



ESCUELA SUPERIOR DE INGENIERÍA

Ingeniero Técnico en Informática de  
Gestión

ikeComercial: Gestión de ventas en  
Android

Curso 2013-2014

Daniel González Jiménez

Cádiz, 1 de Diciembre de 2013





ESCUELA SUPERIOR DE INGENIERÍA

Ingeniero Técnico en Informática de  
Gestión

ikeComercial: Gestión de ventas en  
Android

DEPARTAMENTO: Ingeniería Informática

DIRECTOR DEL PROYECTO: Juan Boubeta Puig

AUTOR DEL PROYECTO: Daniel González Jiménez

Cádiz, 1 de Diciembre de 2013

Fdo.: Daniel González Jiménez



## **Agradecimientos**

Me gustaría agradecer a todas esas personas que siempre han confiado en mí, a mis amigos, compañeros de la facultad, a toda mi familia, a Juan como tutor por su gran ayuda en la elaboración de este proyecto y las personas con las que he compartido mis mejores momentos. Gracias por estar ahí.



# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Motivación . . . . .	7
1.2. Objetivos y Alcance . . . . .	8
1.3. IkeComercial . . . . .	9
1.4. Visión general . . . . .	9
1.5. Glosario . . . . .	10
1.5.1. Acrónimos . . . . .	10
1.5.2. Definiciones . . . . .	11
<b>2. Estado del arte</b>	<b>13</b>
2.1. Comparativa de funcionalidades . . . . .	13
2.2. Comparativa de interfaces de usuario . . . . .	15
2.2.1. Pantalla de Bienvenida. . . . .	15
2.2.2. Pantalla de Menú Principal . . . . .	16
2.2.3. Pantalla de Lista de Clientes . . . . .	17
2.2.4. Pantalla de Ficha del producto . . . . .	18
2.2.5. Pantalla de detalles de Factura . . . . .	19
2.2.6. Pantalla de Estadísticas . . . . .	20
<b>3. Desarrollo del calendario</b>	<b>21</b>
3.1. Fases . . . . .	21
3.1.1. Fase 1: Entrevistas e Identificación de Requisitos . . . . .	21
3.1.2. Fase 2: Estudio de la tecnología Android . . . . .	21
3.1.3. Fase 3: Análisis y Especificación de Requisitos . . . . .	21
3.1.4. Fase 4: Análisis y Diseño de la Base de Datos . . . . .	22
3.1.5. Fase 5: Diseño de la Interfaz de Usuario . . . . .	22
3.1.6. Fase 6: Codificación . . . . .	22
3.1.7. Fase 7: Pruebas . . . . .	22
3.1.8. Fase 8: Redacción de la memoria de proyecto . . . . .	22
3.2. Diagrama de Gantt . . . . .	23
<b>4. Descripción general del proyecto</b>	<b>25</b>
4.1. Perspectiva del producto . . . . .	25

## Índice general

4.1.1.	Entorno del producto . . . . .	25
4.1.2.	Interfaz de usuario . . . . .	25
4.2.	Funciones . . . . .	26
4.3.	Características del usuario . . . . .	27
4.4.	Restricciones generales . . . . .	27
4.4.1.	Control de versiones . . . . .	27
4.4.2.	Lenguajes de programación y tecnologías . . . . .	27
4.4.3.	Herramientas . . . . .	28
4.4.4.	Sistemas operativos y hardware . . . . .	29
<b>5.</b>	<b>Desarrollo del proyecto</b>	<b>31</b>
5.1.	Modelo de ciclo de vida . . . . .	31
5.2.	Herramienta de modelado usada: Dia. . . . .	31
5.3.	Requisitos . . . . .	31
5.3.1.	Funcionales . . . . .	31
5.3.2.	Interfaces de Usuario . . . . .	34
5.3.3.	Informes y ficheros generados. . . . .	34
5.3.4.	No funcionales . . . . .	35
5.4.	Análisis del sistema . . . . .	36
5.4.1.	Casos de uso . . . . .	36
5.4.2.	Especificación de casos de uso . . . . .	44
5.4.3.	Modelo conceptual de datos del dominio . . . . .	74
5.4.4.	Diagramas de secuencia. . . . .	76
5.4.5.	Contrato de las operaciones . . . . .	95
5.4.6.	Diseño Conceptual . . . . .	110
5.5.	Diseño del sistema . . . . .	114
5.5.1.	Diseño Lógico . . . . .	116
5.5.2.	Diagrama de clases de diseño . . . . .	124
5.5.3.	Diagramas de Interacción . . . . .	127
5.5.4.	Contrato de las operaciones . . . . .	134
5.5.5.	Diseño de la capa de presentación . . . . .	137
5.6.	Implementación . . . . .	140
5.7.	Pruebas y validación . . . . .	149
5.7.1.	JUnit . . . . .	149
5.7.2.	UI Automator Viewer y uiautomator . . . . .	154
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>159</b>
6.1.	Conclusiones . . . . .	159
6.2.	Valoración personal . . . . .	160
6.3.	Trabajo futuro . . . . .	161

<b>A. Modelo de negocio</b>	<b>163</b>
<b>B. Manual de instalación</b>	<b>167</b>
B.1. Prerrequisitos. . . . .	167
B.2. Instrucciones para la instalación de ikeComercial. . . . .	168
<b>C. Manual de usuario</b>	<b>171</b>
C.1. Guía de uso. . . . .	171
C.1.1. Pantalla de Bienvenida. . . . .	171
C.1.2. Pantalla de Menú Principal. . . . .	172
C.1.3. Pantalla de Gestión de Clientes. . . . .	172
C.1.4. Pantalla de Gestión de Productos. . . . .	174
C.1.5. Pantalla de Gestión de Rutas. . . . .	176
C.1.6. Pantalla de Inicio de Ventas. . . . .	177
C.1.7. Pantalla de Pedidos y Albaranes . . . . .	179
C.1.8. Pantalla de Facturas. . . . .	179
C.1.9. Pantalla de Cobro de Factura. . . . .	181
C.1.10. Pantalla de Generación de PDF. . . . .	182
C.1.11. Pantalla de Búsqueda. . . . .	183
C.1.12. Pantalla de Cobros. . . . .	184
C.1.13. Pantalla de Carga de vehículo. . . . .	185
C.1.14. Pantalla de Exportación. . . . .	186
C.1.15. Pantalla de Estadísticas. . . . .	187
C.1.16. Pantalla de configuración. . . . .	189
C.2. Información de contacto. . . . .	191
<b>D. Estructura de un proyecto Android en ADT Bundle</b>	<b>193</b>
<b>E. Ficheros contables para importación en Sage ContaPlus.</b>	<b>197</b>
<b>F. Licencia Apache 2.0</b>	<b>201</b>
<b>Bibliografía</b>	<b>205</b>



# Índice de figuras

1.1. Logotipo de ikeComercial . . . . .	9
2.1. Invoice2go & Mobile Biz & Más Ventas . . . . .	13
2.2. Tabla comparativa de aplicaciones. . . . .	14
2.3. Comparativa de Pantalla de Bienvenida. . . . .	15
2.4. Menú Principal de ikeComercial. . . . .	16
2.5. Menú Principal de las demás aplicaciones. . . . .	16
2.6. Lista de clientes de ikeComercial. . . . .	17
2.7. Lista de clientes en las demás aplicaciones. . . . .	17
2.8. Ficha del producto en ikeComercial. . . . .	18
2.9. Ficha del producto en las demás aplicaciones. . . . .	18
2.10. Visualización del detalle de una factura en ikeComercial. . . . .	19
2.11. Visualización del detalle de una factura en las demás aplicaciones. . .	19
2.12. Visualización de una estadística de ventas. . . . .	20
2.13. Visualización de una estadística de ventas en Invoice2go y MobileBiz Co. . . . .	20
3.1. Diagrama de Gantt . . . . .	24
5.1. Diagrama de casos de uso inicial. . . . .	36
5.2. Diagrama de casos de uso del menú principal. . . . .	37
5.3. Diagrama de casos de uso de gestión de clientes. . . . .	37
5.4. Diagrama de casos de uso de gestión de productos. . . . .	38
5.5. Diagrama de casos de uso de gestión de categorías de producto. . . .	38
5.6. Diagrama de casos de uso de gestión de rutas. . . . .	39
5.7. Diagrama de casos de uso de gestión de pedidos. . . . .	39
5.8. Diagrama de casos de uso de gestión de albaranes. . . . .	40
5.9. Diagrama de casos de uso de gestión de PDF albaranes. . . . .	40
5.10. Diagrama de casos de uso de gestión de facturas. . . . .	41
5.11. Diagrama de casos de uso de gestión de PDF facturas. . . . .	41
5.12. Diagrama de casos de uso de gestión de carga de vehículo. . . . .	42
5.13. Diagrama de casos de uso de gestión de exportaciones. . . . .	42
5.14. Diagrama de casos de uso de generación de estadísticas. . . . .	43
5.15. Diagrama de casos de uso de configuración. . . . .	43

## Índice de figuras

5.16. Diagrama de casos de uso de gestión de copias de seguridad. . . . .	44
5.17. Modelo conceptual de datos de dominio. . . . .	75
5.18. Diagrama de Secuencia: Registrar Empresa. . . . .	76
5.19. Diagrama de Secuencia: Recuperar Contraseña. . . . .	76
5.20. Diagrama de Secuencia: Acceder al Sistema. . . . .	77
5.21. Diagrama de Secuencia: Dar de alta a un cliente. . . . .	77
5.22. Diagrama de Secuencia: Dar de baja a un cliente. . . . .	77
5.23. Diagrama de Secuencia: Consultar Cliente. . . . .	78
5.24. Diagrama de Secuencia: Buscar Cliente. . . . .	78
5.25. Diagrama de Secuencia: Modificar Cliente. . . . .	78
5.26. Diagrama de Secuencia: Enviar Correo Electrónico. . . . .	79
5.27. Diagrama de Secuencia: Llamar a un cliente. . . . .	79
5.28. Diagrama de Secuencia: Geoposicionar Cliente. . . . .	79
5.29. Diagrama de Secuencia: Dar de alta un producto. . . . .	80
5.30. Diagrama de Secuencia: Dar de baja un producto. . . . .	80
5.31. Diagrama de Secuencia: Listar Productos. . . . .	80
5.32. Diagrama de Secuencia: Modificar Producto. . . . .	81
5.33. Diagrama de Secuencia: Buscar Producto. . . . .	81
5.34. Diagrama de Secuencia: Crear Categoría. . . . .	81
5.35. Diagrama de Secuencia: Borrar Categoría. . . . .	82
5.36. Diagrama de Secuencia: Dar de Alta una Ruta. . . . .	82
5.37. Diagrama de Secuencia: Dar de Baja una Ruta. . . . .	82
5.38. Diagrama de Secuencia: Modificar Ruta. . . . .	83
5.39. Diagrama de Secuencia: Buscar Ruta. . . . .	83
5.40. Diagrama de Secuencia: Añadir Cliente a Ruta. . . . .	83
5.41. Diagrama de Secuencia: Quitar Cliente a Ruta. . . . .	84
5.42. Diagrama de Secuencia: Ver Mapa Clientes. . . . .	84
5.43. Diagrama de Secuencia: Dar de Alta un Pedido. . . . .	84
5.44. Diagrama de Secuencia: Eliminar Pedido. . . . .	85
5.45. Diagrama de Secuencia: Buscar Pedido. . . . .	85
5.46. Diagrama de Secuencia: Añadir Producto a Pedido. . . . .	85
5.47. Diagrama de Secuencia: Quitar Producto del Pedido. . . . .	86
5.48. Diagrama de Secuencia: Modificar Producto del Pedido. . . . .	86
5.49. Diagrama de Secuencia: Listar Pedidos. . . . .	86
5.50. Diagrama de Secuencia: Guardar Pedido. . . . .	87
5.51. Diagrama de Secuencia: Albaranar Pedido. . . . .	87
5.52. Diagrama de Secuencia: Facturar Pedido. . . . .	87
5.53. Diagrama de Secuencia: Dar de Alta un Albarán. . . . .	88
5.54. Diagrama de Secuencia: Dar de Baja un Albarán. . . . .	88
5.55. Diagrama de Secuencia: Facturar Albarán. . . . .	88

5.56. Diagrama de Secuencia: Crear PDF Albarán. . . . .	89
5.57. Diagrama de Secuencia: Enviar PDF Albarán. . . . .	89
5.58. Diagrama de Secuencia: Visualizar PDF Albarán. . . . .	89
5.59. Diagrama de Secuencia: Dar de Alta una Factura. . . . .	90
5.60. Diagrama de Secuencia: Cobrar Factura. . . . .	90
5.61. Diagrama de Secuencia: Añadir Producto a Vehículo. . . . .	90
5.62. Diagrama de Secuencia: Quitar Producto del Vehículo. . . . .	91
5.63. Diagrama de Secuencia: Modificar Stock Producto. . . . .	91
5.64. Diagrama de Secuencia: Crear Ficheros de Exportación. . . . .	91
5.65. Diagrama de Secuencia: Eliminar Fichero de Exportación. . . . .	92
5.66. Diagrama de Secuencia: Enviar Fichero Exportación. . . . .	92
5.67. Diagrama de Secuencia: Generar Estadística. . . . .	92
5.68. Diagrama de Secuencia: Aviso de Riesgo. . . . .	93
5.69. Diagrama de Secuencia: Controlar Stock de Vehículo. . . . .	93
5.70. Diagrama de Secuencia: Modificar Longitud de Subcuenta. . . . .	93
5.71. Diagrama de Secuencia: Realizar Copia de Seguridad. . . . .	94
5.72. Diagrama de Secuencia: Selecciona Copia de Seguridad. . . . .	94
5.73. Diagrama de Secuencia: Restaurar Copia de Seguridad. . . . .	94
5.74. Diagrama Entidad-Relación. . . . .	113
5.75. Modelo Vista Controlador . . . . .	114
5.76. Diagrama de Clases de Diseño. . . . .	125
5.77. Diagrama de Interacción: Acceso al Sistema. . . . .	127
5.78. Diagrama de Interacción: Opciones Menú Principal. . . . .	128
5.79. Diagrama de Interacción: Alta Cliente. . . . .	129
5.80. Diagrama de Interacción: Baja Cliente. . . . .	129
5.81. Diagrama de Interacción: Consultar Cliente. . . . .	130
5.82. Diagrama de Interacción: Modificar Cliente. . . . .	130
5.83. Diagrama de Interacción: Añadir Cliente a Ruta. . . . .	131
5.84. Diagrama de Interacción: Listar Pedidos. . . . .	131
5.85. Diagrama de Interacción: Añadir Producto a la venta. . . . .	132
5.86. Diagrama de Interacción: Facturar Venta. . . . .	133
5.87. Pantalla de Bienvenida. . . . .	137
5.88. Pantalla del Menú Principal. . . . .	138
5.89. Mensajes de Información. . . . .	138
5.90. Pantallas de confirmación. . . . .	139
5.91. Pantallas de carga. . . . .	139
5.92. Acerca de ADT Bundle. . . . .	140
5.93. Descarga oficial de ADT Bundle. . . . .	141
5.94. Arranque y visualización de Genymotion. . . . .	148
5.95. Detalle de la factura de prueba. . . . .	153

