lenguaje de Programación JSF

Java Server Face es un framework⁴⁰ o ambiente de trabajo, destinado a facilitar la construcción y desarrollo de interfaces para aplicaciones basadas en entornos Web. JSF tiene la función de proveer un grupo muy extenso de herramientas pre desarrolladas para facilitar la creación de interfaces; herramientas tales como: botones, cajas de texto, tablas de datos, etc. Similar a los objetos empleados para la generación de aplicaciones de escritorio (aplicaciones Swing, por ejemplo). Uno de los puntos fuertes hacia los cuales se dirige la arquitectura JSF es la generación de interfaces que puedan ser consumidas por diferentes tipos de dispositivos clientes. De esta manera, la misma interface puede verse desde una PC de escritorio, o una Tablet PC o un PDA, sin tener que crear nuevas interfaces ni desarrollar diferentes procesos para cada uno de los ambientes antes mencionados. Otra de las ventajas para los ambientes de desarrollo que presenta JSF es que muchos de los IDE's actuales utilizados para el desarrollo, incorporan muchos de los controles utilizados por el estándar, de manera que el proceso de la generación de una interfaz de aplicación, puede ser llevado a cabo utilizando ambientes visuales, sin tener que escribir todo el código que se requiere para generar los controles necesarios. (Gonzales, 2013).

⁴⁰ Framework: Marco de Trabajo.

2.6 Patrón de Diseño - MVC (Modelo, Vista, Controlador).

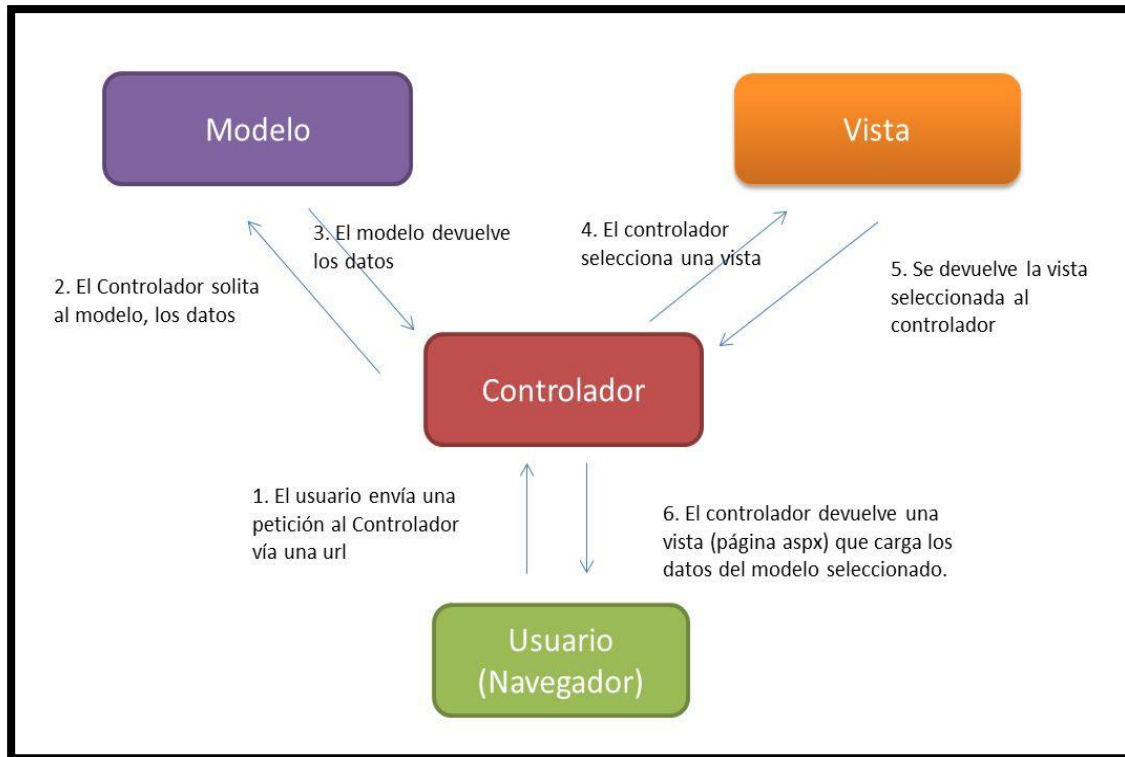


Figura 2.3: Patrón de Diseño (MVC)
Fuente: (Álvarez, 2014)

El patrón de diseño será, **Modelo Vista Controlador (MVC)** que es un esquema de abstracción de desarrollo de software que separa la interfaz de usuario, la lógica del negocio y los datos de una aplicación en módulos.

El patrón de llamada y retorno MVC (según CMU⁴¹), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML⁴² y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. (Álvarez, 2014)

⁴¹ CMU: Universidad Carnegie Mellon.

⁴² HTML: Lenguaje de marcas de hipertexto o lenguaje de programación para desarrollo de páginas web.

2.6.1 Descripción del patrón

- **Modelo:** Es la representación de información específica con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado. (Álvarez, 2014)
- **Vista:** Representa el modelo en un correcto formato para interactuar, usualmente es la interfaz de usuario. (Álvarez, 2014)
- **Controlador:** Invoca peticiones al modelo y probablemente a la vista, responde a eventos, generalmente acciones del usuario. (Álvarez, 2014).

CAPÍTULO III: DESCRIPCIÓN, PROCEDIMIENTO Y FUNCIONAMIENTO

3 Descripción, Procedimiento y Funcionamiento del Sistema

Aquí se detalla una breve descripción de los procedimientos para el desarrollo del sistema y el funcionamiento de acuerdo al problema planteado.

3.1 Visión General

Como visión, este proyecto tiene como objetivo desarrollar e implementar un aplicativo web y móvil, ambos capaces de sustentar la gestión de pedidos de la Distribuidora TierraZoe en donde nuestro objetivo comprende desarrollar un sistema de gestión de pedidos y entrega de productos basado en módulos configurables, que permitan automatizar parte del proceso generado por un cliente: ordenar su pedido, visualizar pedido y recibir pedido.

El módulo de gestión de pedidos está creado para contener el detalle de los pedidos que se encuentren activos mediante la utilización de una base de datos. Dentro de cada pedido se conocerá que productos se ha elegido y sus características. Existirán al menos cuatro tipos de usuarios en donde el administrador estará en la capacidad de modificar en su totalidad la información:

- Administrador: Gestión operativa interna y externa de la aplicación, acceso total.
- Operador: Ingreso de clientes, usuarios, productos, gestión de pedidos, asignación de rutas.
- Vendedor: Entregar pedidos
- Cliente: Creación de pedidos.

3.1.1 Requerimientos funcionales

Describen una interacción entre el sistema y su ambiente, describen cómo debe comportarse el sistema ante determinado estímulo. Son declaraciones de los servicios que debe proporcionar la aplicación, es decir cómo debe comportarse en situaciones específicas en algún caso, también se puede indicar explícitamente lo que el aplicativo no debe hacer.

A continuación, se detallan los requisitos funcionales de cada usuario, con la finalidad de detallar los roles de cada uno de ellos en el proyecto.

1. Usuario Administrador: Acciones que puede realizar:

- Actualiza: Productos, usuarios, clientes, acceso total de la aplicación.
Actualiza cualquier dato en la base de datos para que se pueda ofertar.
- Elimina: Productos, usuarios, clientes, acceso total de la aplicación.
Elimina cualquier dato en la base de datos para que se pueda ofertar.
- Edita: Productos, usuarios, clientes, acceso total de la aplicación.
Edita cualquier dato en la base de datos para que se pueda ofertar.
- Crea: Productos, usuarios, clientes, acceso total de la aplicación.
Crea cualquier dato a la base de datos para que se pueda ofertar.

2. Usuario Administrador: Acciones que no puede realizar:

- Ninguna.

3. Usuario Operador: Acciones que puede realizar:

- Ingreso de clientes.
- Ingreso de productos.
- Ingreso de productos a las bodegas asignadas.
- Ingreso de catálogos.
- Revisión de pedidos.
- Revisión de inventario.
- Asigna rutas.
- Visualiza rutas.
- Genera reportes de pedido.

4. Usuario Operador: Acciones que no puede realizar:

- Modificar el contenido de los pedidos
El operador solo puede visualizar y realizar el cambio de estado de los pedidos generados por el cliente para luego ser despachados.

5. Usuario Vendedor: Acciones que puede realizar:

- Lista los pedidos desde el dispositivo móvil.
- Visualiza las rutas de los pedidos.
- Entrega los pedidos a los clientes.

6. Usuario Vendedor: Acciones que no puede realizar:

- Ingresar o actualizar productos ofertados.

El administrador es el único que tiene los respectivos permisos para ingresar o modificar.

7. Usuario Cliente: Acciones que puede realizar:

- Registrarse.

El cliente deberá estar registrado en el sistema de facturación para obtener los datos y asignarle un usuario y contraseña.

- Crear usuario y contraseña desde el dispositivo móvil.

El cliente al momento de ingresar por el dispositivo móvil el sistema le pedirá por una sola vez que modifique su usuario y contraseña para su ingreso.

- Crear pedido.

El cliente crea un pedido a través del dispositivo móvil.

- Visualizar pedido.

El cliente podrá visualizar el pedido realizado.

- Realizar varios pedidos

El cliente podrá realizar varios pedidos uno a la vez.

8. Usuario Cliente: Acciones que no puede realizar:

- No puede agregar productos nuevos una vez realizado el pedido.

- Modificar o anular el pedido una vez confirmado.

Validados los productos que componen el pedido, este ya no tiene posibilidad de sufrir una alteración o una cancelación.

3.1.2 Requerimientos no funcionales

Describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta, capacidad de almacenamiento.

Los siguientes requerimientos no funcionales no conforman la funcionalidad del sistema, pero son importantes mencionarlos ya que son restricciones que debe cumplir el sistema.

Los requerimientos **no funcionales** ponen límites y restricciones al sistema.

- El sistema debe visualizarse y funcionar correctamente en cualquier navegador.
- El sistema debe cumplir con las respectivas validaciones.
- El sistema debe cumplir con las respectivas seguridades.
- El sistema no debe tardar más de 5 segundos en mostrar los resultados de una búsqueda.
- El sistema será escalable dependiendo de los requerimientos del usuario.
- Ante un fallo en el software del sistema, no se tardará más de 5 minutos en restaurar los datos (en un estado válido) y volver a poner en marcha el sistema.
- La aplicación está desarrollada en software libre.
- La aplicación funcionará sobre PostgreSQL.
- El sistema se debe implementar sobre la infraestructura que posee la empresa.
- La aplicación funcionará sobre cualquier dispositivo móvil que tenga como sistema operativo Android.
- La aplicación estará desarrollada con un modelo MVC.
- El framework de desarrollo utilizado es Java.
- El proceso de desarrollo del sistema y los documentos a entregar deberán ajustarse al proceso y al producto a entregar definidos en el proyecto.

3.2 Definición, procesos y subprocesos a implementar

3.2.1 Procesos y subprocesos a implementar

3.2.1.1 Ambiente físico

- El sistema estará instalado sobre equipos y dispositivos móviles propios de la empresa.

3.2.1.2 Interfaces

- La entrada de los pedidos proviene de uno o varios dispositivos móviles.
- La salida de información va a uno o más dispositivos móviles.

3.2.1.3 Usuarios y factores humanos

- El sistema estará supervisado por un administrador.
- El sistema será usado desde la aplicación web por un usuario para que verifique pedidos y asigne pedidos, usuario destinado para este fin.
- El sistema será utilizado por los usuarios, vendedores y clientes de la empresa.
- El nivel de uso de la aplicación será de acuerdo al rol de cada usuario.
- El sistema será sencillo y amigable de usar por todos quienes lo manejen.
- El usuario dependiendo del rol y con un previo entrenamiento podrá manejar el sistema sin ninguna complejidad.
- El sistema será visualmente amigable y entendible en su uso.

3.2.1.4 Funcionalidad

- El sistema estará dirigido al pedido y entrega de productos a través de dispositivos móviles.
- El sistema gestionará el pedido y la entrega de los productos.
- Existirán varias entradas para realizar un pedido desde la aplicación móvil y desde la aplicación web.
- El sistema debe estar sujeto a futuros cambios dependiendo de los requerimientos.
- El sistema no debe tardar más de 5 segundos en mostrar los resultados de una búsqueda o de gestión.

3.2.1.5 Documentación

- La documentación del aplicativo estará disponible por algunos medios disponibles de información, como manual de usuario como documento y un menú ayuda desde la aplicación web.
- Esta información estará orientada a todos quienes usen la aplicación.

3.2.1.6 Datos

- La entrada de datos será por el teclado y por pantallas móviles.
- Los datos serán enviados y gestionados dependiendo de la prioridad del pedido.
- Los datos deberán tener correlación con los existentes en la base de datos para que se despliegue la lista de productos.

- Los cálculos internos serán precisos a la hora de poner en marcha la aplicación.
- A nivel de base de datos en el almacenamiento habrá un incremento indefinido de acuerdo a la cantidad de registros que se guarden o ejecuten para realizar una transacción.
- Los datos no serán retenidos todo deberá ser conciso y lógicamente direccionado.

3.2.1.7 Seguridad

- Se controla el acceso al sistema de acuerdo al rol.
- Se controla el acceso a la información dependiendo del usuario.
- Los backup se realizarán una vez por semana durante todo el año.
- Se tomarán las debidas precauciones ante una catástrofe natural.

3.2.1.8 Aseguramiento de la calidad

- El aplicativo tendrá consistencia en la interfaz de usuario.
- El aplicativo tendrá protección contra fallos.
- El aplicativo tendrá facilidad de uso.
- La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

3.3 Funcionamiento del sistema

El sistema estará orientado a:

- Personas
- Datos
- Actividades o técnicas de trabajo
- Recursos materiales de comunicación e informáticos.

Estas categorías interactúan para resolver los procesos automáticos y manuales del sistema dando lugar a documentación más elaborada.

3.3.1 Modelo de negocio

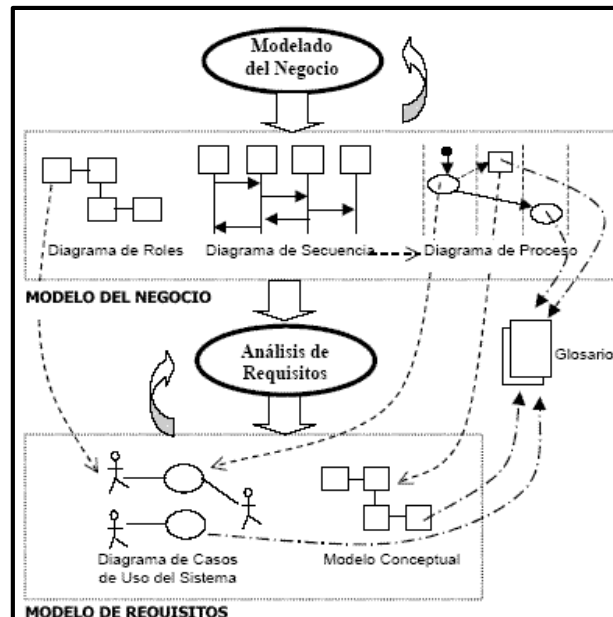


Figura 3.1: Relaciones de trazabilidad entre los modelos de negocio y de requisitos
Fuente: (Joaquin, 2007)

Ilustramos el proceso mediante el ejemplo de la Empresa TierraZoe, una empresa que oferta y distribuye productos bajo demanda. Los objetivos estratégicos de dicha empresa incluyen: Satisfacer los pedidos de clientes, Incrementar los ingresos, Disminuir el tiempo de entrega de los productos. El objetivo Satisfacer los pedidos de clientes puede ser dividido en sub objetivos tales como: Registrar Pedidos de Clientes, Verificar Producto Pedido, Gestionar Almacén, Realizar Pedidos a Proveedores y Entregar los Productos a Tiempo. Éstos serán los objetivos principales a tratar en nuestro modelo del negocio.

3.3.2 Identificación de roles en el entorno del negocio:

1. Rol Cliente-Realiza el Pedido

Identificados los procesos de negocio, se incluye a todos quienes serán parte fundamental del desarrollo del proyecto, en donde éstos desempeñan diferentes roles para continuar con las tareas de los casos de uso según las actividades.

Para lo cual, se debe reconocer los roles que son realizados por quienes conforman la propia empresa o actores externos (ejemplo: clientes).

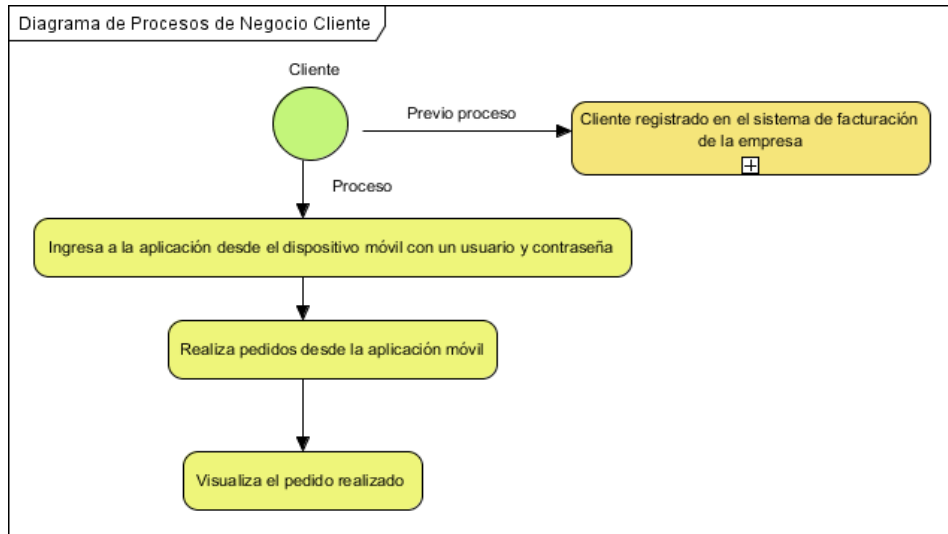


Figura 3.2: Diagrama del negocio-Rol Cliente
Fuente: (Propia)

2. Rol Administrador-Verifica el Pedido

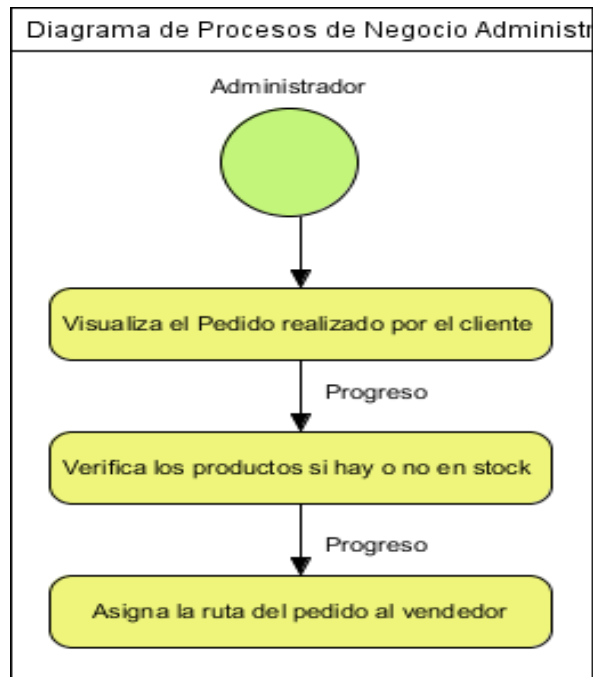


Figura 3.3: Diagrama del negocio-Rol Administrador-Verifica el Pedido
Fuente: (Propia)

3. Rol Vendedor

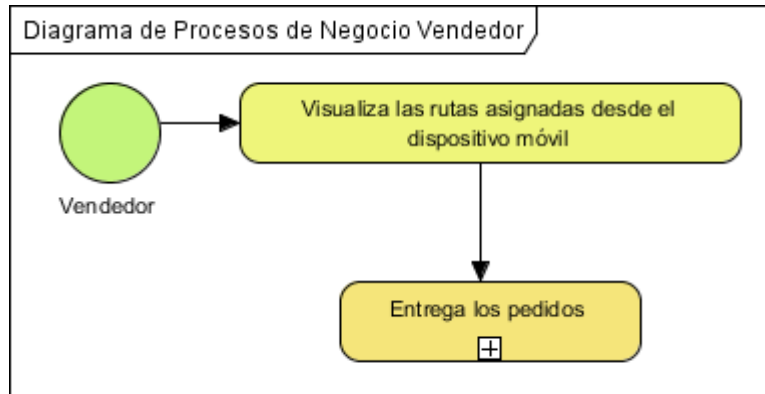


Figura 3.4: Diagrama del negocio-Rol Vendedor
Fuente: (Propia)

3.4 Funcionamiento de procesos

3.4.1 Proceso 1: Funcionamiento de la aplicación

La aplicación móvil está desarrollada con la herramienta Android Studio, esta tiene por defecto una base de datos llamada Sqlite, en donde se almacenarán los datos temporalmente, desde la aplicación móvil el cliente realiza un pedido (petición) hacia el servidor de aplicaciones Jboss Wildfly, este se conecta con la base de datos PostgreSQL para enviar una Respuesta y mostrar los datos.

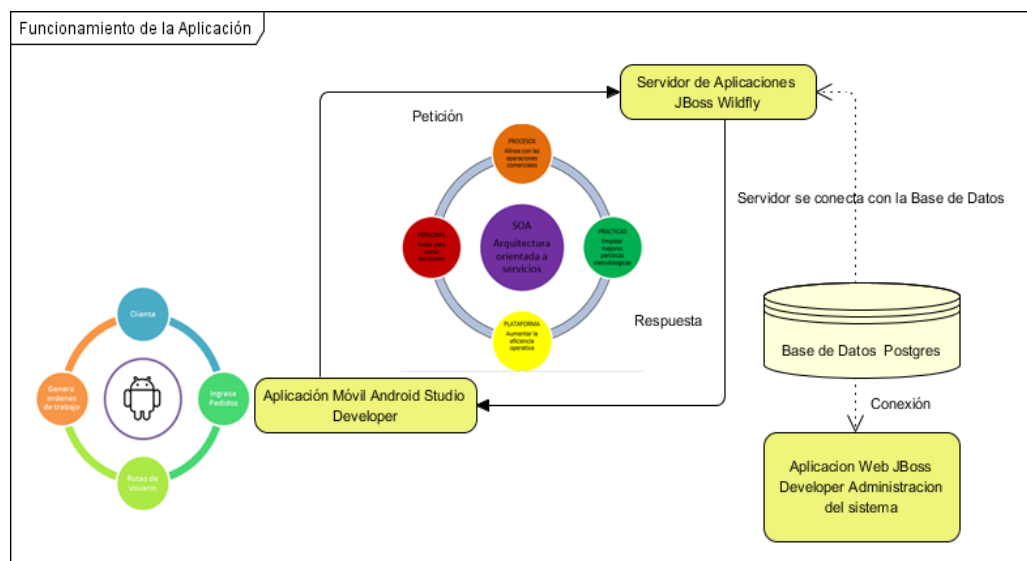


Figura 3.5: Diagrama del negocio/Funcionamiento de la Aplicación
Fuente: (Propia)

3.4.2 Proceso 2: Cliente Realiza Pedidos

El cliente realiza el pedido a través del dispositivo móvil, este se conecta con el servidor de aplicaciones y este a su vez se conecta con la base de datos, y visualiza el pedido en la aplicación web del servidor.

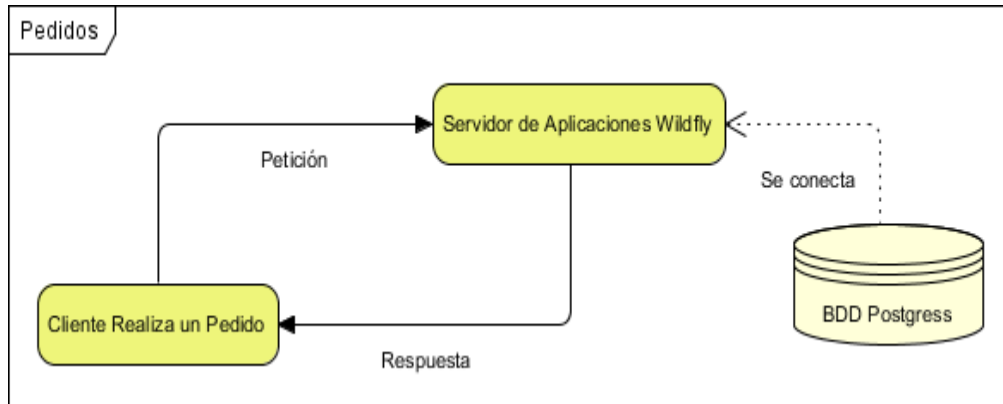


Figura 3.6: Diagrama del negocio/Cliente Realiza Pedidos
Fuente: (Propia)

3.4.3 Proceso 3: Operador/Administrador

El operador o administrador desde la administración del sistema visualiza el pedido realizado por el cliente, luego verifica el stock de los productos, asigna el estado del pedido y la ruta de entrega al vendedor.

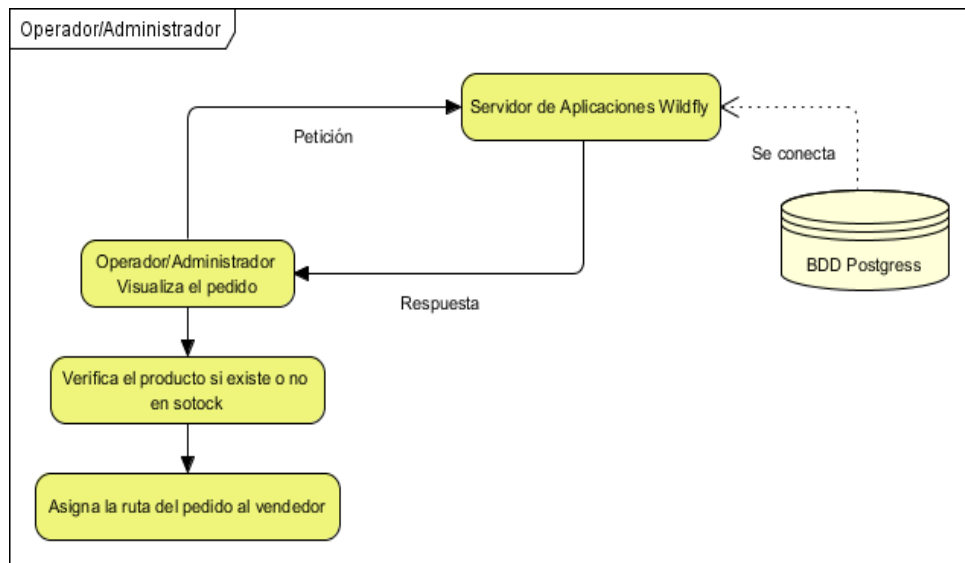


Figura 3.7: Diagrama del negocio/Operador
Fuente: (Propia)

3.4.4 Proceso 4: Vendedor

Vendedor visualiza la ruta asignada desde el dispositivo móvil y realiza la entrega del pedido al cliente.

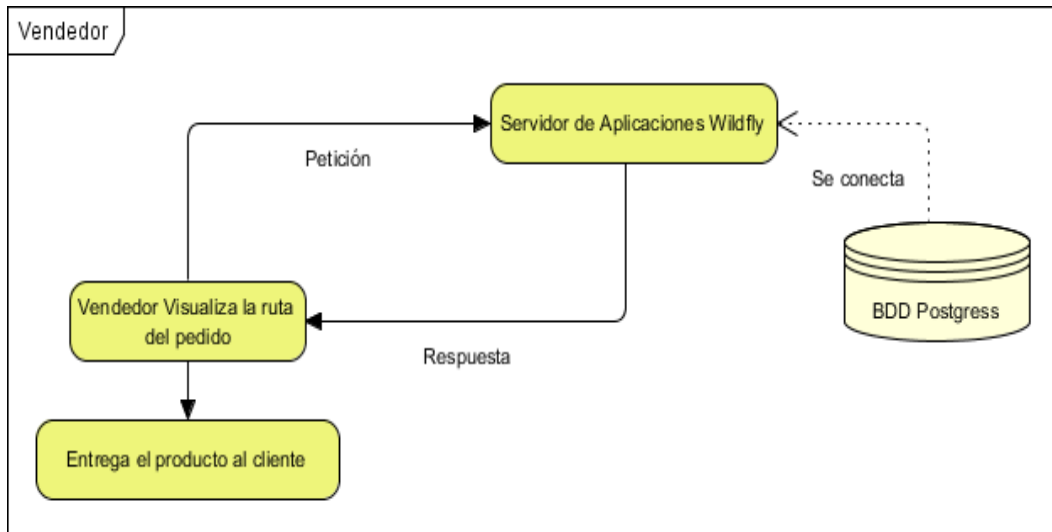


Figura 3.8: Diagrama del negocio/Vendedor
Fuente: (Propia)

3.4.5 Proceso 5: Administrador

Administra en su totalidad la aplicación.

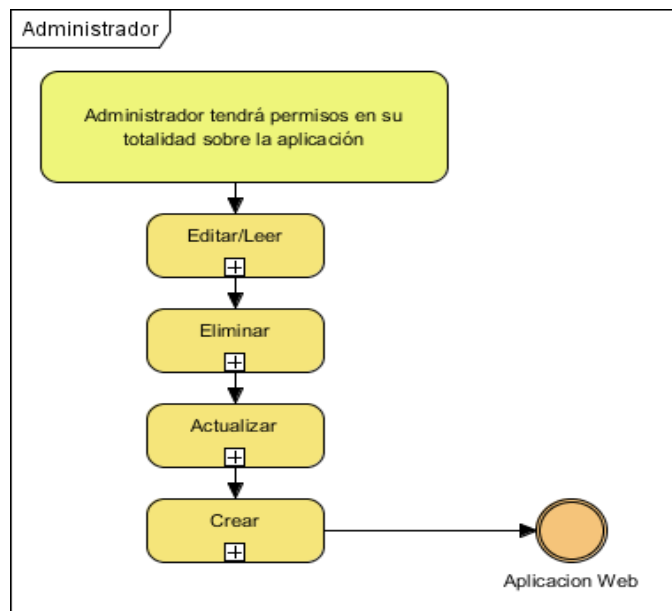


Figura 3.9: Diagrama del negocio/Administrador
Fuente: (Propia)

CAPÍTULO IV: INTEGRACIÓN DE LA APLICACIÓN

4 Integración de la Aplicación: Metodología RUP

4.1 Diseño y Arquitectura del Sistema

En este capítulo, se tratará el diseño del sistema, es decir, se detallarán los componentes de la aplicación y que herramientas se utilizaron.

4.1.1 Selección del entorno de desarrollo: Aplicación WEB

Entorno de desarrollo para el diseño del sistema web.

- IDE-JBoss Developer



JBoss Developer Studio proporciona un soporte para todo el tiempo de vida de desarrollo en una sola herramienta. Incluye un entorno de desarrollo integrado (IDE) certificado y basado en Eclipse para desarrollo, permite probar e implementar aplicaciones web, móviles, empresariales transaccionales y servicios de integración basados en la arquitectura orientada a servicios (SOA). (Developer, 2014)

- Servidor de Aplicaciones Wildfly



Se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Las principales ventajas de los servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de proyectos o aplicaciones.

- Base de Datos PostgreSQL



El aplicativo estará bajo PostgreSQL que es para gestionar la base de datos implementada en el desarrollo del proyecto, este a su vez contiene una interfaz gráfica amigable para crear bases de datos, usuarios, administrar tareas entre otras, en si permite una gestión y configuración del uso de información sobre el sistema.

4.1.2 Selección del entorno de desarrollo: Aplicación Móvil

A continuación, se describen aspectos importantes para el diseño de la aplicación móvil.

- Android Studio: Aplicación Móvil



Android Studio es un entorno de desarrollo integrado (IDE) para la plataforma Android, aquí es donde se desarrollará toda la lógica de la aplicación móvil.

- Servicio Web SOA



La conexión se realiza a través de la Arquitectura Orientada a Servicios (**SOA**), que es la integración a través de sistemas diversos. Utiliza protocolos estándar e interfaces convencionales, llamados Web Services, facilitando el acceso a la lógica de negocios y la información entre diversos servicios. Además, nos brinda los principios y la guía para transformar el conjunto de recursos de TI de la compañía, que son por lo general heterogéneos, distribuidos, inflexibles y complejos, en recursos flexibles, integrados y simplificados, que pueden ser modificados y estructurados para alinearse más fácilmente con los objetivos del negocio. Por lo que se puede decir que SOA no es una herramienta, sino más bien es un conjunto de patrones de construcción de las nuevas aplicaciones de la empresa, más dinámicas y menos dependientes. (Vera, 2013)

4.1.3 Modelo contextual inicial del proyecto

Este diagrama de clases inicial que describe el primer modelo conceptual del proyecto, en el cual se describen los conceptos a identificar y a ser tomados en cuenta para los diagramas UML: Por ejemplo, se puede determinar que un pedido debe ligarse al cliente que lo realiza conjuntamente con los elementos que lo componen. Y este puede ser una clase en el modelo conceptual. Como se muestra en la figura.

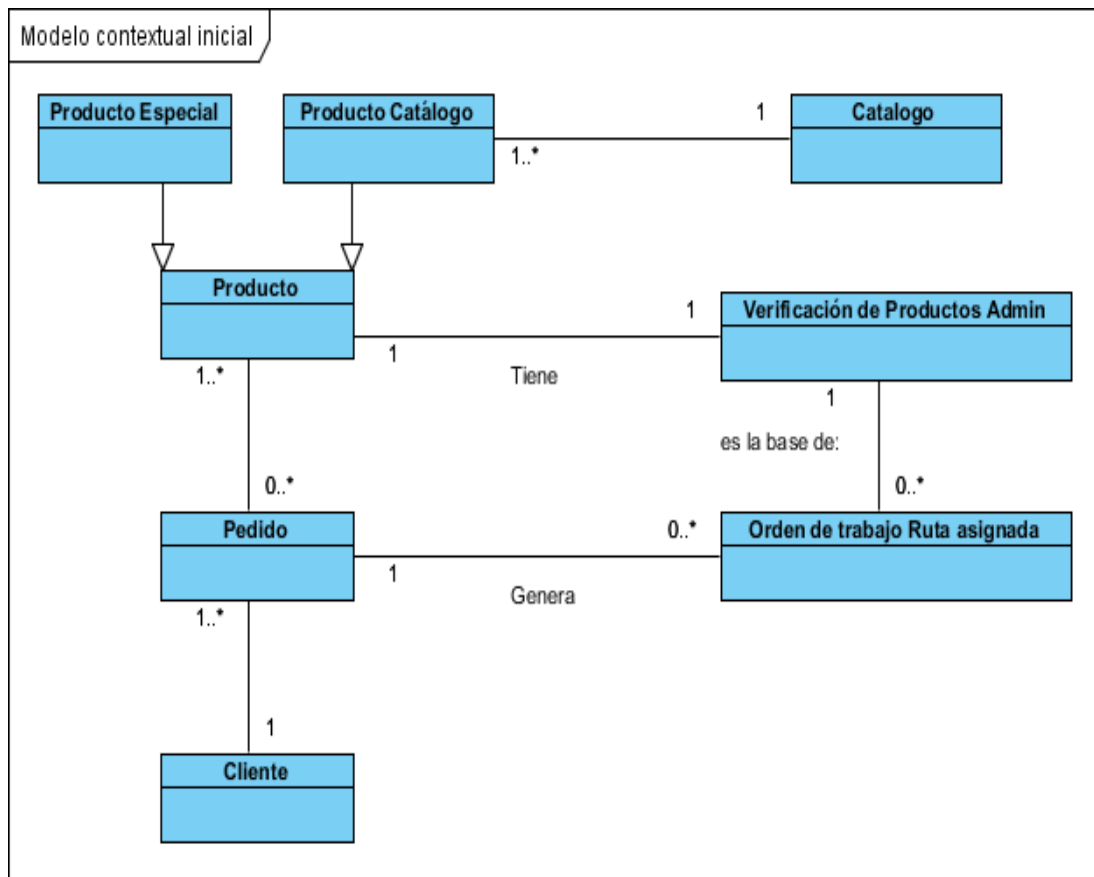


Figura 4.1: Modelo conceptual inicial para el caso de uso del negocio Crear Pedido

Fuente: (Propia)

4.1.4 Arquitectura del sistema

La solución propuesta para la realización del Sistema Informático de Pedidos y Entrega de Productos utilizando Android para la Distribuidora de Alimentos y Productos Veterinarios TierraZoe, consiste en una arquitectura de escritorio y móvil separada en tres capas: capa de presentación, capa de negocio y capa de datos.