## Índice de figuras

5.96. UI Automator Viewer . . . . .	154
A.1. GitHub . . . . .	165
B.1. ikeComercial en Google Play. . . . .	168
B.2. Confirmación de la instalación aceptando los permisos. . . . .	169
B.3. Instalando ikeComercial. . . . .	169
C.1. Primeras pantallas de visualización. . . . .	171
C.2. Visualización del menú principal. . . . .	172
C.3. Visualización de la lista de clientes y sus opciones. . . . .	173
C.4. Vistas de la ficha de clientes. . . . .	173
C.5. Visualizaciones del diálogo de llamada y del mapa de cliente. . . . .	174
C.6. Visualización de la lista de productos. . . . .	175
C.7. Visualización del detalle de producto. . . . .	175
C.8. Visualización de la lista de rutas y el detalle de una de ellas. . . . .	176
C.9. Visualización de opciones de ruta y mapas. . . . .	177
C.10. Pantalla de Inicio de Ventas. . . . .	178
C.11. Añadiendo productos a la venta. . . . .	178
C.12. Visualización de listas de pedidos y albaranes. . . . .	179
C.13. Visualización de listas de facturas pendientes y cobradas. . . . .	180
C.14. Visualización del detalle de una factura. . . . .	180
C.15. Visualización de la pantalla de cobro de factura. . . . .	181
C.16. Visualización de creación de PDF de factura. . . . .	182
C.17. Visualización de las pantallas de búsqueda. . . . .	183
C.18. Pantallas de Gestión de Cobros. . . . .	184
C.19. Pantallas de carga del vehículo. . . . .	185
C.20. Pantallas de exportación. . . . .	186
C.21. Pantallas de estadísticas . . . . .	187
C.22. Ejemplo de visualización de estadísticas. . . . .	188
C.23. Pantallas de configuración. . . . .	190
C.24. Pantalla acerca de. . . . .	190
D.1. Estructura de un proyecto Android. . . . .	194
E.1. Importando/Exportando con Sage ContaPlus. . . . .	199

# Índice de tablas

5.2. Descripción de las entidades del diseño conceptual . . . . .	111
5.4. Tabla Normalizada: Cliente. . . . .	118
5.6. Tabla Normalizada: ClienteCP. . . . .	118
5.8. Tabla Normalizada: ClienteGeo. . . . .	119
5.10. Tabla Normalizada: Teléfonos. . . . .	119
5.12. Tabla Normalizada: Ruta. . . . .	119
5.14. Tabla Normalizada: Pedidos. . . . .	120
5.16. Tabla Normalizada: Albarán. . . . .	120
5.18. Tabla Normalizada: Factura. . . . .	120
5.20. Tabla Normalizada: LíneaDocumento. . . . .	121
5.22. Tabla Normalizada: Productos. . . . .	121
5.24. Tabla Normalizada: Impuestos. . . . .	122
5.26. Tabla Normalizada: Categoría. . . . .	122
5.28. Tabla Normalizada: AsientoContable. . . . .	122
5.30. Tabla Normalizada: CobroFactura. . . . .	123
5.32. Tabla Normalizada: HistorialImpuestos. . . . .	123
E.2. Estructura de fichero Diario. . . . .	198
E.4. Estructura de fichero Subcuenta. . . . .	199

iii

## *Índice de tablas*

# 1. Introducción

En el mundo empresarial, son muy importantes las relaciones que se establecen entre las empresas y los clientes que se asocian a ellas. Esta relación tiene que cultivarse desde el punto de vista del esfuerzo y de la dedicación, con el objetivo de satisfacer y cubrir las necesidades que todo cliente tiene de una manera sencilla y eficaz. Uno de los objetivos que se persiguen es el de fortalecer y mejorar ese vínculo.

Los encargados de hacer el trabajo de seguimiento y atención al cliente en una empresa son los denominados comerciales. Generalmente, los comerciales llevan a cabo su labor realizando visitas a las empresas de sus clientes (una vez por semana, normalmente) para comprobar y poder hacer llegar al cliente los productos que necesitan, sin que éste tenga que trasladarse, para su mayor comodidad. Dado los avances en el mundo de la telefonía móvil, sobre todo con la aparición de los llamados smartphones (teléfonos inteligentes)[19] ó tablets[21] (tabletas móviles), este PFC (Proyecto Fin de Carrera) tiene como objetivo principal el desarrollo de una aplicación móvil que facilite y permita llevar a cabo a los comerciales la gestión de ventas de sus productos de una manera sencilla y cómoda. En concreto, este PFC consiste fundamentalmente en especificar, diseñar, implementar y realizar casos de prueba de una aplicación llamada *ikeComercial*, una aplicación de gestión de ventas que permitirá a los comerciales llevar a cabo su tarea desde un terminal móvil bajo sistema operativo Android[2].

## 1.1. Motivación

MediaStudio es una empresa situada en Chiclana que se dedica al desarrollo de *software* y proporciona soluciones informáticas a, mayormente, empresas locales. En ella podemos encontrar *software* para la gestión de hoteles, TPV (Terminal Punto de Venta) para comercios de hostelería ó *software* de gestión de ventas de comercialización de productos alimenticios. En concreto, esta empresa proporciona *software* para dispositivos PDA (Personal Digital Assistant)[15] para que los comerciales puedan realizar la venta de los productos que provee a sus clientes que visitan periódicamente.

En este PFC se han considerado los smartphones como dispositivos capaces no sólo de desarrollar la labor de una PDA, sino también la de aprovechar la capacidad que hoy en día nos ofrecen para evolucionar el proceso de desarrollo de ventas de los

## 1. Introducción

comerciales hacia una experiencia más agradable, más fácil y más sencilla. Debido a que la plataforma Android[2] se ha consolidado como uno de los referentes en el mercado móvil [33], en este PFC se desarrolla una aplicación Android para dicha empresa que permite almacenar toda la información que el comercial requiere, de forma que puede acceder a la información de todos los productos que provee y almacenar los pedidos que los clientes soliciten. Una vez recopilados todos los pedidos, el comercial descarga esta información al *software* central de ventas de la empresa para su posterior gestión y distribución.

### 1.2. Objetivos y Alcance

El objetivo principal de este proyecto es la creación de una aplicación móvil para Android que permita a los comerciales ejercer su práctica empresarial, en el contexto de movilidad aquí descrito, obteniendo el control total de sus ventas a los clientes que provee.

Para llevar a cabo este objetivo, se proponen los siguientes sub-objetivos:

- La aplicación permitirá gestionar tanto los clientes que la empresa tiene como los productos que suministra.
- Asimismo, permitirá administrar los pedidos del comercial, ya sean en calidad de albarán, factura ó pedido pendiente.
- Otro punto importante serán las gestiones de cobros de las facturas de los clientes. La aplicación permitirá clasificar las facturas pagadas como las facturas pendientes de cobro.
- También permitirá planificar semanalmente las visitas ó rutas que el comercial debe trazar para llegar a sus clientes.
- Además se añadirá un sistema de aviso de cobros para los clientes, así como un sistema de control de stock de los productos que el comercial transporte en su vehículo (el comercial podrá suministrar al momento los productos que lleva en el vehículo ó si no los lleva, podrá guardarlo como pedido para realizar su suministro en su próxima visita).
- La aplicación también tendrá la opción de generar gráficas estadísticas de las ventas realizadas.
- Por último, daremos la opción de exportar los asientos contables (anotaciones realizadas con el fin de reflejar una operación contable) de las facturas emitidas y acoplarlas a un software externo de gestión contable: Sage Contaplus[16].

El alcance de este PFC abarcará la creación de este sistema llamado *ikeComercial* que satisfaga todos los objetivos aquí descritos. Además, una vez obtenida la licencia de desarrollador de Android, se trabajará para su posterior publicación en *Google Play* [6], mercado de aplicaciones de Android. Para más información, se ha incluido en esta memoria el futuro modelo de negocio elegido para *ikeComercial A*.

### 1.3. IkeComercial

Ante la necesidad de crear un sistema que permita hacer a los comerciales en movimiento su trabajo de gestión de ventas, nace *IkeComercial*, una aplicación para Android con el objetivo principal de facilitar el trabajo de gestión de ventas a clientes de cara en su día a día laboral de una manera cómoda y sencilla.



Figura 1.1.: Logotipo de *ikeComercial*

### 1.4. Visión general

Empezamos con una breve introducción al PFC como primer capítulo, en el que se relatan, entre otros aspectos, los objetivos, alcance y motivación que nos ha llevado a realizarlo. A continuación se detalla el estado del arte con respecto a otras aplicaciones del mercado (véase capítulo 2), se describen, posteriormente, las fases que han constituido el proyecto (véase capítulo 3 y una descripción general de *ikeComercial*).

## 1. Introducción

A lo largo del resto de la memoria se detalla el proceso de análisis, diseño, codificación y pruebas que se ha seguido al realizar el proyecto.

Los manuales de instalación, de usuario y del desarrollador se incluyen tras un resumen de los aspectos más destacables del proyecto y las conclusiones.

## 1.5. Glosario

### 1.5.1. Acrónimos

**PFC** Proyecto Fin de Carrera

**UCA** Universidad de Cádiz

**ER** Entidad Relación

**SDK** Software Development Kit

**IVA** Impuesto Válido Adicional

**RE** Recuento de Equivalencia

**MVC** Modelo Vista Controlador

**TPV** Terminal Punto de Venta

**PDA** Personal Digital Assistant

## 1.5.2. Definiciones

**Integrated Development Environment(IDE)** Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (-de uno o varios lenguajes de programación), un compilador, un depurador y una interfaz gráfica (GUI).

**Sage Contaplus** Programa de gestión contable de la empresa Sage.

**Asiento contable** Anotaciones o registros que se hacen en el Libro diario de contabilidad, que se realizan con la finalidad de registrar un hecho económico que provoca una modificación cuantitativa o cualitativa en la composición del patrimonio de una empresa y por tanto un movimiento en las cuentas de una empresa.

**Android** Es un sistema operativo móvil basado en Linux, enfocado para ser implantado en dispositivos móviles como, teléfonos inteligentes o tabletas. Perteneció a Google Inc.

**Google Play** Mercado de aplicaciones Android.

**Modelo de datos relacional.** Modelo que organiza y representa los datos en forma de tablas o relaciones.

**SQLite** Sistema de gestión de bases de datos relacional multiplataforma escrita en C, compatible con CRUD, y de dominio público.

**CRUD** En el contexto de los casos de uso, emplearemos «Gestionar X» para unificar los 5 casos de uso básicos sobre la clase de objetos X: Crear X, Modificar X, Listar X, Buscar X, Borrar X.

**PDF** Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto, en los que se puede incluir imagen vectorial, mapa de bits y texto, del que se disponen lectores para todas las plataformas.

**UML** Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

**OOP** Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos.

## 1. Introducción

**Modelo de datos relacional** Diagrama que utiliza un grupo de tablas para representar los datos y las relaciones entre ellos.

**Google Nexus 4** Dispositivo móvil diseñado por Google Inc.

**Mockups** Prototipo a escala o tamaño real de un diseño o un dispositivo, utilizado para la demostración, evaluación del diseño, promoción, y para otros fines.

**SDK** Un kit de desarrollo de software es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.

**Tablet** Una tablet es una computadora portátil con la que se puede interactuar a través de una pantalla táctil. El usuario puede utilizar una pluma o los dedos para trabajar con el ordenador sin necesidad de teclado físico, o ratón.

**Java** Lenguaje de programación creada por la empresa Sun Microsystems.

**XML** Lenguaje estándar de marcado con múltiples aplicaciones para la descripción y/o definición de documentos estructurados.

**Software** Parte lógica de un sistema informático.

**Hardware** Parte física de un sistema informático.

**Activity** Es un componente de desarrollo en Android que provee pantallas que pueden interactuar con los usuarios para realizar una acción determinada.

**Launcher** Es la parte encargada de la interfaz de usuario de Android. Totalmente personalizable.

## 2. Estado del arte

En este capítulo, nos vamos a centrar en destacar las funcionalidades de ike-Comercial frente a otras alternativas que nos ofrece el mercado. En concreto, tres aplicaciones de facturación y ventas en sistemas Android: *Invoice2go*[9], *MobileBiz Co*[13] y *Más Ventas* [12].

### 2.1. Comparativa de funcionalidades

La primera de nuestras opciones a la hora de buscar una aplicación de facturación en Google Play[6], es *Invoice2go*[9]. Con *Invoice2go* podemos hacer presupuestos y facturas sencillas pudiendo elegir entre diferentes plantillas de facturas, llevar al día todas aquellas facturas que has entregado, cobrado o anulado, etc. Una vez que tengas tu factura la puedes enviar por email o visualizarla en el dispositivo. Actualmente se encuentra en la versión 7.1.8 y con más de 10000 descargas. Existen una aplicación gratuita (limitada) y otra de pago (9,99 €).

En segundo lugar hemos analizado como segunda opción a *MobileBiz Co*[13], una aplicación de presupuestos y facturas en la que destacamos la funcionalidad de sincronizar con la nube y poder tener nuestros documentos disponibles cuando los necesitamos. Se encuentra en la versión 1.19.14, posee más de 5000 descargas y sólo está disponible en inglés.

Por último, hemos evaluado una aplicación llamada *Más Ventas*[12]. Esta aplicación está sujeta a contratación por suscripción pero, en ella podemos ver, funcionalidades de reparto de pedidos y entrega de mercancías.



Figura 2.1.: Invoice2go & Mobile Biz & Más Ventas

## 2. Estado del arte

En la Tabla 2.2 se muestra la comparativa que hemos realizado atendiendo a las funcionalidades proporcionadas por dichas aplicaciones.

				
	ikeComercial	Invoice2go	MobileBiz Co	Más Ventas
Pedidos, Albaranes Facturas	Sí	Sí	Sí	Sí
Email, llamada y foto de Cliente	Sí	No	No	Sí
Foto y Geoposicionamiento de Cliente	Sí	No	No	No
Descuentos y devoluciones de producto.	Sí	Sí	Sí	Sí
Control de Stock	Sí	Sí	Sí	No
Aviso de Riesgo	Sí	No	Sí	Sí
Control de Rutas y Geoposicionamiento	Sí	No	No	Sí
Control de Carga de Vehículo	Sí	No	No	No
Creación e impresión de documentos PDF.	Sí	Sí	Sí	Sí
Copias de Seguridad	Sí	Sí	Sí	No
Liquidación de productos.	No	No	Sí	Sí
Estadísticas	Sí	Sí	Sí	No
Exportación a Sage Contaplus	Sí	No	No	No

Figura 2.2.: Tabla comparativa de aplicaciones.

## 2.2. Comparativa de interfaces de usuario

Ahora vamos a tratar la interfaz de usuario de las distintas aplicaciones aquí descritas para contrastar las principales pantallas de cada una de ellas. Estas comparaciones estarán basadas en cómo se ilustrarían en una maqueta de un dispositivo real ampliamente conocido en la actualidad, el Google Nexus 4 (4.7 pulgadas)[5]. Dado el tamaño de este terminal, las dimensiones que se usarán en esta memoria serán más reducidas que las originales.

### 2.2.1. Pantalla de Bienvenida.

Comenzaremos con la pantalla de bienvenida que es aquella que vemos por primera vez cuando arrancamos la aplicación en cuestión. Se ha optado por una pantalla sencilla en la que podemos destacar el logo de la aplicación y, a continuación, el formulario de acceso a ésta. Como no nos hemos registrado todavía y es la primera vez que arrancamos la aplicación, deberemos de registrarnos previamente (véase Figura 2.3).



Figura 2.3.: Comparativa de Pantalla de Bienvenida.

## 2. Estado del arte

### 2.2.2. Pantalla de Menú Principal

En este apartado nos vamos a centrar en la visualización del menú principal en las diferentes aplicaciones. Se ha apostado por distribuir en 3 pantallas el menú principal de la aplicación ikeComercial que pueden ser fácilmente visualizados con el deslizamiento de izquierda a derecha o viceversa de nuestro dedo por la pantalla con el objetivo de mejorar la navegación para el usuario (véase Figura 2.4).



Figura 2.4.: Menú Principal de ikeComercial.

Ahora vamos a visualizar cómo se vería el menú principal en las demás aplicaciones que estamos tratando en este documento (véase Figura 2.5).

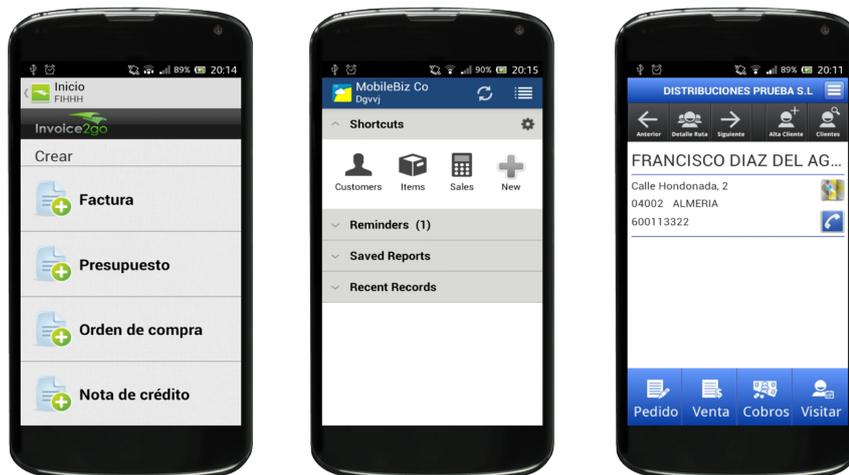


Figura 2.5.: Menú Principal de las demás aplicaciones.

### 2.2.3. Pantalla de Lista de Clientes

En este apartado nos vamos centrar en la visualización de la lista de clientes en las diferentes aplicaciones. En nuestra aplicación se ha optado por una lista sencilla indexada por orden alfabético en la cuál podemos deslizarla verticalmente por ella. Se ha incluido la opción de buscar en la barra de acciones para encontrar más eficazmente al usuario en cuestión, si se desea (véase Figura 2.6).

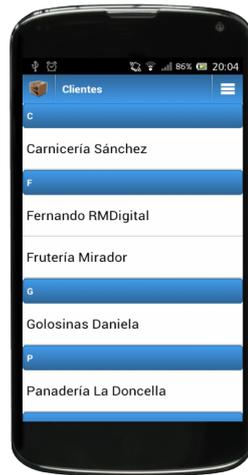


Figura 2.6.: Lista de clientes de ikeComercial.

Ahora vamos a visualizar cómo se vería la lista de clientes en las demás aplicaciones que estamos tratando en este documento (véase Figura 2.7).

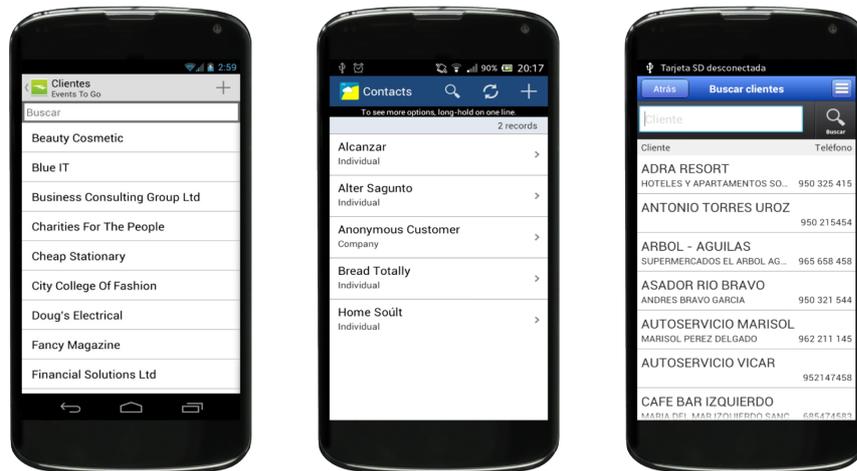


Figura 2.7.: Lista de clientes en las demás aplicaciones.

## 2. Estado del arte

### 2.2.4. Pantalla de Ficha del producto

En este apartado nos vamos centrar en la visualización de la ficha del producto en las diferentes aplicaciones. Se ha apostado por una lista en la que podemos deslizarlos verticalmente para ver toda la información acerca de éste como puede ser el precio o la categoría en la que está incluida. En la parte superior destacamos la imagen de dicho producto (véase Figura 2.8).



Figura 2.8.: Ficha del producto en ikeComercial.

Ahora vamos a visualizar cómo se vería la ficha del producto en las demás aplicaciones que estamos tratando en este documento (véase Figura 2.9).

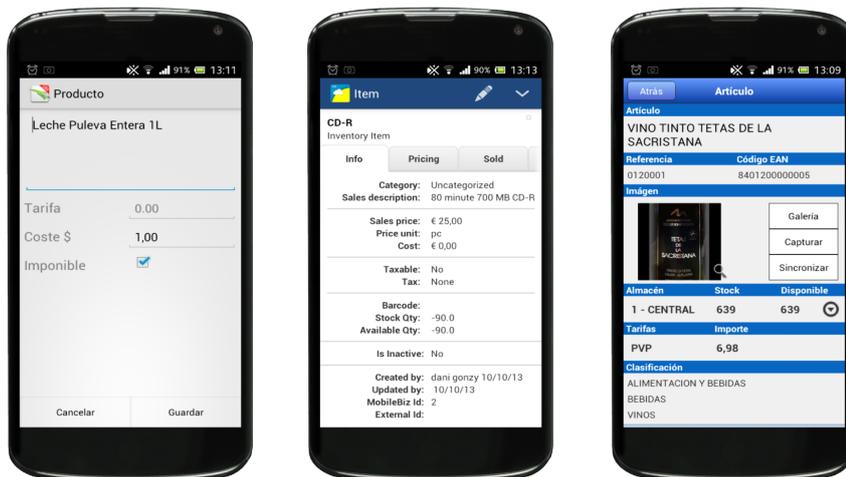


Figura 2.9.: Ficha del producto en las demás aplicaciones.

### 2.2.5. Pantalla de detalles de Factura

Como podemos ver, se ha optado por una distribución de 3 pantallas. En la primera de ellas veremos información general acerca de la factura. En la segunda pantalla vemos los artículos que se citan en la factura y en la tercera pantalla vemos una tabla que indica los impuestos incluidos así como el cálculo del total de la factura (véase Figura 2.10).



Figura 2.10.: Visualización del detalle de una factura en ikeComercial.

Ahora vamos a visualizar el detalle de la factura en las demás aplicaciones (véase Figura 2.11).

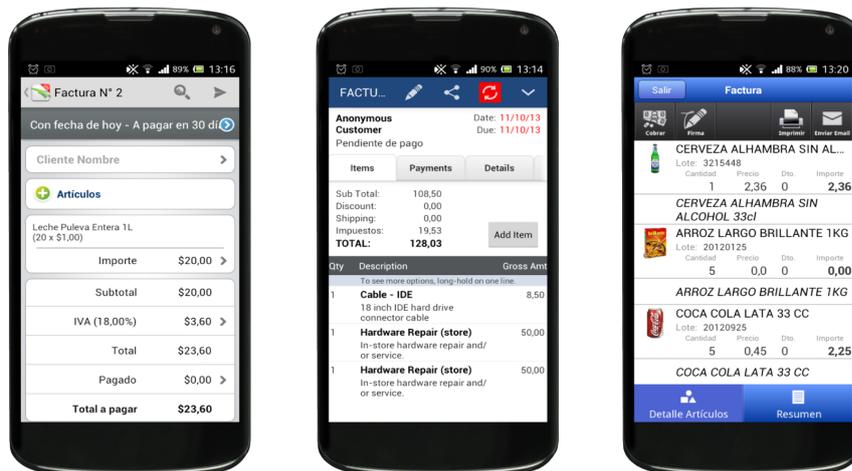


Figura 2.11.: Visualización del detalle de una factura en las demás aplicaciones.

## 2. Estado del arte

### 2.2.6. Pantalla de Estadísticas

En este apartado nos vamos a centrar en la visualización de las estadísticas de ventas en las diferentes aplicaciones. Para la visualización de la estadística de ventas (ya sea por día, mes o por año) se ha optado por un gráfico de barras que ocupa toda la pantalla con la capacidad de poder interactuar con él, pudiéndonos acercar y alejar para apreciar mejor los detalles (véase Figura 2.12).

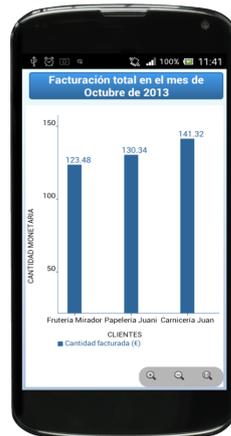


Figura 2.12.: Visualización de una estadística de ventas.

Como la aplicación Más Ventas no posee ningún tipo de estadísticas gráficas, mostraremos las que sí lo tienen (véase Figura 2.13).

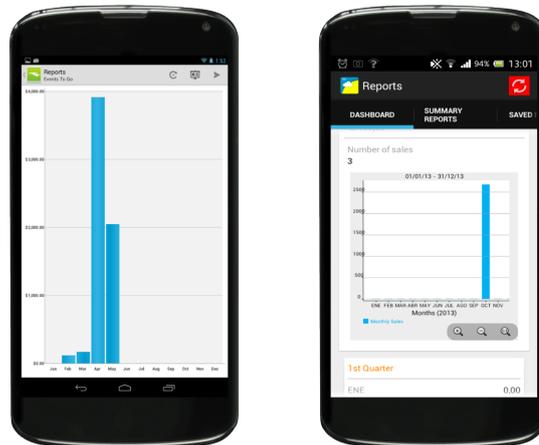


Figura 2.13.: Visualización de una estadística de ventas en Invoice2go y MobileBiz Co.

## 3. Desarrollo del calendario

En este capítulo se van a describir las fases que se han llevado a cabo en la realización de este proyecto fin de carrera. Algunas de ellas se pueden solapar en el tiempo, incluso, en otras se han tenido que realizar en repetidas ocasiones, por lo tanto, no es necesario la finalización de una fase para el comienzo de la siguiente.

### 3.1. Fases

#### 3.1.1. Fase 1: Entrevistas e Identificación de Requisitos

Esta es la fase inicial del proyecto en la que se toma contacto con el jefe de la empresa para poder realizar la lista de posibles requisitos de este PFC; tener una idea globalizada de qué es lo que se pretende conseguir con la aplicación.

Se estudian las conclusiones extraídas de las entrevistas, y se comparan con algunas de las posibles soluciones existentes en el mercado.

#### 3.1.2. Fase 2: Estudio de la tecnología Android

En esta fase se procedió a conocer la tecnología Android[26] compuesta fundamentalmente del lenguaje Java y XML[11], además de la plataforma SQLite[22], las cuales eran prácticamente desconocidas por el alumno. En un primer lugar, se buscó bibliografía acerca de esta plataforma tanto en libros como en Internet, recopilando información de todo tipo [27].

Posteriormente se realizaron tutoriales para su mejor comprensión así como ejemplos prácticos basados en ellos para afianzar los conocimientos que se iban aprendiendo durante esta etapa[34, 35].

#### 3.1.3. Fase 3: Análisis y Especificación de Requisitos

Una vez que se llegó a un nivel aceptable de solvencia con la tecnología, se procedió a elicitar los requisitos funcionales que la aplicación debe cumplir además de la información a mostrar y del comportamiento del sistema. En particular, esta fase es muy importante para el posterior desarrollo de la base de datos y de los posibles diseños de interfaz de usuario.

### 3. Desarrollo del calendario

#### 3.1.4. Fase 4: Análisis y Diseño de la Base de Datos

En esta fase se realizó un análisis exhaustivo y detallado de cuál es la información que necesitamos almacenar para el uso de nuestra aplicación. Hay que destacar que esta, desde mi punto de vista, es la fase más importante del proyecto ya que, si no se hace correctamente, es posible que la aplicación nos de muchos problemas durante el transcurso de las siguientes fases.

Para ello, se realizó el diagrama ER (Entidad Relación) para representar gráficamente las entidades relevantes y sus relaciones en este sistema así como toda la información que se almacena. A continuación, se realizó la transformación de este modelo de datos al modelo de datos relacional para su posterior normalización en base a las formas normales de Edgar F. Codd.

#### 3.1.5. Fase 5: Diseño de la Interfaz de Usuario

En este punto, se diseña las diferentes pantallas de la aplicación así como su interacción entre ellas. Para esta finalidad, se hacen mockups de la aplicación, con lápiz y papel, describiendo su orden de visualización y contenido.

#### 3.1.6. Fase 6: Codificación

Aquí haremos la codificación del sistema para que se comporte como se ha especificado en la etapa de diseño, como pueden ser clases, métodos ó incluso, librerías externas utilizadas. Incluimos en esta fase la implementación de la base de datos basada en *SGBD SQLite*[22]. En esta fase también se diseña la interfaz de usuario como también el logo de la aplicación con *GIMP*[32] e *Inkscape*[8].

#### 3.1.7. Fase 7: Pruebas

Pasaremos una serie de pruebas en la que probaremos todas las funcionalidades de nuestra aplicación y anotaremos los resultados. Utilizaremos las herramientas que el SDK (Software Development Kit) de Android nos facilita para este objetivo. Algunas de ellas son *UI Automator Viewer* ó *uiautomator*[29].

#### 3.1.8. Fase 8: Redacción de la memoria de proyecto

La redacción de la memoria del PFC comenzó desde la fase del análisis. A lo largo del desarrollo del proceso de *software* se ha ido completando la memoria hasta su finalización a finales de diciembre de 2013.

## 3.2. Diagrama de Gantt

Se ha elaborado un diagrama Gantt para poder visualizar con mayor facilidad la distribución de las tareas. Se ha empleado la herramienta Gantt Project para dibujar el diagrama que vemos a continuación.

Puede descargarse aquí:<http://ganttproject.sourceforge.net>

### 3. Desarrollo del calendario

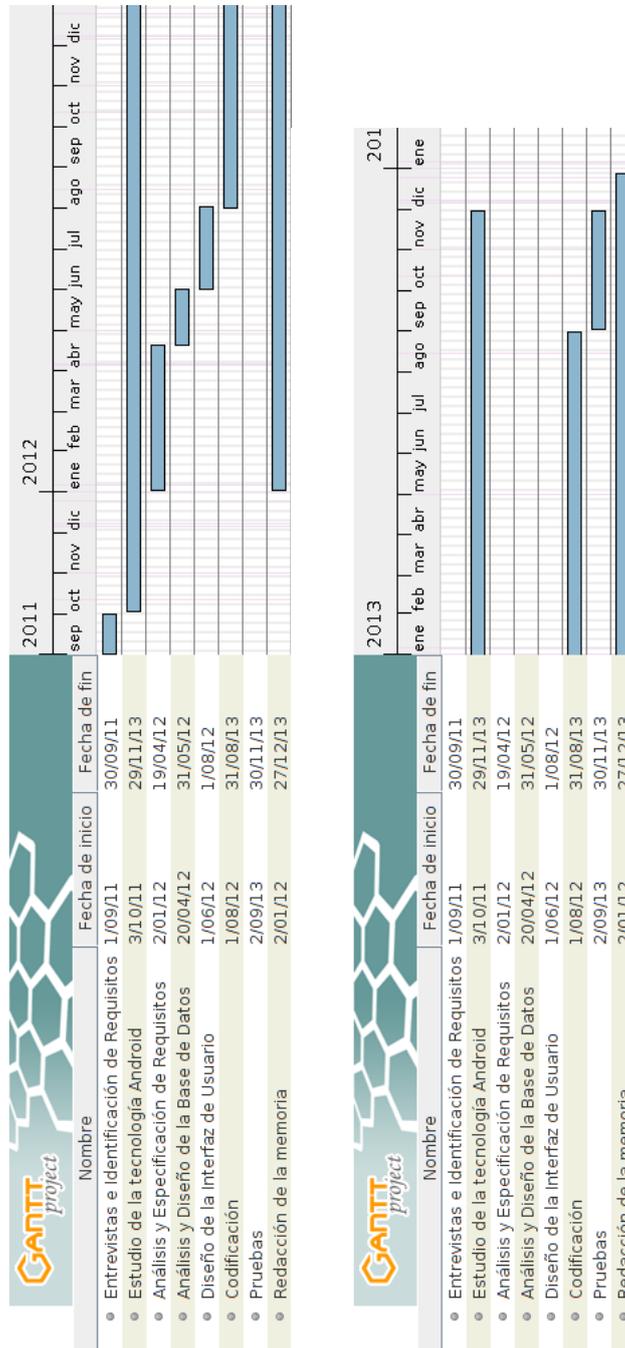


Figura 3.1.: Diagrama de Gantt

# 4. Descripción general del proyecto

## 4.1. Perspectiva del producto

### 4.1.1. Entorno del producto

La aplicación ikeComercial está pensada para aquellos comerciales que tienen que moverse hacia sus clientes facilitando su tarea de gestión de ventas. Se pretende publicar la aplicación en Google Play[6] así como producto de la empresa *MediaStudio* para su comercialización y distribución.