El hecho de dividir el sistema en capas permite tener un sistema bien estructurado, donde cada capa es independiente del resto. Con ello consigue un sistema estandarizado, reutilizable y con un bajo acoplamiento.

- **Capa de Presentación**

La capa de presentación es la que se conoce como interfaz de usuario, ayuda a interactuar con la aplicación, permite visualizar la información ingresada por el usuario para evitar errores, así como los resultados de la aplicación. Esta capa debe ser diseñada de forma entendible, amigable y fácil de utilizar para el usuario final. Esta capa está desarrollada en Android Studio.

- **Capa de Negocio**

La capa de negocio también conocida como capa de dominio, es la que mantiene toda la lógica y las reglas del negocio. Se encarga de validar datos, realizar las operaciones y cálculos con los datos y gestionar las interacciones con la capa de datos y la capa de presentación. Esta capa está desarrollada en Java.

- **Capa de Datos**

La capa de datos es la que se encarga de la persistencia y recuperación de los datos desde la base de datos relacional. Para la comunicación entre la capa de presentación y la capa de negocio se utiliza el Servicio Web SOA, que es un puente que permite la comunicación a través del uso de los mensajes entre aplicaciones Android Studio y programas escritos en Java.

La siguiente figura indica los componentes y arquitectura del presente sistema tanto de la aplicación Web como de la Aplicación Móvil.

Arquitectura

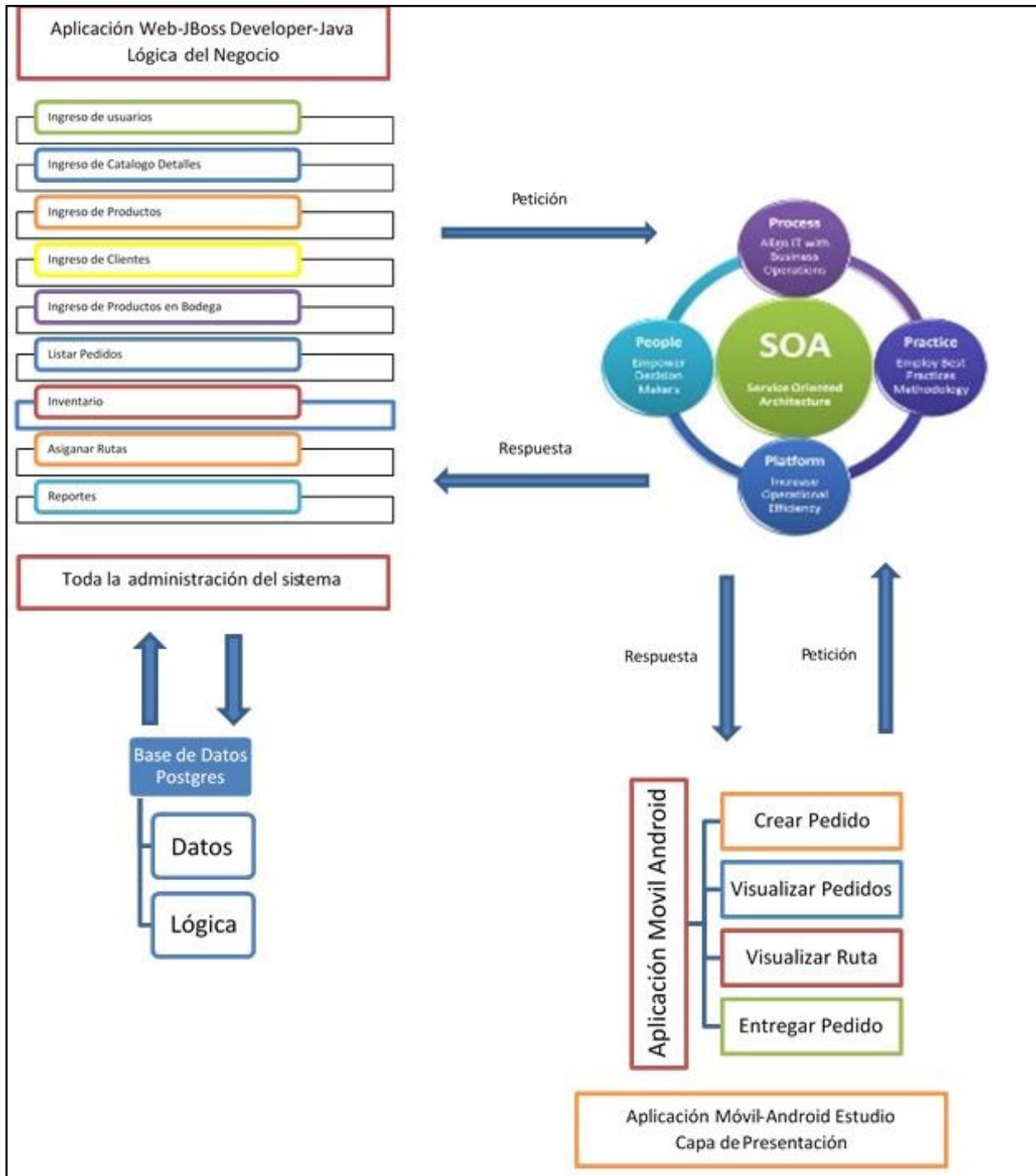


Figura 4.2: Arquitectura del Sistema
Fuente: (Propia)

4.2 Construcción del Sistema: Fase de Inicio

4.2.1 Plan de desarrollo.

Este Plan es una versión inicial preparada para ser adjuntada en la propuesta elaborada como respuesta al proyecto **SISTEMA DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID**, para la Distribuidora de Alimentos y Productos Veterinarios TIERRAZOE.

El proyecto ha sido implementado en base a la metodología RUP con el fin de implantar un compendio inicial para posteriores desarrollos.

La orientación del desarrollo propuesto compone una alineación del proceso RUP de acuerdo a las características del proyecto, recopilando los roles, actividades y documentos entregables. En si este documento o artefacto es parte de la metodología RUP.

4.2.1.1 Propósito

El propósito será suministrar toda la información requerida para llevar un registro y control del proyecto.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto quien se encargará de llevar un control y seguimiento del proyecto.
- Los miembros del equipo de desarrollo quienes se encargarán de entender las funcionalidades a implementar cuando y como hacerlo y que actividades dependen de las mismas.

4.2.1.2 Alcance

Para el Plan de Desarrollo del Software, se basa en la información obtenida por medio del (cliente) del área de ventas de la Distribuidora TIERRAZOE, para hacer una estimación inicial del proyecto, generando la primera versión del artefacto “Visión”, el cual se utilizará para depurar este documento. Consecuentemente, el avance y el seguimiento en cada una de las iteraciones producirán un reajuste de este documento.

4.2.2 Vista General.

4.2.2.1 Propósito, objetivos y alcance

La información inicial indicada en este artefacto se obtuvo de reuniones que se han realizado con el Cliente de la Distribuidora TIERRAZOE, desde el comienzo del proyecto.

A partir de los ordenamientos ya establecidos en el área de Ventas de la Distribuidora TIERRAZOE, y como parte del proceso de automatización requerido, se determina la creación del Sistema DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID, que permita mejorar y automatizar la gestión de las actividades relacionadas con las ventas de la empresa.

El proyecto debe proporcionar una propuesta para el desarrollo de un sistema de pedidos y entrega de productos utilizando la tecnología Android para el mejoramiento de los procesos dentro de la empresa, a continuación se exterioriza los sub-procesos a implementar.

Usuarios.

- Ingreso de usuarios.
- Ingreso de clientes.

Ingreso de catálogo detalles

- Ingreso de catálogo productos.
- Ingreso de catálogo provincias.
- Ingreso de catálogo unidades de medida.
- Ingreso de catálogo bodega.
- Ingreso de catálogo identificación.

Ingreso de productos.

- Ingreso de productos.

Bodega.

- Ingreso de productos en bodega.
- Listar pedidos de bodega.
- Inventario de bodega.
- Impresión.

Rutas.

- Asignar rutas al vendedor.
- Visualizar el detalle de la ruta asignada.

Reporte

- Reporte de pedidos.
- Impresión.

4.2.2.2 Suposiciones y Restricciones

Estas se originan directamente de las conversaciones con los clientes de la Distribuidora TierraZoe.

El sistema será implementado sobre la plataforma Android y Java cumpliendo con los patrones de excelencia actuales para desarrollo de software. Utilizando la metodología RUP para la fase de ingeniería de software, también cumpliendo con la norma para dirección de proyectos PMI⁴³, además de Java y Android para el desarrollo de la aplicación.

Generalmente, la lista de supuestos y limitaciones se extenderá durante el desarrollo del aplicativo, una vez establecido el artefacto “Visión”.

4.2.2.3 Entregables del proyecto

El aplicativo será desarrollado acorde a la metodología RUP, cuyas recomendaciones incluyen la generación de artefactos que se constituyen en los documentos entregables.

Todos los documentos estarán sujetos a variaciones durante el proceso de desarrollo, por lo que una versión concluyente y completa de cada uno de ellos se obtendría al finalizar el proyecto. Por lo que, el resultado de cada iteración estará apuntando en conseguir un nivel aceptable de los documentos.

A continuación, se presenta la lista de artefactos propuesta para el proyecto:

1. Plan de desarrollo de software.

Es el presente documento.

⁴³ PMI: Estándares que proveen las mejores guías para la realización de proyectos.

2. *Visión.*

Este artefacto detalla la visión del proyecto desde el enfoque del cliente, definiendo los requisitos y particularidades del mismo siendo la base en cuanto a los requerimientos del sistema a desarrollarse.

3. *Modelo de casos de uso.*

Representado mediante Diagramas de Casos de Uso (plasma las funcionalidades del sistema y quienes hacen uso de las mismas).

4. *Especificaciones de casos de uso.*

Para aquellos casos de uso cuando no es tan conclusa su explicación, se realiza una descripción detallada por medio de documentos diseñados para este fin, donde comprendan: requisitos no funcionales, post y pre condiciones, flujo de eventos. Además, para casos cuyas circunstancias sean complejas se añadirá un esquema gráfico.

5. *Modelo de análisis y diseño.*

Constituye la elaboración de los casos de uso redirigiendo una representación en términos de análisis, hacia una de diseño orientado hacia un entorno de implementación, según el desarrollo del proyecto.

6. *Modelo de datos.*

Puntualiza la representación lógica de los datos persistentes. Se utiliza un Diagrama de Clases (donde se utiliza un Lenguaje de Modelado Unificado, para representar tablas, claves, entre otros.)

7. *Lista de riesgos.*

Este artefacto presenta una lista de riesgos señalados y relevantes en el desarrollo del proyecto, regulados de acuerdo a la importancia y con tareas específicas de contingencia o para su mitigación.

8. *Material de apoyo al usuario final.*

Conjunto de documentos para el uso y manejo del sistema, que incluyen: Guías de Usuario, Operación, Mantenimiento, Instalación etc.

9. *Producto*

Empacados y recopilados en un CD o memoria USB con los componentes apropiados para facilitar su implementación.

4.2.2.4 Evolución del plan de desarrollo de software

La evolución del plan será periódicamente y se depurará antes del comienzo de cada iteración.

4.2.3 Organización Esquema del Proyecto

4.2.3.1 Copartícipes en el Proyecto

Inicialmente no se incluye al que se designará Responsable del Proyecto, Comité de Control y Seguimiento, además de colaboradores necesarios para suministrar los requerimientos y aprobar el sistema.

El resto de los contribuyentes del proyecto, en donde se considera las fases de Inicio, Elaboración y las Iteraciones necesarias en la fase de Construcción, estará estructurado por Ingenieros en Sistemas:

- **Jefe de Proyecto.** Con experiencia en metodologías de desarrollo, herramientas CASE y notaciones, en particular notación UML y RUP.
- **Analista de Sistemas.** Con conocimientos de UML, y poseer experiencia en sistemas afines a la línea del proyecto
- **Analistas - Programadores.** Con experiencia en el entorno de desarrollo y manejo de herramientas para este fin, este trabajo ha sido encomendado a Daniel Omar Villarreal Caluqui.
- **Ingeniero de Software.** El cual realiza labores de gestión de requerimientos, gestión de configuración, documentación y diseño de datos. Encargado de las pruebas del sistema, realizará la labor de Tester⁴⁴.

4.2.3.2 Interfaces externas

Se define los participantes del proyecto que proporcionarán los requisitos del sistema y entre ellos quiénes serán los responsables de evaluar los artefactos o documentos de acuerdo a cada módulo y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes para especificación y validación de los artefactos generados.

⁴⁴ Tester: Investiga un producto de software con el objetivo de obtener información acerca de su calidad y el valor que representa para quienes lo utilizan.

4.2.3.3 Roles y responsabilidades

A continuación, se describen las principales responsabilidades de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	Gestiona los recursos, los riesgos, planifica y realiza un seguimiento del proyecto, coordina las interacciones con los usuarios, asegura la correcta ejecución de los artefactos del proyecto también supervisa la arquitectura del sistema entre otros.
Analista de Sistemas	Valida los requisitos, interactúa con el cliente y los usuarios mediante entrevistas. Colaboración en el modelo de datos y la preparación de las pruebas funcionales.
Programador	Elaboración de prototipos. Contribución en el modelo de datos y pruebas funcionales.
Ingeniero de Software	Construcción del modelo de datos mediante la gestión de requisitos, elaboración de la documentación y modelos de implementación y desarrollo además prepara las pruebas funcionales entre otros.

Tabla 4.1: Roles y Responsabilidades del Proyecto
Fuente: (Propia)

4.2.4 Gestión del Esquema del Proyecto

4.2.4.1 Plan del proyecto

En este plan se presenta la alineación en fases e iteraciones, conjuntamente con la agenda para llevar a cabo el desarrollo del proyecto.

4.2.4.2 Plan de las fases

Por consiguiente, la tabla adjunta del plan de fases indica el número de iteraciones y el manejo de los tiempos de cada fase.

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	6 semanas
Fase de Elaboración	1	9 semanas
Fase de Construcción	3	14 semanas
Fase de Transición	1	6 semanas

Tabla 4.2: Plan de Fases del Proyecto
Fuente: (Propia)

En la siguiente tabla se indican los hitos que finalizan en cada fase.

Descripción	Hito
Fase de Inicio	En esta fase se establece el contexto de los requerimientos del proyecto o el producto desde el punto de vista del cliente, señalados en el documento Visión. Para lograr esto se debe conocer el ámbito del proyecto, definiendo los límites del mismo para establecer una arquitectura, riesgos, identificación e iteración de actores participativos en el proyecto entre otras.
Fase de Elaboración	En esta fase se elabora un prototipo de arquitectura y se estudian los requerimientos. Además, se desarrolla el plan y se mitigan los riesgos, se completa el diseño también se puede determinar el costo para la realización entre otras. El reconocimiento y aceptación del esquema de la arquitectura del proyecto marca el final de esta fase.
Fase de Construcción	Consiste en desarrollar el proyecto hasta que esté listo para la primera versión, refinando el Modelo de Análisis y Diseño. El problema principal es si la versión está o no lista para las pruebas iniciales en caso de haber aceptación en las pruebas se pasa a la siguiente etapa.
Fase de Transición	En esta fase se traslada el sistema al ambiente del usuario, se realiza la verificación de la versión y se realizan las correcciones necesarias generando la versión final, debe estar completo y en un nivel aceptable, se hace la entrega de informes de instalación, manuales, material de apoyo y capacitación sobre el funcionamiento y manejo del sistema a los usuarios.

Tabla 4.3: Plan de Fases: Hitos del Proyecto
Fuente: (Propia)

4.2.4.3 Objetivos de las iteraciones

Fase	Iteración	Descripción	Hitos Asociados	Riesgos direccionados
Incepción	Iteración Preliminar	Define el modelo del negocio, requerimientos del producto, plan de proyecto.	Revisión de Casos de Uso	Aclarar los requerimientos. Desarrollar planes de proyecto realista y alcance. Determina la viabilidad de los proyectos desde el punto de vista empresarial.
Fase de Elaboración	Desarrollar una arquitectura prototipo.	Completar el análisis y desarrollo de todos los casos de Usos. Desarrollar arquitectura prototipo.	Arquitectura Prototipo	Arquitectura del Software clarificada. Técnicas de mitigación de riesgos. Prototipo para que el usuario revise.
Fase de Construcción	1. Iteración desarrollo Beta	Implementar y probar casos de uso para proveer la versión Beta.	Beta	Todas las características claves y arquitectura prospectiva usadas en la versión Beta. Opinión de los usuarios antes de lanzar la versión del software.
	2. Iteración desarrollo versión inicial	Implementar y probar casos de uso restantes, Corregir defectos en versión Beta, e incorporar los comentarios del usuario. Desarrollar el sistema inicial.	Software	Software revisado por los usuarios. Producto de alta calidad Minimizar defectos Costes de calidad reducido.

	3. Iteración desarrollar módulo completo	Incorporar mejoras a los defectos en la versión inicial. Desarrollar sistema completo.	Software	Liberación rápida asegurando la satisfacción del cliente. Todas las funciones claves proporcionando el módulo completo
Fase de Transición	Sistema	Empaquetar, distribuir, instalar el sistema	Sistema	El sistema debe estar completo en un nivel aceptable de calidad para ser llevado al entorno del usuario con resultados positivos.

Tabla 4.4: Plan de Fases: Iteraciones del Proyecto
Fuente: (Propia)

4.2.4.4 Calendario

A continuación, se muestra la planificación de las tareas importantes del proyecto. Puesto que anteriormente se había mencionado, el proceso de la metodología RUP caracterizada por el desarrollo en equivalente de todas las fases a lo largo del proyecto, por lo que la documentación inicial estará sujeta a cambios de acuerdo al alcance del proyecto e iteración de las fases. La siguiente figura ilustra este enfoque. (Camargo, 2014)

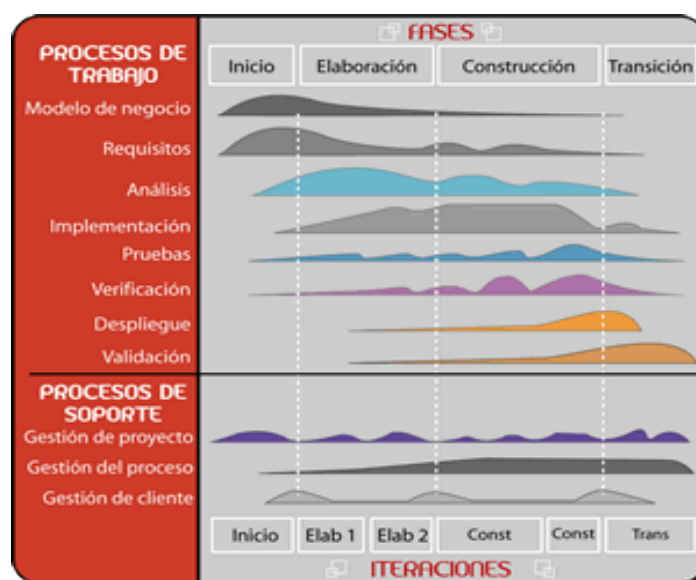


Figura 4.3: Descripción de la metodología RUP
Fuente: (Camargo, 2014)

Para este proyecto la planificación de acuerdo a las fechas establecidas será de la siguiente manera: fecha de aprobación indica cuándo el documento tiene un estado completo suficiente para ser evaluado y aprobado, sin embargo, existe la posibilidad de estar sujeto a cambios posteriormente.

Disciplinas/Artefactos modificados o generados durante la Fase de Inicio	Comienzo	Aprobación
Modelo del negocio		
• Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 4	Semana 5
Requisitos		
• Glosario	Semana 2	Semana 5
• Visión	Semana 2	Semana 4
• Modelo de Casos de Uso	Semana 3	Siguiente fase
• Especificación de Casos de Uso	Semana 5	Siguiente fase
• Especificaciones Adicionales	Semana 5	Siguiente fase
Análisis/Diseño		
• Modelo de Análisis/Diseño	Semana 6	Siguiente fase
• Modelo de Datos	Semana 7	Siguiente fase
Implementación		
• Prototipos de interfaces de usuario	Semana 8	Semana 9
• Modelo de implementación	Semana 10	Siguiente fase
Pruebas		
• Casos de pruebas funcionales	Semana 11	Siguiente fase
Despliegue		
• Modelo de despliegue	Semana 11	Siguiente fase
• Gestión de cambios y configuración	Durante todo el proyecto	
Gestión del proyecto		
• Plan de desarrollo de software en su versión 1.0 y planes de iteraciones	Semana 1	Semana 2
• Ambiente: Durante todo el proyecto		

Tabla 4.5: Roles y Responsabilidades Fase de Inicio
Fuente: (Propia)

Disciplinas/Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelo del negocio		
• Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 4	Aprobado
Requisitos		
• Glosario	Semana 2	Aprobado
• Visión	Semana 2	Aprobado
• Modelo de Casos de Uso	Semana 3	Aprobado
• Especificación de Casos de Uso	Semana 5	Aprobado
• Especificaciones Adicionales	Semana 5	Aprobado
Análisis/Diseño		
• Modelo de Análisis/Diseño	Semana 6	Revisar en cada iteración
• Modelo de Datos	Semana 7	Revisar en cada iteración
Implementación		
• Prototipos de interfaces de usuario	Semana 8	Revisar en cada iteración
• Modelo de implementación	Semana 10	Revisar en cada iteración
Pruebas		
• Casos de pruebas funcionales	Semana 11	Revisar en cada iteración
Despliegue		
• Modelo de despliegue	Semana 11	Revisar en cada iteración
• Gestión de cambios y configuración	Durante todo el proyecto	
Gestión del proyecto		
• Plan de desarrollo de software en su versión 1.0 y planes de iteraciones	Semana 1	Revisar en cada iteración
• Ambiente: Durante todo el proyecto		

Tabla 4.6: Roles y Responsabilidades Fase de Elaboración
Fuente: (Propia)

4.2.5 Seguimiento y control del desarrollo del proyecto

4.2.5.1 Gestión de requerimientos del proyecto

Cada requisito tendrá una serie de particularidades tales como: nivel de importancia, estado, iteración, implementación etc. Donde permitirá hacer un seguimiento de cada uno de ellos. De existir cambio estos serán solucionados mediante una Solicitud de Cambio, las cuales serán analizadas para asegurar la funcionalidad y esencia del sistema.

4.2.5.2 Control de plazos

El calendario o planificación del proyecto tendrá un seguimiento y evaluación semanal o mensual por el director de proyecto y por la Junta de Seguimiento y Control.

4.2.5.3 Control de calidad

En esta etapa el control de calidad se hará de acuerdo a las fallas detectadas en el desarrollo del proyecto con una Solicitud de Cambio, y para garantizar la calidad serán necesarias las guías y checklist⁴⁵ incluidas en RUP.

4.2.5.4 Gestión de riesgos

Se deberá determinar y evaluar en cada iteración todos los riesgos que se asocien al desarrollo del proyecto y se deberá tomar alternativas para solucionarlos.

4.2.5.5 Gestión de configuración

Básicamente en esta etapa se lleva un registro de versiones generadas en el desarrollo del proyecto a través de documentos o artefactos. Además, se incluye las Solicitudes de Cambio y de las alteraciones que se presenten, los cuales deberán ser transmitidos a todos quienes conformen el equipo de trabajo. Al final de cada iteración se establecerá un registro de la versión del estado de cada documento, la cual podrá ser reformada sólo por una Solicitud de Cambio aceptada.

⁴⁵ Checklist: Lista de verificación.

4.2.6 Lista de riesgos

Los riesgos relacionados a este proyecto son evaluados por lo menos una vez en cada iteración y son documentados en esta tabla. Han sido clasificados de acuerdo a su magnitud desde el que podría tener mayor impacto hasta el menor.

Puntaje-Magnitud	Descripción del riesgo e impacto	Estrategia de mitigación y/o plan de contingencia
7	Que la aplicación no esté disponible a tiempo estimado.	Monitorear constantemente el progreso y el cumplimiento de metas en el cronograma.
5	Surgimiento de requerimientos extras del sistema.	Investigación en alternativas externas para mejoramiento de los diversos requerimientos, en pro del proyecto.
3	Responsables de la guía para realizar el Proyecto tanto en desarrollo y documentación estén en desacuerdo en el proyecto. Implica tiempos de retraso en las diferentes fases de la implantación.	Tomar en cuenta varios aspectos del proyecto analizarlos y llegar a un acuerdo antes que abandonen el proyecto. Mantener toda la documentación disponible para una rápida ayuda para el usuario final.

Tabla 4.7: Lista de Riesgos
Fuente: (Propia)

4.3 Fase de Elaboración

Especificación de los casos de uso

Esta sección presenta el modelo de casos de uso del presente sistema, en la cual se muestra las iteraciones y los casos de uso. Este Modelo de casos de Uso es un modelo de las funciones esperadas del sistema y su ambiente, y sirve como un contrato entre el cliente y los desarrolladores. Este modelo es utilizado como una entrada fundamental para las actividades de análisis, diseño y prueba. El rol más importante de un Modelo de Casos de Uso es identificar el comportamiento deseado del sistema con clientes y usuarios finales. En consecuencia, el modelo debe ser fácil de entender.

Se debe tomar en cuenta como punto importante, que los clientes ya se encuentran registrados en el sistema de facturación de la Distribuidora, por lo que el Administrador del sistema de Pedidos y Entrega de Productos, mitigará los datos de los clientes al aplicativo para así poder asignar el rol al cliente y que pueda ingresar desde cualquier dispositivo móvil con su usuario y contraseña a través de su número de ruc, pasaporte o cédula, para luego por única vez ingresar a una ventana de cambio de contraseña en caso de que desee cambiar la misma y para luego comenzar a realizar los pedidos, con esta aclaración se establecen los casos de uso del aplicativo.

Casos de Uso-Administrador/Operador-Aplicación Web

4.3.1 Especificación caso de uso: Ingreso Datos Empresa

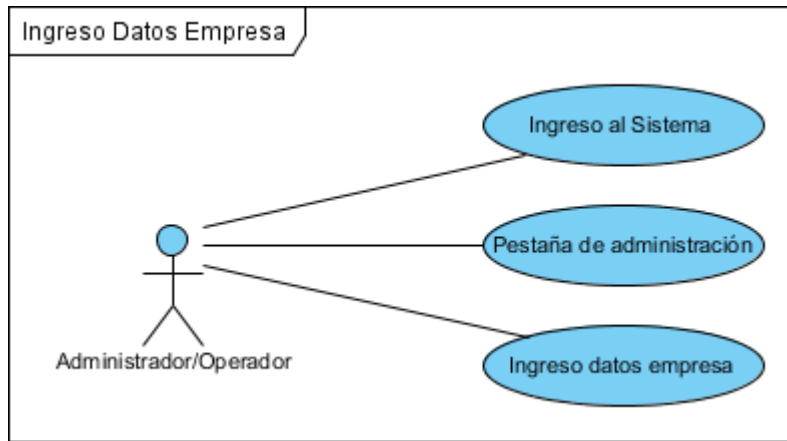


Figura 4.4: CU-001-Ingreso Datos Empresa
Fuente: (Propia)

Caso de uso	Ingreso Datos Empresa
Código	CU-001
Descripción	En este menú se realiza el ingreso de todos los datos de la empresa, ubicación, teléfonos, dirección, etc. Como punto de referencia de la aplicación web.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
1. Se levanta el servidor de aplicaciones wildfly.	
2. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
3. Se selecciona la pestaña Administración.	
4. Se da clic en el botón Empresa.	
5. Se edita los datos de la Empresa.	

Tabla 4.8: CU-001-Ingreso Datos Empresa
Fuente: (Propia)

4.3.2 Especificación caso de uso: Ingreso de Usuarios

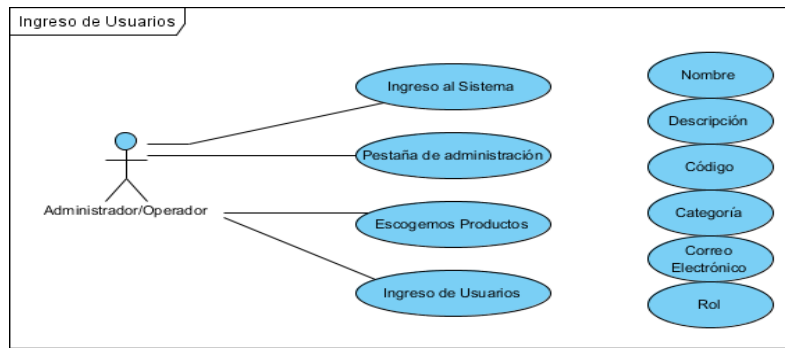


Figura 4.5: CU-002-Ingreso de Usuarios
Fuente: (Propia)

Caso de uso	Ingreso de Usuarios
Código	CU-002
Descripción	En este menú se realiza el ingreso de todos los usuarios de la aplicación, con su respectivo rol: Administrador, Vendedor y Cliente.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Usuario.	
4. Se ingresa los datos de los Usuarios y se asigna el rol.	

Tabla 4.9: CU-002-Ingreso Usuarios
Fuente: (Propia)

Previo al ingreso de los catálogos detalles, se ha creado a nivel de base de datos los catálogos correspondientes de acuerdo a la lógica del negocio, que en este caso son 5 tipos de catálogos: Provincia, Categoría Producto, Bodega, Unidad de Medida e Identificación.

4.3.3 Especificación caso de uso: Ingreso de Catálogos Detalles-Provincia

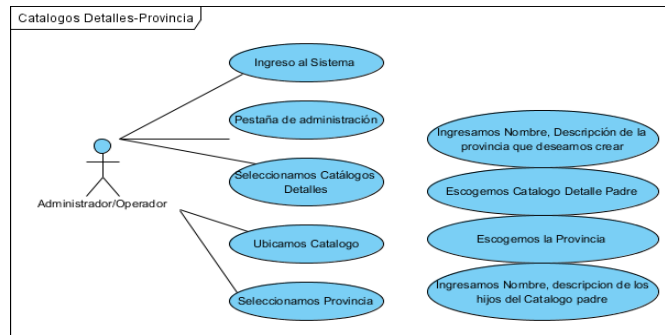


Figura 4.6: CU-003-Ingreso de Catálogos Detalles-Provincia
Fuente: (Propia)

Caso de uso	Ingreso de Catálogos Detalles-Provincia
Código	CU-003
Descripción	En este menú se realiza el ingreso de los catálogos detalles, con sus respectivos Detalle Padre y Detalles Hijos.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Catálogos Detalles.	
4. Seleccionar un catálogo.	
5. Se ingresa nombre y descripción del detalle Padre.	
Flujo Alternativo	
6. Si se desea ingresar hijos al detalle Padre escoger catálogo detalle padre.	
7. Se ingresa nombre y descripción del detalle Hijo.	
8. Guardar	

Tabla 4.10: CU-003-Ingreso de Catálogos Detalles-Provincia
Fuente: (Propia)

4.3.4 Especificación caso de uso: Ingreso de Catálogos Detalles-Categoría Producto.

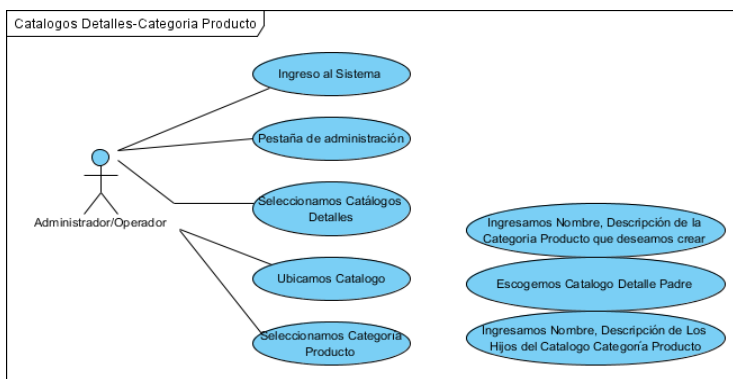


Figura 4.7: CU-004-Ingreso de Catálogos Detalles-Categoría Producto
Fuente: (Propia)

Caso de uso	Ingreso de Catálogos Detalles-Categoría Producto
Código	CU-004
Descripción	En este menú se realiza el ingreso de los catálogos detalles, con sus respectivos Detalle Padre y Detalles Hijos.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1.	Se abre la aplicación Sistema de Pedidos y Entrega de Productos.
2.	Se selecciona la pestaña Administración.
3.	Se da clic en el botón Catálogos Detalles.
4.	Seleccionar un catálogo.
5.	Ingresar nombre y descripción del detalle Padre.
Flujo Alternativo	
6.	Si se desea ingresar hijos al detalle Padre escoger catálogo detalle padre.
7.	Ingresar nombre y descripción del detalle Hijo.
8.	Guardar

Tabla 4.11: CU-004-Ingreso de Catálogos Detalles-Categoría Producto
Fuente: (Propia)

4.3.5 Especificación caso de uso: Ingreso de Catálogos Detalles-Bodega.

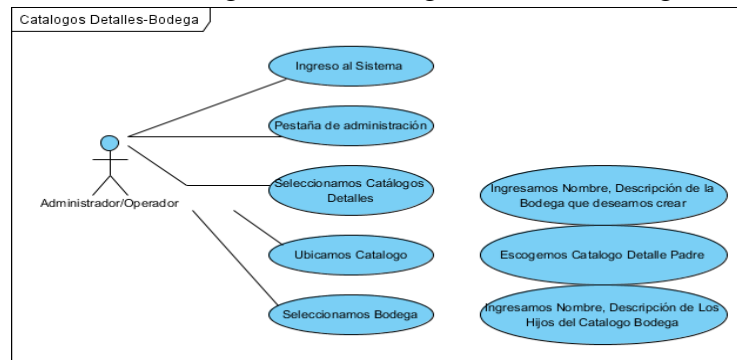


Figura 4.8: CU-005-Ingreso de Catálogos Detalles-Categoría Bodega
Fuente: (Propia)

Caso de uso	Ingreso de Catálogos Detalles-Categoría Bodega
Código	CU-005
Descripción	En este menú se realiza el ingreso de los catálogos detalles, con sus respectivos Detalle Padre y Detalles Hijos.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Catálogos Detalles.	
4. Seleccionar un catálogo.	
5. Ingresar nombre y descripción del detalle Padre.	
Flujo Alternativo	
6. Si se desea ingresar hijos al detalle Padre escoger catálogo detalle padre.	
7. Ingresar nombre y descripción del detalle Hijo.	
8. Guardar	

Tabla 4.12: CU-005-Ingreso de Catálogos Detalles-Categoría Bodega
Fuente: (Propia)