La aplicación puede ser instalada en cualquier dispositivo con sistema Android[2], aunque no está optimizada para todos los tamaños de pantalla que existen actualmente en el mercado. Inicialmente se pensó partir de la base de un dispositivo móvil con pantalla de 4" a 4.7" pulgadas aproximadamente, por lo tanto, a partir de este tamaño es posible que no aproveche completamente la pantalla o los elementos de visualización no estén de la mejor manera ordenados. Esto no impedirá, como quiero recalcar, que se puedan instalar y usar, sino que más bien, no se adapten en tamaños de pantalla superiores a la medida indicada, como pueden ser los *tablets*[21]. Esto se ofrecerá como una mejora futura.

Por supuesto, el sistema estará pendiente a evolucionar conforme se vayan recibiendo comentarios con deseadas mejoras, funcionalidades o errores que puedan producirse en el manejo de la aplicación.

### 4.1.2. Interfaz de usuario

Como característica significativa, podemos comunicarnos con los *smartphones*[19] por medio de lo que se llama el multitouch. Si a esta nueva forma de interactuar le añadimos un buen diseño de interfaz gráfica, tenemos todo para conseguir que nuestros usuarios se encuentren cómodos utilizando nuestra aplicación.

En todo el proceso de diseño de la interfaz de usuario se ha perseguido un mismo objetivo, la sencillez. Esto implica, entre otras cosas, no sobrecargar con demasiada información cada una de las vistas, dar un tamaño adecuado a los componentes (como pueden ser botones o cajas de texto) que forman una vista ó diseñar diálogos interactivos para una mejor experiencia de usuario. Además, se han elegido básicamente dos colores para toda la aplicación, el azul y el blanco (y algunas tonalidades de éstas).

#### 4. Descripción general del proyecto

Un objetivo que también se ha perseguido es el de poder movernos por la aplicación fácilmente e ir desde un punto hacia otro lo más rápido posible. Así pues se han diseñado menús desplegados en cada pantalla, donde aparecen todas las acciones que puedan realizarse en ella, así como, una barra de acciones con un icono que representa el logotipo de ikeComercial que nos va a llevar siempre, estemos donde estemos, al menú principal de la aplicación.

## 4.2. Funciones

Las funciones de ikeComercial son las siguientes:

- Gestión de clientes y de productos.
- Captura fotográfica de imágenes en las fichas de clientes y productos.
- Planificación de Visitas a clientes.
- Capacidad de realización de pedidos, albaranes y facturas.
- Agrupación de artículos por categorías.
- Descuentos en artículos.
- Geolocalización de clientes en mapa.
- Carga y Control de Stock del Vehículo.
- Avisos pendientes de albaranes o cobro de facturas.
- Creación de documentos PDF (albaranes y facturas).
- Opción de incluir la imagen de la empresa en la factura.
- Envío por email de documentos en PDF (albaranes y facturas).
- Exportación a Sage Contaplus.
- Copias de Seguridad de la base de datos.
- Estadísticas de ventas.

## 4.3. Características del usuario

El perfil de usuario en el que se ha basado para la realización de este proyecto son los comerciales de empresas. Estas personas son las que se dedican a la comercialización de los productos mediante relaciones directas con los clientes, es decir, son los que ofrecen los productos al cliente en persona.

## 4.4. Restricciones generales

### 4.4.1. Control de versiones

Para el control de las distintas versiones de la aplicación ikeComercial que iban surgiendo, he utilizado el servicio de alojamientos online *Skydrive* y también *Ubuntu One*.

En el comienzo, cuando se estaba aún en la investigación de la tecnología y se realizaban tutoriales en Android[35, 27, 34], se utilizaba *Skydrive*[18] para alojar las copias de seguridad de los primeros proyectos que iba creando. Era un lugar seguro de tener a mano todos los proyectos iniciales que se realizaban para consultar el código u otras cuestiones. A medida que el tiempo y los conocimientos fueron avanzando, *Skydrive* se convirtió en una herramienta insuficiente una vez que se empezaba a materializar las primeras líneas de código de ikeComercial con el objetivo de tener un control de versiones de seguridad en caso de supuesta pérdida ó volver a una versión anterior en algún momento. A partir de ese momento se empezó a usar *Ubuntu One*[23].

*Ubuntu One*[23], incorporado en las últimas versiones de Ubuntu (sistema operativo GNU/Linux) es un servicio de alojamiento gratuito que te permite sincronizar muy fácilmente tus ficheros o carpetas como copias de seguridad ya sea en un dispositivo o en varios conectados a Internet. Se realizaban copias de seguridad periódicas, normalmente en períodos de una semana y han sido de mucha utilidad.

### 4.4.2. Lenguajes de programación y tecnologías

El sistema está desarrollado fundamentalmente en dos lenguajes: *Java* y XML[11]. El lenguaje XML se utiliza para crear las vistas, es decir, la interfaz de diseño, la cual es muy sencilla de usar. Podemos utilizar el código directamente para crear nuestras interfaces ó también tenemos la posibilidad de crearlas mediante un visualizador de pantallas interactivo en el cual podremos añadir nuestras vistas utilizando un menú lateral de iconos que de todas las componentes que podamos añadir (sólo los componentes oficiales de Android) haciendo *drag & drop* (pica y arrastra). Utilizaremos Java cuando queramos crear clases ó métodos y así poder interactuar con el diseño

#### 4. Descripción general del proyecto

aplicándole lo que queremos que se haga cuando, por ejemplo, toquemos un botón o aparezca una animación cuando se inicie una determinada pantalla.

Estos lenguajes se han desarrollado mediante el ADT Bundle[1], el IDE creado por los ingenieros de Google basado en Eclipse.

##### 4.4.3. Herramientas

Las herramientas que se han utilizado para el desarrollo de este PFC son:

- **ADT Bundle:** un IDE basado en Eclipse que contiene todo lo necesario para empezar a desarrollar en Android.
- **Skydrive:** es un servicio de alojamiento de archivos online de Microsoft.
- **Ubuntu One:** es un servicio de alojamiento de archivos de Canonical integrado con el entorno de escritorio del sistema operativo Ubuntu, enfocado a la copia de seguridad y a la sincronización de archivos e información entre computadores conectados a Internet.
- **UI Automator Viewer:** herramienta que nos permite identificar los elementos de vista de una pantalla en Android.
- **uiautomator:** es un framework de pruebas que te permite crear casos de pruebas automáticas de la interfaz gráfica capaces de correr en tu teléfono o emulador Android.
- **Ganttproject:** un programa gratuito para la administración de proyectos utilizando diagramas de Gantt.
- **Genymotion:** es una versión x86 del sistema operativo Android, anteriormente llamado AndroVM, ejecutado en VirtualBox, lo cual le permite ejecutarse de manera mucho más rápida que un emulador.
- **Inkscape:** editor de gráficos en formato vectoriales SVG, gratuito, libre y multiplataforma.
- **Lyx:** programa gráfico multiplataforma creado por Matthias Ettrich que permite la edición de texto usando  $\text{\LaTeX}$ .
- **Dia:** programa gratuito para la creación de diagramas.
- **IconFinder:** una excelente página de búsqueda de iconos, con filtrado por licencia.

#### 4.4.4. Sistemas operativos y hardware

Esta aplicación esta diseñada para funcionar en sistemas Android con versión 2.2 o superior. Hemos instalado, por medio del emulador incorporado en el SDK de Android, la aplicación en las siguientes versiones de Android:

- *Android Froyo (2.2.x)*
- *Android GingerBread (2.3.x)*
- *Android HoneyComb (3.x)*
- *Android Ice Cream Sandwich (4.0.x)*
- *Android Jelly Bean (4.1, 4.2, 4.3)*

Además del emulador, las pruebas también se han realizado en un dispositivo Android con *Ice Cream Sandwich versión 4.0.4*. En concreto estamos hablando de un *Sony Xperia Arc S[20]* (un móvil de gama media con procesador *Qualcomm MSM8255T Snapdragon* a 1.4GHzm y una pantalla de 4.2”).

Con lo que se refiere a las conexiones de bases de datos, la hemos realizado con SGBD SQLite versión 3. También interactúa con ficheros en formato *.pdf* y ficheros *.dbf* para exportación de asientos contables.

Este software es capaz de interactuar con diferentes componentes hardware, como son, la cámara integrada en el teléfono móvil, el *GPS* ó utilizar la antena para llamar a los clientes. Además, también podemos conectarnos mediante conexión *WIFI* a Internet, con lo cuál podemos, entre otras cosas, mandar mensajes de correo electrónico.

#### 4. Descripción general del proyecto

## 5. Desarrollo del proyecto

Procederemos a describir el análisis, diseño, implementación y realización de pruebas del proyecto.

### 5.1. Modelo de ciclo de vida

Para desarrollar este sistema se ha utilizado el modelo de ciclo de vida de desarrollo software lineal secuencial, también llamado modelo en cascada[14].

Como veremos más adelante, se parten de unos requisitos bien definidos en la etapa inicial de desarrollo para posteriormente realizar las fases de análisis, diseño, implementación y pruebas, por lo tanto, este modelo es el que mejor se adapta a nuestras necesidades.

### 5.2. Herramienta de modelado usada: Dia.

Dia es un programa de creación de diagramas, similar al programa Visio de la suite de ofimática de Microsoft Office. Está basado en GTK+, biblioteca con objetos y funciones para la interfaz gráfica de usuario, y tiene licencia GPL. Dispone de una gran serie de extensiones que permiten la elaboración de diagramas entidad-interrelación, UML, flujo de datos, diagramas de red, entre otros.

Puede ser descargado desde esta página de SourceForge:

<http://sourceforge.net/projects/dia-installer/>

### 5.3. Requisitos

#### 5.3.1. Funcionales

A continuación se especifican los requisitos de la aplicación:

- Acceso al sistema
  - El usuario accederá al sistema mediante su correo electrónico y su contraseña, ambas previamente especificadas en el formulario de registro.

## 5. Desarrollo del proyecto

- Registro de la empresa
  - El usuario registrará la empresa con la que se desea trabajar mediante un formulario de registro.
- Recordatorio de contraseña
  - El usuario podrá recuperar su contraseña mediante una pregunta de seguridad previamente pedida en el formulario de registro de la empresa.
- Clientes
  - El usuario podrá consultar, modificar, listar, dar de alta o baja a un cliente.
  - El usuario podrá realizar una llamada al cliente.
  - El usuario podrá enviar un correo electrónico al cliente.
  - El usuario podrá localizar geográficamente al cliente en un mapa físico.
- Productos
  - El usuario podrá consultar, modificar, listar, dar de alta o baja a un producto.
- Categorías de productos.
  - El usuario podrá crear, modificar ó dar de baja una categoría de producto.
- Rutas
  - El usuario podrá consultar, modificar, listar, dar de alta o baja a una ruta de visita.
  - El usuario podrá añadir ó quitar clientes a una ruta de visita.
  - El usuario podrá visualizar en un mapa físico las localizaciones de los clientes que formen la ruta.
- Pedidos
  - El usuario podrá crear, consultar, modificar, listar ó borrar un pedido.
  - El usuario podrá añadir, modificar ó quitar productos al pedido.
  - El usuario podrá albaranar ó facturar el pedido.
- Albarán
  - El usuario podrá crear, consultar, modificar, listar ó borrar un albarán.

- El usuario podrá añadir, modificar ó quitar productos al albarán.
  - El usuario podrá facturar el albarán.
  - El usuario podrá generar un PDF del albarán.
  - El usuario podrá visualizar el PDF generado del albarán.
  - El usuario podrá enviar el PDF generado de la factura por correo electrónico.
- Factura
    - El usuario podrá crear, consultar ó listar tanto una factura pendiente como una factura cobrada.
    - El usuario podrá cobrar la factura.
    - El usuario podrá generar un PDF de la factura.
    - El usuario podrá visualizar el PDF generado del factura.
    - El usuario podrá enviar el PDF generado de la factura por correo electrónico.
  - Carga del vehículo.
    - El usuario podrá agregar, modificar el stock ó quitar productos del vehículo.
  - Exportación
    - El usuario podrá generar y listar ficheros de exportación de los asientos contables de las facturas emitidas y cobradas.
    - El usuario podrá enviar por correo electrónico los ficheros de exportación.
  - Estadísticas
    - El usuario podra generar reportes de estadísticas de las facturas realizadas y los productos más vendidos en un determinado intervalo de tiempo.
  - configuración
    - El usuario podrá modificar los datos personales de la empresa.
    - El usuario podrá modificar los impuestos aplicados a las facturas, entre los que figuran el IVA y el RE.
    - El usuario podrá modificar la longitud de la subcuenta de los asientos contables de las facturas.
    - El usuario podrá activar el aviso de riesgo en las ventas a clientes.

## 5. Desarrollo del proyecto

- El usuario podrá activar el control de stock de productos en las ventas a clientes.
  - El usuario podrá realizar y restaurar copias de seguridad de la base de datos.
  - El usuario podrá acceder a la pantalla de Acerca de, que contendrá información acerca de la aplicación.
- Acerca de
    - El usuario podrá reportar un error al desarrollador mediante correo electrónico.
    - El usuario podrá calificar la aplicación en el mercado de aplicaciones Google Play.
    - El usuario podrá visualizar las principales librerías utilizadas en la aplicación.

### 5.3.2. Interfaces de Usuario

A continuación se exponen los requisitos de la interfaz de usuario:

- **Interfaz intuitiva, clara y sencilla:** la interfaz cumple tres características principales, sencillez, claridad y por último, se ha pretendido que resulte intuitiva para el usuario.
- **Barra de Acciones:** en la aplicación se ha utilizado la librería de Cyril Mottier llamada *GreenDroid*[7] que hace mostrar en cada una de las pantallas de la aplicación una barra de acciones en la parte superior en la cual podremos ver todas las opciones disponibles en cada pantalla y, también, un acceso directo al menú principal.
- **Mensajes de la aplicación:** los mensajes tanto críticos, como informativos de la aplicación, serán mostrados conforme se puede apreciar en la Figura 5.89.

### 5.3.3. Informes y ficheros generados.

Con respecto a los informes que debe generar la aplicación, destacar que son documentos que trabajan directamente contra la base de datos. Algunas características destacables de los informes son:

1. **Claridad:** una de las características más importantes, deben resultar claros para facilitar la lectura por parte del usuario.

2. **Precisión:** no deben contener más información de la realmente necesaria.
3. **Formato universal:** se utilizará el formato ampliamente distribuido y usado ".pdf".

Con respecto a los ficheros de exportación, se ha acompañado un anexoE explicando detenidamente qué son exactamente y qué es lo que contienen. Se ha preferido generarlos en formato ".dbf".