4.3.6 Especificación caso de uso: Ingreso de Catálogos Detalles-Unidad de Medida.

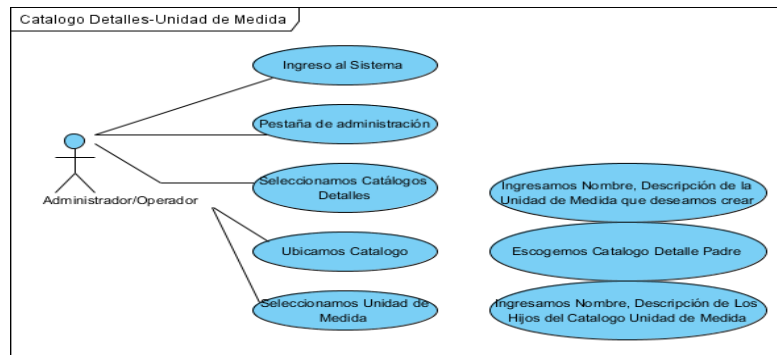


Figura 4.9: CU-006-Ingreso de Catálogos Detalles-Unidad de Medida
Fuente: (Propia)

Caso de uso	Ingreso de Catálogos Detalles-Unidad de Medida
Código	CU-006
Descripción	En este menú se realiza el ingreso de los catálogos detalles, con sus respectivos Detalle Padre y Detalles Hijos.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Catálogos Detalles.	
4. Seleccionar un catálogo.	
5. Ingresar nombre y descripción del detalle Padre.	
Flujo Alternativo	
6. Si se desea ingresar hijos al detalle Padre escoger catálogo detalle padre.	
7. Ingresar nombre y descripción del detalle Hijo.	
8. Guardar	

Tabla 4.13: CU-006-Ingreso de Catálogos Detalles-Unidad de Medida
Fuente: (Propia)

4.3.7 Especificación caso de uso: Ingreso de Catálogos Detalles-Identificación.

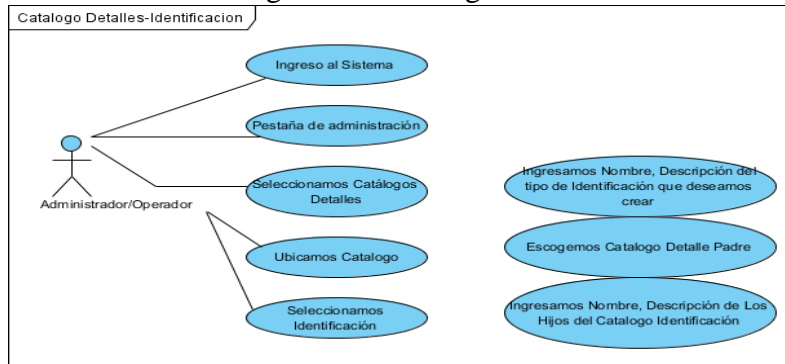


Figura 4.10: CU-007-Ingreso de Catálogos Detalles-Identificación
Fuente: (Propia)

Caso de uso		Ingreso de Catálogos Detalles-Identificación	
Código		CU-007	
Descripción		En este menú se realiza el ingreso de los catálogos detalles, con sus respectivos Detalle Padre y Detalles Hijos.	
Autor/es		Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.	
Precondiciones		Debe estar subido el servidor de aplicaciones wildfly.	
Post Condición		Ninguna	
Flujo del evento			
Flujo Principal			
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.			
2. Se selecciona la pestaña Administración.			
3. Se da clic en el botón Catálogos Detalles.			
4. Seleccionar un catálogo.			
5. Ingresar nombre y descripción del detalle Padre.			
Flujo Alternativo			
6. Si se desea ingresar hijos al detalle Padre escoger catálogo detalle padre.			
7. Ingresar nombre y descripción del detalle Hijo.			
8. Guardar			

Tabla 4.14: CU-007-Ingreso de Catálogos Detalles-Unidad de Medida
Fuente: (Propia)

4.3.8 Especificación caso de uso: Ingreso de Productos

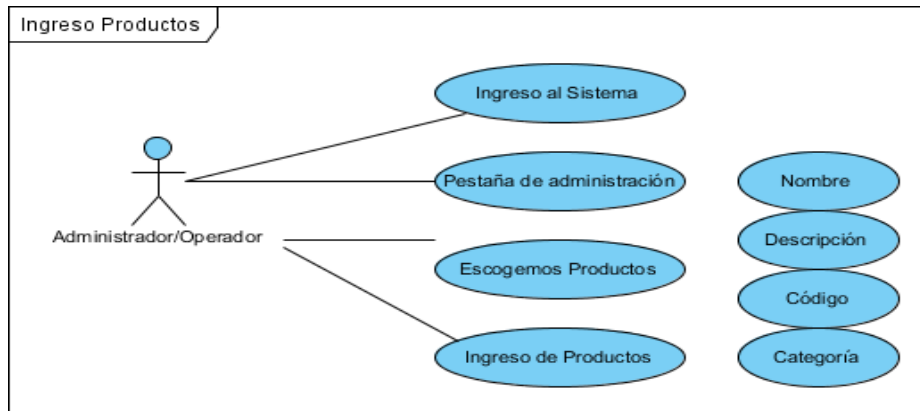


Figura 4.11: CU-008-Ingreso de Productos
Fuente: (Propia)

Caso de uso	Ingreso de Productos
Código	CU-008
Descripción	En este menú se realiza el ingreso de los Productos, con sus respectivos datos y asignación de categoría producto.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Productos.	
4. Ingresar nombre, descripción y código del Producto.	
5. Ingresar la categoría a donde pertenecerá ese Producto.	
6. Guardar	

Tabla 4.15: CU-008-Ingreso de Catálogos Detalles-Unidad de Medida
Fuente: (Propia)

4.3.9 Especificación caso de uso: Ingreso de Clientes

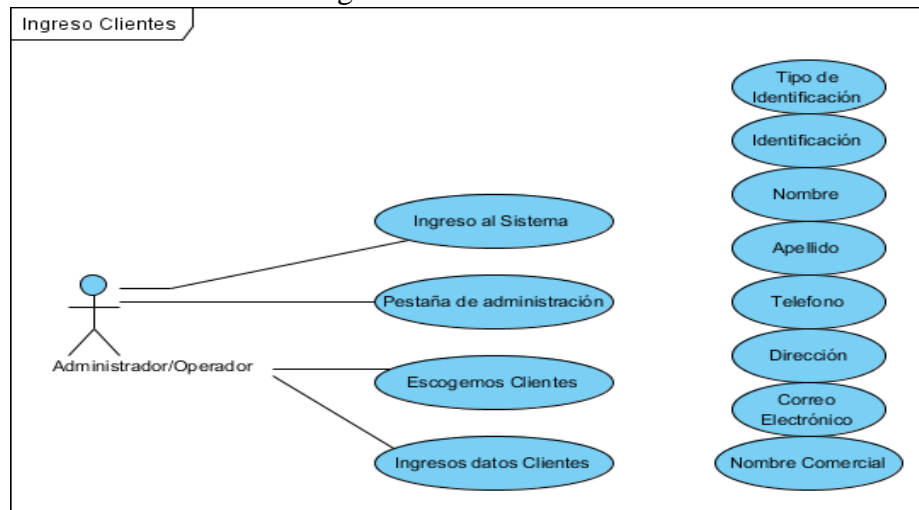


Figura 4.12: CU-009-Ingreso de Clientes
Fuente: (Propia)

Caso de uso	Ingreso de Clientes
Código	CU-009
Descripción	En este menú se realiza el ingreso de los Clientes, con sus respectivos datos.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Administración.	
3. Se da clic en el botón Clientes.	
4. Ingresar Tipo de Identificación, Nombre, Apellido, Teléfono, Dirección, Correo Electrónico, Nombre Comercial.	
5. Guardar	

Tabla 4.16: CU-009-Ingreso de Clientes
Fuente: (Propia)

4.3.10 Especificación caso de uso: Ingreso de Productos en Bodega.

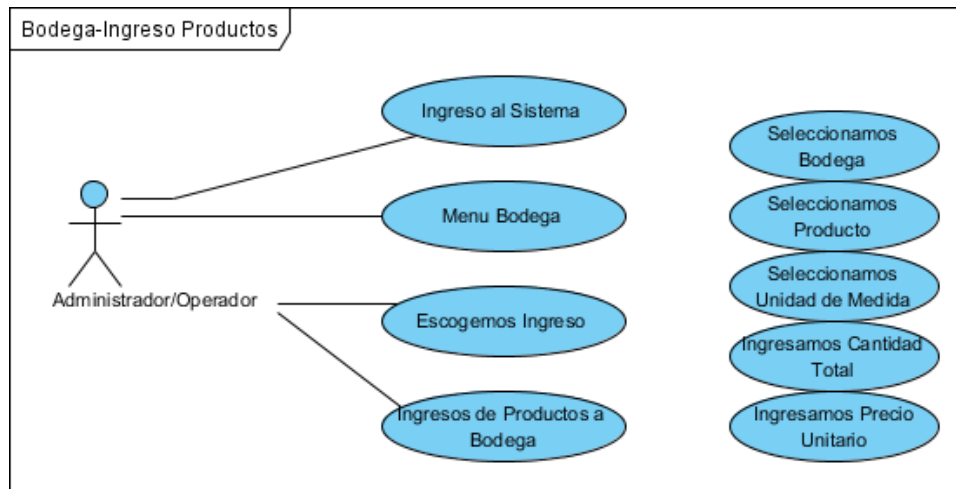


Figura 4.13: CU-010-Ingreso de Productos en Bodega

Fuente: (Propia)

Caso de uso	Ingreso de Productos en Bodega
Código	CU-010
Descripción	En este menú se realiza el ingreso de los Productos en Bodega, con su respectivo precio y cantidad en stock.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Bodega.	
3. Se da clic en el botón Ingreso.	
4. Ingresar: Bodega, Producto, Unidad de Medida, Cantidad Total y Precio U.	
5. Guardar	

Tabla 4.17: CU-010-Ingreso de Productos en Bodega

Fuente: (Propia)

4.3.11 Especificación caso de uso: Visualización de Lista de Pedidos.

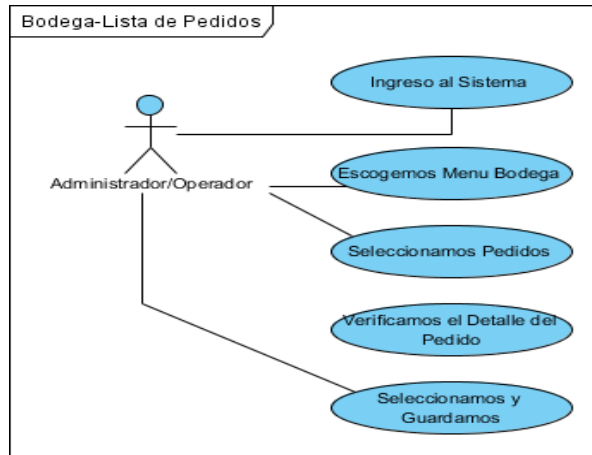


Figura 4.14: CU-011-Lista de Pedidos
Fuente: (Propia)

Caso de uso	Lista de Pedidos
Código	CU-011
Descripción	Se realiza la verificación del detalle del pedido y se selecciona para ingresar el pedido.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Bodega.	
3. Se da clic en el botón Pedidos.	
4. Verificar el detalle del pedido.	
5. Una vez verificado el detalle del pedido se selecciona para ingresar dicho pedido, para luego asignar la ruta al vendedor y que realice la entrega del pedido.	
6. Guardar	

Tabla 4.18: CU-011-Lista de Pedidos
Fuente: (Propia)

4.3.12 Especificación caso de uso: Inventario-Bodega.

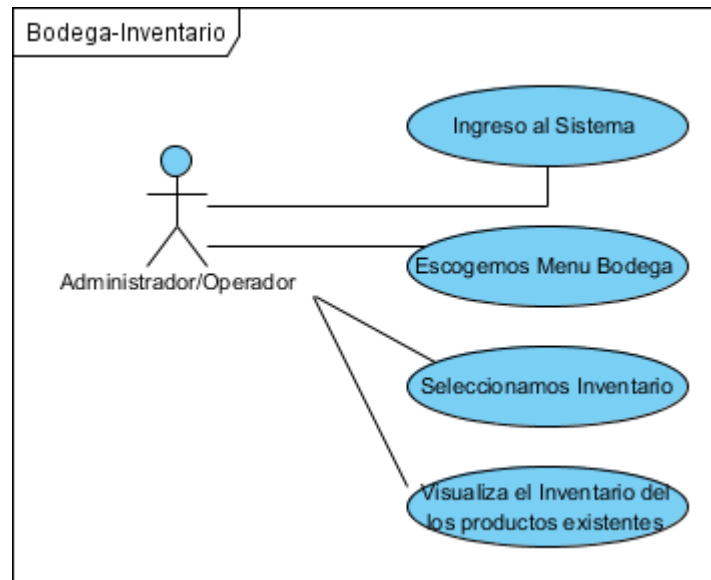


Figura 4.15: CU-012-Inventario-Bodega
Fuente: (Propia)

Caso de uso	Inventario Bodega
Código	CU-012
Descripción	Se visualiza el inventario de los productos existentes.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Bodega.	
3. Se da clic en el botón Inventario.	
4. Se visualiza el inventario de los productos existentes.	

Tabla 4.19: CU-012-Inventario-Bodega
Fuente: (Propia)

4.3.13 Especificación caso de uso: Asignar Ruta.

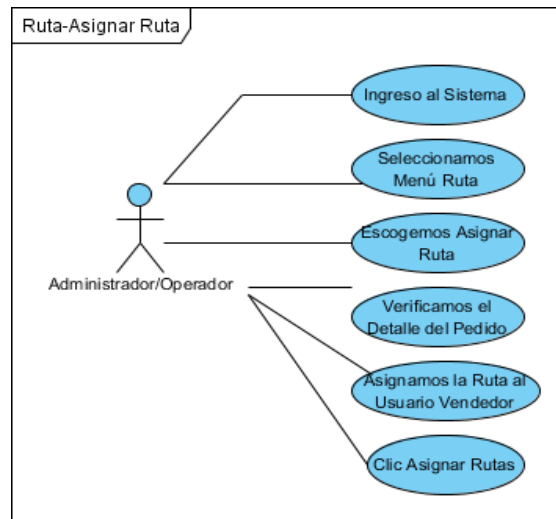


Figura 4.16: CU-013-Ruta-Asignar Ruta
Fuente: (Propia)

Caso de uso	Asignar Ruta
Código	CU-013
Descripción	Se asigna al vendedor la Ruta del pedido generado por el cliente, previa verificación del detalle del pedido.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Rutas.	
3. Dar clic en el botón Asignar Ruta.	
4. Seleccionar al Usuario Vendedor.	
5. Clic en el Botón Asignar la Ruta	

Tabla 4.20: CU-013-Asignar Ruta
Fuente: (Propia)

4.3.14 Especificación caso de uso: Ver Ruta.

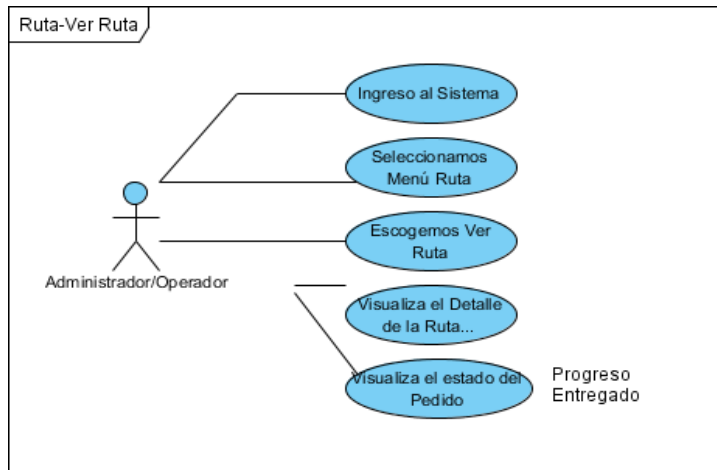


Figura 4.17: CU-014-Ruta-Ver Ruta
Fuente: (Propia)

Caso de uso	Ver Ruta
Código	CU-014
Descripción	Se visualiza el detalle de la ruta y el estado del pedido.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Rutas.	
3. Dar clic en el botón Ver Ruta.	
4. Se visualiza el detalle de la ruta y el estado del pedido	

Tabla 4.21: CU-014-Ver Ruta
Fuente: (Propia)

4.3.15 Especificación caso de uso: Reporte-Pedidos.

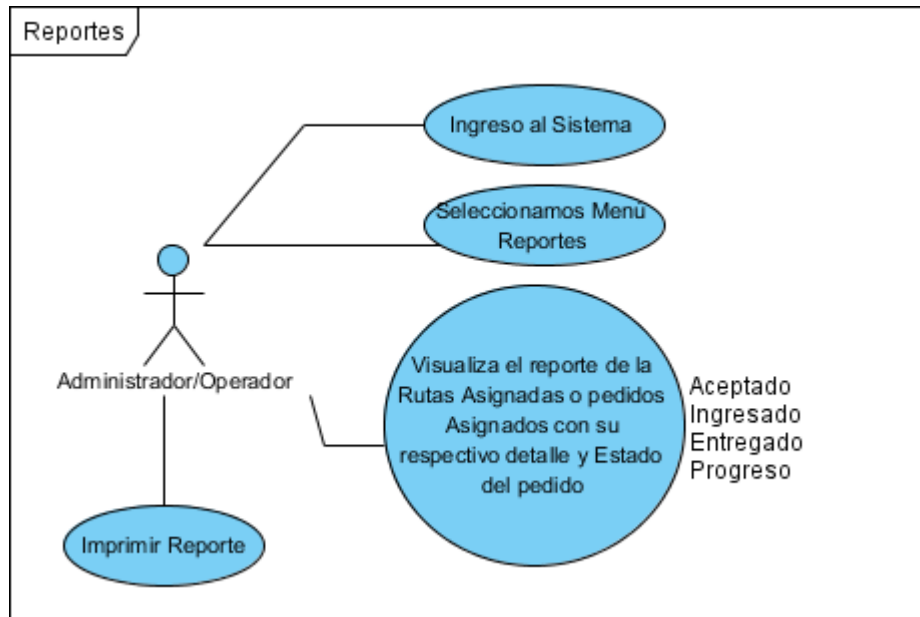


Figura 4.18: CU-015-Reporte-Pedidos
Fuente: (Propia)

Caso de uso	Reporte-Pedidos
Código	CU-015
Descripción	Se genera el reporte del detalle de la ruta y el estado del pedido.
Autor/es	Usuario - Personal de la Distribuidora TierraZoe, encargados del sistema.
Precondiciones	Debe estar subido el servidor de aplicaciones wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se abre la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se selecciona la pestaña Reportes.	
3. Se genera el reporte del detalle de la ruta y el estado del pedido.	

Tabla 4.22: CU-015-Reporte-Pedidos
Fuente: (Propia)

Casos de Uso-Cliente/Vendedor-Applicación Móvil

4.3.16 Especificación caso de uso: Cliente-Crear Pedido

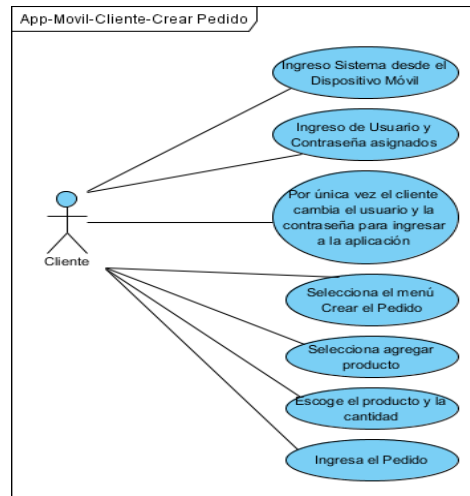


Figura 4.19: CU-016-Cliente-Crear Pedido-App-Móvil
Fuente: (Propia)

Caso de uso	Crear Pedido
Código	CU-016
Descripción	El cliente desde el dispositivo móvil crea el pedido.
Autor/es	Cliente de la Distribuidora TierraZoe.
Precondiciones	Debe estar subido el servidor de aplicaciones Wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se ingresa desde el dispositivo móvil a la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se ingresa con su usuario y contraseña.	
3. Se selecciona la opción Crear Pedido.	
4. Se ingresa los productos.	
5. Se ingresa la Cantidad.	
6. Finalmente, clic en Ingresar.	

Tabla 4.23: CU-016-Cliente-Crear Pedido-App-Móvil
Fuente: (Propia)

4.3.17 Especificación casos de uso: Cliente-Visualizar Pedidos Ingresados

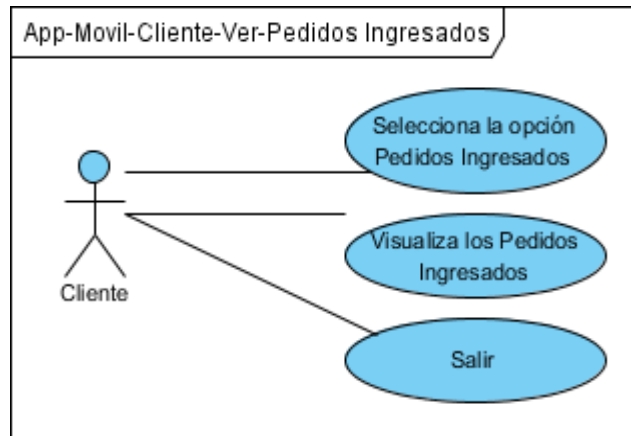


Figura 4.20: CU-017-Cliente-Visualizar-Pedidos Ingresados-App-Móvil
Fuente: (Propia)

Caso de uso		Visualizar Pedidos Ingresados	
Código	CU-017		
Descripción	El cliente desde el dispositivo móvil visualiza el pedido ingresado.		
Autor/es	Cliente de la Distribuidora TierraZoe.		
Precondiciones	Debe estar subido el servidor de aplicaciones Wildfly.		
Post Condición	Ninguna		
Flujo del evento			
Flujo Principal			
1. Se ingresa desde el dispositivo móvil a la aplicación Sistema de Pedidos y Entrega de Productos.			
2. Se ingresa con su usuario y contraseña.			
3. Se selecciona la opción Pedidos Ingresados.			
4. Se visualiza los pedidos ingresados.			
5. Regresar.			

Tabla 4.24: CU-017-Cliente-Visualiza-Pedidos Ingresados-App-Móvil
Fuente: (Propia)

4.3.18 Especificación caso de uso: Vendedor-Visualiza y Entrega Pedidos

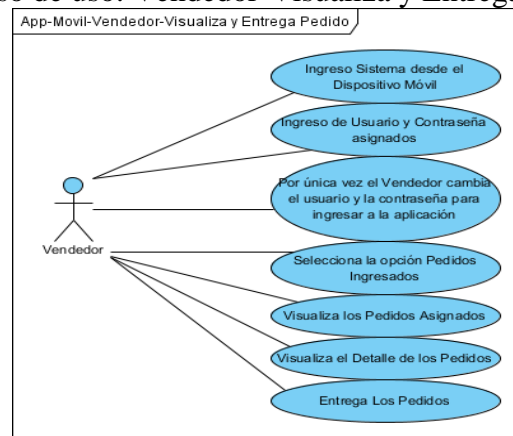


Figura 4.21: CU-018-Vendedor-Visualiza y Entrega Pedidos-App-Móvil
Fuente: (Propia)

Caso de uso	Visualiza y Entrega Pedidos
Código	CU-018
Descripción	El vendedor desde el dispositivo móvil visualiza el detalle del pedido ingresado y hace la respectiva entrega.
Autor/es	Vendedor de la Distribuidora TierraZoe.
Precondiciones	Debe estar subido el servidor de aplicaciones Wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Se ingresa desde el dispositivo móvil a la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Se ingresa con su usuario y contraseña.	
3. Se selecciona la opción Pedidos Ingresados.	
4. Se visualiza los pedidos ingresados.	
5. Visualizar el detalle del pedido.	
6. Realizar la respectiva entrega.	
7. Dar clic en Entregar.	
8. El Pedido se entrega y en la aplicación web sale pedido entregado.	

Tabla 4.25: CU-018-Vendedor-Visualiza y Entrega Pedidos-App-Móvil
Fuente: (Propia)

4.3.19 Especificación caso de uso: Vendedor-Crear Cliente

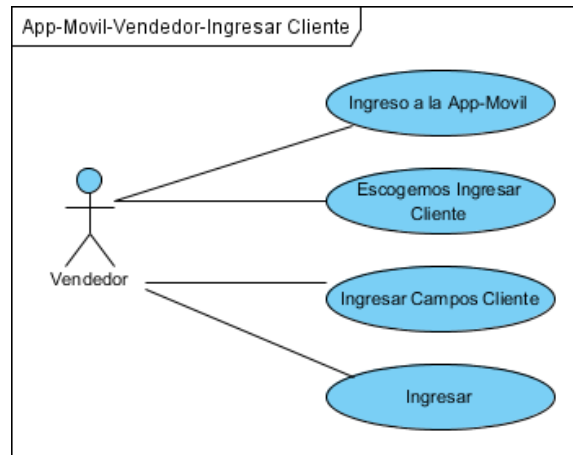


Figura 4.22: CU-019-Vendedor-Ingresar Clientes -App-Móvil
Fuente: (Propia)

Caso de uso	Vendedor-Ingresar Clientes
Código	CU-018
Descripción	El vendedor desde el dispositivo móvil podrá ingresar clientes.
Autor/es	Vendedor de la Distribuidora TierraZoe.
Precondiciones	Debe estar subido el servidor de aplicaciones Wildfly.
Post Condición	Ninguna
Flujo del evento	
Flujo Principal	
1. Ingresar desde el dispositivo móvil a la aplicación Sistema de Pedidos y Entrega de Productos.	
2. Ingresar con su usuario y contraseña.	
3. Seleccionar la opción Ingresar Clientes.	
4. Ingresar todos los campos.	
5. Verificar.	
6. Ingresar.	

Tabla 4.26: CU-019-Vendedor-Ingresar Clientes -App-Móvil
Fuente: (Propia)

4.4 Fase de construcción

VISTA LÓGICA

4.4.1 Diagrama de Clases

Debido a que más y más clases son añadidas a un modelo, una representación textual de las clases no es suficiente. En un sistema Orientado a Objetos, su visión lógica se construye, fundamentalmente, definiendo las clases. Los objetos identificados en el dominio de aplicación son agrupados en clases, cada una de las cuales representa una categoría de objetos que tienen iguales propiedades (atributos) y comportamientos (operaciones, métodos o servicios). La clase como componente principal del sistema, y las relaciones entre ésta y otras clases, hacen posible la estructuración lógica de un sistema Orientado a Objetos. Los Diagramas de Clase son creados para proveer una visualización de algunas o todas las clases en el modelo.

En UML (Unified Modeling Language), los diagramas que permiten visualizar la descomposición del sistema en clases mostrando su estructura genérica, se denominan Diagramas de Clase.

El presente sistema está diseñado para realizar pedidos desde cualquier dispositivo móvil con sistema operativo Android, todos estos pedidos se administrarán a través de la aplicación web instalada en un servidor de la empresa para gestionar los pedidos y entregar los mismos a través de rutas asignadas a los vendedores, que al igual que el cliente crea su pedido, el vendedor también podrá visualizar las rutas asignadas de los pedidos a través de su dispositivo móvil asignado.

A continuación, se presenta el diagrama de clase de la aplicación.

DIAGRAMA DE CLASES

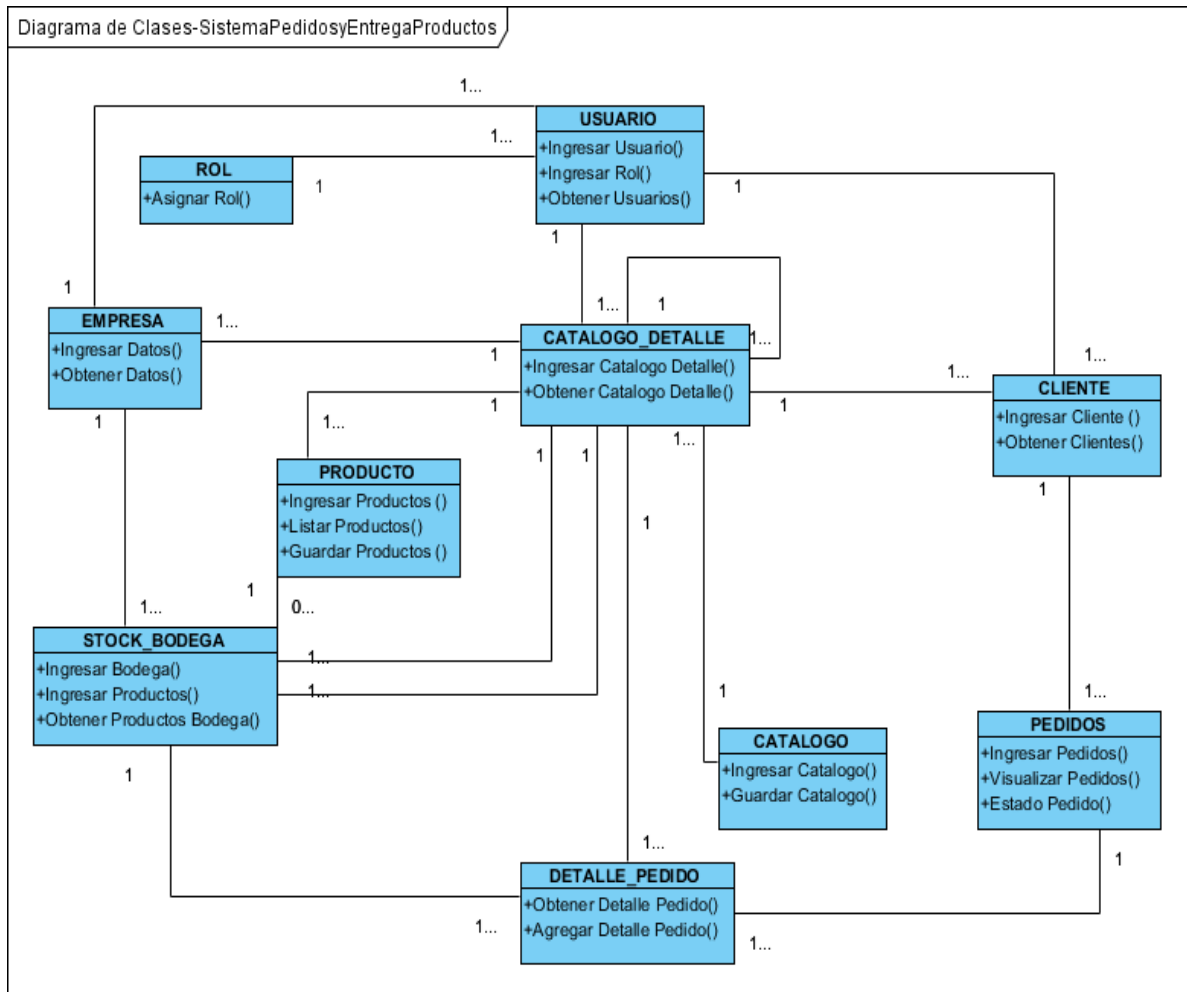


Figura 4.23: Diagrama de Clases-SPEP-UA
Fuente: (Propia)

4.4.2 Diagrama Entidad Relación

La base de datos es el lugar donde la información quedará almacenada tras la interacción entre el usuario y el sistema. Esta sin duda es una pieza clave de la aplicación puesto que, sin un gestor de base de datos no se podría dejar constancia de los posibles cambios producidos.

En el diagrama entidad-relación con sus tablas y sus relaciones a implementar en el Sistema Informático de Pedidos y Entrega de Productos utilizando Android es el siguiente.

DIAGRAMA ENTIDAD RELACIÓN

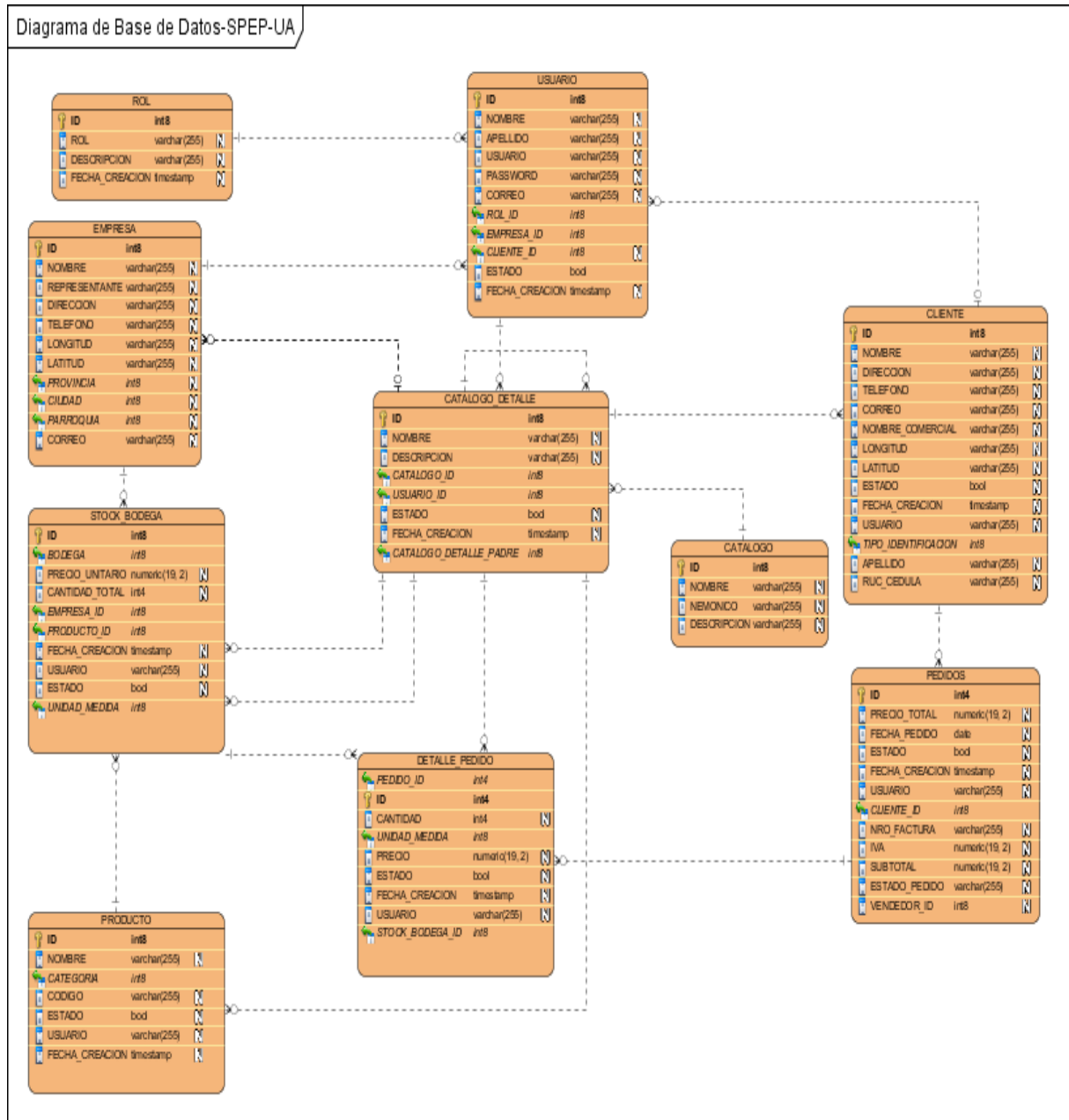


Figura 4.24: Diagrama Entidad Relación-SPEP-UA
Fuente: (Propia)

4.5 Fase de Implementación

ESPECIFICACIONES DE CASOS DE PRUEBA

4.5.1 Especificación de caso de prueba: Ingreso Datos Empresa

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso Datos Empresa”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Empresa.
- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Empresa y comprobar agregando datos: Nombre, Representante, Dirección, Correo Electrónico, Teléfono, etc., verificando si se ingresa o no correctamente todos los campos de Empresa.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar, todos los campos ingresados de Empresa a la aplicación.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Administración.
 3. Seleccionar Empresa.
 4. Agregar los campos.
 5. Guardar y verificar.
- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.
- Evaluación de la prueba.