#### 5.3.4. No funcionales

Cualquier sistema software a desarrollar, debe poseer las siguientes propiedades:

- **Fiabilidad:** este producto trata con datos muy importantes para la empresa, por tanto, una de sus características más importantes debe ser la fiabilidad. Por ello, todas las operaciones críticas de la base de datos se han controlado debidamente, y se han capturado a través de mensajes al usuario todas las excepciones que se pudieran producir. Si se produjese algún error se recomienda al usuario, ponerse en contacto con el desarrollador del software.
- **Disponibilidad:** el sistema está totalmente disponible localmente en nuestro terminal. Este es un requisito muy importante, ya que se ha preferido trabajar sin uso de la red de datos ya que, puede darse el caso, de estar realizando una venta en la que no se tenga una buena cobertura de red.
- **Seguridad:** en ikeComercial, la seguridad la marca el acceso a la base de datos, por ello, se han encriptado las contraseñas de los usuarios con el algoritmo de encriptación AES, así como para acceder a la aplicación, es necesario introducir el correo electrónico y la contraseña descritos en el formulario de registro. También se ha introducido una pregunta de seguridad para poder restaurar la clave de acceso en el caso de no recordar la contraseña.
- **Mantenibilidad:** esta aplicación no necesita la realización de pasos previos para su utilización, simplemente podemos instalarla desde el mercado de aplicaciones y empezar a usarla instantáneamente. Por supuesto, es fundamental la organización de la implementación del producto para que posteriores ampliaciones y correcciones se hagan de forma rápida y sencilla. Para cualquier duda, no dude en contactar con el desarrollador.

## 5.4. Análisis del sistema

En esta sección, seguiremos la notación del lenguaje de modelado de sistemas software UML[24].

### 5.4.1. Casos de uso

Los casos de uso describen la secuencia de acciones, incluyendo variantes, que ejecuta un sistema y que producen unos resultados observables para un actor particular. Estos casos de uso nos permiten analizar de manera general y abstracta qué es lo que se requiere en nuestro sistema.

#### Diagramas de casos de uso

En la Figura 5.1 se muestra el diagrama de casos de uso de registro de empresa, recuperar contraseña y acceso al sistema.

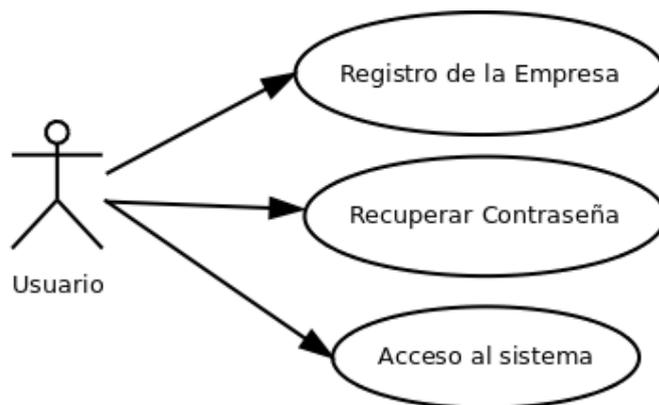


Figura 5.1.: Diagrama de casos de uso inicial.

En la Figura 5.2 se muestra el diagrama de casos de uso del menú principal.

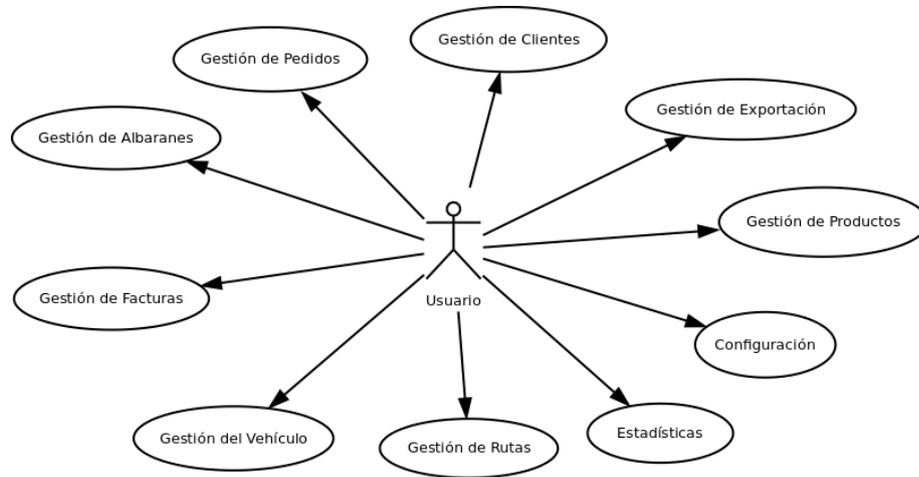


Figura 5.2.: Diagrama de casos de uso del menú principal.

En la Figura 5.3 se muestra el diagrama de casos de uso de gestión de clientes.

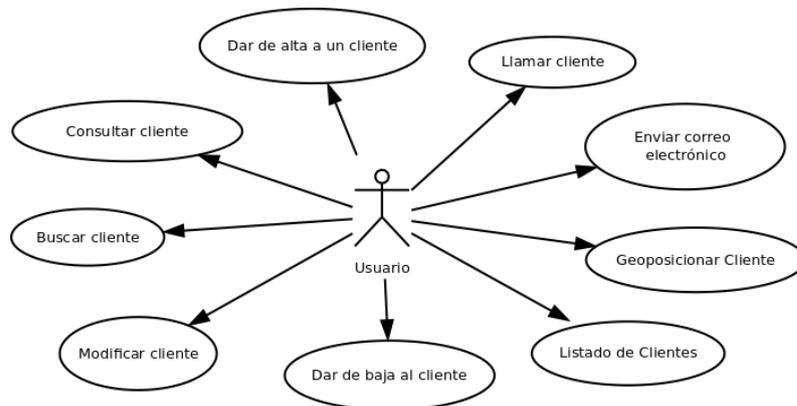


Figura 5.3.: Diagrama de casos de uso de gestión de clientes.

## 5. Desarrollo del proyecto

En la Figura 5.4 se muestra el diagrama de casos de uso de gestión de productos.

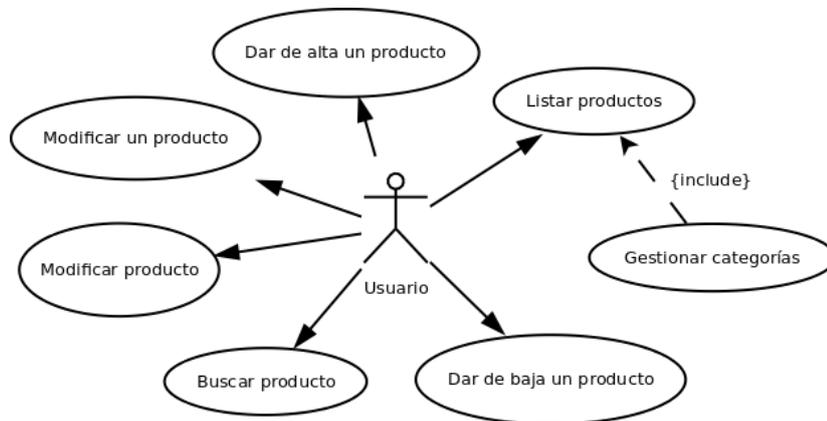


Figura 5.4.: Diagrama de casos de uso de gestión de productos.

En la Figura 5.5 se muestra el diagrama de casos de uso de la gestión de categorías de producto.

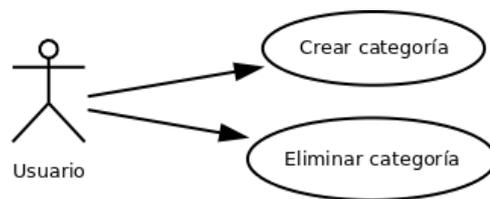


Figura 5.5.: Diagrama de casos de uso de gestión de categorías de producto.

En la Figura 5.6 se muestra el diagrama de casos de uso de gestión de rutas.

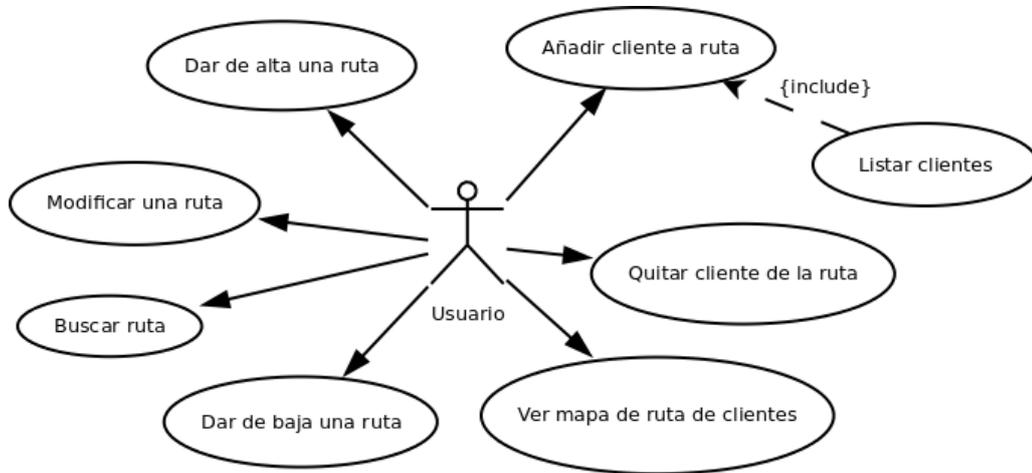


Figura 5.6.: Diagrama de casos de uso de gestión de rutas.

En la Figura 5.7 se muestra el diagrama de casos de uso de gestión de pedidos.

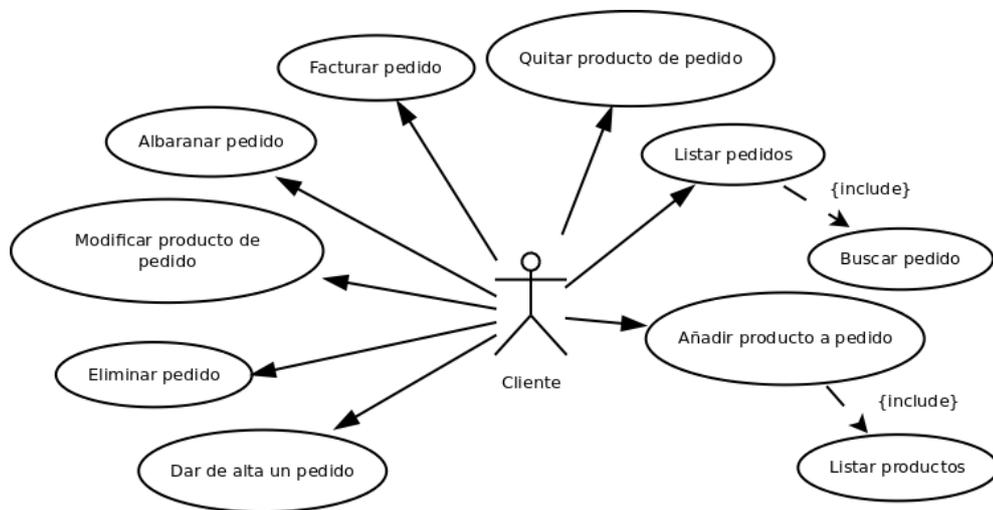


Figura 5.7.: Diagrama de casos de uso de gestión de pedidos.

## 5. Desarrollo del proyecto

En la Figura 5.8 se muestra el diagrama de casos de uso de gestión de albaranes.

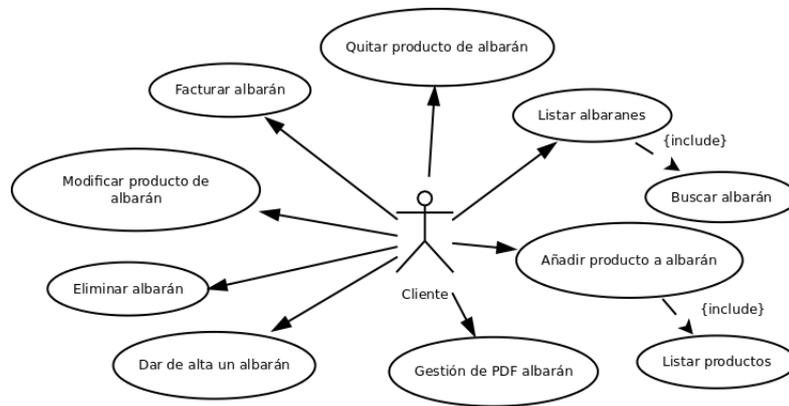


Figura 5.8.: Diagrama de casos de uso de gestión de albaranes.

En la Figura 5.9 se muestra el diagrama de casos de uso de gestión de PDF albaranes.

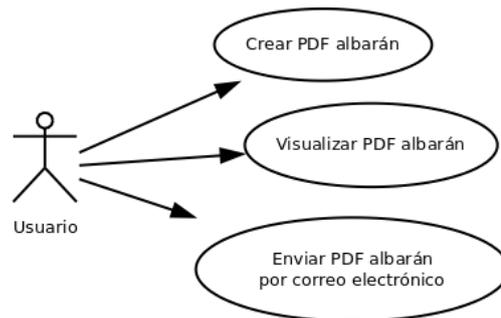


Figura 5.9.: Diagrama de casos de uso de gestión de PDF albaranes.

En la Figura 5.10 se muestra el diagrama de casos de uso de gestión de facturas.

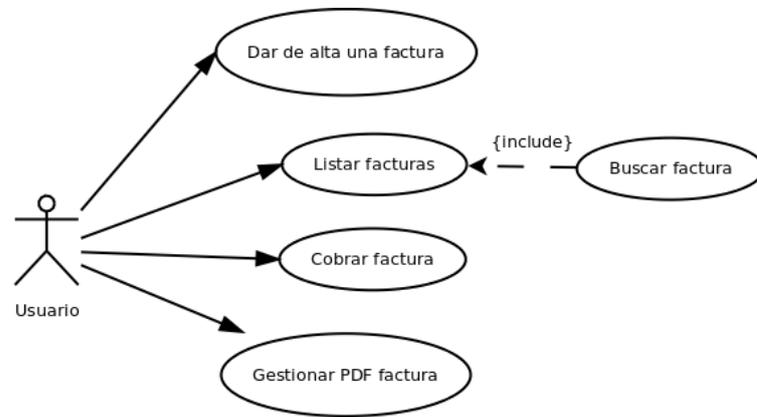


Figura 5.10.: Diagrama de casos de uso de gestión de facturas.

En la Figura 5.11 se muestra el diagrama de casos de uso de gestión de PDF facturas.

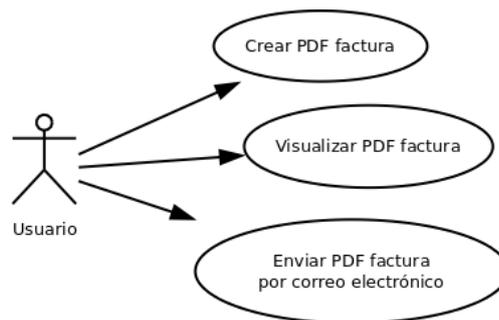


Figura 5.11.: Diagrama de casos de uso de gestión de PDF facturas.

## 5. Desarrollo del proyecto

En la Figura 5.12 se muestra el diagrama de casos de uso de gestión de carga de vehículo.

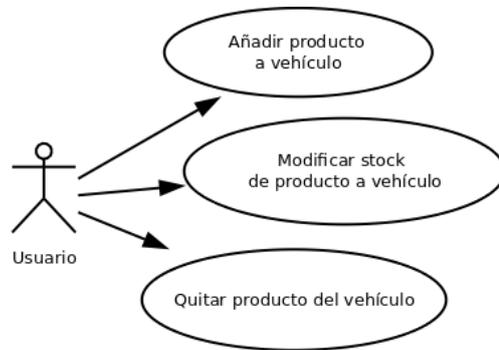


Figura 5.12.: Diagrama de casos de uso de gestión de carga de vehículo.

En la Figura 5.13 se muestra el diagrama de casos de uso de gestión de exportaciones.

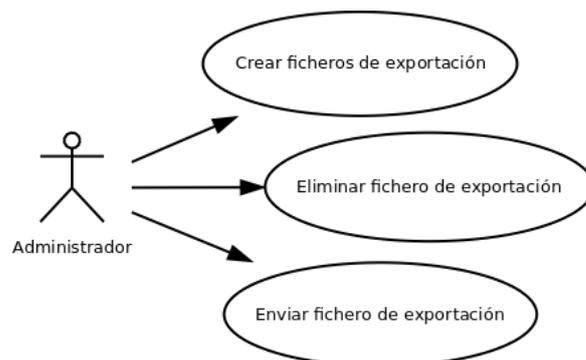


Figura 5.13.: Diagrama de casos de uso de gestión de exportaciones.

En la Figura 5.14 se muestra el diagrama de casos de uso de generación de estadísticas.

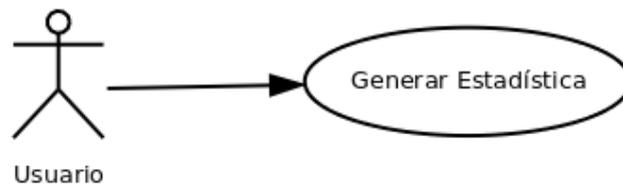


Figura 5.14.: Diagrama de casos de uso de generación de estadísticas.

En la Figura 5.15 se muestra el diagrama de casos de uso de configuración.

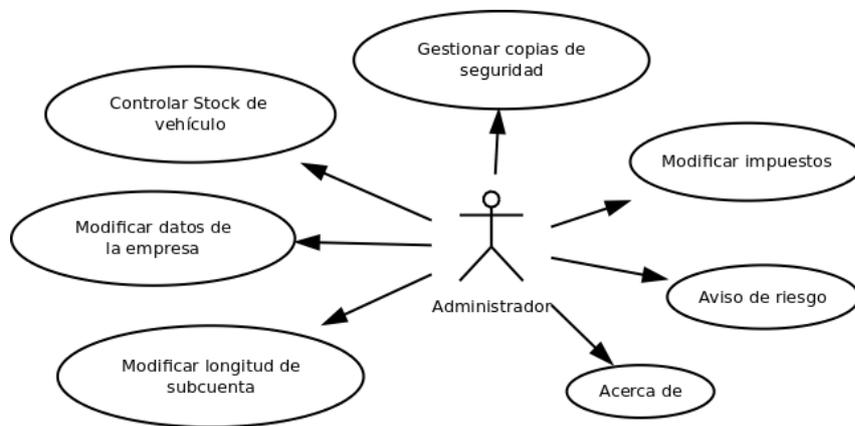


Figura 5.15.: Diagrama de casos de uso de configuración.

## 5. Desarrollo del proyecto

En la Figura 5.16 se muestra el diagrama de casos de uso de gestión de copias de seguridad.

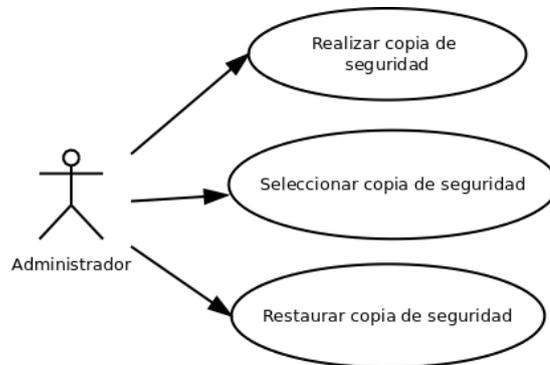


Figura 5.16.: Diagrama de casos de uso de gestión de copias de seguridad.

### 5.4.2. Especificación de casos de uso

#### Registro de la empresa

- **Actor principal** Usuario: desea registrar a su empresa en el sistema.
- **Precondiciones** Ninguna
- **Postcondiciones** Se realiza el registro de la empresa en el sistema.
- **Escenario principal**
  1. El usuario accede a la pantalla de bienvenida de la aplicación.
  2. El usuario pulsa el boton "Regístrese" de la pantalla de bienvenida.
  3. El usuario rellena el formulario de registro y lo acepta.
  4. El sistema registra los datos de la empresa en el sistema y vuelve a la pantalla de bienvenida.
- **Variaciones**
  - 3a. El sistema detecta que uno de los campos del formulario es incorrecto.
    1. El sistema muestra los campos con valores no válidos mediante un mensaje.
    2. El usuario modifica el formulario introduciendo valores válidos para los campos incorrectos.
    3. El usuario acepta el formulario de registro de nuevo.

### Recuperar contraseña de acceso

- **Actor principal** Usuario: desea recuperar la contraseña de acceso al sistema.
- **Precondiciones** El usuario se ha registrado previamente en el sistema.
- **Postcondiciones** El usuario recupera la contraseña de acceso para entrar en el sistema.
- **Escenario principal**
  1. El usuario desea recuperar la contraseña de acceso.
  2. El usuario pulsa sobre la pregunta "¿ha olvidado su contraseña?"
  3. El sistema muestra un cuadro de diálogo pidiendo la respuesta a la pregunta de seguridad.
  4. El usuario introduce la respuesta de seguridad y la acepta.
  5. El sistema muestra un cuadro de diálogo mostrando su contraseña de acceso.
- **Variaciones**
  - 4a. El sistema detecta que no es la respuesta correcta a la pregunta de seguridad.
    1. El sistema muestra un diálogo mostrando un mensaje describiendo que la respuesta no es la correcta.
    2. Se vuelve al paso 1.

### Acceso al sistema

- **Actor principal** Usuario: se desea acceder al sistema. Para ello introduce el email de registro y la contraseña, el sistema valida al usuario y muestra la pantalla del menú principal.
- **Precondiciones** el usuario está registrado en el sistema.
- **Postcondiciones** se accederá al sistema.
- **Escenario principal**
  1. El usuario desea acceder al sistema.
  2. El usuario introduce sus datos de login (email y contraseña).
  3. El sistema valida que los datos introducidos por el usuario son correctos.

## 5. Desarrollo del proyecto

4. El sistema muestra el menú principal de la aplicación.

### ■ Variaciones

3a. El sistema detecta que los datos de entrada del usuario son incorrectos.

1. El sistema muestra un mensaje de error.
2. Vuelve al paso 2.

## Dar de alta a un cliente

■ **Actor principal** Usuario: desea dar de alta a un nuevo cliente

■ **Precondiciones** Ninguna.

■ **Postcondiciones** Se registrará un nuevo cliente en el sistema

### ■ Escenario principal

1. El usuario desea registrar un nuevo cliente.
2. El sistema muestra el formulario de registro de un nuevo cliente.
3. El usuario introduce todos los datos de registro de cliente.
4. El sistema valida los datos introducidos por el usuario
5. El sistema registra al nuevo cliente en el sistema.

### ■ Variaciones

4a. El sistema detecta que los datos introducidos no son correctos.

1. El sistema muestra mensajes de error en los campos incorrectos del formulario.
2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
3. El sistema vuelve a validar los datos.

4b. El sistema detecta que el número de cliente ya existe.

1. El sistema muestra un mensaje de error contemplando el error.
2. El usuario introduce un nuevo número de cliente.
3. El sistema valida el nuevo número de cliente introducido.

4c. El sistema detecta que el dni está registrado y que también pertenece a un cliente activo.

1. El sistema muestra un mensaje confirmando que el dni introducido está ya dado de alta.

2. El usuario volverá a introducir un nuevo dni para el cliente.
3. El sistema valida el nuevo dni introducido.
- 4d. El sistema detecta que el dni está registrado y que pertenece a un cliente que no está activo.
  1. El sistema muestra un mensaje confirmando que el dni ya se encuentra en el sistema y que pertenece a un cliente inactivo.
  2. El sistema muestra un cuadro de diálogo preguntando si se desea activar nuevamente a este cliente
  3. En caso afirmativo, el cliente quedará visible de nuevo desde el listado de clientes. En caso contrario, el cliente actual no se dará de alta.
- 4e. El sistema detecta que el correo electrónico introducido está registrado.
  1. El sistema muestra un mensaje de error.
  2. El usuario podrá volver a introducir un nuevo correo electrónico.
  3. El sistema valida el nuevo correo electrónico introducido.

### **Dar de baja a un cliente**

- **Actor principal** Usuario: desea dar de baja a un cliente en el sistema.
- **Precondiciones** El usuario debe estar registrado en el sistema.
- **Postcondiciones** Se dará de baja al cliente en el sistema
- **Escenario principal**
  1. El usuario desea dar de baja a un cliente.
  2. El usuario selecciona la opción "Eliminar Cliente" en la ficha de cliente que desea dar de baja.
  3. El sistema muestra un mensaje de confirmación.
  4. El usuario acepta el mensaje de confirmación.
  5. El sistema dará de baja al cliente en el sistema.
- **Variaciones**
  - 3a. El usuario deniega la petición de eliminación de cliente.
    1. El sistema vuelve a la pantalla de la ficha de cliente.

## 5. Desarrollo del proyecto

### Modificar Cliente

- **Actor principal** Usuario: desea modificar un dato de la ficha de cliente.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificará los datos de la ficha de cliente.
- **Escenario principal**
  1. El usuario desea modificar los datos de la ficha de cliente.
  2. El usuario introduce los nuevos datos modificados en la ficha de cliente.
  3. El usuario pulsa sobre el botón "Actualizar Cliente".
  4. El sistema comprueba los datos introducidos y son correctos.
  5. El sistema actualiza la ficha del cliente.
- **Variaciones**
  - 4a. El sistema detecta que un dato introducido no es válido.
    1. El sistema muestra un mensaje de error.
    2. El usuario podrá cambiar los datos nuevamente.
    3. Se vuelve al paso 3.

### Consultar Cliente

- **Actor principal** Usuario: desea consultar la ficha de cliente.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán los datos de la ficha de cliente.
- **Escenario principal**
  1. El usuario desea mostrar los datos de la ficha de cliente.
  2. El usuario pulsa sobre el cliente a mostrar
  3. El sistema visualiza la ficha del cliente seleccionado.

### Buscar Cliente

- **Actor principal** Usuario: desea buscar un cliente.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán los clientes que se asemejen a la búsqueda realizada.
- **Escenario principal**
  1. El usuario desea buscar cliente.
  2. El usuario pulsa sobre la opción "Buscar Cliente"
  3. El usuario escribe en el campo de búsqueda la razón social del cliente a buscar.
  4. El sistema visualiza los clientes con la razón social indicada.

### Llamar a un cliente

- **Actor principal** Usuario: desea realizar una llamada telefónica al cliente
- **Precondiciones** El usuario debe de estar registrado en el sistema.
- **Postcondiciones** Se realizará una llamada al número especificado para el cliente.
- **Escenario principal**
  1. El usuario desea realizar una llamada al cliente.
  2. El usuario pulsa sobre el botón "Llamada".
  3. El sistema realizará una llamada al cliente.
- **Variaciones**
  - 3a. El sistema detecta que no hay ningún número de teléfono especificado para el cliente.
    1. El sistema mostrará un mensaje de error.
    2. El usuario podrá introducir un número de teléfono para el cliente.
  - 3b. El sistema detecta que hay dos números de teléfono para el cliente.
    1. El sistema mostrará un cuadro de diálogo con los número de teléfonos disponibles para el cliente.
    2. El usuario podrá elegir entre los números disponibles y escogerá uno de ellos.

## 5. Desarrollo del proyecto

### Enviar correo electrónico a cliente

- **Actor principal** Usuario: desea enviar un correo electrónico al cliente
- **Precondiciones** Es necesario haber instalado una aplicación gestora de correo electrónico.
- **Postcondiciones** USe abrirá la aplicación de correo electrónico con la dirección de correo electrónico del cliente.
- **Escenario principal**
  1. El usuario desea enviar un correo electrónico a un cliente.
  2. El usuario pulsa sobre "Enviar Email" en la ficha del cliente.
  3. La aplicación mostrará en una lista seleccionable los gestores de correo instalados en el terminal.
  4. El usuario elegirá uno de ellos y abrirá la aplicación de correo electrónico con la dirección de correo del cliente.
- **Variaciones**
  - 3a. La aplicación detecta que no hay ningún gestor de correo electrónico compatible.
    1. El sistema muestra un mensaje conforme a la imposibilidad de realizar esta acción.

### Geolocalizar al cliente

- **Actor principal** Usuario: desea localizar geográficamente al cliente en un mapa.
- **Precondiciones** USe tiene que tener instalada la aplicación Google Play Services para el uso de Google Maps v2.
- **Postcondiciones** Se guardará la ubicación en forma de coordenadas geográficas al cliente.
- **Escenario principal**
  1. El usuario desea localizar geográficamente al cliente.
  2. El usuario presiona sobre el botón "Geolocalización" en la ficha de cliente.
  3. El sistema visualiza Google Maps en pantalla.
  4. El usuario toca la pantalla y se señala un nuevo punto de geolocalización.

5. El usuario presiona sobre el botón "Capturar ubicación".
6. El sistema guarda la ubicación del cliente y vuelve a la ficha de cliente.

■ **Variaciones**

- 3a. El sistema detecta que no se dispone de acceso a Internet.
  1. El sistema muestra la opción para activar la red de datos ó activar la red inalámbrica.
  2. El usuario tiene acceso a Internet.
- 4a. El sistema no acepta la ubicación señalada como un punto de geolocalización válido.
  1. El usuario vuelve a presionar en la pantalla para obtener un nuevo punto de geolocalización.
- 5a. El sistema detecta que no hay una nueva ubicación establecida para el cliente.
  1. El sistema vuelve a la ficha del cliente.

**Dar de alta a un producto**

- **Actor principal** Usuario: desea dar de alta un nuevo producto en el sistema.
- **Precondiciones** .Ninguna
- **Postcondiciones** .El sistema registrará un nuevo producto en el sistema.
- **Escenario principal**
  1. El usuario desea registrar un nuevo producto en el sistema.
  2. El usuario presiona la opción "Nuevo Producto".
  3. El usuario introduce todos los datos de registro en el formulario y los acepta.
  4. El sistema valida que los datos introducidos son correctos.
  5. El sistema registra el nuevo producto en el sistema.
- **Variaciones**
  - 4a. El sistema detecta que los datos de registro son incorrectos.
    1. El sistema muestra un mensaje de error en aquellos campos no válidos.
    2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
    3. El sistema vuelve a validar los datos.

## 5. Desarrollo del proyecto

### Dar de baja a un producto

- **Actor principal** Usuario: desea dar de baja un producto en el sistema.
- **Precondiciones** .El producto deberá estar registrado previamente.
- **Postcondiciones** .El sistema dará de baja un producto en el sistema.
- **Escenario principal**
  1. El usuario desea dar de baja un producto del sistema.
  2. El usuario visualiza la ficha del producto y presiona la opción "Eliminar Producto".
  3. El sistema muestra un mensaje de confirmación y el usuario acepta.
  4. El sistema elimina el producto indicado del sistema.

### Listar Productos

- **Actor principal** Usuario: desea listar los productos registrados.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán los productos registrados en el sistema.
- **Escenario principal**
  1. El usuario desea mostrar todos los productos.
  2. El usuario pulsa sobre la opción "Productos" del menú principal.
  3. El sistema visualiza los productos registrados en pantalla.