Prueba superada con éxito.

4.5.2 Especificación de caso de prueba: Ingreso Datos Usuarios

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso Datos Usuarios”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Usuario.
- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Usuario y comprobar agregando datos: Nombre, Apellido, Usuario, Contraseña, Correo Electrónico y Rol, verificando si se ingresa o no correctamente todos los campos de Usuario.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar, todos los campos ingresados de Usuario a la aplicación.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Administración.
 3. Seleccionar Usuario.
 4. Agregar los campos.
 5. Guardar y verificar.
- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.
- Evaluación de la prueba.

Prueba superada con éxito.

4.5.3 Especificación de caso de prueba: Catálogos Detalles: Provincia

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Catálogos Detalles: Provincia”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Catálogos Detalles.

- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Catálogos Detalles, escoger Provincia y comprobar agregando datos si se ingresa o no correctamente todos los campos, además se puede insertar sub ítems o hijos del detalle padre como ejemplo: de Padre que sería Provincia se agrega las ciudades o cantones que componen a esa provincia, que serían los hijos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar, todos los campos ingresados de Provincia y sus respectivos sub ítems a la aplicación.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Catálogos Detalles.
4. Agregar los campos del detalle padre.
5. Agregar sub ítems del detalle padre.
6. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.4 Especificación de caso de prueba: Catálogos Detalles: Categoría Producto

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Catálogos Detalles: Categoría Producto”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Catálogos Detalles.

- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Catálogos Detalles, escoger Categoría Producto y comprobar agregando datos si se ingresa o no correctamente todos los campos, además se puede insertar sub ítems o hijos del detalle padre como ejemplo: de padre, que sería Balanceados se agrega los productos, balanceado para aves, balanceado para cerdos etc., que serían los hijos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar, todos los campos ingresados de Categoría Producto y sus respectivos sub ítems a la aplicación.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Catálogos Detalles.
4. Agregar los campos del detalle padre.
5. Agregar sub ítems del detalle padre.
6. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.5 Especificación caso de prueba: Catálogos Detalles: Bodega

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Catálogos Detalles: Bodega”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Catálogos Detalles.

- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Catálogos Detalles, escoger Bodega y comprobar agregando datos si se ingresa o no correctamente todos los campos, además se puede insertar sub ítems o hijos del detalle padre como ejemplo: de padre, que sería Bodega1 agregar los productos que serían los hijos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar, todos los campos ingresados de Bodega y sus respectivos sub ítems a la aplicación.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Catálogos Detalles.
4. Agregar los campos del detalle padre.
5. Agregar sub ítems del detalle padre.
6. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.6 Especificación caso de prueba: Catálogos Detalles: Unidad de Medida

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Catálogos Detalles: Unidad de Medida”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Catálogos Detalles.

- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Catálogos Detalles, escoger Unidad de Medida y comprobar agregando datos si se ingresa o no correctamente todos los campos, además se puede insertar sub ítems o hijos del detalle padre, como ejemplo: de padre, que sería Unidades agregar, unidad, 6 unidades, 12 unidades, que serían los hijos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los campos ingresados de Unidad de Medida y sus respectivos sub ítems a la aplicación.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Catálogos Detalles.
4. Agregar los campos del detalle padre.
5. Agregar sub ítems del detalle padre.
6. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.7 Especificación caso de prueba: Catálogos Detalles: Identificación.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Catálogos Detalles: Identificación”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Catálogos Detalles.

- Comprobar si se puede ingresar datos a la aplicación.

Ingresar a Catálogos Detalles, escoger Identificación y comprobar agregando datos si se ingresa o no correctamente todos los campos, además se puede insertar sub ítems o hijos del detalle padre.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los campos ingresados de Identificación y sus respectivos sub ítems a la aplicación.

Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Catálogos Detalles.
4. Agregar los campos del detalle padre.
5. Agregar sub ítems del detalle padre.
6. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.8 Especificación caso de prueba: Ingreso de Productos.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Productos”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Productos.

- Comprobar si se puede ingresar datos a la aplicación.

Escoger el menú Administrador, ingresar a Productos y comprobar agregando datos: Nombre, Descripción, Código, Categoría, verificando si se ingresa o no correctamente todos los campos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los campos ingresados de Producto.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Productos.
4. Agregar los campos.
5. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: el producto fue ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.9 Especificación caso de prueba: Ingreso de Clientes.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Clientes”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Administración-Clientes.

- Comprobar si se puede ingresar datos a la aplicación.

Escoger el menú Administrador, escoger Clientes y comprobar agregando datos: Tipo de Identificación, Identificación, Nombre, Apellido, Teléfono, Dirección, Correo Electrónico, Nombre Comercial, verificando si se ingresa o no correctamente todos los campos.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los campos ingresados de Cliente.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Administración.
3. Seleccionar Clientes.
4. Agregar los campos.
5. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.10 Especificación caso de prueba: Ingreso de Productos en Bodega.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ingreso de Productos en Bodega”. El objetivo será verificar que un usuario pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Bodega-Ingreso.
- Comprobar si se puede ingresar datos a la aplicación.

Escoger el menú Bodega, escoger Ingreso y comprobar agregando datos: Bodega, Producto, Unidad de Medida, Cantidad Total y Precio Unitario, verificando si se ingresa o no correctamente todos los campos.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los Productos ingresados a Bodega.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Bodega.
 3. Seleccionar Ingreso.
 4. Agregar los campos.
 5. Guardar y verificar.
- Resultado esperado.

El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.
- Evaluación de la prueba.

Prueba superada con éxito

4.5.11 Especificación caso de prueba: Visualización de Lista de Pedidos.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Visualización Lista de Pedidos”. El objetivo será verificar que un usuario pueda visualizar correctamente los datos ingresados a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Bodega-Pedidos.

- Comprobar si se puede visualizar los datos ingresados a la aplicación.

Escoger el menú Bodega, escoger Pedidos y comprobar los datos ingresados a la aplicación.

- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda visualizar todos los Pedidos ingresados a la aplicación para luego verificar el detalle y seleccionar el ingreso del pedido, para finalmente guardarlo.

- Entrada.

1. Ingresar a la aplicación.
2. Escoger el menú Bodega.
3. Seleccionar Pedido.
4. Visualizar los campos.
5. Verificar el detalle del pedido.
6. Seleccionar para ingresar el pedido.
7. Guardar y verificar.

- Resultado esperado.

El sistema muestra una pantalla donde se visualiza la lista de Pedidos ingresados en la aplicación desde el dispositivo móvil.

- Evaluación de la prueba.

Prueba superada con éxito

4.5.12 Especificación caso de prueba: Inventario Bodega.

- Descripción.
Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Inventario Bodega”. El objetivo será verificar que un usuario pueda visualizar el inventario de los productos ingresados a la Bodega asignada. El entorno del cual se iniciará para realizar la prueba será el menú Bodega-Inventario.
- Comprobar si se puede visualizar los datos ingresados a la aplicación.
Escoger el menú Bodega, escoger Inventario y comprobar los productos ingresados a la aplicación.
- Condiciones de ejecución.
Las condiciones de ejecución del caso de prueba es que se pueda visualizar todos los productos ingresados a la Bodega asignada.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Bodega.
 3. Seleccionar Inventario.
 4. Visualizar el Inventario.
- Resultado esperado.
El sistema muestra una pantalla donde se visualiza el inventario de los productos ingresados a la respectiva bodega.
- Evaluación de la prueba.
Prueba superada con éxito

4.5.13 Especificación caso de prueba: Asignar Ruta.

- Descripción.
Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Asignar Ruta”. El objetivo será verificar que un usuario pueda asignar correctamente las rutas a los vendedores quienes se encargaran de verificar y hacer la entrega del pedido. El entorno del cual se iniciará para realizar la prueba será el menú Rutas-Asignar Ruta.
- Comprobar si se puede visualizar los datos ingresados a la aplicación.
Escoger el menú Rutas, escoger Asignar Rutas y realizar la respectiva asignación de la ruta al vendedor.
- Condiciones de ejecución.
Las condiciones de ejecución del caso de prueba es que se pueda asignar todas las rutas de los Pedidos ingresados a los vendedores, previa verificación del detalle del mismo, para finalmente dar clic sobre el botón Asignar Ruta.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Rutas.
 3. Seleccionar Asignar Ruta.
 4. Visualizar los campos.
 5. Verificar el detalle del pedido.
 6. Seleccionar para ingresar el pedido.
 7. Clic en el Botón Asignar Ruta
- Resultado esperado.
El sistema muestra una pantalla donde se visualiza la lista de Pedidos ingresados para luego hacer la respectiva asignación de la ruta al vendedor en donde muestra el mensaje: Ruta Asignada Exitosamente.
- Evaluación de la prueba.
Prueba superada con éxito

4.5.14 Especificación caso de prueba: Ver Rutas.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “Ver Rutas”. El objetivo será verificar que un usuario pueda visualizar correctamente las rutas ingresados a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Rutas-Ver Rutas.
- Comprobar si se puede visualizar las rutas ingresados a la aplicación.

Escoger el menú Rutas, escoger Ver Rutas y verificar el estado del pedido de las rutas deberán estar en progreso esto indica que las rutas están asignadas, pero aun no son entregadas, esto se verifica en el reporte de pedidos en donde muestra el pedido con su estado Entregado, realizado por el vendedor.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda visualizar todas las rutas de los Pedidos ingresados a la aplicación, para luego verificar el detalle y su estado.
- Entrada.
 1. Ingresar a la aplicación.
 2. Escoger el menú Rutas.
 3. Seleccionar Ver Rutas.
 4. Visualizar los campos.
 5. Verificar el detalle del pedido.
 6. Verificar el Estado.
 7. Realizar la respectiva transacción de gestión de pedidos de acuerdo al estado.
- Resultado esperado.

El sistema muestra una pantalla donde se visualiza las rutas y su respectivo estado.
- Evaluación de la prueba.

Prueba superada con éxito

4.5.15 Especificación caso de prueba: App-Móvil-Cliente-Crear Pedido.

- Descripción.
Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “App-Móvil-Cliente-Crear Pedido”. El objetivo será verificar que el cliente pueda Crear el Pedido correctamente a través de la aplicación móvil el entorno del cual se iniciará para realizar la prueba será el menú Crear Pedido desde el dispositivo móvil.
- Comprobar si se puede Crear el Pedido desde la aplicación móvil.
Escoger el menú Crear Pedido, ingresar el pedido, asignar los productos con su respectiva cantidad para finalmente dar clic en Ingresar para luego automáticamente el pedido se ingresa en la aplicación web y luego ser gestionada a través del administrador del sitio.
- Condiciones de ejecución.
Las condiciones de ejecución del caso de prueba es que se pueda Crear el Pedido correctamente.
- Entrada.
 1. Ingresar a la aplicación móvil.
 2. Escoger el Crear Pedido.
 3. Ingresar los productos.
 4. Ingresar la cantidad a ser adquirida de dichos productos.
 5. Ingresar el pedido.
- Resultado esperado.
El sistema muestra una pantalla donde se visualiza que se ha creado el pedido exitosamente.
- Evaluación de la prueba.
Prueba superada con éxito

4.5.16 Especificación caso de prueba: App-Móvil-Cliente-Ver Pedidos Ingresados.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “App-Móvil-Cliente-Ver Pedidos Ingresados”. El objetivo será verificar que el cliente pueda Visualizar el Pedido Ingresado a través de la aplicación móvil el entorno del cual se iniciará para realizar la prueba será el menú, Pedidos Ingresados desde el dispositivo móvil.
- Comprobar si se puede Visualizar el Pedido Ingresado desde la aplicación móvil.

Escoger el menú Pedidos Ingresados, visualizar los mismos.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda Visualizar el Pedido Ingresado correctamente.
- Entrada.
 1. Ingresar a la aplicación móvil.
 2. Escoger Pedidos Ingresados.
 3. Visualizar los Pedidos Ingresados
 4. Además, se visualiza el estado del pedido si este fue entregado o ingresado.
- Resultado esperado.

El sistema muestra una pantalla donde se visualiza el estado del pedido entregado o ingresado exitosamente.
- Evaluación de la prueba.

Prueba superada con éxito

4.5.17 Especificación caso de prueba: App-Móvil-Vendedor-Ver Pedido Asignado.

- Descripción.
Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “App-Móvil-Vendedor-Ver Pedidos Asignado”. El objetivo será verificar y visualizar el detalle de los pedidos asignados a través de la aplicación móvil, el entorno del cual se iniciará para realizar la prueba será el menú, Pedidos Ingresados desde el dispositivo móvil.
- Comprobar si se puede Visualizar el Pedido Asignado desde la aplicación móvil.
Escoger el menú Pedidos Asignados, visualizar los mismos.
- Condiciones de ejecución.
Las condiciones de ejecución del caso de prueba es que se pueda Visualizar el Pedido asignado con su respectivo detalle.
- Entrada.
 1. Ingresar a la aplicación móvil.
 2. Escoger Pedidos Asignados.
 3. Visualizar todos los pedidos asignados.
 4. Visualizar el detalle de cada pedido.
- Resultado esperado.
El sistema muestra una pantalla donde se visualiza el estado del pedido asignado exitosamente.
- Evaluación de la prueba.
Prueba superada con éxito

4.5.18 Especificación caso de prueba: App-Móvil-Vendedor-Entregar Pedido.

- Descripción.

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “App-Móvil-Vendedor-Entregar Pedido”. El objetivo será verificar y visualizar el detalle de los pedidos asignados para luego ser entregados a sus respectivos clientes, a través de la aplicación móvil, el entorno del cual se iniciará para realizar la prueba será el menú, Pedidos Ingresados desde el dispositivo móvil.
- Comprobar si se puede Visualizar el Pedido Asignado desde la aplicación móvil.

Escoger el menú Pedidos Asignados, visualizar los mismos.
- Condiciones de ejecución.

Las condiciones de ejecución del caso de prueba es que se pueda Visualizar el Pedido Asignado con su respectivo detalle y luego ser entregados a sus respectivos clientes.
- Entrada.
 1. Ingresar a la aplicación móvil.
 2. Escoger Pedidos Asignados.
 3. Visualizar los Pedidos Asignados
 4. Visualizar el detalle del pedido
 5. Entregar el pedido al cliente.
- Resultado esperado.

El sistema muestra una pantalla del pedido asignado para su respectiva verificación del detalle y el botón de entregar que al momento de darle clic se asume que fue entregado.
- Evaluación de la prueba.

Prueba superada con éxito

4.5.19 Especificación caso de prueba: App-Móvil-Vendedor-Ingresa Clientes

- Descripción.
Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso “App-Móvil-Vendedor-Ingresa Clientes”. El objetivo será verificar que un vendedor pueda ingresar los datos a la aplicación. El entorno del cual se iniciará para realizar la prueba será el menú Ingresar Clientes desde el dispositivo móvil.
- Comprobar si se puede ingresar clientes desde el dispositivo móvil.
Escoger el menú Ingresar Cliente y comprobar agregando datos: Nombre, Apellido, Teléfono, Dirección, Correo Electrónico, Nombre Comercial, Tipo de Identificación, Identificación, verificando si se ingresa o no correctamente todos los campos.
- Condiciones de ejecución.
Las condiciones de ejecución del caso de prueba es que se pueda guardar todos los campos ingresados de Cliente.
- Entrada.
 1. Ingresar al dispositivo móvil con su usuario y contraseña.
 2. Escoger Ingresar Cliente.
 3. Agregar los campos.
 4. Guardar y verificar.
- Resultado esperado.
El sistema muestra una pantalla donde se visualiza los datos guardados en la aplicación y el mensaje: los datos fueron ingresados exitosamente.
- Evaluación de la prueba.
Prueba superada con éxito

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES.

5.1 CONCLUSIONES

1. El uso de software libre diseñado para dispositivos móviles ha permitido la realización e implementación de este proyecto, lo cual representa un apoyo estratégico tanto para el desarrollo del mismo como para las organizaciones encargadas de este tipo de actividades como es la distribución de productos, permitiendo la reducción en costos de recursos y optimizando los procesos que incluyen dichas actividades.
2. Las aplicaciones de este tipo tienen la posibilidad de ser modificadas para ser utilizadas por otras empresas con actividades similares, de acuerdo a características propias y sus necesidades.
3. La utilización de software libre como Android Studio y Java permitieron el desarrollo de la aplicación “SISTEMA DE PEDIDOS Y ENTREGA DE PRODUCTOS UTILIZANDO ANDROID”, herramientas adaptables al entorno y sin costo, a través de las cuales se gestionó la administración de los pedidos, entrega y asignación de rutas. Estas herramientas poseen una serie de características que las hacen sólidas frente a aplicaciones de escritorio, éstas a su vez integradas con un servidor web denominado SOA y como gestor de base de datos POSTGRESQL.
4. El manejo de la metodología RUP para el desarrollo de software, ha sido de gran ayuda, ya que permite tener una idea clara en todas las fases del desarrollo del proyecto, su estructura obliga al desarrollador a documentar todo el proceso desde el inicio, elaboración, pruebas y puesta a producción, contando con un registro detallado de todos los cambios realizados en el transcurso hasta la finalización del mismo.

5. El desarrollo de prototipos ayuda a visualizar de mejor manera los requerimientos del usuario ya que nos permite realizar pruebas prematuras del sistema con los usuarios reales.

6. El sistema ofrece las funcionalidades necesarias para convertir este proyecto en un producto comercial que puede ser utilizado en grandes, medianas y pequeñas empresas.

5.2 RECOMENDACIONES

Al realizar la implementación del Sistema Informático de Pedidos y Entrega de Productos utilizando Android, se debe tener en consideración las siguientes recomendaciones que ayudará al buen funcionamiento y desempeño del Sistema:

1. Se recomienda instalar la aplicación en un Servidor que maneje únicamente la gestión y administración del sistema ya que esto permitirá un seguimiento eficiente de la distribución de los pedidos que se generen por parte de los clientes.
2. Es importante que en el desarrollo del proyecto se cuente con la participación activa de todos los entes que componen la empresa para solucionar todos los riesgos que se vayan generando en la realización del proyecto, esto con la finalidad de entregar una versión óptima para su implementación.
3. Es recomendable realizar un estudio antes de empezar a desarrollar un sistema, para entender el comportamiento del mercado y la demanda, dentro de este estudio podemos medir el nivel de demanda y aceptación del presente sistema.
4. La metodología implementada RUP cuenta con un enfoque disciplinado en la asignación de tareas y responsabilidades dentro del desarrollo del proyecto, con la cual se puede mantener una fácil administración de todo este proceso.

GLOSARIO

• ARQ:	(Automatic Repeat-reQuest) son protocolos utilizados para el control de errores en la transferencia de datos, garantizando la integridad de los mismos.
• API:	(Application Programming Interface) es un conjunto de librerías de código JAVA compilado que se incluyen en la Java Runtime Environment.
• Bits:	Es un dígito de sistema de numeración binaria.
• Bluetooth:	Es una especificación para redes inalámbricas, que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura (2,4 GHz). (Lebedev, 2012)
• EDGE:	Tasas de Datos Mejoradas para la evolución del sistema global de dispositivos móviles GMS.
• Gateway:	Es una puerta de enlace, un nodo en una red informática que sirve de punto de acceso a otra red.
• Globalización:	Es una interdependencia económica creciente entre los países, provocada por el aumento del volumen y la variedad de las transacciones de bienes y servicios.
• Hibernate:	Herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.
• ISM:	Industrial Scientific and Medical, son bandas reservadas internacionalmente para el uso no comercial de radiofrecuencia electromagnética en el área industrial, científica y médica. (Lebedev, 2012)

• JBOSS	La comunidad de código abierto es amplia y colaborativa. A ella contribuyen una amplia variedad de organizaciones y desarrolladores. El código fuente de libre acceso permite a los usuarios explorar, experimentar, probar, y proporcionar comentarios sobre las funciones más novedosas del software.
• J2EE:	(Java 2 Enterprise Edition) define un estándar para el desarrollo de aplicaciones empresariales multicapas diseñada por Sun Microsystems.
• JSP:	Tecnología y marco de trabajo para aplicaciones java basadas en web, simplifica el desarrollo de interfaces cliente en aplicaciones Java EE. (Gonzales, Información de tecnología, 2015)
• MVC	Es un modelo empleado por desarrolladores de software: Modelo-Vista-Controlador.
• MMS:	Servicio de mensajería Multimedia.
•	
• SMS	Servicio de mensajes cortos o servicio de mensajes simples.
• VGA:	Adaptador gráfico de video.
• WILDFLY:	Servidor de aplicaciones escrito en código Java. Conocido antes como JBoss AS.

BIBLIOGRAFÍA

- Álvarez, M. A. (02 de 01 de 2014). *DesarrolloWeb*. Recuperado el 19 de 02 de 2015, de Qué es MVC: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>
- Camargo, O. (12 de 06 de 2014). *Los Hijos de Bracamontes*. Obtenido de Los Hijos de Bracamontes: Modelo de Proceso Unificado: <http://loshijosdebraca.blogspot.com/2010/12/modelo-de-proceso-unificado.html>
- Carlos Silva, H. K. (26 de Noviembre de 2011). *PostgreSQL Ventajas-Desventajas* . Obtenido de <http://postgresql-adsi.blogspot.com/2011/11/ampliamente-popular-ideal-para.html>
- Developer, J. (10 de 11 de 2014). *Red Hat JBoss Developer Studio*. Recuperado el 18 de 02 de 2015, de Visión de conjunto: <http://www.jboss.org/products/devstudio/overview/&prev=search>
- EcuRed. (04 de Noviembre de 2015). *EcuRed/Servidor de Aplicaciones*. Recuperado el 15 de 02 de 2015, de Servidor_de_aplicaciones: https://www.ecured.cu/Servidor_de_Aplicaciones
- García Cancela, L., & Ostos Lobo, S. (3 de 3 de 2014). *Software de Comunicaciones*. Recuperado el 14 de 5 de 2014, de Programación en dispositivos móviles portables: <https://sites.google.com/site/swcuc3m/>
- García, L. (15 de Diciembre de 2015). *Manual de InstalaciónJava* . Obtenido de <http://leonardogarciamartinez.blogdiario.com/i2015-12/>
- Gonzales, G. (27 de 01 de 2013). *Información de Tecnología*. Recuperado el 19 de 02 de 2015, de JavaServer Faces (JSF): <https://kalistog.wordpress.com/javaserver-faces-jsf/>
- Gonzales, G. (30 de Noviembre de 2015). *Información de tecnología*. Obtenido de <https://kalistog.wordpress.com/javaserver-faces-jsf/>
- Joaquin, B. A. (07 de 12 de 2007). *Artículo: De los Procesos del Negocio a los Casos de Uso*. Recuperado el 2 de 04 de 2015, de De los Procesos del Negocio a los Casos de Uso: <http://www.cyta.com.ar/ta0604/v6n4a1.htm>
- Lebedev, S. (3 de Marzo de 2012). *Historia de la Informática*. Obtenido de <http://histinf.blogs.upv.es/2012/12/03/smartphones/>
- Loor, J. M. (15 de 18 de 2014). *nDeveloper*. Recuperado el 18 de 02 de 2015, de JSF Java: http://www.ndeveloper.com/ndeveloperDocuments/documents/nDeveloper_JavaServerFaces.pdf

- Martinez, R. (12 de Octubre de 2012). *PostgreSQL-es*. Obtenido de http://www.postgresql.org/es/sobre_postgresql
- Montero, J. (05 de Marzo de 2013). *Instalación del Kit de Desarrollo Java (JDK) en Windows*. Obtenido de <http://elclubdelautodidacta.es/wp/2013/03/instalacion-del-kit-de-desarrollo-java-jdk-en-windows/>
- Oliver, G. (28 de Diciembre de 2014). *Estructura de un proyecto Android (Android Studio)*. Obtenido de <http://www.sgoliver.net/blog/estructura-de-un-proyecto-android-android-studio/>
- Orozco, D. (1 de Mayo de 2011). *Concepto Definición de*. Recuperado el 6 de 05 de 2014, de <http://conceptodefinicion.de/android>
- Robledo Fernández, D. (2014). *Desarrollo de aplicaciones para Android II* (2 ed.). España, Madrid.: Ministerio de Educación de España.
- Robledo Sacristán, D. (2011). *Programación en Android*. España, Madrid.: Ministerio de Educación de España.
- Tellez Ariza, F. (16 de 5 de 2013). *Herramientas de Desarrollo* . Obtenido de Android Introduccion .
- Tomás, J. (Mayo de 2015). *Las versiones de Android y niveles de API* . Recuperado el 16 de 11 de 2014, de <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/146-las-versiones-de-android-y-niveles-de-api>
- Vera, M. (7 de Noviembre de 2013). *Intelligence to bussines*. Obtenido de <http://www.i2btech.com/blog-i2b/tech-deployment/que-se-entiende-por-soa-y-cuales-son-sus-beneficios/>

ANEXOS

ANEXO A. DICCIONARIO DE DATOS

Nombre de las tablas del sistema.

- ❖ ROL
- ❖ USUARIO
- ❖ EMPRESA
- ❖ CLIENTE
- ❖ CATÁLOGO
- ❖ CATÁLOGO_DETALLE
- ❖ PEDIDOS
- ❖ DETALLE_PEDIDOS
- ❖ PRODUCTO
- ❖ STOCK_BODEGA

Explicación de campos de cada tabla del sistema.

❖ TABLA ROL

Descripción: Esta tabla guarda el listado de todos los roles que sirve para describir a los usuarios que accederán a la aplicación.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
ROL	2		N	VARCHAR	NOMBRE DEL ROL
DESCRIPCIÓN	3		N	VARCHAR	DESCRIPCIÓN DEL ROL
FECHA_CREACIÓN	4		N	TIME STAMP	FECHA DE CREACIÓN ROL

Tabla 5.1: Tabla Rol

Fuente: (Propia)

❖ TABLA USUARIO

Descripción: Esta tabla guarda a todos los usuarios que accederán a la aplicación, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE DEL USUARIO
APELLIDO	3		N	VARCHAR	APELLIDO DEL USUARIO
USUARIO	4		N	VARCHAR	TIPO DE USUARIO
PASSWORD	5		N	VARCHAR	CONTRASEÑA USUARIO
CORREO	6		N	VARCHAR	CORREO USUARIO
ROL_ID	7	FK		INT	CÓDIGO ROL
EMPRESA_ID	8	FK		INT	CÓDIGO EMPRESA
CLIENTE_ID	9	FK	N	INT	CÓDIGO CLIENTE
ESTADO	10			BOOL	VERDADERO-FALSO
FECHA_CREACION	11		N	TIME_STAMP	FECHA INGRESO

Tabla 5.2: Tabla Usuario

Fuente: (Propia)

❖ TABLA EMPRESA

Descripción: Esta tabla guarda todos los datos de la Empresa, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE DE LA EMPRESA
REPRESENTANTE	3		N	VARCHAR	REPRESENTANTE
DIRECCIÓN	4		N	VARCHAR	DIRECCIÓN
TELÉFONO	5		N	VARCHAR	TELÉFONO
LONGITUD	6		N	VARCHAR	LONGITUD
LATITUD	7		N	VARCHAR	LATITUD
PROVINCIA	8	FK	N	INT	CÓDIGO PROVINCIA
CIUDAD	9	FK	N	INT	CÓDIGO CIUDAD
PARROQUIA	10	FK	N	INT	CÓDIGO PARROQUIA
CORREO	11		N	VARCHAR	CORREO ELECTRÓNICO

Tabla 5.3: Tabla Empresa

Fuente: (Propia)

❖ **TABLA CLIENTE**

Descripción: Esta tabla guarda todos los datos de la Empresa, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE
APELLIDO	2		N	VARCHAR	APELLIDO
RUC_CEDULA	3		N	VARCHAR	CÉDULA-RUC-PASAPORT
DIRECCIÓN	4		N	VARCHAR	DIRECCIÓN
TELÉFONO	5		N	VARCHAR	TELÉFONO
CORREO	6		N	VARCHAR	CORREO ELECTRÓNICO
NOMBRE_COMERCIAL	7		N	VARCHAR	NOMBRE COMERCIAL
LONGITUD	8		N	VARCHAR	LONGITUD
LATITUD	9		N	VARCHAR	LATITUD
ESTADO	10		N	BOOL	VERDADERO – FALSO
FECHA_CREACION	11		N	VARCHAR	FECHA DE CREACIÓN
USUARIO	12		N	VARCHAR	USUARIO
TIPO_IDENTIFICACION	13	FK		INT	TIPO IDENTIFICACIÓN

Tabla 5.4: Tabla Cliente

Fuente: (Propia)

❖ **TABLA CATÁLOGO**

Descripción: Esta tabla guarda todos los tipos de catálogos que existirán en el proyecto, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE
NEMÓNICO	2		N	VARCHAR	NEMÓNICO
DESCRIPCIÓN	3		N	VARCHAR	DESCRIPCIÓN

Tabla 5.5: Tabla Catálogo

Fuente: (Propia)

❖ TABLA CATÁLOGO_DETALLE

Descripción: Esta tabla guarda todos los detalles de la tabla catálogo, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE
DESCRIPCIÓN			N	VARCHAR	DESCRIPCIÓN
CATALOGO_ID				INT	CÓDIGO CATALOGO
USUARIO_ID				INT	CÓDIGO USUARIO
ESTADO			N	VARCHAR	VERDADERO – FALSO
FECHA_CREACION			N	VARCHAR	FECHA DE CREACIÓN
CATALOGO_DETALLE_PADRE				INT	CÓDIGO PADRE

Tabla 5.6: Tabla Catálogo Detalle

Fuente: (Propia)

❖ TABLA PEDIDOS

Descripción: Esta tabla guarda todos los datos de los pedidos, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
PRECIO_TOTAL	2		N	NUMERIC	PRECIO PRODUCTO
FECHA_PEDIDO	2		N	DATE	FECHA VERIFICACIÓN
FECHA_CREACION	3		N	TIME_STAMP	FECHA CREACIÓN PEDIDO
USUARIO	4		N	VARCHAR	USUARIO
NRO_FACTURA	5		N	VARCHAR	NÚMERO DE FACTURA
IVA	6		N	NUMERIC	IVA
SUBTOTAL	7		N	NUMERIC	SUBTOTAL
ESTADO_PEDIDO	8		N	VARCHAR	ESTADO DEL PEDIDO
VENDEDOR_ID	9	FK		INT	CÓDIGO VENDEDOR
CLIENTE_ID	10	FK		INT	CÓDIGO CLIENTE
ESTADO	11		N	BOOLEAN	VERDADERO-FALSO

Tabla 5.7: Tabla Pedidos

Fuente: (Propia)

❖ TABLA DETALLE_PEDIDOS

Descripción: Esta tabla guarda todos los detalles de los pedidos, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
PEDIDO_ID	2	FK		INT	CÓDIGO PEDIDO
CANTIDAD	2		N	INT	CANTIDAD
UNIDAD DE MEDIDA	3		N	INT	UNIDAD DE MEDIDA
PRECIO	4		N	NUMERIC	PRECIO
ESTADO	5		N	BOOL	ESTADO PEDIDO
FECHA_CREACION	6		N	TIME_STAMP	FECHA
USUARIO	7		N	VARCHAR	USUARIO
STOCK_BODEGA_ID	8	FK		INT	STOCK

Tabla 5.8: Tabla Detalle Pedidos
Fuente: (Propia)

❖ TABLA PRODUCTO

Descripción: Esta tabla guarda todos los productos ingresados al sistema, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
NOMBRE	2		N	VARCHAR	NOMBRE PRODUCTO
CATEGORIA	2	FK		INT	CATEGORÍA
CODIGO	3		N	VARCHAR	CÓDIGO DEL PRODUCTO
ESTADO	4		N	BOOLEAN	ESTADO
USUARIO	5		N	VARCHAR	USUARIO
FECHA_CREACION	6		N	TIME_STAMP	FECHA CREACIÓN

Tabla 5.9: Tabla Producto
Fuente: (Propia)

❖ TABLA STOCK_BODEGA

Descripción: Esta tabla guarda todos los detalles de los pedidos, con sus respectivas claves foráneas.

NOMBRE COLUMNA	NRO. COLUMNA	CLAVE PRIMARIA	NULL	TIPO DATO	DESCRIPCIÓN
ID	1	1		INT	CAMPO NUMÉRICO SECUENCIAL
BODEGA	2			INT	ESPECIFICACIÓN BODEGA
PRECIO_UNITARIO	2		N	NUMERIC	PRECIO UNITARIO
CANTIDAD_TOTAL	3		N	INT	CANTIDAD TOTAL
EMPRESA_ID	4	FK		INT	CÓDIGO EMPRESA
PRODUCTO_ID	5	FK		INT	CÓDIGO PRODUCTO
FECHA_CREACION	6		N	DATE	FECHAR CREACIÓN
USUARIO	7		N	VARCHAR	USUARIO
ESTADO	8		N	BOOLEAN	ESTADO
UNIDAD DE MEDIDA	9			INT	UNIDAD DE MEDIDA

Tabla 5.10: Tabla Stock Bodega
Fuente: (Propia)

ANEXO B. MANUAL DE USUARIO

- Introducción

Las aplicaciones y dispositivos móviles han cambiado el entorno en el cual vivimos. Este cambio no ha hecho más que empezar, lo que permite que puedan ser utilizados para un sin número de tareas como, por ejemplo: leer el correo, visualizar la agenda entre otras. El lanzamiento del sistema operativo Android como nueva plataforma para el desarrollo de aplicaciones móviles, ha tenido una creciente aceptación en usuarios y empresas.

Por lo que Android, fue una alternativa inicial para desarrollar este proyecto por sus características y funcionalidades, además de estar al alcance de millones de personas y de aprovechar el beneficio en costo, ya que las herramientas de desarrollo son totalmente gratuitas, como de igual forma para el cliente en donde el sistema operativo de Android se encuentra en la mayoría de dispositivos móviles.

- Aplicación web

La pantalla de administración para gestionar y asignar los pedidos, estará desarrollada en JBoss Developer que es un Entorno de desarrollo integrado (IDE) certificado y basado en Eclipse para desarrollar, probar e implementar aplicaciones web, aplicaciones web móviles, aplicaciones empresariales transaccionales y servicios de integración basados en la arquitectura orientada a servicios (SOA). Esta aplicación estará instalada en un servidor de la Empresa, para gestionar toda la lógica del sistema, Pedidos y Entrega de Productos utilizando Android.

- Aplicación móvil

La aplicación móvil estará desarrollada en Android Studio y se podrá descargar la aplicación desde la app store de google.

- Requisitos

1. Para la aplicación web se debe instalar PostgreSQL y subir el servidor de aplicaciones Wildfly-8. (Administrador).
2. Para la aplicación móvil, descargar el aplicativo desde la app store de google, para la creación de pedidos. (Cliente).