### Buscar Producto

- **Actor principal** Usuario: desea buscar un producto.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán los productos que se asemejen a la búsqueda realizada.
- **Escenario principal**
  1. El usuario desea buscar un producto.
  2. El usuario pulsa sobre la opción "Buscar Producto"
  3. El usuario escribe en el campo de búsqueda el nombre completo del producto a buscar.
  4. El sistema visualiza los productos que se asemejan al nombre indicado.

### **Modificar Producto**

- **Actor principal** Usuario: desea modificar datos de la ficha de producto.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificará los datos de la ficha de producto.
- **Escenario principal**
  1. El usuario desea modificar los datos de la ficha de producto.
  2. El usuario introduce los nuevos datos modificados en la ficha de producto.
  3. El usuario pulsa sobre el botón "Actualizar Producto".
  4. El sistema comprueba los datos introducidos y son correctos.
  5. El sistema actualiza la ficha del producto.
- **Variaciones**
  - 4a. El sistema detecta que un dato introducido no es válido.
    1. El sistema muestra un mensaje de error.
    2. El usuario podrá cambiar los datos nuevamente.
    3. Se vuelve al paso 3.

### **Crear una nueva categoría de producto**

- **Actor principal** Usuario: desea dar de alta una nueva categoría de producto.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se dará de alta una nueva categoría de producto.
- **Escenario principal**
  1. El usuario desea dar de alta una nueva categoría de producto.
  2. El usuario introduce el nombre de la nueva categoría de producto y presiona "Añadir".
  3. El sistema realiza el alta de la nueva categoría de producto.
- **Variaciones**
  - 2a. El sistema detecta que la categoría introducida ya existe.
    1. El sistema muestra un mensaje de error y no registra la categoría.

## 5. Desarrollo del proyecto

### Eliminar una categoría de producto

- **Actor principal** Usuario: desea eliminar una categoría de producto
- **Precondiciones** La categoría de producto esta registrada.
- **Postcondiciones** Se eliminará la categoría de producto del sistema.
- **Escenario principal**
  1. El usuario desea eliminar una categoría de producto.
  2. El usuario presiona prolongadamente una categoría de producto en la lista de categorías.
  3. El sistema mostrará un mensaje de confirmación de eliminación de categoría.
  4. El usuario acepta la confirmación.
  5. El sistema elimina la categoría de producto seleccionada.

### Dar de alta una ruta

- **Actor principal** Usuario: desea dar de alta a una nueva ruta.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se registrará un nueva ruta en el sistema.
- **Escenario principal**
  1. El usuario desea registrar una nueva ruta en el sistema.
  2. El usuario introduce el nombre de la ruta y el día de la semana que se realiza.
  3. El sistema valida los datos introducidos por el usuario.
  4. El sistema registra la nueva ruta en el sistema.
- **Variaciones**
  - 3a. El sistema detecta que los datos introducidos no son correctos.
    1. El sistema muestra mensajes de error en los campos incorrectos del formulario.
    2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
    3. El sistema vuelve a validar los datos.

### **Dar de baja una ruta**

- **Actor principal** Usuario: desea dar de baja una ruta en el sistema.
- **Precondiciones** La ruta está registrada en el sistema.
- **Postcondiciones** Se dará de baja la ruta en el sistema
- **Escenario principal**
  1. El usuario desea dar de baja una ruta.
  2. El sistema muestra un mensaje de confirmación de eliminación.
  3. El usuario acepta el mensaje de confirmación.
  4. El sistema dará de baja la ruta y los clientes que estaban asociados a la ruta dejarán de estarlo.

### **Modificar Ruta**

- **Actor principal** Usuario: desea modificar datos de la ficha de ruta.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificará los datos de la ficha de ruta.
- **Escenario principal**
  1. El usuario desea modificar los datos de la ficha de ruta.
  2. El usuario introduce los nuevos datos modificados en la ficha de clieruta.
  3. El sistema comprueba los datos introducidos y son correctos.
  4. El sistema actualiza la ficha de la ruta.
- **Variaciones**
  - 4a. El sistema detecta que un dato introducido no es válido.
    1. El sistema muestra un mensaje de error.
    2. El usuario podrá cambiar los datos nuevamente.
    3. Se vuelve al paso 3.

## 5. Desarrollo del proyecto

### Buscar Ruta

- **Actor principal** Usuario: desea buscar una ruta en un día determinado.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán las rutas registradas que se realicen ese día.
- **Escenario principal**
  1. El usuario desea buscar una ruta de un determinado día de la semana.
  2. El usuario escoge un día de la semana.
  3. El sistema visualiza las rutas del día de la semana indicado.

### Añadir cliente a ruta

- **Actor principal** Usuario: desea añadir un cliente a la ruta.
- **Precondiciones** Ninguna
- **Postcondiciones** El cliente será añadido a la ruta.
- **Escenario principal**
  1. El usuario desea añadir a un cliente en la ruta.
  2. El usuario selecciona a un cliente de la ruta a agregar.
  3. El sistema agrega al cliente en la ruta.
- **Variaciones**
  - 3a. El cliente ya está agregado a la ruta.
    1. Se muestra un mensaje de aviso mostrando que el cliente ya está agregado.

### Quitar cliente a ruta

- **Actor principal** Usuario: desea quitar un cliente de la ruta.
- **Precondiciones** El cliente debe de estar añadido a dicha ruta.
- **Postcondiciones** El sistema quitará al cliente de la ruta.
- **Escenario principal**
  1. El usuario desea quitar un cliente de la ruta.
  2. El usuario presiona prolongadamente al cliente que desea quitar.

3. El sistema mostrará un mensaje de confirmación y el usuario acepta.
4. El sistema quita al cliente de la ruta.

▪ **Variaciones**

- 3a. El usuario deniega la petición.
  1. El sistema no quita al cliente de la ruta.

**Ver mapa de ruta de clientes.**

- **Actor principal** Usuario: desea visualizar la ruta de clientes en un mapa.
- **Precondiciones** Los clientes deberán estar geolocalizados.
- **Postcondiciones** Se visualiza por pantalla el mapa de ruta de los clientes asociados a ella.
- **Escenario principal**
  1. El usuario desea ver el mapa de ruta de los clientes asociados a ella.
  2. El sistema muestra el mapa de ruta de los clientes asociados a ella.

**Dar de alta a un pedido**

- **Actor principal** Usuario: desea dar de alta un nuevo pedido.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se registrará un nuevo pedido en el sistema.
- **Escenario principal**
  1. El usuario desea registrar un nuevo pedido.
  2. El usuario inicia una nueva venta en el sistema.
  3. El sistema muestra la lista de clientes disponibles.
  4. El usuario selecciona un cliente dentro de la lista.
  5. El usuario introduce todos los datos del pedido.
  6. El sistema valida los datos de pedido y lo registra en el sistema.
- **Variaciones**
  - 6a. El sistema detecta que los datos introducidos no son correctos.

## 5. Desarrollo del proyecto

1. El sistema muestra mensajes de error en los campos incorrectos del formulario.
2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
- 6b. El sistema detecta que no hay ningún producto añadido al pedido.
  1. El sistema muestra un mensaje pidiendo, al menos, un producto en la lista.

### Eliminar a un pedido

- **Actor principal** Usuario: desea eliminar un pedido.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se dará de baja un nuevo pedido en el sistema.
- **Escenario principal**
  1. El usuario desea dar de baja un pedido.
  2. El usuario indica cuál es el pedido a eliminar.
  3. El sistema valida el pedido y lo elimina.
- **Variaciones**
  - 4a. El sistema detecta que los datos introducidos no son correctos.
    1. El sistema muestra mensajes de error y no se elimina el pedido.

### Buscar pedido

- **Actor principal** Usuario: desea buscar un pedido.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrará el pedido buscado.
- **Escenario principal**
  1. El usuario desea buscar un pedido.
  2. El usuario indica los datos del pedido a buscar.
  3. El sistema busca, encuentra y visualiza el pedido que se deseaba encontrar.

- Variantes
  - 3a. El sistema no encuentra el pedido buscado.
    1. Muestra un mensaje de que no se ha encontrado el pedido buscado

### **Listar Pedido**

- **Actor principal** Usuario: desea listar los pedidos registrados.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán los pedidos registrados en el sistema.
- **Escenario principal**
  1. El usuario desea mostrar todos los pedidos y lo indica.
  2. El sistema visualiza los pedidos registrados en pantalla.

### **Añadir producto a pedido**

- **Actor principal** Usuario: desea añadir un producto a pedido.
- **Precondiciones** Ninguna
- **Postcondiciones** El producto será añadido al pedido.
- **Escenario principal**
  1. El usuario desea añadir un producto al pedido
  2. El usuario selecciona el producto que desea añadir de la lista de productos registrados.
  3. El sistema le pide al usuario la cantidad de productos que desea añadir de ese tipo.
  4. El usuario le indica el número de unidades a añadir y lo acepta.
  5. El sistema añade el producto al pedido.
- **Variaciones**
  - 4a. El sistema detecta que el número de unidades a añadir es incorrecta
    1. Se muestra un mensaje de error y se le da la oportunidad de volver a insertar una nueva cantidad.

## 5. Desarrollo del proyecto

### Quitar producto del pedido

- **Actor principal** Usuario: desea quitar un producto del pedido
- **Precondiciones** El producto debe pertenecer al pedido.
- **Postcondiciones** El sistema quitará el producto del pedido.
- **Escenario principal**
  1. El usuario desea quitar un producto del pedido.
  2. El usuario le indica el producto a quitar.
  3. El sistema mostrará un mensaje de confirmación y el usuario acepta.
  4. El sistema quita el producto del pedido.
- **Variaciones**
  - 3a. El usuario deniega la petición.
    1. El sistema no quita el producto del pedido.

### Albaranar Pedido

- **Actor principal** Usuario: desea albaranar el pedido.
- **Precondiciones** El pedido debe de tener, al menos, un producto.
- **Postcondiciones** Se creará un albarán a partir del pedido seleccionado.
- **Escenario principal**
  1. El usuario desea crear un albarán a partir de un pedido.
  2. El usuario le indica al sistema el pedido a albaranar.
  3. El sistema albarana el pedido indicado.

### Facturar Pedido

- **Actor principal** Usuario: desea facturar un pedido.
- **Precondiciones** El pedido debe tener, al menos, un producto.
- **Postcondiciones** Se facturará el pedido y se generará su correspondiente asiento contable.
- **Escenario principal**

1. El usuario desea facturar un pedido.
2. El usuario indica el pedido a facturar.
3. El sistema factura el pedido indicado y genera los asientos contables pertinentes.

### **Dar de alta un albarán**

- **Actor principal** Usuario: desea dar de alta un nuevo albarán.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se dará de alta un nuevo albarán.
- **Escenario principal**
  1. El usuario desea dar de alta un nuevo albarán.
  2. El sistema muestra la lista de clientes disponibles.
  3. El usuario selecciona un cliente dentro de la lista.
  4. El usuario introduce todos los datos del albarán.
  5. El sistema valida los datos de pedido y lo registra en el sistema.
- **Variaciones**
  - 5a. El sistema detecta que los datos introducidos no son correctos.
    1. El sistema muestra mensajes de error en los campos incorrectos del formulario.
    2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
  - 5b. El sistema detecta que no hay ningún producto añadido al albarán
    1. El sistema muestra un mensaje pidiendo, al menos, un producto en la lista.

### **Eliminar albarán**

- **Actor principal** Usuario: desea eliminar un albarán.
- **Precondiciones** El albarán debe de estar registrado previamente en el sistema.
- **Postcondiciones** Se eliminará el albarán indicado.

## 5. Desarrollo del proyecto

### ■ Escenario principal

1. El usuario desea eliminar un albarán.
2. El usuario indica el albarán a a eliminar.
3. El sistema elimina el albarán indicado.

### Añadir producto a albarán

- **Actor principal** Usuario: desea añadir un producto al albarán.

- **Precondiciones** Ninguna.

- **Postcondiciones** Se añadirá el producto al albarán.

### ■ Escenario principal

1. El usuario desea añadir un producto al albarán.
2. El usuario indica el producto a añadir de la lista de productos.
3. El sistema le pide al usuario la cantidad de productos que desea añadir de ese tipo.
4. El usuario le indica el número de unidades a añadir y lo acepta.
5. El sistema añade el producto al albarán.

### ■ Variaciones

- 4a. El sistema detecta que el número de unidades a añadir es incorrecta
  1. Se muestra un mensaje de error y se le da la oportunidad de volver a insertar una nueva cantidad.

### Quitar producto en albarán

- **Actor principal** Usuario: desea quitar un producto del albarán.

- **Precondiciones** Ninguna.

- **Postcondiciones** Se quitará el producto del albarán.

### ■ Escenario principal

1. El usuario le indica el producto a quitar.
2. El sistema mostrará un mensaje de confirmación y el usuario acepta.
3. El sistema quita el producto del albarán.

- **Variaciones**

3a. El usuario deniega la petición.

1. El sistema no quita el producto del albarán.

### **Facturar albarán**

- **Actor principal** Usuario: desea facturar un albarán

- **Precondiciones** El albarán debe de tener, al menos, un producto.

- **Postcondiciones** Se facturará el albarán indicado por el usuario.

- **Escenario principal**

1. El usuario desea facturar un albarán
2. El usuario indica el albarán a facturar.
3. El sistema factura el albarán y genera los asientos contables pertinentes.

### **Listar albaranes**

- **Actor principal** Usuario: desea buscar una ruta en un día determinado.

- **Precondiciones** Ninguna.

- **Postcondiciones** Se mostrarán los albaranes registrados.

- **Escenario principal**

1. El usuario desea mostrar todos los albaranes y lo indica.
2. El sistema visualiza los albaranes registrados en pantalla.

### **Buscar albarán**

- **Actor principal** Usuario: desea buscar un albarán.

- **Precondiciones** Ninguna.

- **Postcondiciones** Se mostrará el albarán buscado.

- **Escenario principal**

1. El usuario desea buscar un albarán.
2. El usuario indica los datos del albarán a buscar.

## 5. Desarrollo del proyecto

3. El sistema busca, encuentra y visualiza el albarán que se deseaba encontrar.

- Variantes

3a. El sistema no encuentra el albarán buscado.

1. Muestra un mensaje de que no se ha encontrado el albarán buscado.

### Crear PDF albarán

- **Actor principal** Usuario: desea crear un fichero PDF de un albarán.

- **Precondiciones** Ninguna.

- **Postcondiciones** Se creará el fichero PDF del albarán.

- **Escenario principal**

1. El usuario desea crear un fichero PDF de un albarán.

2. El usuario indica el albarán a partir del cuál se va a crear el fichero PDF.

3. El sistema crea el fichero PDF del albarán.

### Visualizar PDF albarán

- **Actor principal** Usuario: desea visualizar el PDF del albarán.

- **Precondiciones** Se ha de tener instalado, al menos, un visor de documentos PDF y se deberá de haber creado previamente.

- **Postcondiciones** Se mostrará el albarán en el visor de documentos PDF indicado.

- **Escenario principal**

1. El usuario desea visualizar el PDF de un albarán.

2. El usuario indica el albarán a mostrar.

3. El sistema visualiza en pantalla el fichero PDF del albarán en el visor de documentos especificado.

### Enviar PDF albarán por correo electrónico

- **Actor principal** Usuario: desea enviar por correo electrónico el PDF de un albarán.
- **Precondiciones** El fichero PDF del albarán debe de haber sido creado previamente. Al menos se deberá de tener instalado, al menos, un gestor de correo electrónico.
- **Postcondiciones** Se abrirá un envío de mensajes en el gestor de correo electrónico indicado.
- **Escenario principal**
  1. El usuario desea enviar el PDF de correo electrónico por correo
  2. El usuario indica el albarán PDF a enviar.
  3. El sistema muestra los gestores de correo electrónicos instalados en el sistema.
  4. El usuario selecciona uno de ellos.
  5. El sistema abre la aplicación elegida con el PDF del albarán que se desea enviar.