MANUAL DE USUARIO

• Ingreso Aplicación Web

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none">1. Ingreso al sistema.2. Ingresar Usuario y Contraseña-Rol Administrador/Operador.3. Se accede a la administración del aplicativo.	<ul style="list-style-type: none">• Acceso

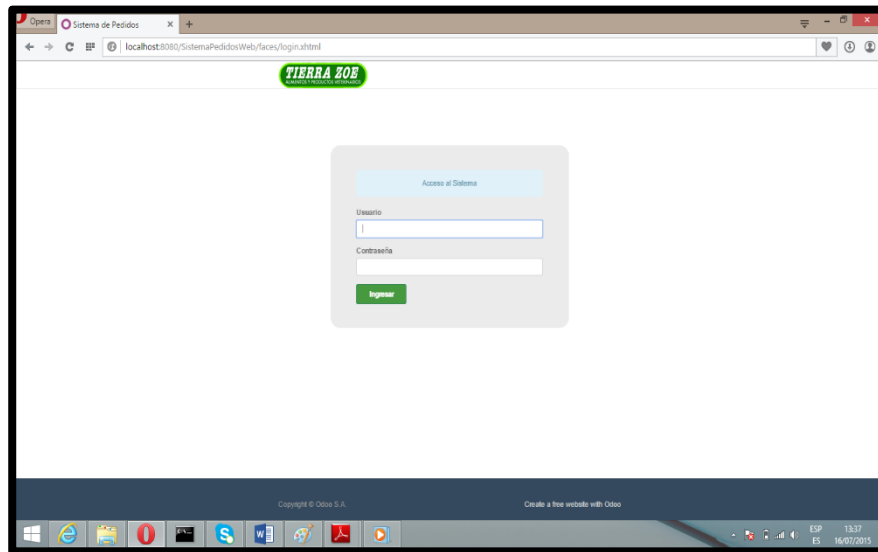


Figura 5.1: Ingreso Aplicación Web
Fuente: (Propia)

- Ingreso Datos Empresa

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Administración. 2. Escoger la opción Empresa. 3. Ingresar todos los campos que pertenecen a Empresa. 4. Finalmente pulsar clic en el botón guardar. Ver Figura. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

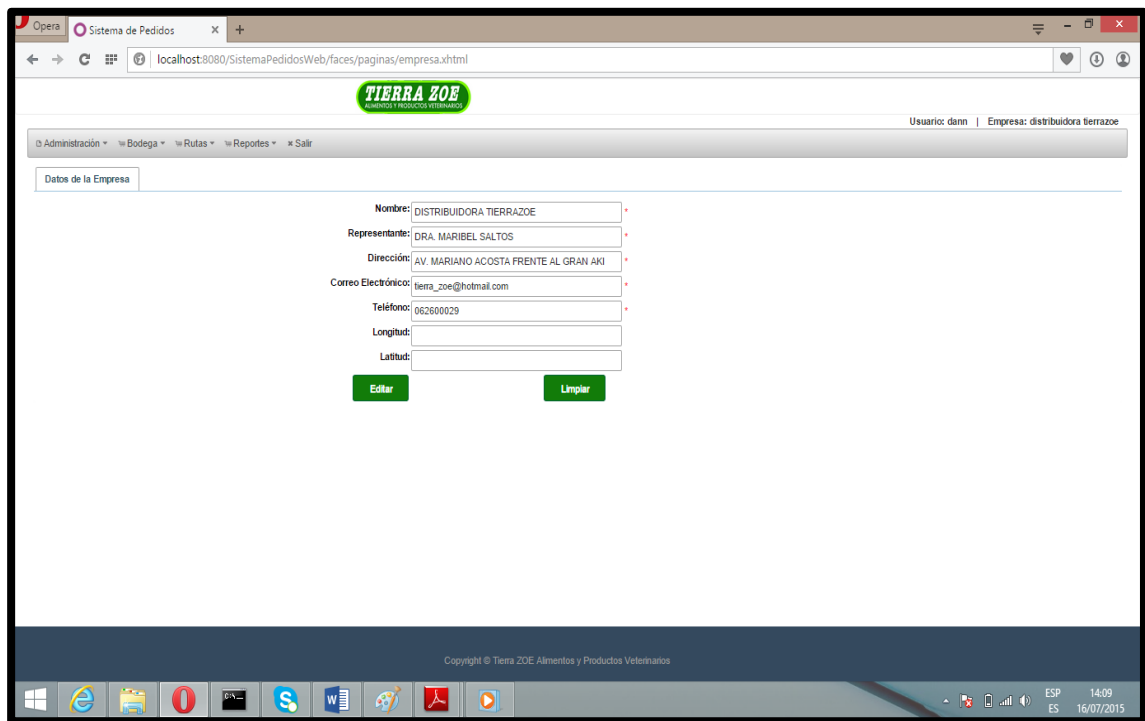


Figura 5.2: Ingreso Datos Empresa
Fuente: (Propia)

- Ingreso Usuarios

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Administrador. 2. Escoger la opción Usuarios. 3. Ingresar todos los campos que pertenecen a Usuario. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

The screenshot shows a web browser window with the URL `localhost:8080/SistemaPedidosWeb/faces/paginas/usuarios.xhtml`. The page features the 'TIERRA ZOE' logo and a navigation menu. The main content area contains a form for adding a new user with the following fields:

- Nombre:
- Apellido:
- Usuario:
- Contraseña:
- Correo Electrónico:
- Rol:

Below the form are two buttons: 'Guardar' and 'Limpiar'. Underneath is a table listing existing users:

Nombres	Usuario	Correo	Rol	Editar	Eliminar
AMABLE MEJIA	1001988763001	amable_meja2000@hotmail.com	ROL CLIENTE		
ANDRES CARRILLO	andres	andres@gmail.com	ROL VENDEDOR		
ANDRES ARTEAGA	1091701673001	bensoninst@hotmail.com	ROL CLIENTE		
ANTONIO NEGRETE	0460001020001	antonionegrete11@gmail.com	ROL CLIENTE		
CARLOS GUZMAN	1090054100001	carlosguzman54@hotmail.com	ROL CLIENTE		
DANIEL VILLARREAL	dann	daniel_omar00@hotmail.com	ROL ADMINISTRADOR		
DIANA RODRIGUEZ	diana	diana@gmail.com	ROL VENDEDOR		
FERNANDO MADERA	1791819980001	haciendasanjose@hotmail.com	ROL CLIENTE		
		femandocavajal@gmail.com	ROL CLIENTE		

Figura 5.3: Ingreso Datos Usuario
Fuente: (Propia)

- Ingreso Catálogo Detalles

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Administrador. 2. Escoger la opción Catálogos Detalles. 3. Escoger un Catálogo. 4. Crear los catálogos y guardar. 5. Seleccionar catálogo padre. 6. Ingresar nombre, descripción de los catálogos hijos. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

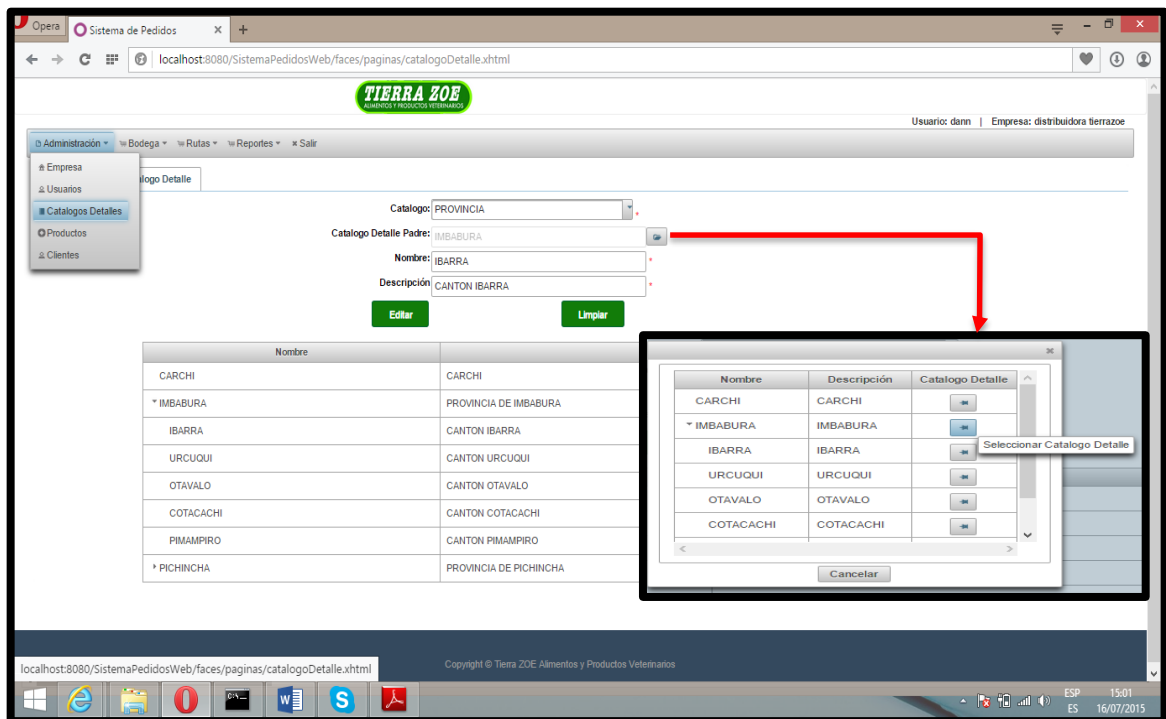


Figura 5.4: Ingreso Catálogos Detalles
Fuente: (Propia)

- Ingreso Productos

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Administrador. 2. Escoger la opción Productos. 3. Ingresar todos los campos que pertenecen a los Productos. 4. Seleccionar la categoría a la que pertenecerá ese producto. 5. Finalmente guardar. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

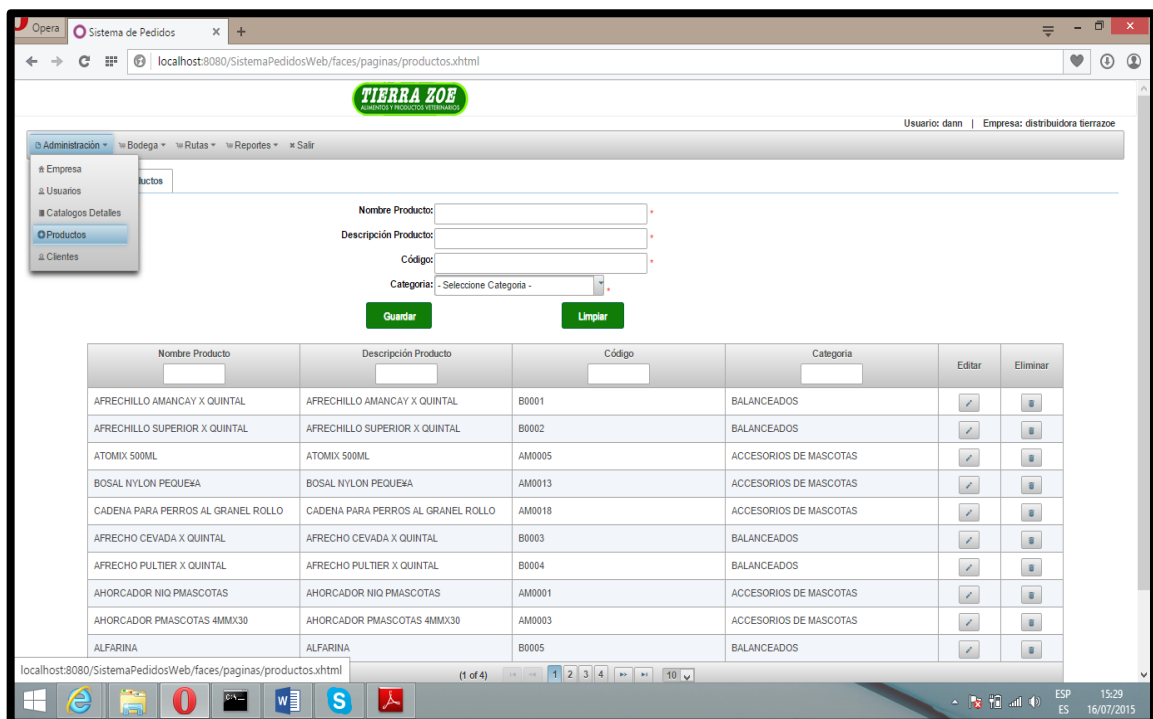


Figura 5.5: Ingreso Productos

Fuente: (Propia)

- Ingreso Clientes

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Administrador. 2. Escoger la opción Clientes. 3. Seleccionar tipo de identificación: Cédula, Ruc o Pasaporte. 4. Ingresar todos los campos que pertenecen a Clientes. 5. Guardar. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

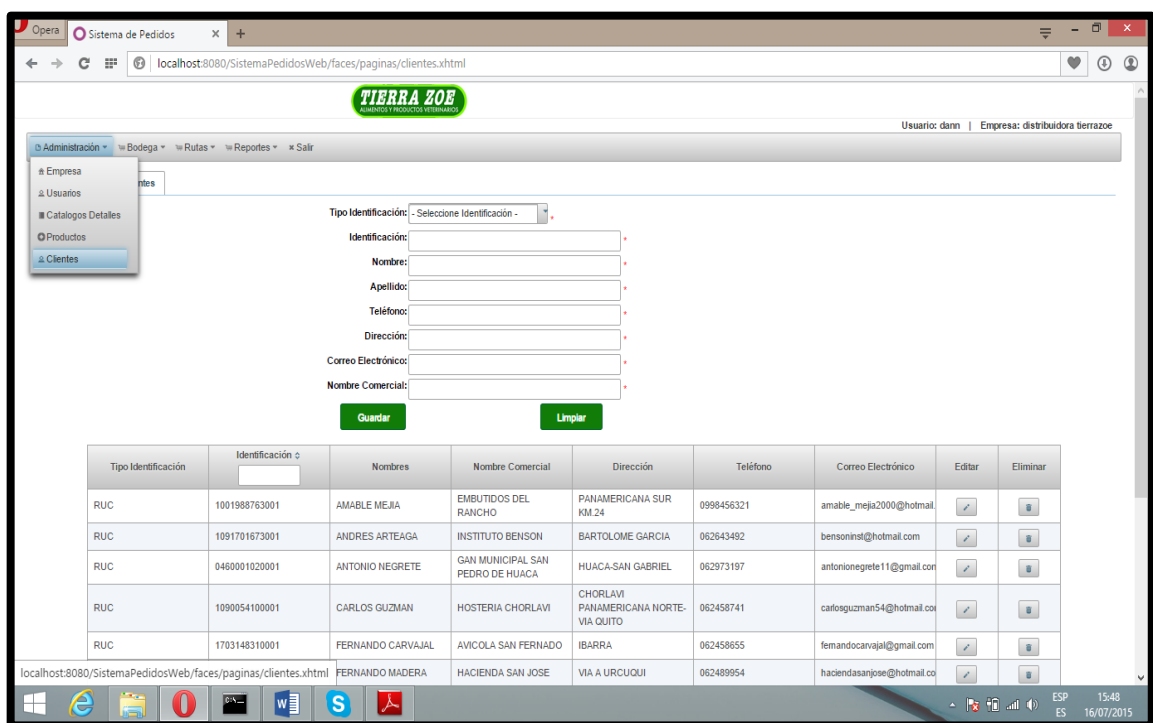


Figura 5.6: Ingreso Clientes
Fuente: (Propia)

- Bodega-Ingreso

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Bodega. 2. Escoger la opción Ingreso. 3. Seleccionar la categoría o la bodega a la cual se va a ingresar los productos. 4. Seleccionar el producto. 5. Seleccionar la Unidad de Medida. 6. Ingresar cantidad total. 7. Ingresar Precio. 8. Guardar. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar

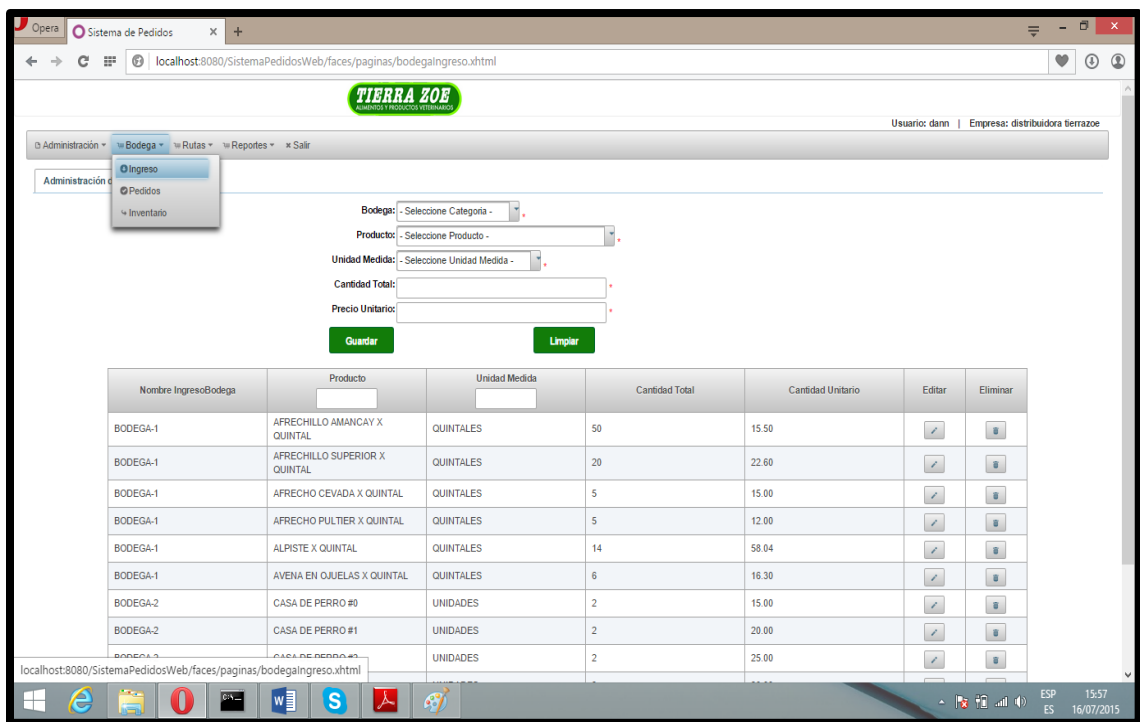


Figura 5.7: Ingreso Productos a Bodega asignada
Fuente: (Propia)

- Bodega-Pedidos

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Bodega. 2. Escoger la opción Pedidos. 3. Verificar el detalle del pedido ingresado por el cliente. 4. Seleccionar el pedido, esto hace que el pedido que el cliente creó sea ingresado. 5. Finalmente guardar el pedido. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar • Seleccionar

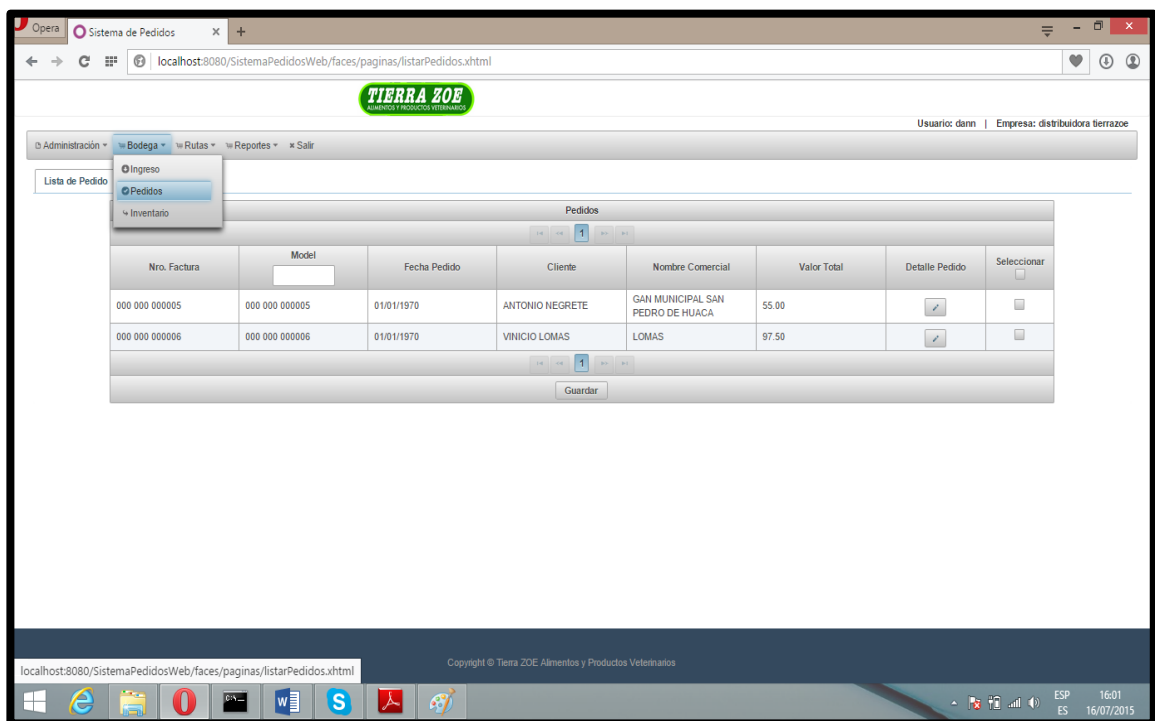


Figura 5.8: Revisar Pedidos
Fuente: (Propia)

- Bodega-Inventario

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Bodega. 2. Escoger la opción Inventario. 3. Visualizar el inventario general de los productos ingresados al sistema. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar • Mostrar

Nombre IngresoBodega	Producto	Unidad Medida	Cantidad Total	Cantidad Unitario
BODEGA-1	AFRECHILLO AMANCAY X QUINTAL	QUINTALES	21	15.50
BODEGA-1	AFRECHILLO SUPERIOR X QUINTAL	QUINTALES	8	22.60
BODEGA-1	AFRECHO CEVADA X QUINTAL	QUINTALES	-6	15.00
BODEGA-1	AFRECHO PULTIER X QUINTAL	QUINTALES	5	12.00
BODEGA-1	ALPISTE X QUINTAL	QUINTALES	10	58.04
BODEGA-1	AVENA EN OJUELAS X QUINTAL	QUINTALES	-2	16.30
BODEGA-2	CASA DE PERRO #0	UNIDADES	1	15.00
BODEGA-2	CASA DE PERRO #1	UNIDADES	2	20.00
BODEGA-2	CASA DE PERRO #2	UNIDADES	2	25.00
BODEGA-2	CASA DE PERRO #3	UNIDADES	3	30.00

Figura 5.9: Revisar Inventario

Fuente: (Propia)

- Rutas-Asignar Ruta

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Ruta. 2. Escoger la opción Asignar Ruta. 3. Verificar el detalle del pedido. 4. Asignar la ruta al vendedor. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar • Asignar Ruta

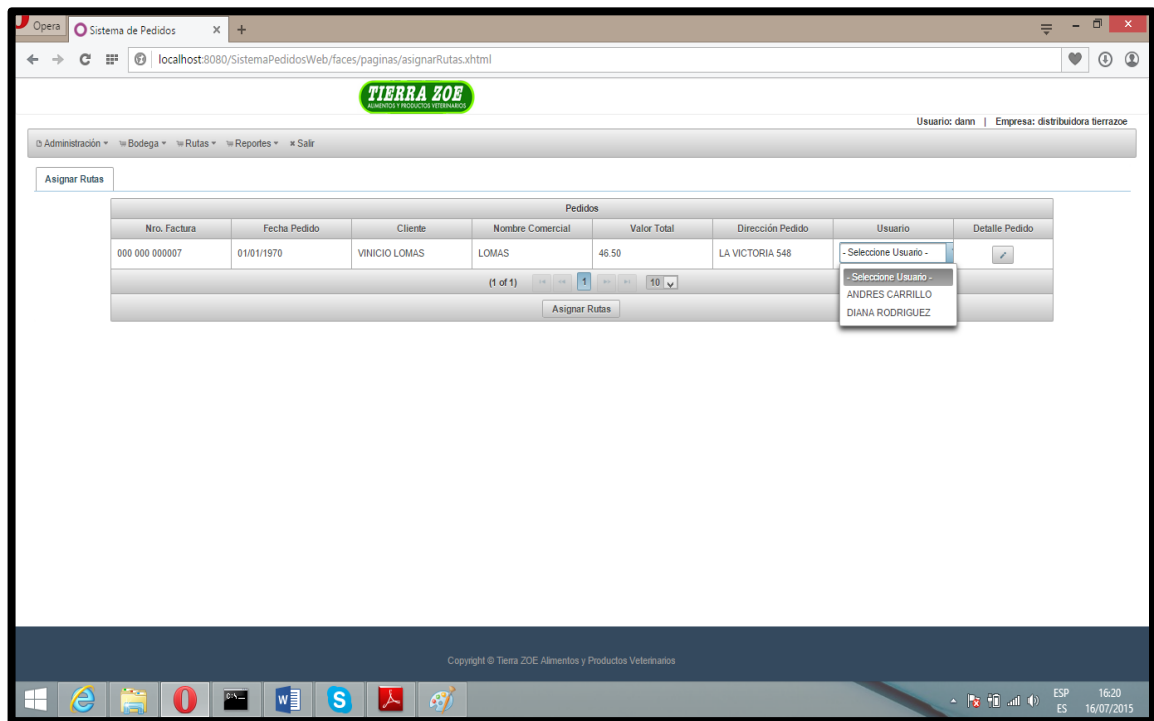


Figura 5.10: Rutas-Asignar Rutas
Fuente: (Propia)

- Rutas-Ver Ruta

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Ruta. 2. Escoger la opción Ver Ruta. 3. Verificar las rutas asignadas al vendedor. 4. Y verificar el estado de la ruta. 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar • Ver Ruta

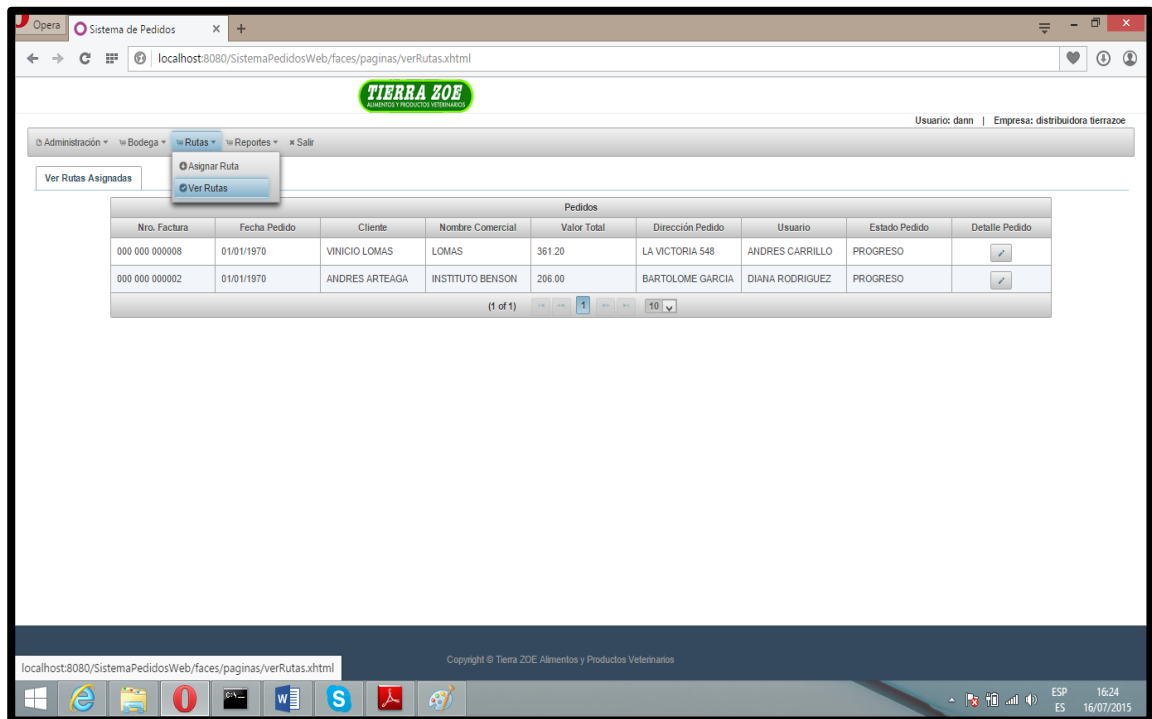


Figura 5.11: Rutas-Ver Rutas
Fuente: (Propia)

- Reporte-Pedidos

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al menú Reporte. 2. Escoger la opción Pedidos. 3. Generar el reporte. 4. Verificar el estado del pedido si este fue ingresado, aceptado por el administrador previa verificación de si existe o no el detalle del pedido (es decir si existe el producto en bodega), además si esta ya entregado o está en progreso de entrega. 5. Por último, verificar el estado de la ruta. 6. Imprimir 	<ul style="list-style-type: none"> • Guardar • Editar • Eliminar • Actualizar • Reportes • Imprimir

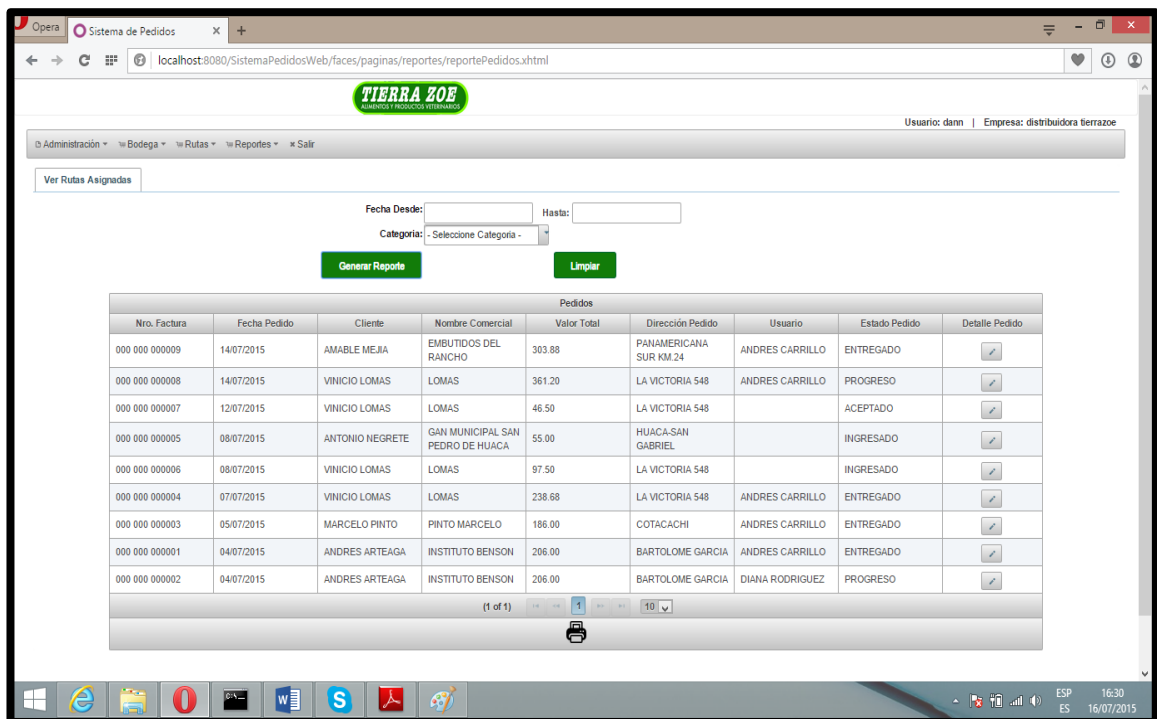
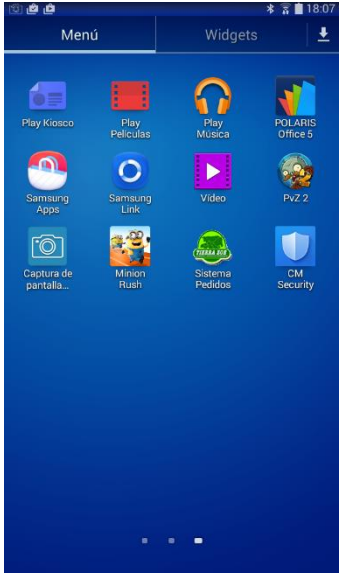


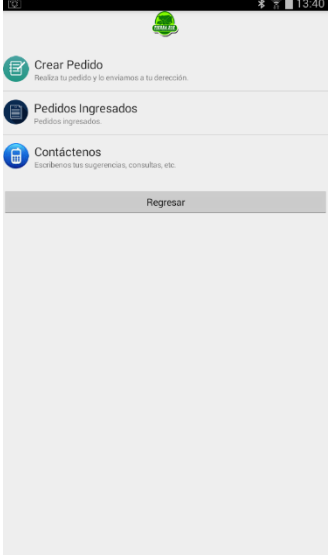
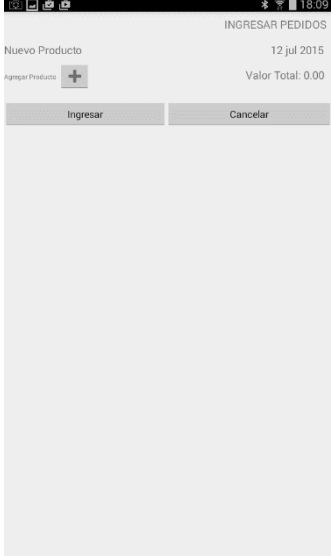
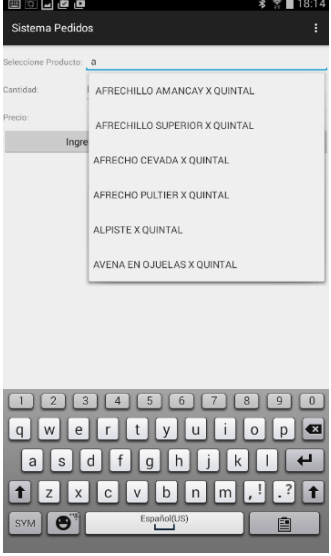


Figura 5.12: Reporte-Pedidos
Fuente: (Propia)

- Manual de Usuario Aplicación Móvil- Cliente.

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none">1. Ingresar al dispositivo móvil.2. Abrir la aplicación Sistema de Pedidos.3. Ingresar con nuestro usuario y contraseña cliente.4. Por única vez el sistema pide cambiar la contraseña con el cual podrá acceder a la aplicación.5. Ingresar y escoger la opción Crear Pedido.6. Ingresar el producto.7. Ingresar la cantidad.8. Si se desea ingresar más productos repetir el proceso desde el paso 6.9. Ingresar el pedido a realizar.10. Al dar clic en el botón Regresar se vuelve a la pantalla de las opciones, se escoge Pedidos Ingresados esto con la finalidad de visualizar todos los pedidos que se ha realizado con su respectiva fecha de ingreso y el estado previa verificación si ha sido entregado o no.11. Al dar clic sobre el Pedido Ingresado, se visualiza el detalle del mismo.12. Finalmente, en la opción Contáctenos, se puede visualizar los detalles de la Empresa.13. Ver figuras.	<ul style="list-style-type: none">• Crear Pedido• Visualizar pedido ingresado.

		
<p>Ingresar al sistema desde el dispositivo móvil</p>	<p>Paso 3: Ingresar Cambiar contraseña si lo desea</p>	<p>Ingresar con su usuario y contraseña</p>
		
<p>Crear Pedido</p>	<p>Agregar Productos al pedido</p>	<p>Seleccionar los productos</p>



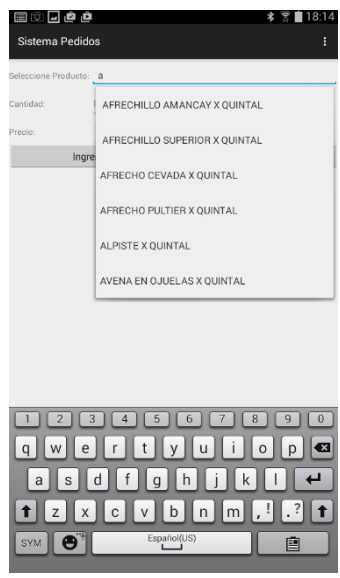
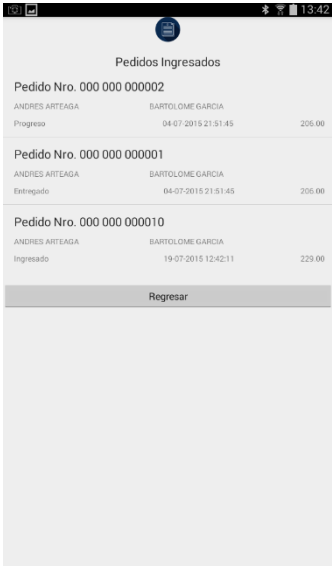
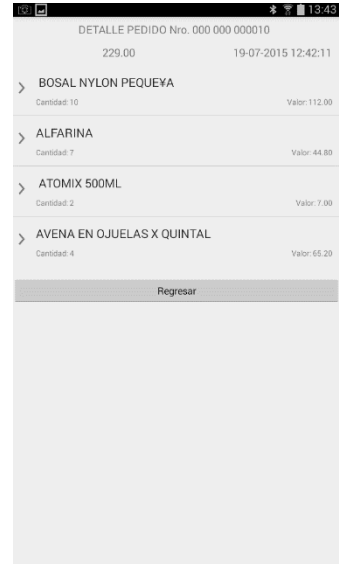
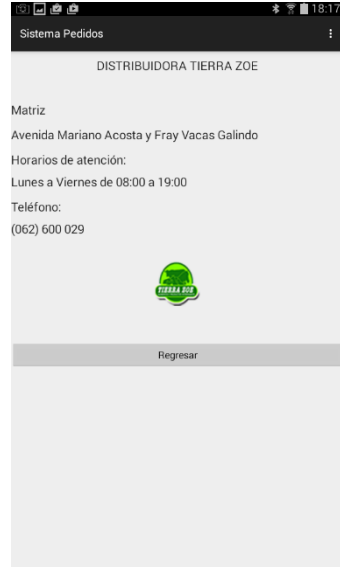
		
<p>Ingresar la cantidad de productos a pedir</p>	<p>Si se desea ingresar más productos</p>	<p>Paso 6: Ingresar más Productos si fuese el caso</p>
		
<p>Paso 10: Ver Pedidos Ingresados</p>	<p>Ver detalle del Pedido</p>	<p>Visualizar Contacto de la Empresa</p>

Figura 5.13: App. Móvil-Cliente-Crear Pedido
Fuente: (Propia)

- Manual de Usuario Aplicación Móvil- Vendedor-Visualiza y Entrega Pedido

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al dispositivo móvil. 2. Abrir la aplicación Sistema de Pedidos. 3. Ingresar con su usuario y contraseña, vendedor. 4. Por única vez el sistema pide cambiar la contraseña con el cual podrá acceder a la aplicación. 5. Ingresar y escoger la opción Pedidos Ingresados. 6. Verificar el nombre y la ruta del cliente. 7. Procede a entregar el pedido y al momento de dar clic en el botón entregar se envía el mensaje a la aplicación web y se cambia el estado del pedido de progreso ha entregado. 8. Entrega al cliente el pedido proceso manual. 9. Ver figuras. 	<ul style="list-style-type: none"> • Visualizar pedido ingresado. • Verificar el detalle. • Entregar Pedido.

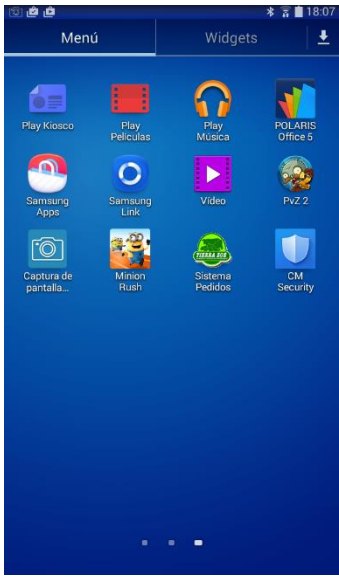
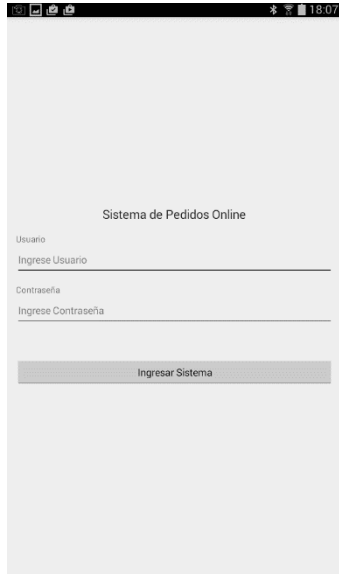

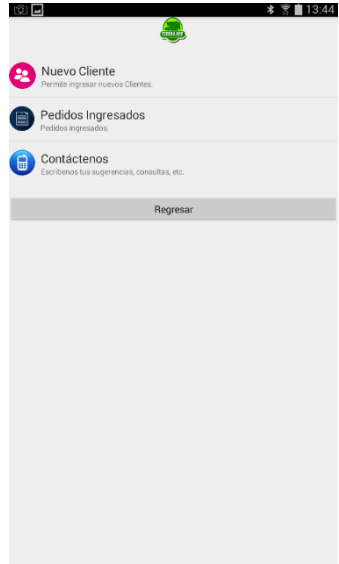
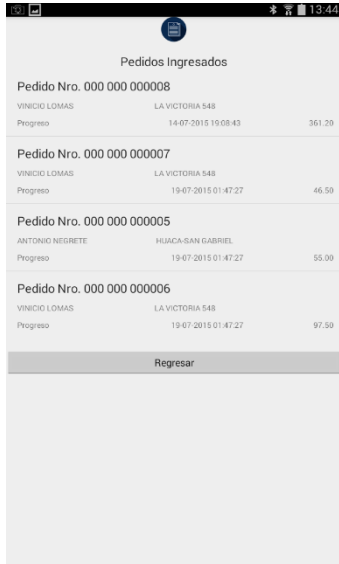

		
<p>Ingresar al sistema desde el dispositivo móvil</p>	<p>Paso 3: Ingreso Cambiar contraseña</p>	<p>Ingresar con su usuario y contraseña</p>
		
<p>Ingresar a pedidos Ingresados</p>	<p>Verificar el pedido con su respectivo nombre y dirección del cliente.</p>	<p>Entrega Pedido-Proceso Manual</p>

Figura 5.14: App. Móvil-Vendedor-Visualiza y Entrega Pedido
Fuente: (Propia)

- Manual de Usuario Aplicación Móvil- Vendedor-Ingresa Clientes.

DESCRIPCIÓN	EVENTOS
<ol style="list-style-type: none"> 1. Ingresar al dispositivo móvil. 2. Abrir la aplicación Sistema de Pedidos. 3. Ingresar con su usuario y contraseña vendedor. 4. Ingresar y escoger la opción Ingresar Cliente. 5. Ingresar los campos correspondientes a cliente. 6. Ingreso Clientes. 	<ul style="list-style-type: none"> • Ingresar Clientes.

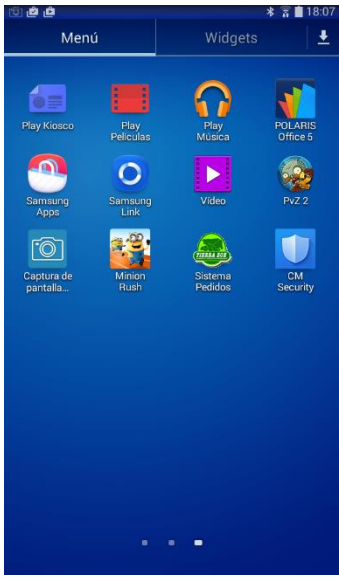
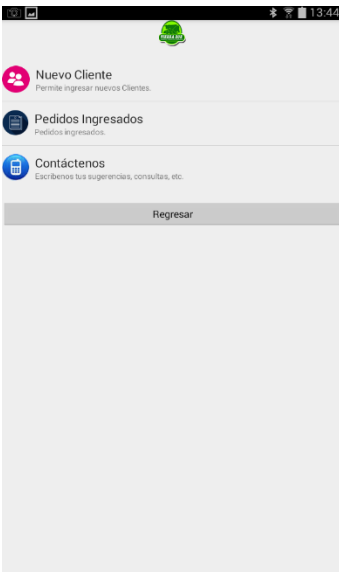

		
<p>Ingresar al sistema desde el dispositivo móvil</p>	<p>Escoger Nuevo Cliente</p>	<p>Ingresar los campos correspondientes a cliente y finalmente ingresar</p>

Figura 5.15: App. Móvil-Vendedor-Ingresa Cliente

Fuente: (Propia)

ANEXO C. MANUAL TÉCNICO

- Introducción

El presente manual técnico proporcionará a los usuarios toda la lógica del desarrollo o elaboración del presente Sistema Informático de Pedidos y Entrega de Productos utilizando Android para la Empresa Tierra Zoe,

Este a su vez estará en lo posible debidamente documentado y para su fácil entendimiento, obviamente el lector deberá profundizar ciertos temas que no estarán especificados completamente ya que este artefacto no es un manual de aprendizaje sino más bien una guía de las herramientas empleadas en el desarrollo del proyecto.

- Objetivo

Proporcionar una guía del desarrollo de la aplicación para el usuario o lector.

INSTALACIÓN DE JAVA



Como primer paso será preparar el entorno de desarrollo. En el artefacto se indicará **cómo instalar el JDK** (*Java Development Kit*) sobre una plataforma Windows.

Se deberá comenzar accediendo a la página oficial para la descarga <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html> desde la cual se descargará el JDK de **Java SE 8u45**, es decir, la versión 8, actualización 45 (la actual en el momento de escribir el manual técnico) de la edición estándar de Java. (Montero, 2013)

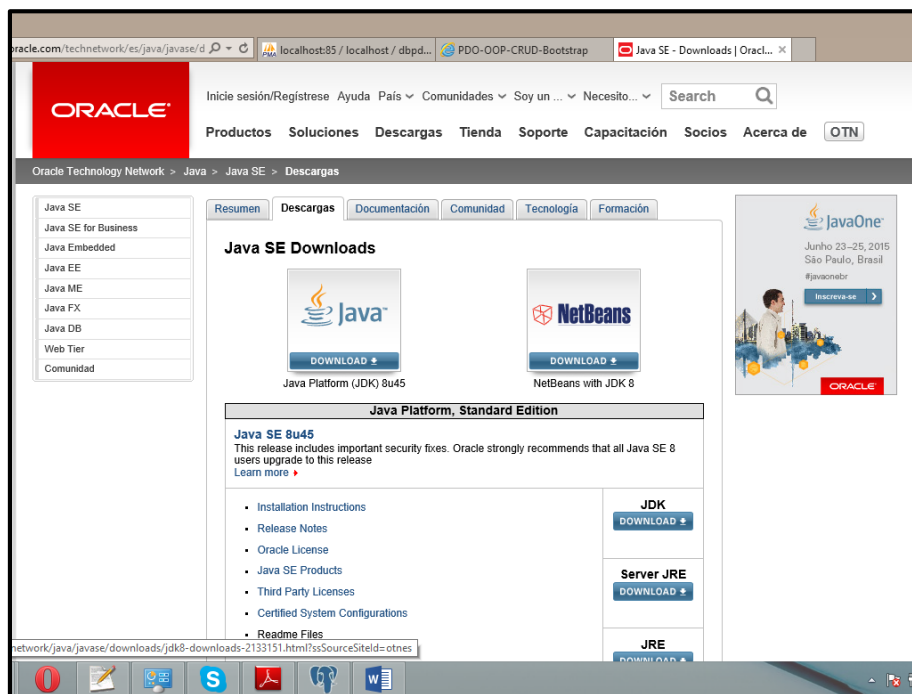
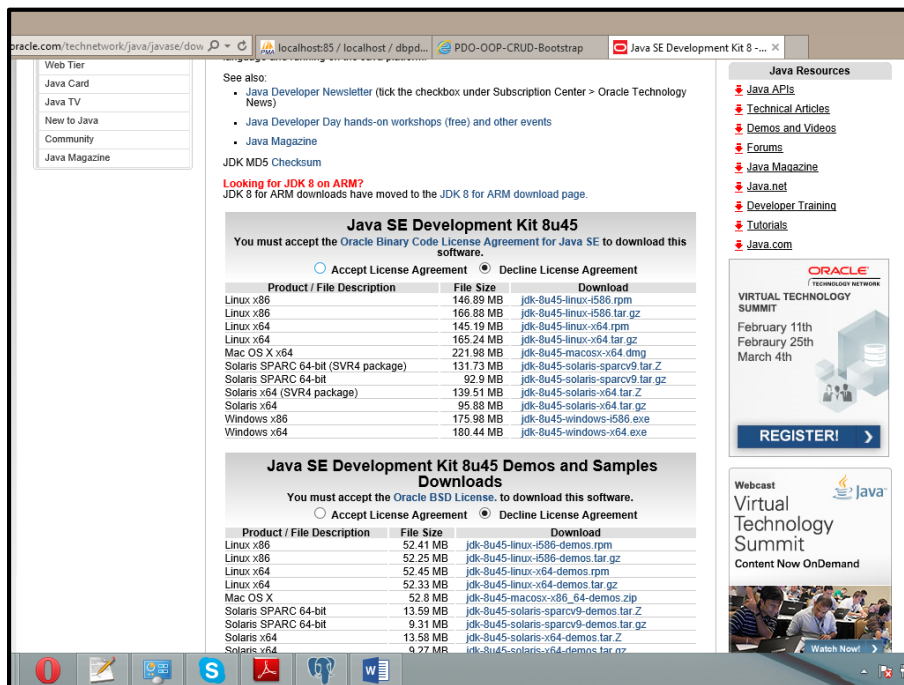




Figura 5.16: Instalación de Java
Fuente: (Propia)

En esta parte se debe tener clara la diferencia entre el **JDK** y el **JRE**. El **JRE** (*Java Runtime Environment*) contiene la **máquina virtual Java**, y es necesario para poder ejecutar aplicaciones creadas en Java. Pero, como programador, lo que se necesita es el kit de desarrollo, el **JDK**, que además incluye el propio **JRE**. (Montero, 2013)

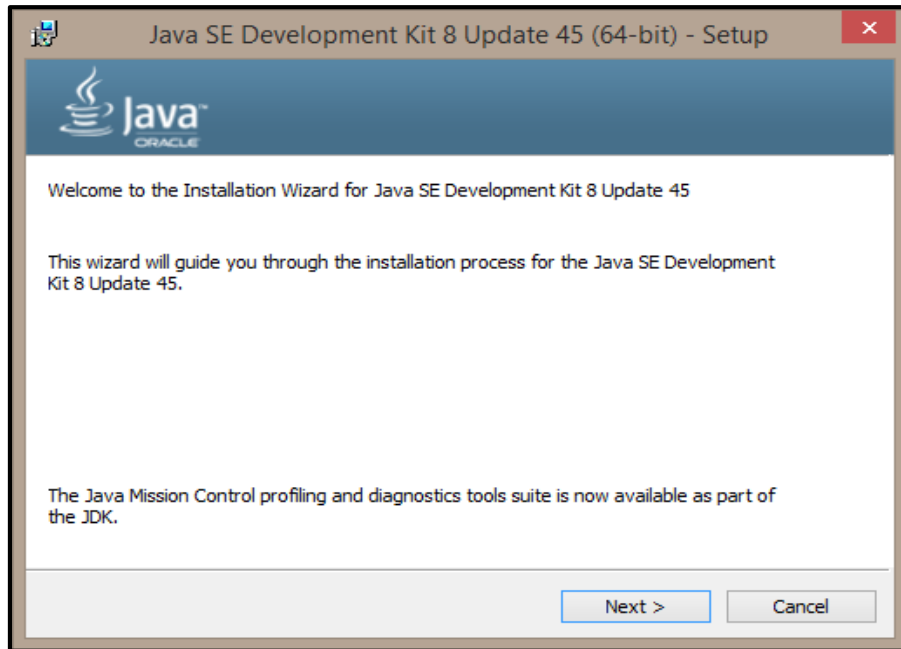
- Se deberá escoger el icono que corresponde al **JDK** para acceder a la página que contiene los enlaces de descarga.



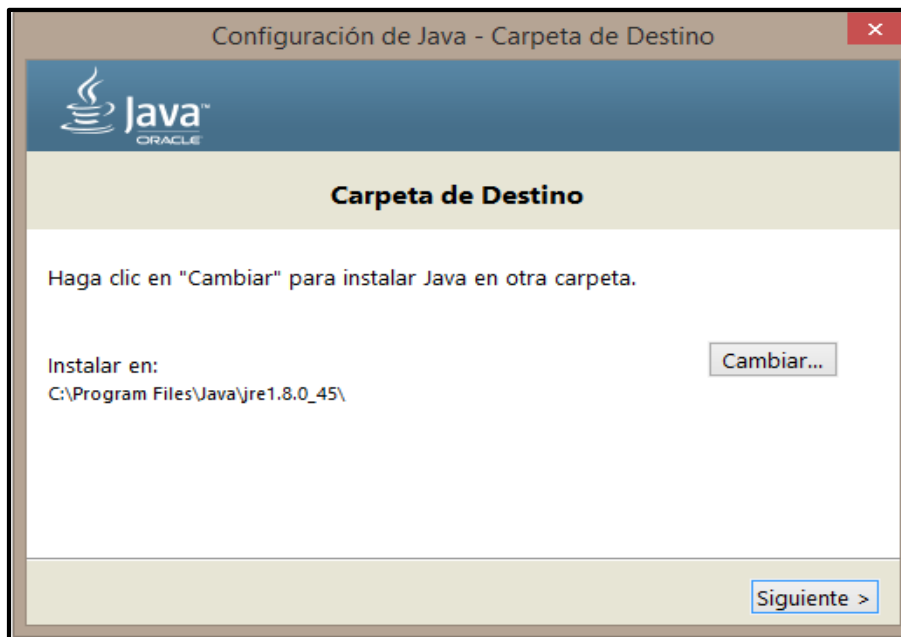
- Aceptar los términos de licencia de *Oracle* para que los enlaces se activen.
- Elige la opción adecuada en función del sistema operativo ya sea de 32 o 64 bits.

Windows x86	88.75 MB	 jdk-7u17-windows-i586.exe
Windows x64	90.42 MB	 jdk-7u17-windows-x64.exe

- Posteriormente se deberá abrir la ubicación del archivo donde se guardó el ejecutable de la descarga y se procede a la instalación.



- Como siguiente paso se deberá dar clic en Next y aparecerá una pantalla en la cual indica la ubicación donde el archivo se instalará, clic en siguiente.



- Se procede a la instalación y esta recordará que 3 billones de dispositivos utilizan Java, y dar una idea de la magnitud del despliegue de Java en la actualidad. (Montero, 2013)



CONFIGURACIÓN DE LA VARIABLE DE ENTORNO

Java requiere una configuración inicial para poder usarlo en nuestro equipo o ambiente de trabajo. Ten en cuenta que puede haber pequeñas diferencias según la versión que se use (W10, W8, W7, Windows Vista, etc.). Vamos a ver paso a paso cómo configurar las variables de entorno del sistema necesarias para poder ejecutar Java.

Existen una gran variedad de variables de entorno para diferentes propósitos, entre ellos la comunicación de Java con el sistema operativo. Se verá sólo dos, las más importantes para nuestra configuración. Estas son las variables que informan al Sistema Operativo dónde y cómo ubicar Java dentro del mismo. Estas variables son: “**JAVA_HOME**” y “**PATH**”. Navega hasta Archivos de Programa | Java | jdk1.8.0_45 | bin, donde se ubican los ejecutables. (Montero, 2013)

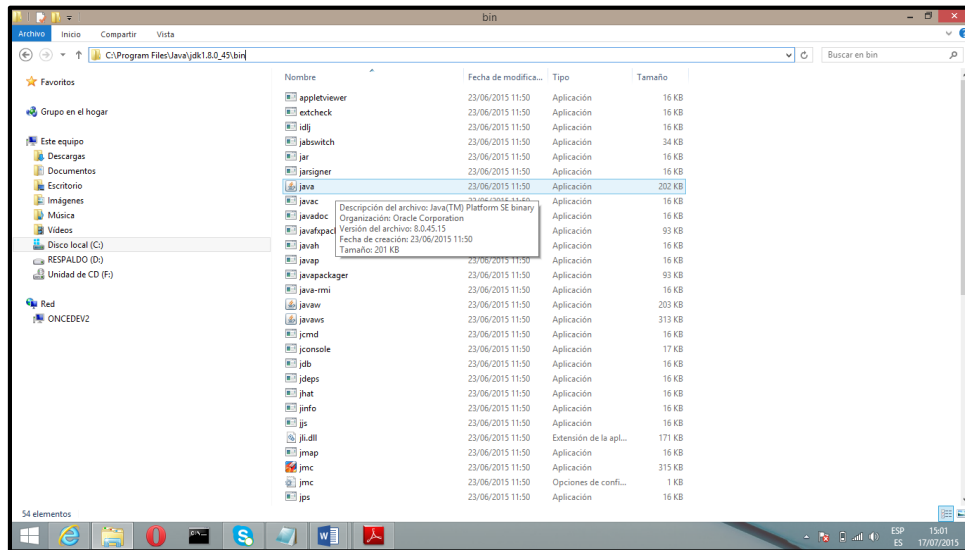


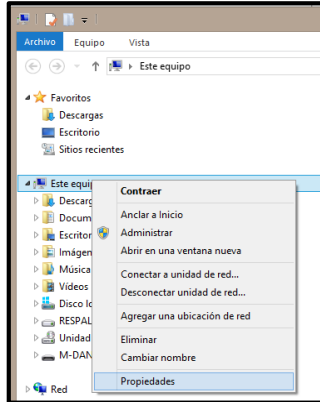
Figura 5.17: Configuración de la Variable de Entorno
Fuente: (Propia)

Los ficheros *javac.exe*, el **compilador** y *java.exe*, la **máquina virtual** Java conforman el corazón del kit. Ver figura.

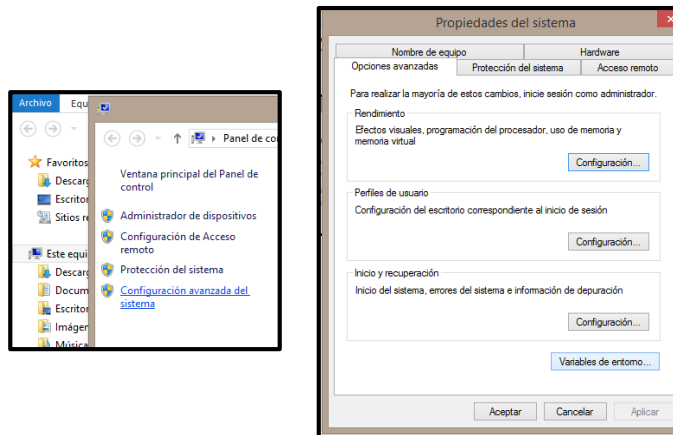
Por consiguiente, para llamar al compilador desde cualquier ubicación en la línea de comandos, es necesario que la carpeta *bin* de arriba figure en el *PATH* del sistema. (Garcia, 2015)

Es recomendable **Copiar la dirección como texto** ubicándose en la parte superior del explorador de Windows. De tal forma la ruta se copiará al portapapeles, a que la agreguemos a la variable *PATH*. (Garcia, 2015)

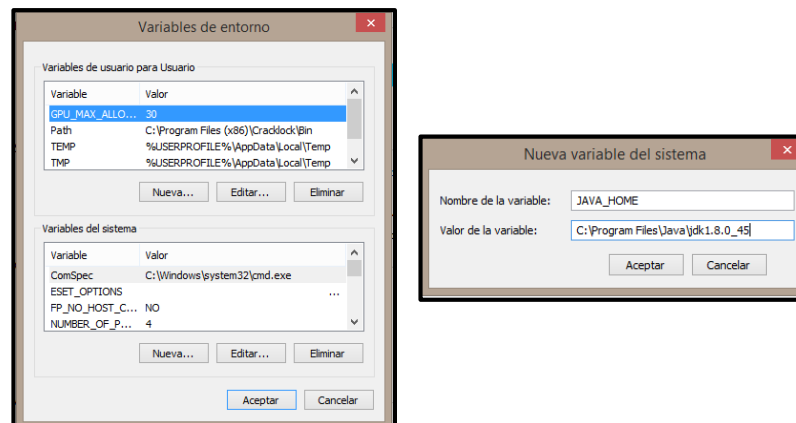
- Configurar la variable `JAVA_HOME`, es una variable de entorno del sistema que informa al sistema operativo sobre la ruta donde se encuentra instalado Java. Seguir la siguiente secuencia de pasos para configurar esta variable: abrir el explorador de Windows y situarse en Este equipo, clic derecho y seleccionar propiedades.



- Se escoge la opción configuración avanzada del sistema, luego en opciones avanzadas, inicio y recuperación y se escoge variables de entorno.



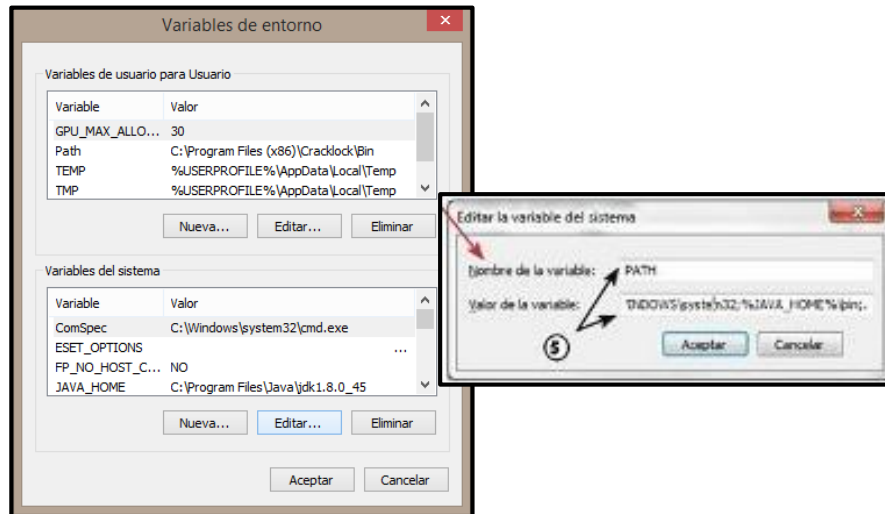
- Se escoge Nueva y aquí se escribe JAVA_HOME y la ruta en que se haya instalado Java. Puedes consultarla en el propio explorador de Windows buscando la carpeta en que se ha instalado, que habitualmente será del tipo: C:\Program Files\Java\jdk1.8.0_45, finalmente aceptar.



- Seguidamente se configura la variable PATH: se escoge la opción configuración avanzada del Sistema, luego opciones avanzadas, inicio y recuperación, variables de entorno y finalmente en el botón Editar dentro de Variables del Sistema para proceder a llenar los campos:

Nombre de variable: PATH

Valor de variable: C:\WINDOWS;C:\WINDOWS\system32;%JAVA_HOME%\bin



- Para verificar que la máquina virtual y el compilador se encuentran accesibles desde cualquier ubicación y comprobar que las versiones son las correctas, se debe abrir el compilador cmd y ejecutar los siguientes comandos
- javac -version
- java -version

```

Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>javac -version
javac 1.8.0_45

C:\Users\Usuario>javac -version
javac 1.8.0_45

C:\Users\Usuario>java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

C:\Users\Usuario>

```

INSTALACIÓN DE POSTGRES

- Para comenzar con la instalación se debe ubicar en la ruta del instalador para luego ejecutarlo.

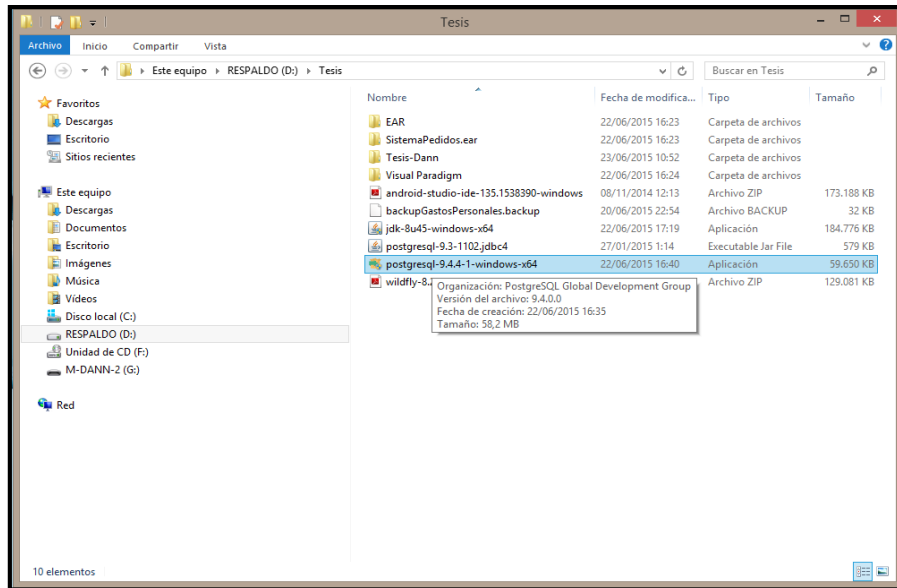
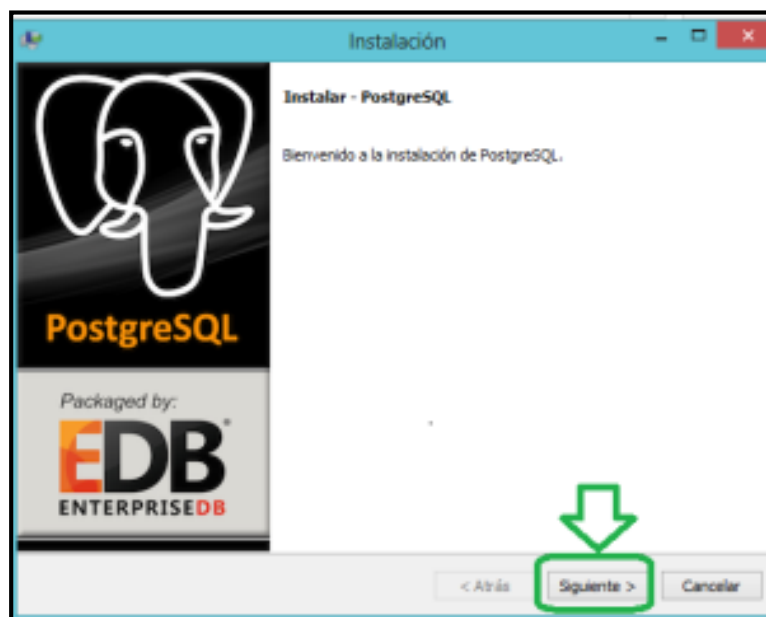


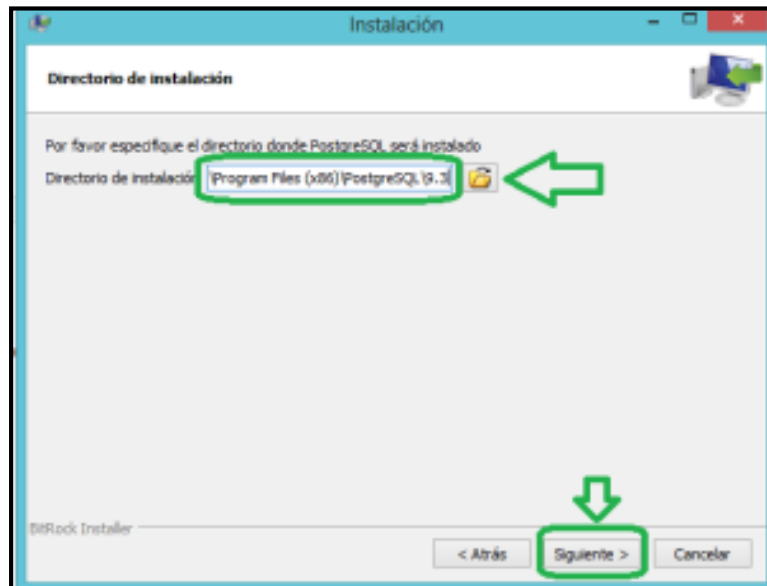
Figura 5.18: Instalación de PostgreSQL

Fuente: (Propia)

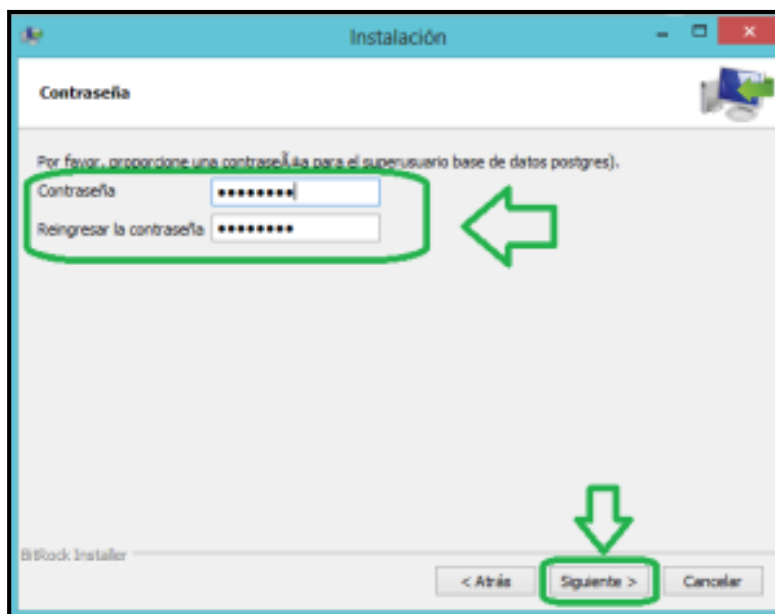
- Aparece una ventana de confirmación si se desea instalar el programa pulsar **Si**, y continuar con la instalación siguiendo las instrucciones que van apareciendo en la pantalla. Las ventanas que van apareciendo a medida que se avanza con la instalación permiten configurar el programa **PostgreSql**. Pulsar el Botón **Siguiente**.



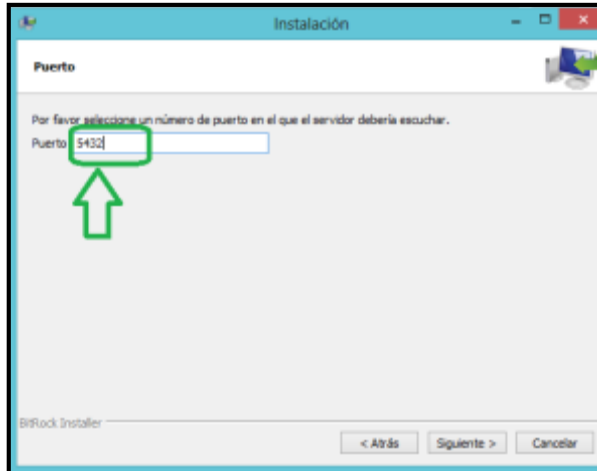
- En esta parte se puede configurar la carpeta donde se ubicará el programa instalado, se puede observar que se instala por defecto en **Archivos de Programas**, sin embargo, esta ubicación se puede personalizar. Pulsar el botón **Siguiente** y continuar la instalación.



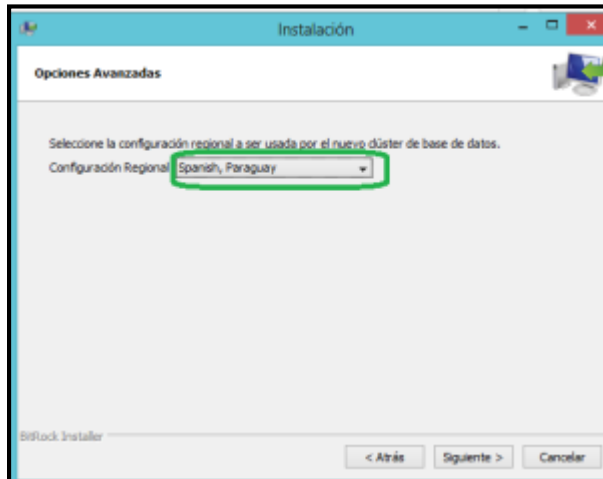
- Se le asigna una contraseña para ingresar a la base de datos. Es necesario poner una que sea fácil de recordar, generalmente se pone **postgres**. Ya que si se olvida la contraseña no se podrá acceder a la base de datos, se tiene que desinstalar y volver a instalar de cero (cosa que a veces puede dificultar su buen funcionamiento sino se hace bien). Pulsar **Siguiente** para continuar con la instalación.



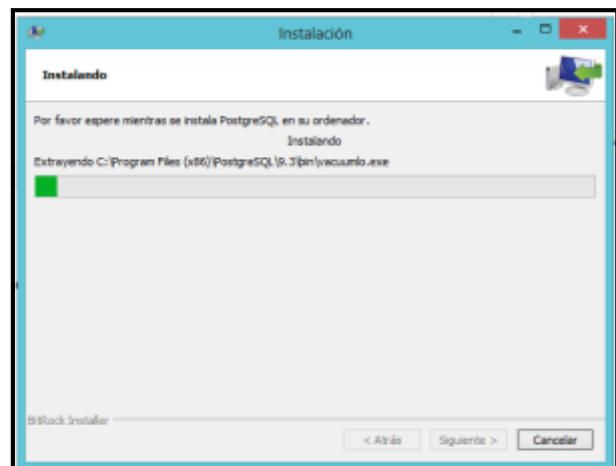
- El número de puerto que trae por defecto es **5432**, se puede cambiar a otro número de puerto o dejarlo así.



- Se elige el idioma en que se va a usar la base de datos.



- Una vez terminada la configuración pulsar el botón **Siguiente** para iniciar la instalación. Esperar hasta que termine la instalación.



- Al terminar la instalación aparece una ventana con la opción de ejecutar la base de datos

INSTALACIÓN DE ANDROID STUDIO

En esta parte se instalará la herramienta en la cual se va a desarrollar nuestro proyecto móvil, descargar de la página oficial: <https://developer.android.com/sdk/index.html>

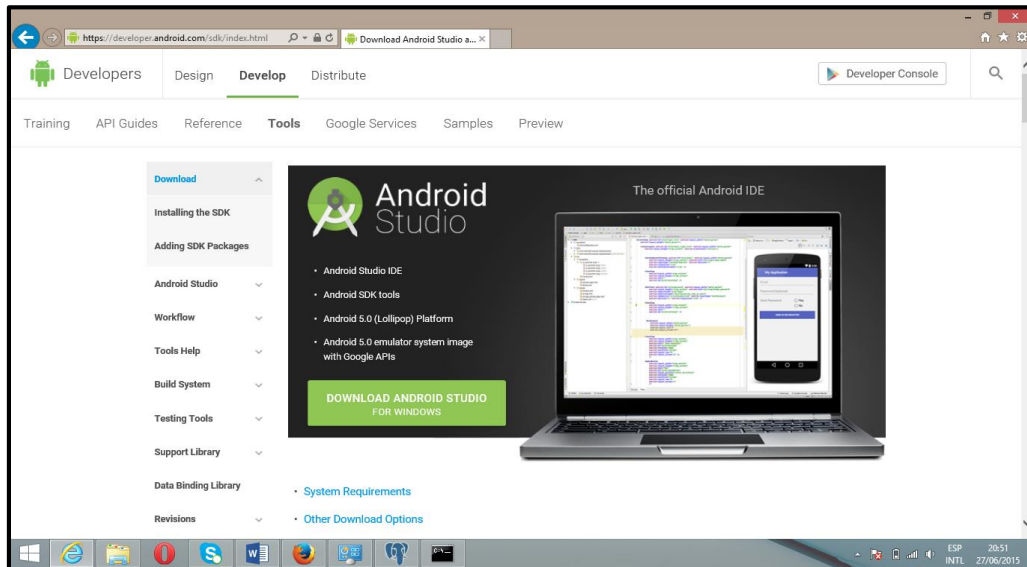
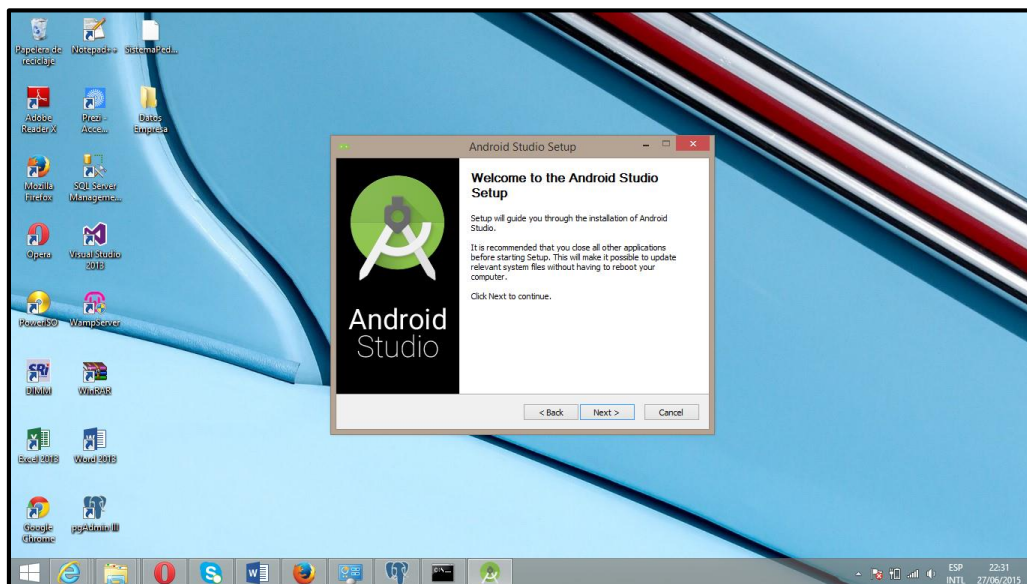
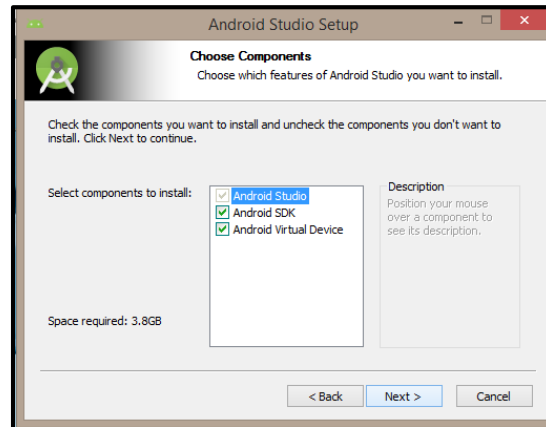


Figura 5.19: Instalación de Android Studio
Fuente: (Propia)

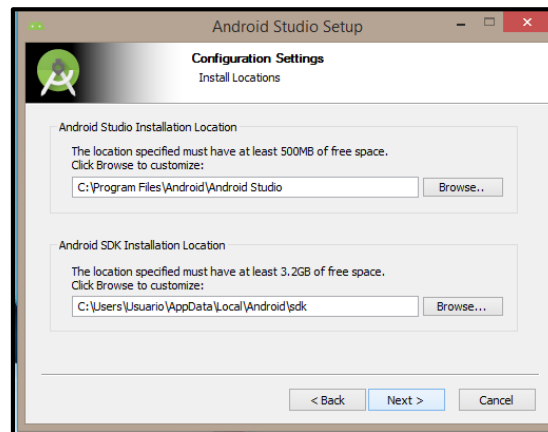
- Una vez descargado el archivo hay que proceder a la instalación.



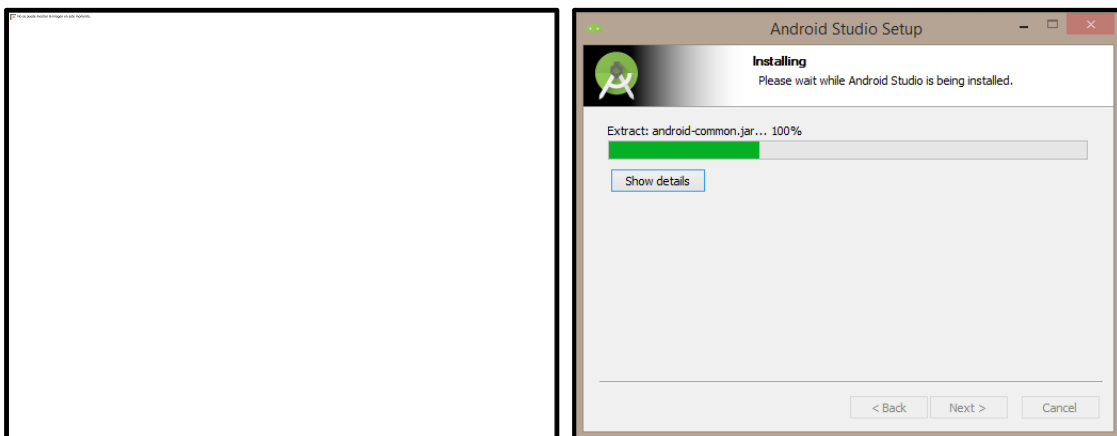
- Seleccionar los componentes.



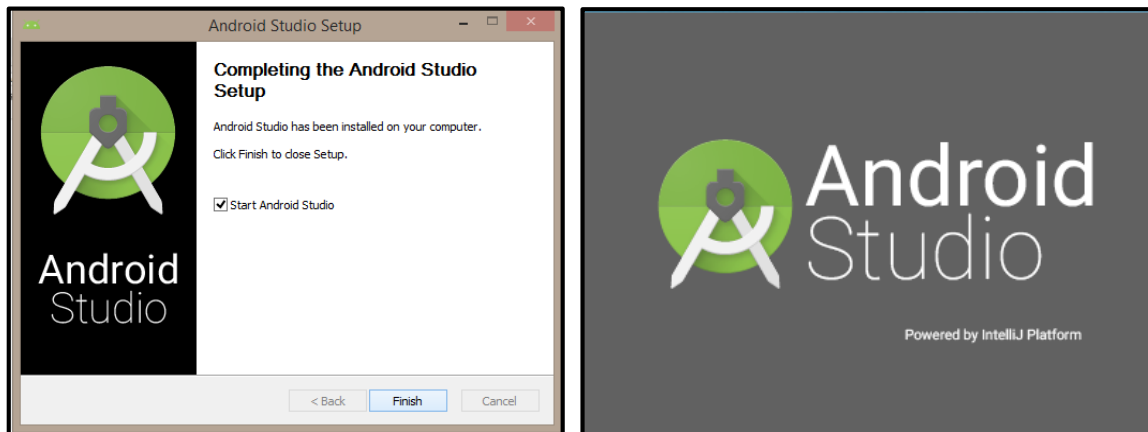
- A continuación, se configura la ubicación donde se instalará Android Studio



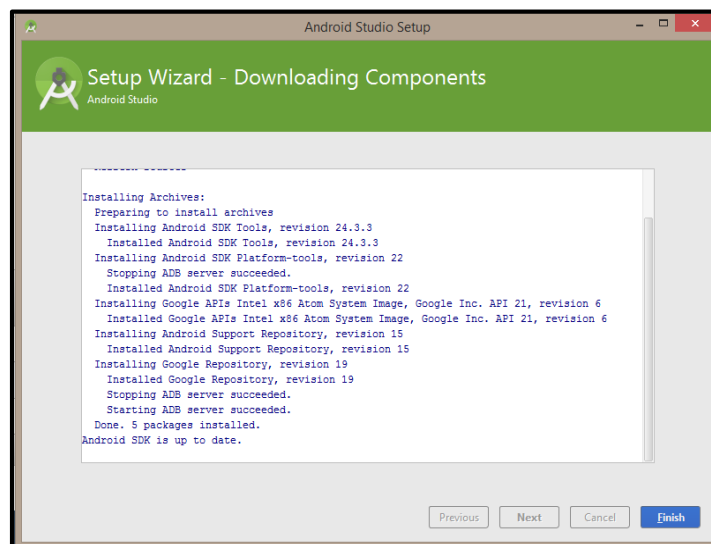
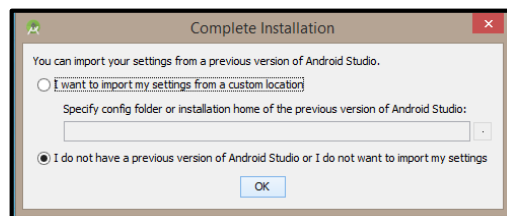
- Luego seleccionar la carpeta de inicio de Android Studio, esta carpeta por lo general ya viene seleccionada por lo que simplemente se debe presionar en el botón instalar.



- Una vez completada la instalación se abre Android Studio



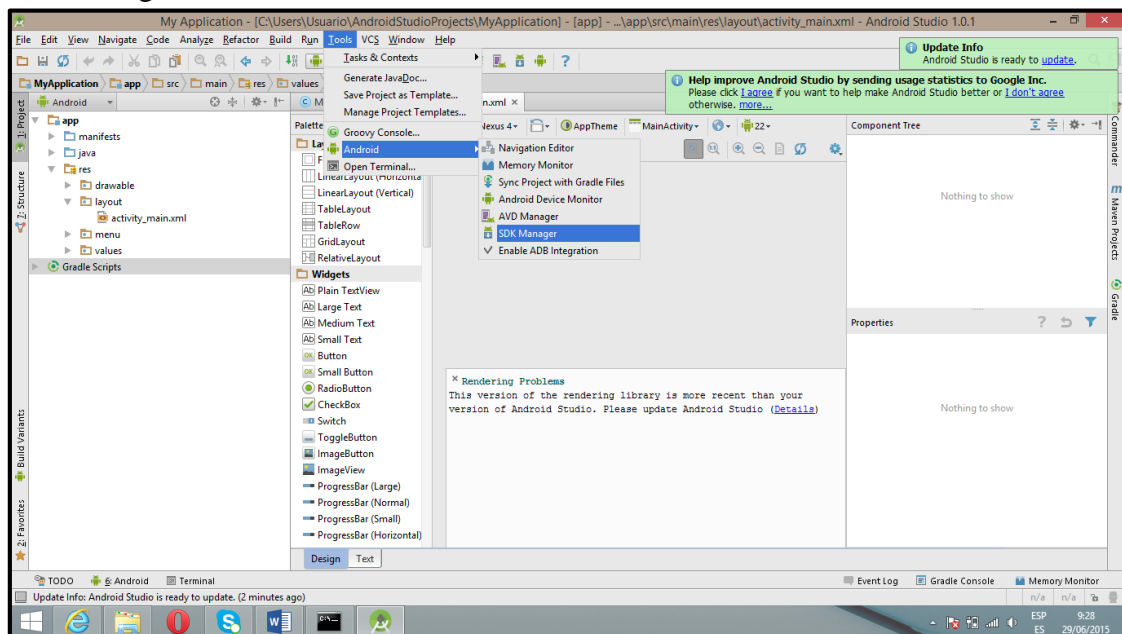
- Aparecerá una pantalla en la cual pregunta si desea importar la configuración desde una ubicación personalizada, dar clic en no y presionamos OK, seguidamente nos muestra una pantalla donde se instalarán los componentes necesarios para el funcionamiento de Android por lo que se espera un tiempo hasta que termine de instalarse, dar clic en finalizar y finalmente abrir Android Studio.



- Abrir y presionar en Iniciar un Nuevo Proyecto de Android Studio, pero antes de comenzar a utilizar esta herramienta, se debe instalar los extras de Android SDK Manager.



- Para instalar los extras de Android SDK Manager, ubicarnos en herramientas-Android SDK Manager.



- Ubicarse en la parte inferior y buscar la carpeta Extras, abrir y seleccionar todos los extras, luego dar clic en instalar Nro. Packages⁴⁶, esperar un tiempo ya que estos paquetes se descargan directamente desde el internet, por lo que es recomendable tener una conexión óptima, al final aparecerá una ventana indicando que se han instalado los extras satisfactoriamente.

⁴⁶ Packages: Paquetes

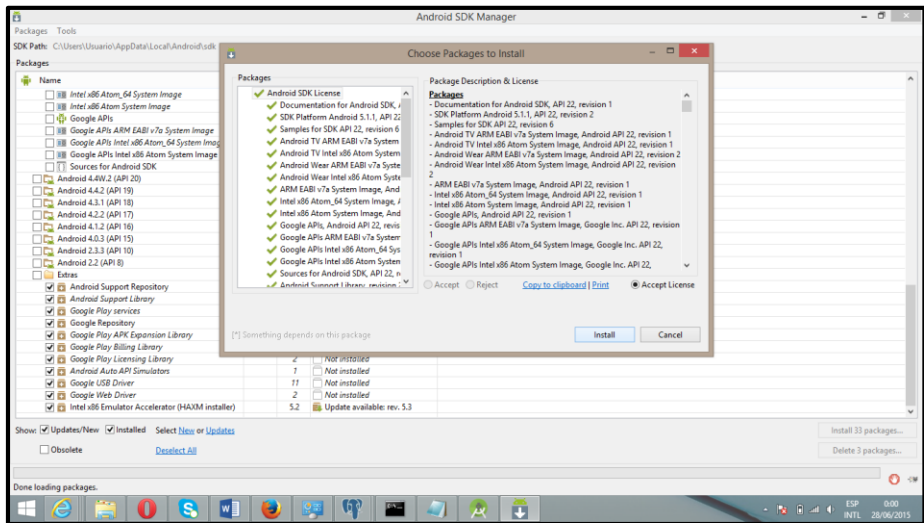
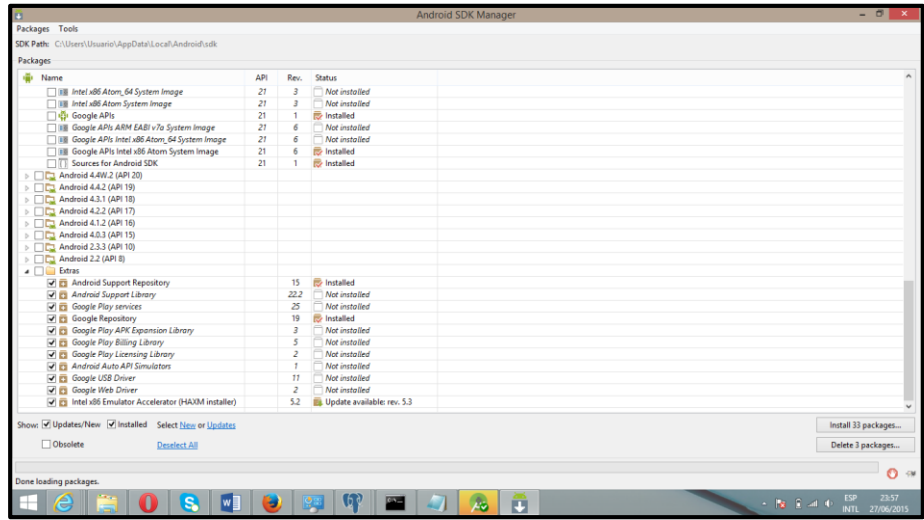
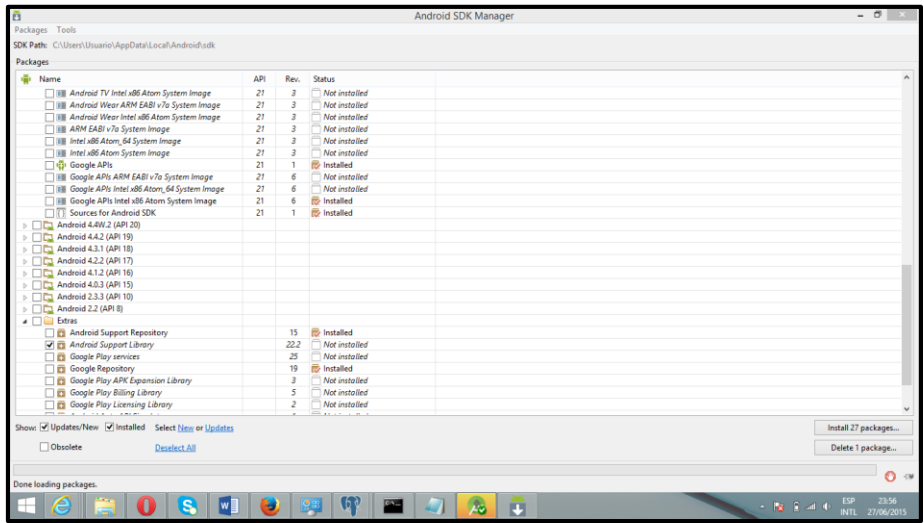


Figura 5.20: Instalación de Paquetes de Android Studio
Fuente: (Propia)

ESTRUCTURA DEL PROYECTO ANDROID

- Para crear un nuevo proyecto se debe ejecutar Android Studio y desde la pantalla de bienvenida se escoge la opción “Start a new Android Studio project” para ejecutar el asistente de creación de nuevo proyecto.

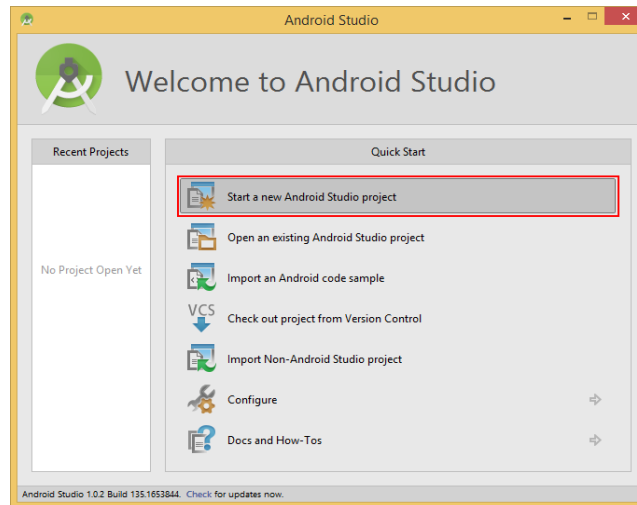
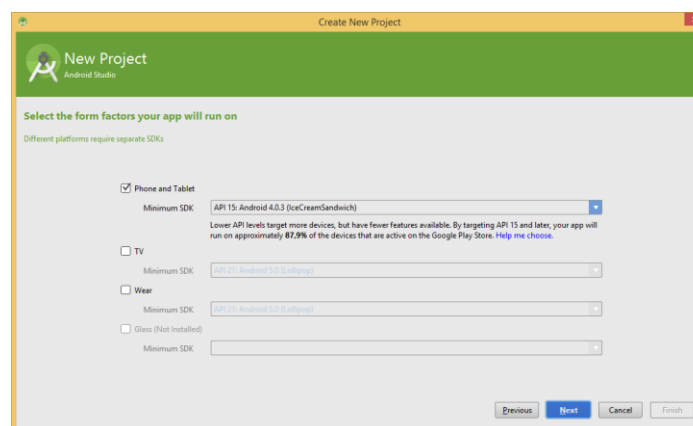


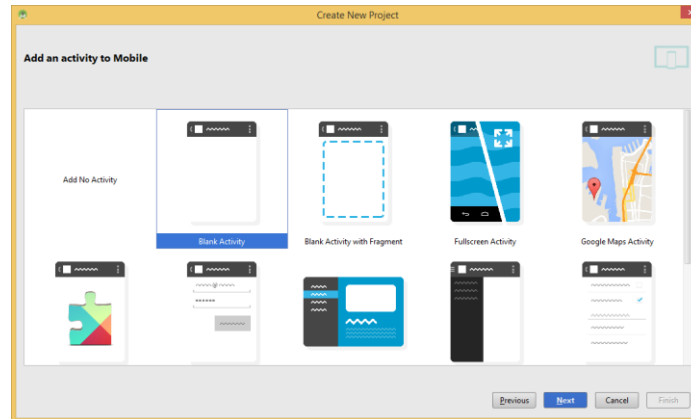
Figura 5.21: Estructura de Un Proyecto Android
Fuente: (Propia)

- En la pantalla inicial se indica, por el siguiente orden, el nombre de la aplicación, el dominio de la compañía y la ruta donde crear el proyecto. Seguidamente el segundo de los datos que muestra tan sólo se utilizará como paquete de nuestras clases java.

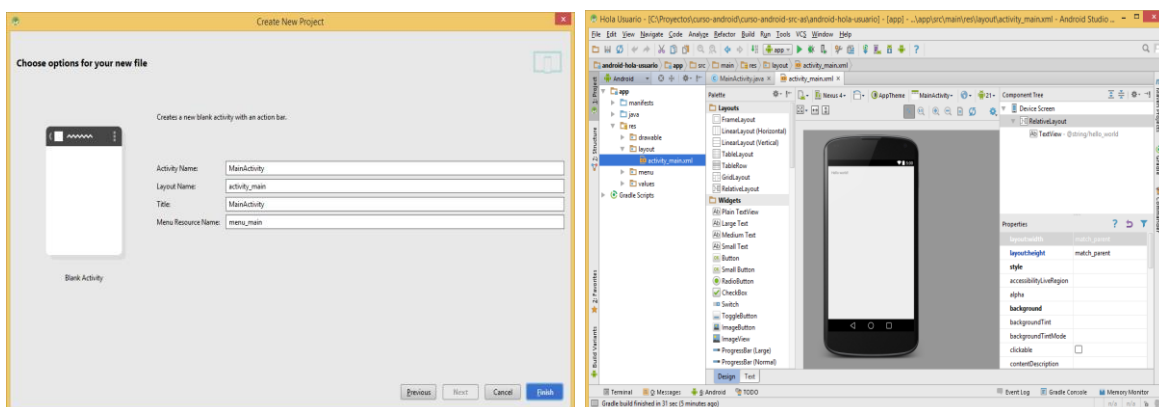


- En la siguiente pantalla del asistente configurar las plataformas y APIs que va a utilizar la aplicación. En este caso enfocarse, en aplicaciones para teléfonos y tablets, en cuyo caso tan sólo se debe seleccionar la API mínima (es decir, la versión mínima de Android) que soportará la aplicación.

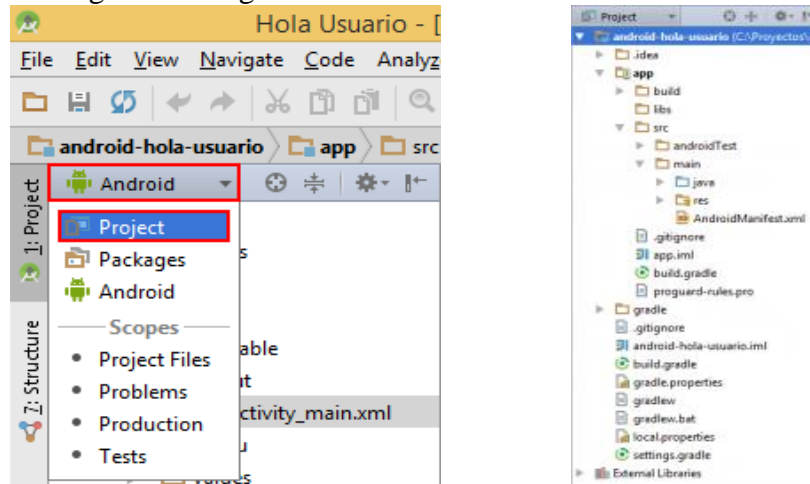
- En la siguiente pantalla del asistente elegir el tipo de *actividad* principal de la aplicación. Entender por ahora que una *actividad* es una “ventana” o “pantalla” de la aplicación. Para empezar, seleccionar *BlankActivity*, que es el tipo más sencillo.



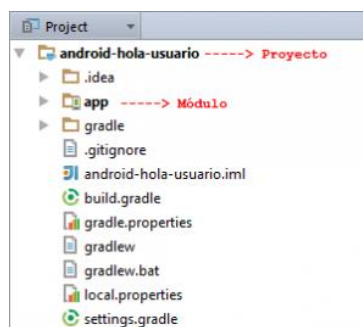
- Por último, en el siguiente paso del asistente se indica los datos asociados a esta actividad principal que se acaba de elegir, donde se indica el nombre de su clase java asociada (*Activity Name*) y el nombre de su *layout xml* (algo así como la interfaz gráfica de la actividad, se verá más adelante), su título, y el nombre del recurso XML correspondiente a su menú principal. No se debe preocupar mucho por ahora de todos estos datos por lo que se puede dejar todos los valores por defecto.
- Una vez configurado todo pulsar el botón *Finish* y Android Studio creará toda la estructura del proyecto y los elementos indispensables que debe contener. Si todo va bien aparecerá la pantalla principal de Android Studio con el nuevo proyecto creado.



- En la parte izquierda, se puede observar todos los elementos creados inicialmente para el nuevo proyecto Android, Para entender mejor la estructura del proyecto cambiar momentáneamente la forma en la que Android Studio nos la muestra. Para ello, pulsar sobre la lista desplegable situada en la parte superior izquierda, y cambiar la vista de proyecto a “Project”.
- Tras hacer esto, la estructura del proyecto cambia un poco de aspecto y pasa a ser como se observa en la siguiente imagen:



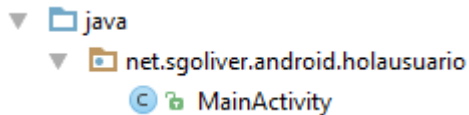
- Lo inicial que se debe diferenciar son los conceptos de *proyecto* y *módulo*. La entidad *proyecto* encierra a todos los elementos y es única. Se puede incluir dentro de un proyecto diversos *módulos*, como, por ejemplo: versiones diferentes de una misma aplicación, o distintos componentes de un sistema (aplicación móvil, aplicación servidor, librerías). En la mayoría de los casos, se trabajará con un proyecto que contendrá un sólo módulo que corresponde a la aplicación principal. Por ejemplo, se tiene el proyecto “android-hola-usuario” que contiene al módulo “app” que contendrá todo el software de la aplicación de ejemplo.



Seguidamente, se describen los contenidos primordiales del módulo principal.

- Carpeta /app/src/main/java

Contiene clases auxiliares y todo el código fuente del proyecto, entre otras. En primera instancia Android Studio creará el código básico de la pantalla principal (*actividad* o *activity*), recordar que en este caso era MainActivity y siempre bajo la estructura del paquete java definido durante la creación del proyecto.



- Carpeta /app/src/main/res/

Contendrá todos los ficheros de recursos necesarios para el proyecto: imágenes, layouts, cadenas de texto, etc. Los diferentes tipos de recursos se pueden distribuir entre las siguientes subcarpetas:

Carpeta	Descripción
/res/drawable/	Contiene las imágenes y otros elementos gráficos. Para poder definir diferentes recursos dependiendo de la resolución y densidad de la pantalla se suele dividir en varias subcarpetas: <ul style="list-style-type: none"> • /drawable (recursos independientes de la densidad) • /drawable-ldpi (densidad baja) • /drawable-mdpi (densidad media) • /drawable-hdpi (densidad alta) • /drawable-xhdpi (densidad muy alta) • /drawable-xxhdpi (densidad muy muy alta :)
/res/layout/	Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Para definir distintos layouts dependiendo de la orientación del dispositivo se puede dividir también en subcarpetas: <ul style="list-style-type: none"> • /layout (vertical) • /layout-land (horizontal)
/res/anim/ /res/animator/	Contienen la definición de las animaciones utilizadas por la aplicación.
/res/color/	Contiene ficheros XML de definición de colores según estado.
/res/menu/	Contiene la definición XML de los menús de la aplicación.
/res/xml/	Contiene otros ficheros XML de datos utilizados por la aplicación.
/res/raw/	Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.
/res/values/	Contiene otros ficheros XML de recursos de la aplicación, como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), arrays de valores (arrays.xml), tamaños (dimens.xml), etc.

Tabla 6.11: Contenido de Subcarpetas de un Proyecto Android

Fuente: (Oliver, 2014)

Como se observa en la tabla, existen carpetas cuyo nombre incluye un sufijo adicional, como ejemplo “values-w820dp”. Estos y otros sufijos, se aplican para definir recursos independientes para determinados dispositivos según sus características. (Oliver, 2014)

De esta forma, los recursos incluidos en la carpeta “values-**w820dp**” se aplicarían sólo a pantallas con más de 820dp de ancho, o los incluidos en una carpeta llamada “values-**v11**” se aplicarían tan sólo a dispositivos cuya versión de Android sea la 3.0 (API 11) o superior. (Oliver, 2014)

Entre estas carpetas se tiene:

- Fichero /app/src/main/AndroidManifest.xml

Comprende la definición en XML, como por ejemplo su identificación (icono, nombre, etc), sus componentes (servicios, pantalla), o los permisos para su funcionamiento. (Oliver, 2014)

- Fichero /app/build.gradle

Contiene información necesaria para la compilación del proyecto, por ejemplo, la versión del SDK de Android utilizada para compilar, la mínima versión de Android que soportará la aplicación, referencias a las librerías externas utilizadas, etc. (Oliver, 2014)

- Carpeta /app/libs

Comprende las librerías javas externas (ficheros *.jar*) que utilice la aplicación. Normalmente se hace referencia a dichas librerías en el fichero *build.gradle* descrito en el punto anterior, de forma que se incorporen en el proceso de compilación de la aplicación. (Oliver, 2014)

- Carpeta /app/build/

Comprende una serie de elementos de código generados automáticamente al compilar el proyecto, tras cada compilación es importante no modificar de forma manual bajo ninguna circunstancia. (Oliver, 2014)

INSTALACIÓN DE JBOSS DEVELOPER

Se procede a la instalación de Jboss Developer en su versión 8, para esto se debe situar en la página oficial: <http://www.jboss.org/products/devstudio> y finalmente comenzar con la descarga.

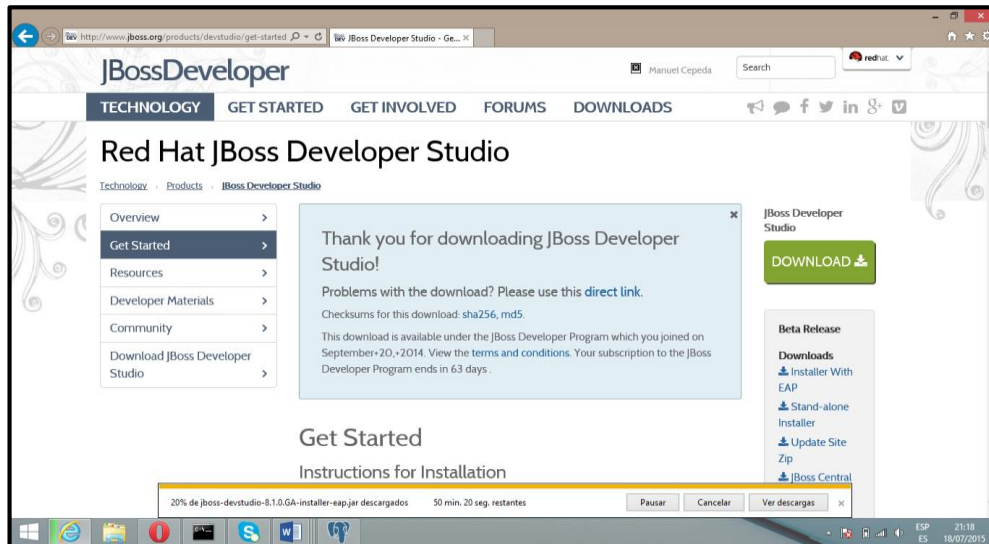
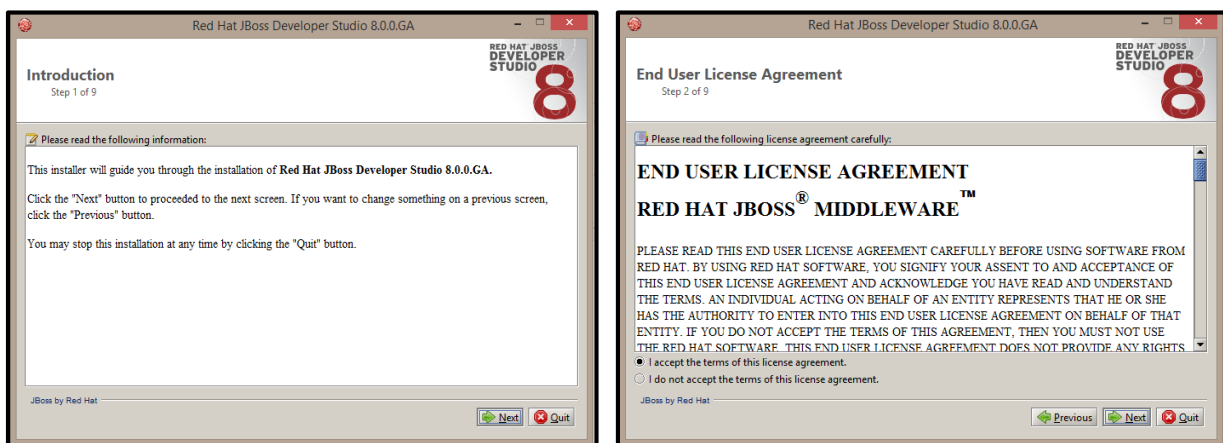
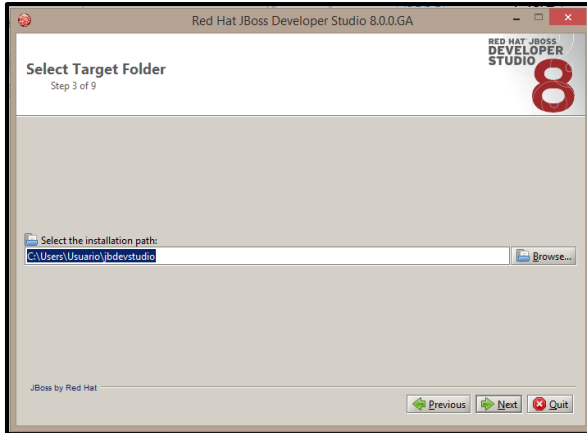


Figura 5.22: Instalación de Jboss Developer Studio
Fuente: (Propia)

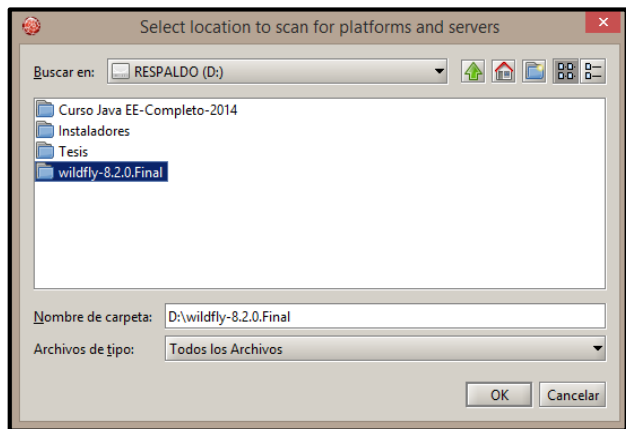
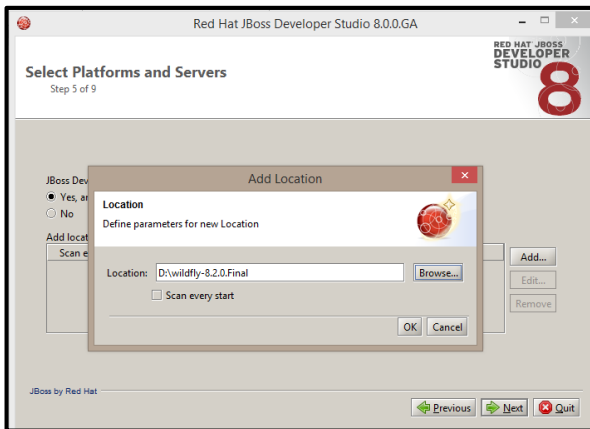
- Una vez terminada la descarga abrir la ubicación y comenzar con la instalación, se da clic en siguiente, aceptar.



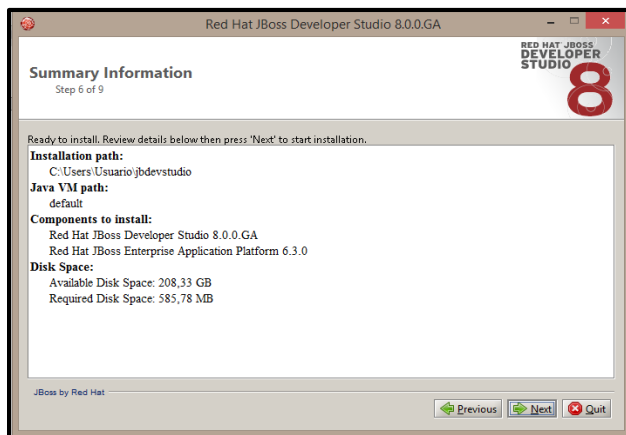
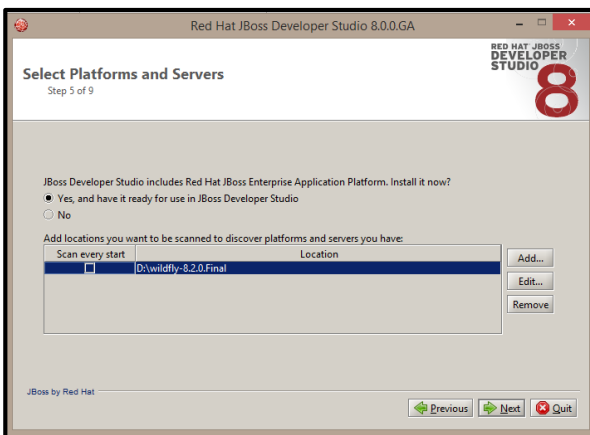
- Se selecciona el destino de la carpeta y la ubicación del JDK de la máquina virtual de Java.



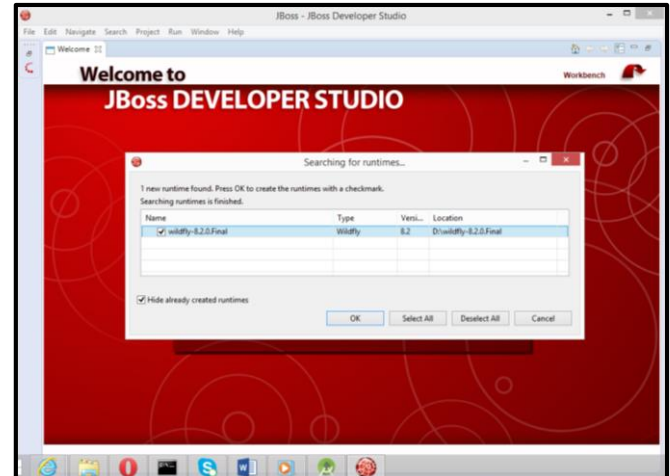
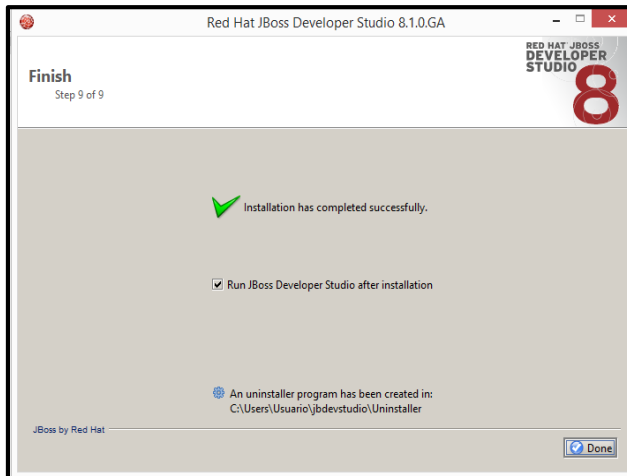
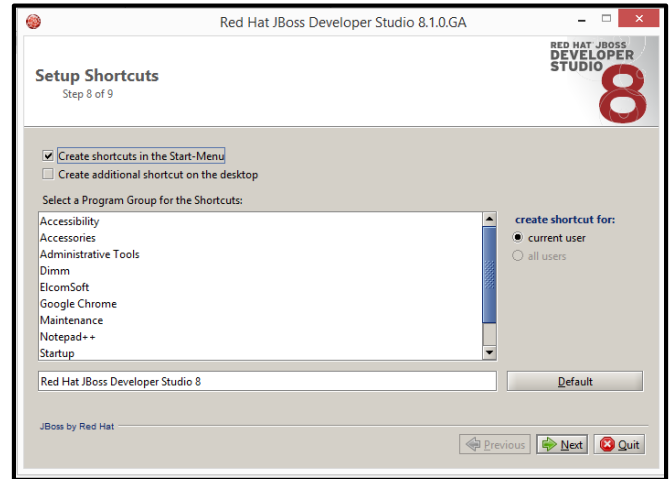
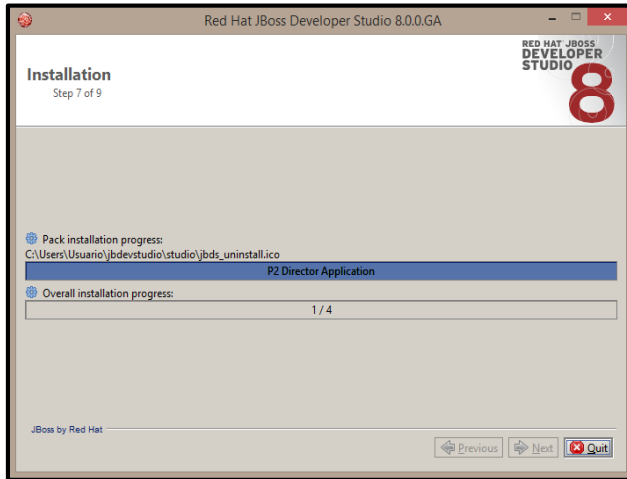
- Se selecciona la ubicación para agregar el servidor Wildfly y agregar.



- Finalmente se da clic en siguiente y se muestra un resumen de los componentes instalados.



- A continuación, se selecciona en siguiente y comenzará la instalación, luego se visualiza una pantalla, la cual muestra si se desea crear accesos directos en el escritorio, esto se deja por defecto, clic en siguiente y por último muestra una pantalla de instalación exitosa.



- Finalmente, al abrir Jboss pide pulsar OK para crear tiempos de ejecución con el servidor de aplicaciones Wildfly.

COPIAMOS EL JDBC PARA CONECTARNOS A LA BDD

- Como siguiente paso será descargar el jdbc dependiendo de la versión de PostgreSQL que se tenga instalado, en este caso se descarga el jdbc de la página: <https://jdbc.postgresql.org/download.html>

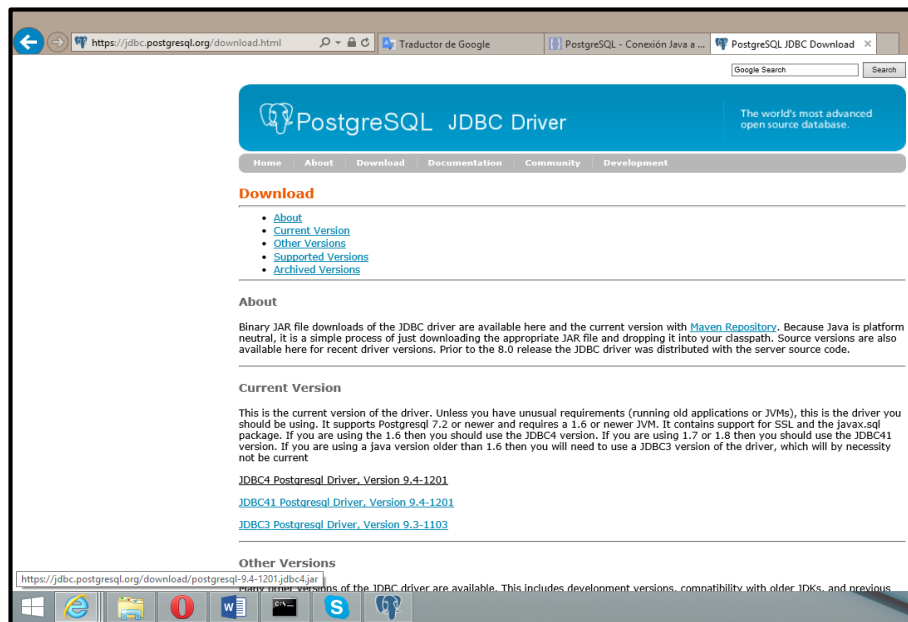
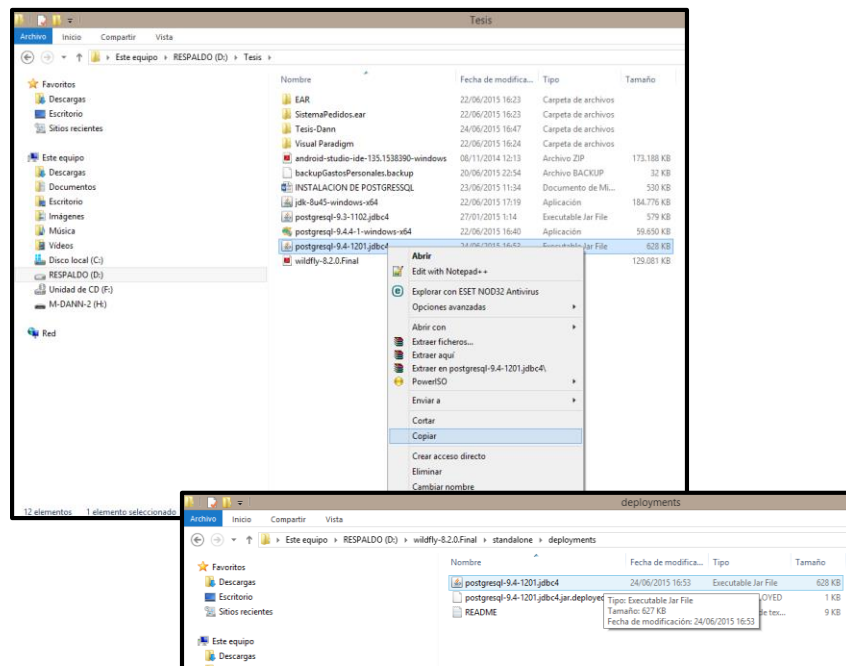


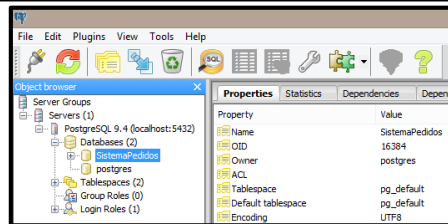
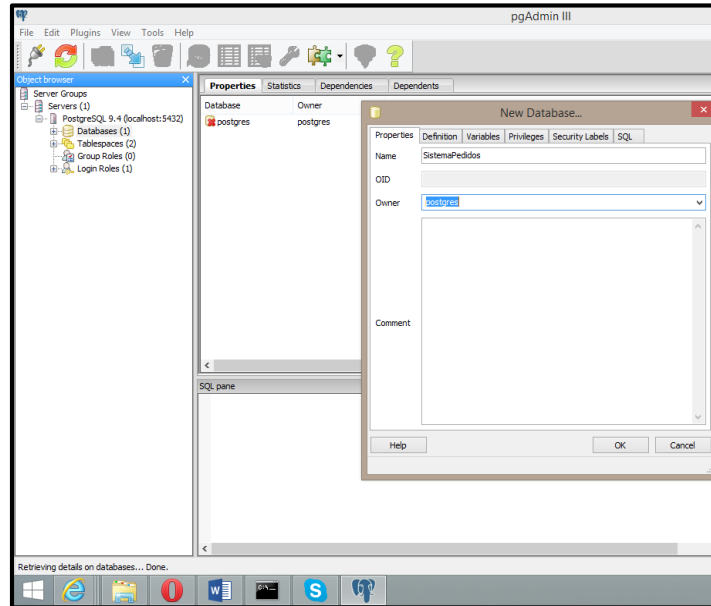
Figura 5.23: Agregar Jdbc para conectarnos a la BDD
Fuente: (Propia)

- Luego el jdbc descargado, se lo copia dentro de la carpeta del servidor de aplicaciones WildFly ubicada en D: Wilfly-8.2.0. Final – standalone – deployments.



RESTAURAR LA BASE DE DATOS

- Se abre Postgres, seguidamente dar clic derecho sobre Base de Datos y escoger Nueva Base de Datos por consiguiente colocar el nombre: Sistema Pedidos, clic en OK.



- Restaurar el Backup, clic derecho sobre la BDD que se creó anteriormente y escoger Backup, ubicar el archivo: SistemaPedidos.sql, y agregarlo.

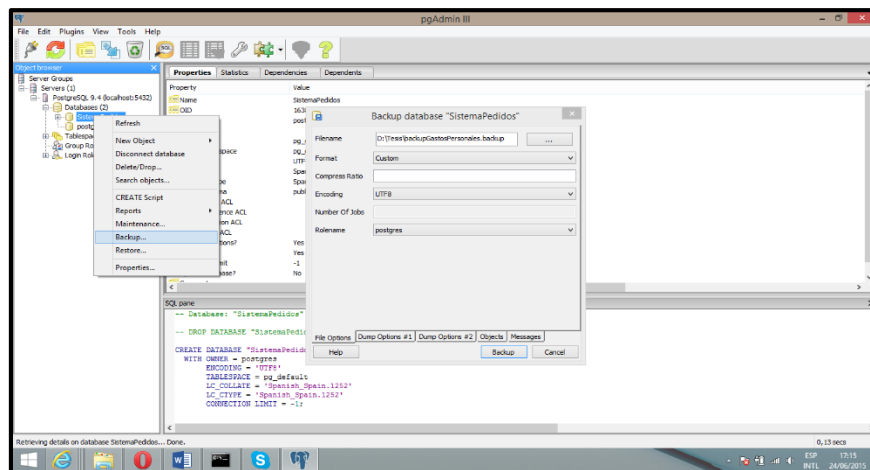
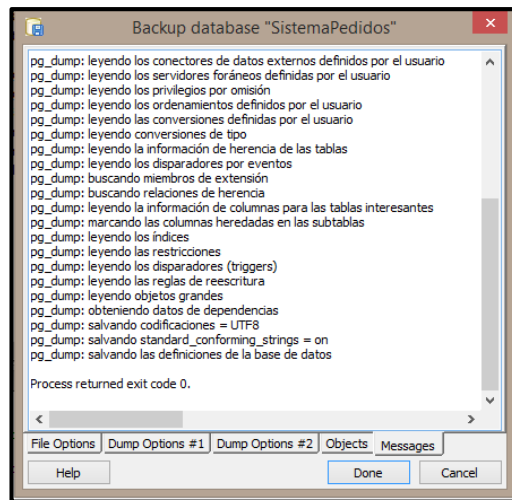


Figura 5.24: Restaurar la Base de Datos
Fuente: (Propia)

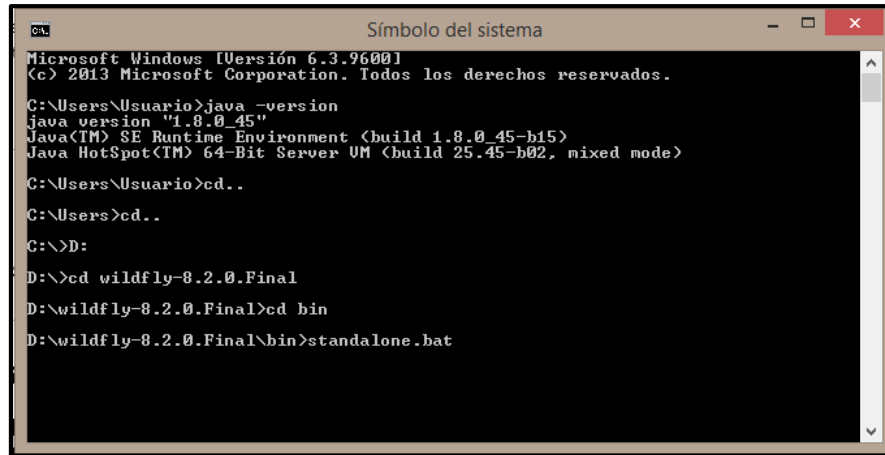
- Una vez realizado el Backup⁴⁷, aparecerá la siguiente pantalla para luego dar clic en Done, y listo se creará la Base de Datos.



⁴⁷ Backup: Se refiere a la copia y archivo de datos de la computadora de modo que se puede utilizar para restaurar la información original después de una eventual pérdida de datos.

LEVANTAR EL SERVIDOR DE APLICACIONES JBOSS/WILDFLY

- Se abre símbolo del sistema y se debe ubicar en la ruta donde está el servidor de aplicaciones Wildfly-8.2.0.Final, en este caso está ubicado en la unidad D.




```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b15)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

C:\Users\Usuario>cd..
C:\Users>cd..
C:\>D:
D:\>cd wildfly-8.2.0.Final
D:\wildfly-8.2.0.Final>cd bin
D:\wildfly-8.2.0.Final\bin>standalone.bat
```

Figura 5.25: Levantar el Servidor de Aplicaciones Wilffly
Fuente: (Propia)

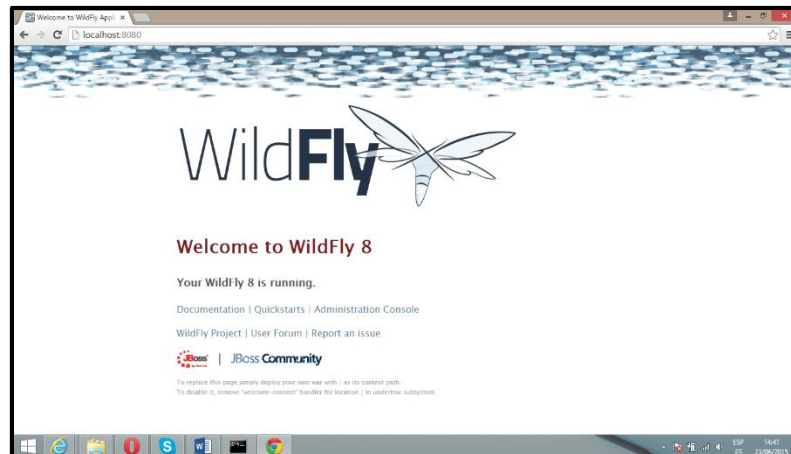
- Ejecutar el script, esto permitirá el acceso a la conexión del dispositivo móvil al servidor JbossWildFly.



```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>cd..
C:\Users>cd..
C:\>D:
D:\>cd wildfly-8.2.0.Final
D:\wildfly-8.2.0.Final>cd bin
D:\wildfly-8.2.0.Final\bin>standalone.bat -b 0.0.0.0_
```

- Una vez ejecutado el script ya se puede ingresar por cualquier navegador al servidor Wildfly por medio de <http://localhost:8080>.



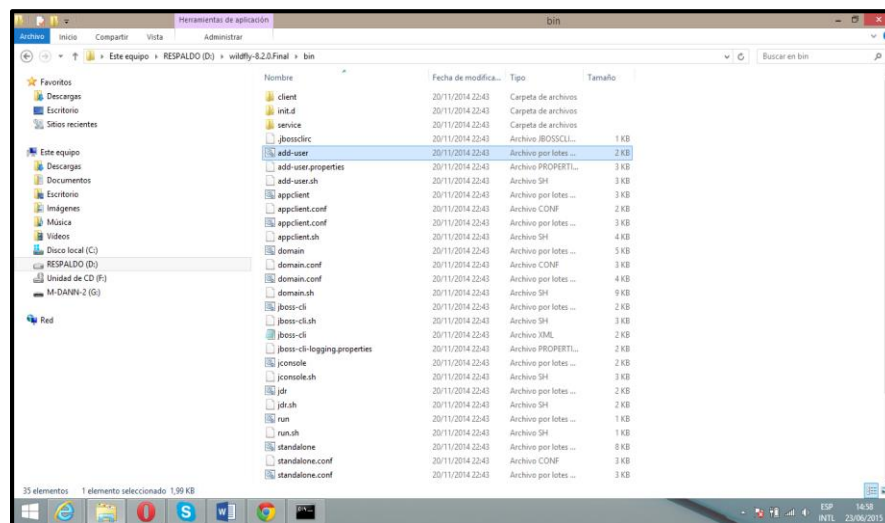
CREACIÓN DEL USUARIO POR MEDIO DE LA ADMINISTRACIÓN DE CONSOLA

- Dar clic en Administración de Consola de Wildfly y proceder a la creación del Usuario.



Figura 5.26: Creación del Usuario Consola de Administración
Fuente: (Propia)

- Luego ubicarse dentro de la carpeta donde se copió el servidor Wildfly-8.2.0.Final-bin-add user, para agregar el usuario y la contraseña.



- Se asigna el usuario: **root** y la contraseña: **dann**, luego de asignar el usuario pregunta si está seguro de asignar el usuario root, por lo que se debe escribir yes, ingresar la contraseña presionar enter, finalmente pregunta si es la contraseña que desea utilizar escribir yes, y al final pregunta si este usuario es con el que se trabajará, escribir yes.

```

C:\Windows\system32\cmd.exe
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : root_

```

```

C:\Windows\system32\cmd.exe
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : root
[The username 'root' is easy to guess.
Are you sure you want to add user 'root'? yes/no? yes
Password recommendations are listed below. To modify these restrictions edit the
add-user.properties configuration file.
- The password should not be one of the following restricted values (root, admin,
administrator)
- The password should contain at least 8 characters, 1 alphabetic character(s),
1 digit(s), 1 non-alphanumeric symbol(s)
- The password should be different from the username
Password : _

```

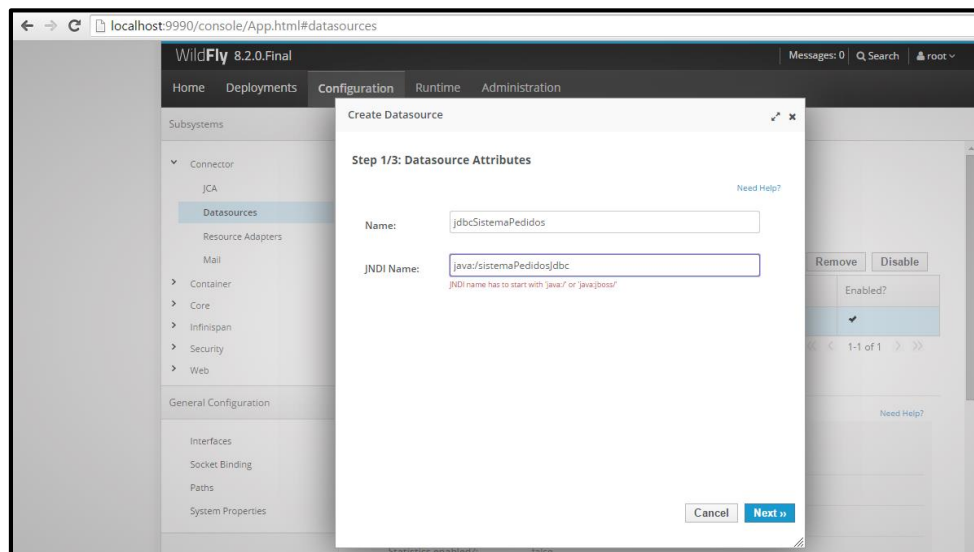
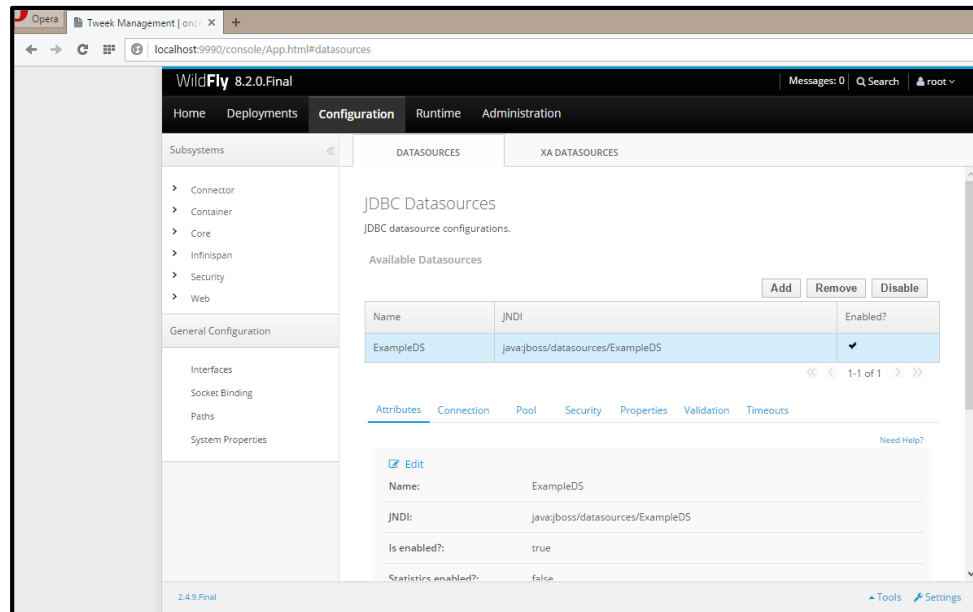
```

C:\Windows\system32\cmd.exe
- The password should be different from the username
Password : JBAS015269: Password must have at least 8 characters!
Are you sure you want to use the password entered yes/no? yes
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated
list, or leave blank for none)l l:
about to add user 'root' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'root' to file 'D:\wildfly-8.2.0.Final\standalone\configuration\mgmt-
users.properties'
Added user 'root' to file 'D:\wildfly-8.2.0.Final\domain\configuration\mgmt-user
s.properties'
Added user 'root' with groups to file 'D:\wildfly-8.2.0.Final\standalone\config
uration\mgmt-groups.properties'
Added user 'root' with groups to file 'D:\wildfly-8.2.0.Final\domain\configur
ation\mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS pr
ocess?
e.g. for a slave host controller connecting to the master or for a Remoting conn
ection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <sec
ret value="ZGFubG==" />
Presione una tecla para continuar . . .

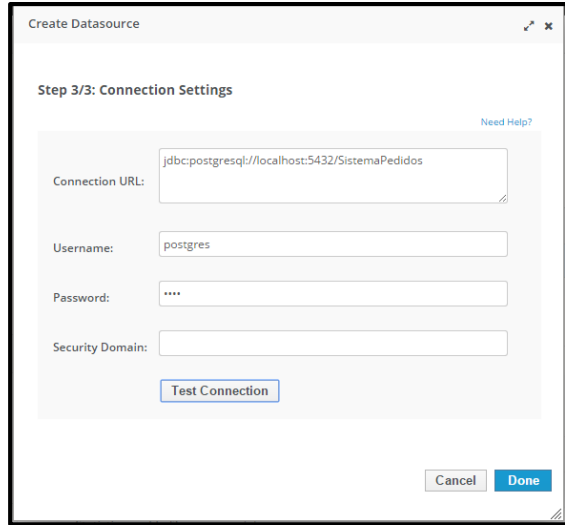
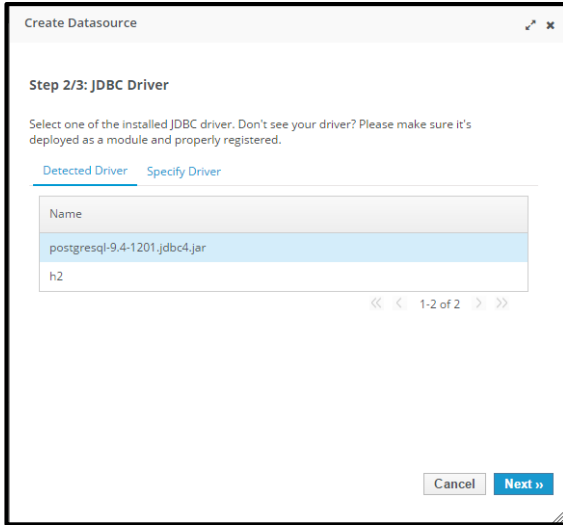
```

- Se debe ubicar nuevamente en la página de administración del servidor de aplicaciones Wildfly, luego dar clic sobre Administración de Consola, muestra una pantalla en donde se ingresa el Usuario y la Contraseña que se creó anteriormente: Usuario: root contraseña: dann, para proceder a crear el Datasource.

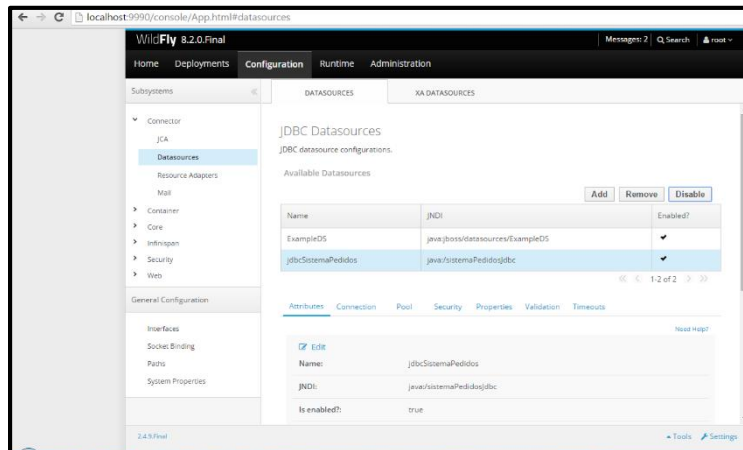
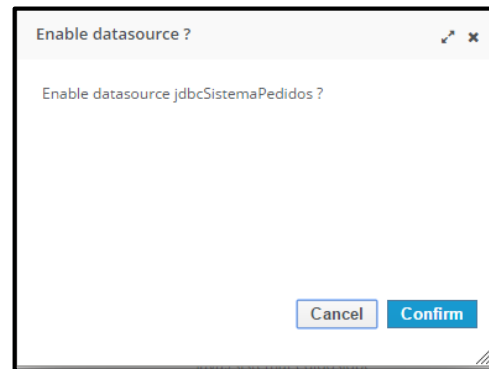
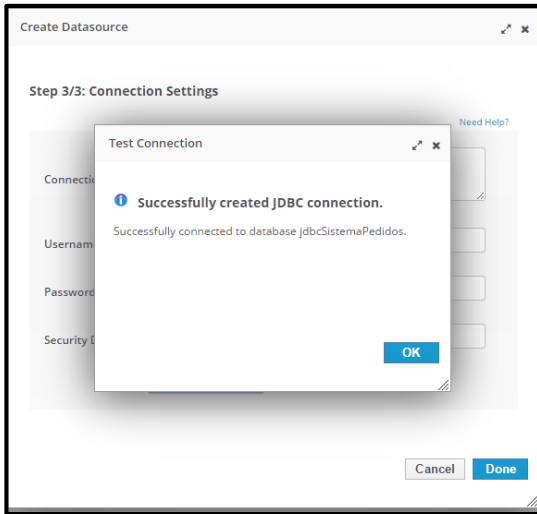
- Pulsar clic en Crear Datasource, luego clic en Add, ingresar Name y el JNDI Name para conectarse con la base de datos a través de este Datasource.



- Seguidamente dar clic en siguiente y escoger el Driver Jdbc, que se copió en la carpeta Wildfly en la unidad D, luego pulsar clic en siguiente y seguir al paso 3, finalmente asignar la conexión URL-Username-Password.



- Verificar la conexión con Test Connection, luego pulsar clic en siguiente y activar el Datasource.



- Finalmente, una vez concluidos todos los pasos, ya se puede conectar al sistema por medio de <http://localhost:8080/SistemaPedidosWeb>