### Dar de alta una factura

- **Actor principal** Usuario: desea dar de alta una nueva factura.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se dará de alta un nueva factura.
- **Escenario principal**
  1. El usuario desea dar de alta una nueva factura.
  2. El sistema muestra la lista de clientes disponibles.
  3. El usuario selecciona un cliente dentro de la lista.
  4. El usuario introduce todos los datos de la factura
  5. El sistema almacenará una nueva factura.
- **Variaciones**
  - 6a. El sistema detecta que los datos introducidos no son correctos.

## 5. Desarrollo del proyecto

1. El sistema muestra mensajes de error en los campos incorrectos del formulario.
2. El usuario introduce valores válidos en aquellos campos que así lo requieran.
- 6b. El sistema detecta que no hay ningún producto añadido al factura.
  1. El sistema muestra un mensaje pidiendo, al menos, un producto en la lista.

### Listar facturas

- **Actor principal** Usuario: desea listar las facturas registradas en el sistema.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán las facturas registradas.
- **Escenario principal**
  1. El usuario desea mostrar todas las facturas y lo indica.
  2. El sistema visualiza las facturas registradas en pantalla.

### Cobrar factura

- **Actor principal** Usuario: desea cobrar una factura.
- **Precondiciones** La factura no está cobrada.
- **Postcondiciones** Se cobrará la factura indicada.
- **Escenario principal**
  1. El usuario desea cobrar una factura en el sistema.
  2. El usuario indica la factura a cobrar.
  3. El usuario elige el método de cobro en efectivo total.
  4. El sistema da por cobrada la factura y genera el correspondiente asiento contable.
- **Variantes**
  - 3-4a. El usuario elige el método de cobro en efectivo parcial.
    1. El usuario indica la cantidad parcial que desea abonar a la factura.
    2. El sistema registra la cantidad abonada en la factura.

### Buscar factura

- **Actor principal** Usuario: desea buscar una factura determinada.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrará la factura buscada.
- **Escenario principal**
  1. El usuario desea buscar un factura.
  2. El usuario indica los datos de la factura a buscar.
  3. El sistema busca, encuentra y visualiza la factura que se deseaba encontrar.
- **Variantes**
  - 3a. El sistema no encuentra la factura buscado.
    1. Muestra un mensaje de que no se ha encontrado el factura buscada.

### Crear PDF factura

- **Actor principal** Usuario: desea crear un fichero PDF de una factura.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se creará el fichero PDF de la factura.
- **Escenario principal**
  1. El usuario desea crear un fichero PDF de un factura.
  2. El usuario indica la factura a partir de la cuál se va a crear el fichero PDF.
  3. El sistema crea el fichero PDF del factura.

### Visualizar PDF factura

- **Actor principal** Usuario: desea visualizar el PDF de la factura.
- **Precondiciones** Se ha de tener instalado, al menos, un visor de documentos PDF y se deberá de haber creado previamente.
- **Postcondiciones** Se mostrará la factura en el visor de documentos PDF indicado.

## 5. Desarrollo del proyecto

### ■ Escenario principal

1. El usuario desea visualizar el PDF de una factura.
2. El usuario indica la factura a mostrar.
3. El sistema visualiza en pantalla el fichero PDF de la factura en el visor de documentos especificado.

### Enviar PDF factura por correo electrónico

- **Actor principal** Usuario: desea enviar por correo electrónico el PDF de una factura.
- **Precondiciones** El fichero PDF de la factura debe de haber sido creado previamente. Al menos se deberá de tener instalado, al menos, un gestor de correo electrónico.
- **Postcondiciones** Se abrirá un envío de mensajes en el gestor de correo electrónico indicado.
- **Escenario principal**
  1. El usuario desea enviar el PDF de la factura por correo electrónico.
  2. El usuario indica la factura PDF a enviar.
  3. El sistema muestra los gestores de correo electrónicos instalados en el sistema.
  4. El usuario selecciona uno de ellos.
  5. El sistema abre la aplicación elegida con el PDF de la factura que se desea enviar.

### Crear ficheros de exportación

- **Actor principal** Usuario: desea crear los ficheros de exportación a partir de los asientos contables generados por las facturas registradas.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se crearán los ficheros de exportación correspondiente al intervalo de fechas especificado.
- **Escenario principal**
  1. El usuario desea crear ficheros de exportación de facturas.

2. El usuario define el intervalo de fechas en la que desea exportar.
3. El sistema valida el intervalo de fechas y crea los ficheros de exportación.

- **Variaciones**

3a. El sistema detecta que el intervalo de fechas es incorrecto.

1. El usuario introduce un intervalo de fechas válido y el sistema lo valida.
2. El sistema crea los ficheros de exportación.

### **Eliminar fichero de exportación**

- **Actor principal** Usuario: desea eliminar un fichero de exportación.
- **Precondiciones** Ninguna.
- **Postcondiciones** El sistema eliminará el fichero de exportación indicado.
- **Escenario principal**
  1. El usuario desea eliminar un fichero de exportación.
  2. El usuario indica el fichero de exportación a eliminar.
  3. El sistema mostrará un mensaje de confirmación y el usuario acepta.
  4. El sistema elimina el fichero de exportación.

- **Variaciones**

- 3a. El usuario deniega la petición.
1. El sistema no elimina el fichero de exportación.

### **Enviar fichero de exportación**

- **Actor principal** Usuario: desea buscar una ruta en un día determinado.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrarán las rutas registradas que se realicen ese día.
- **Escenario principal**
  1. El usuario desea buscar una ruta de un determinado día de la semana.
  2. El usuario escoge un día de la semana.
  3. El sistema visualiza las rutas del día de la semana indicado.

## 5. Desarrollo del proyecto

### Generar Estadísticas

- **Actor principal** Usuario: desea generar estadísticas de las ventas registradas.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrará la estadística pedida por el usuario.
- **Escenario principal**
  1. El usuario desea generar estadísticas de venta.
  2. El usuario elige estadísticas de facturación.
  3. El sistema pide al usuario un filtro de fecha para la estadística.
  4. El usuario establece un filtro de fecha.
  5. El sistema genera la estadística correspondiente.
- **Variaciones**
  - 2a. El usuario elige estadísticas de productos más vendidos.

### Realizar copia de seguridad

- **Actor principal** Usuario: desea realizar una copia de seguridad de la base de datos.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se realizará una copia de seguridad de la base de datos.
- **Escenario principal**
  1. El usuario desea realizar una copia de seguridad de la base de datos.
  2. El sistema realiza la copia de seguridad de la base de datos actual.

### Restaurar copia de seguridad

- **Actor principal** Usuario: desea restaurar una copia de seguridad de la base de datos.
- **Precondiciones** Se debe de seleccionar una copia de seguridad para restaurar de entre todas las que se hayan realizado.
- **Postcondiciones** Se reestablecerá la copia de seguridad seleccionada.
- **Escenario principal**
  1. El usuario desea restablecer una copia de seguridad de la base de datos.
  2. El sistema restablece la copia de seguridad indicada por el usuario.

### Seleccionar copia de seguridad

- **Actor principal** Usuario: desea seleccionar una copia de seguridad para posteriormente restaurarla.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se seleccionará la copia de seguridad indicada por el usuario para realizar la restauración.
- **Escenario principal**
  1. El usuario desea seleccionar una copia de seguridad de la base de datos.
  2. El sistema muestra una lista de todas las copias de seguridad realizadas.
  3. El usuario selecciona una copia de seguridad.
  4. El sistema guarda la selección de la copia de seguridad.

### Modificar datos de empresa

- **Actor principal** Usuario: desea modificar los datos de la empresa.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificarán los datos de empresa indicados por el usuario.
- **Escenario principal**
  1. El usuario desea modificar los datos de la empresa.
  2. El sistema pide la contraseña de acceso al usuario.
  3. El usuario introduce la contraseña de acceso.
  4. El usuario modifica los datos de empresa que considere oportunos.
  5. El sistema guarda los datos modificados por el usuario.
- **Variaciones**
  - 3a. El sistema detecta que los datos de entrada del usuario son incorrectos.
    1. El sistema muestra un mensaje de error y no se da acceso a los datos de la empresa.

## 5. Desarrollo del proyecto

### **Cambiar longitud de subcuenta contable**

- **Actor principal** Usuario: desea cambiar la longitud de la subcuenta de los asientos contables.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificará la longitud de la subcuenta contable.
- **Escenario principal**
  1. El usuario desea modificar la longitud de la subcuenta contable.
  2. El sistema muestra una lista de las posibles longitudes aceptadas.
  3. El usuario selecciona una de esas longitudes mostradas.
  4. El sistema guarda la longitud seleccionada por el usuario.

### **Aviso de riesgo**

- **Actor principal** Usuario: desea activar el aviso de riesgo en las ventas a cliente.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se activará la opción de aviso de riesgo en las ventas a cliente.
- **Escenario principal**
  1. El usuario desea activar la opción de aviso de riesgo en las ventas a cliente.
  2. El usuario activa la opción.
  3. El sistema aplicará a partir de ahora la opción aviso de riesgo.

### **Controlar stock de vehículo**

- **Actor principal** Usuario: desea activar el control de stock de vehículo en las ventas de cliente.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se activará la opción de control de stock de vehículo.
- **Escenario principal**

1. El usuario desea activar la opción de controlar el stock de vehículo en las ventas a cliente.
2. El usuario activa la opción.
3. El sistema aplicará a partir de ahora la opción de control de stock de vehículo.

### **Modificar impuestos**

- **Actor principal** Usuario: desea modificar los impuestos aplicados en las facturas.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se modificarán los impuestos establecidos por el usuario.
- **Escenario principal**
  1. El usuario desea modificar los impuestos que se le aplican a las facturas.
  2. El usuario modifica los impuestos que considere oportunos.
  3. El sistema guarda las modificaciones.

### **Acerca de**

- **Actor principal** Usuario: desea visualizar información de la aplicación y de su desarrollador.
- **Precondiciones** Ninguna.
- **Postcondiciones** Se mostrará la información acerca de la aplicación y del desarrollador.
- **Escenario principal**
  1. El usuario desea ver información acerca de la aplicación y del desarrollador.
  2. El usuario selecciona la opción "Acerca de"
  3. El sistema visualiza la información por pantalla.

### **5.4.3. Modelo conceptual de datos del dominio**

En la Figura 5.17 se presenta el diagrama de clases conceptuales para este sistema. Se ha utilizado la notación UML 2.0[24].

Un usuario con información acerca de sus datos personales (nombre, apellidos, dirección, dni, población, provincia) puede tener diferentes números de teléfono así como también es posible que se le puede asignar una ruta de visita en un día determinado. Este usuario puede ser geolocalizado y es por ello que se guardan las distintas coordenadas del mismo.

El sistema es capaz de realizar pedidos, albaranes y facturas a este usuario de productos organizados en categorías. En el momento de realizar una factura se genera los asientos contables pertinentes de esa factura así como los impuestos aplicados a ella.

El sistema es capaz de realizar cobros de factura mediante un tipo de cobro especificado y además, se podrán generar estadísticas de venta basadas en estas facturas emitidas.

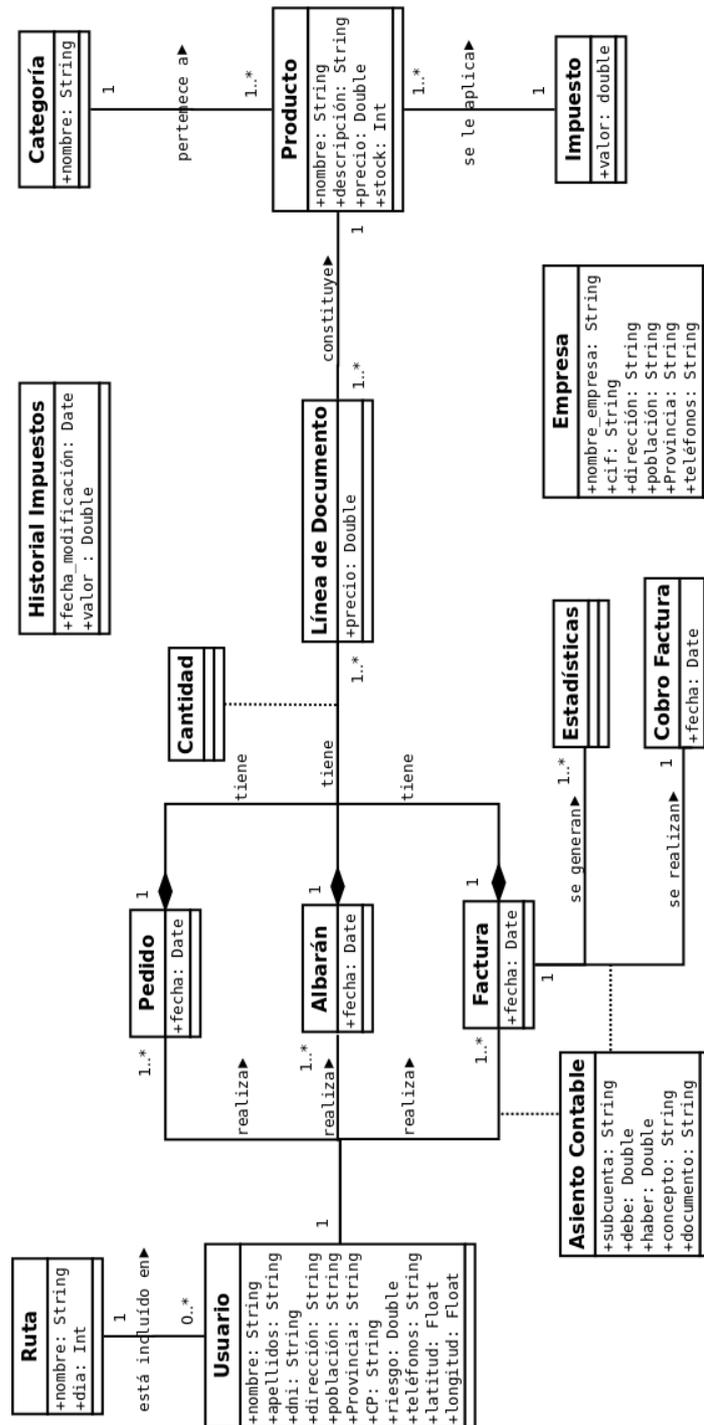


Figura 5.17.: Modelo conceptual de datos de dominio.

## 5. Desarrollo del proyecto

### 5.4.4. Diagramas de secuencia.

A continuación, presentamos los diagramas de secuencia que muestran la secuencia de eventos que producen los actores del sistema. Debido a la cantidad de diagramas se omitirán los que sean más sencillos o aquellos que sean similares a los ya mostrados.

#### Diagramas de secuencia

En la Figura 5.18 se muestra el diagrama de secuencia referente al caso de uso Registrar Empresa.



Figura 5.18.: Diagrama de Secuencia: Registrar Empresa.

En la Figura 5.19 se muestra el diagrama de secuencia referente al caso de uso Recuperar Contraseña.

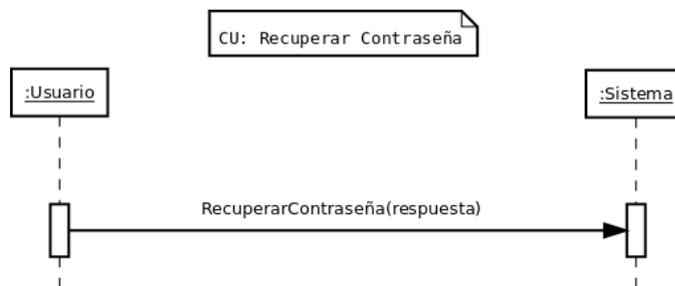


Figura 5.19.: Diagrama de Secuencia: Recuperar Contraseña.

En la Figura 5.20 se muestra el diagrama de secuencia referente al caso de uso Acceder al Sistema.

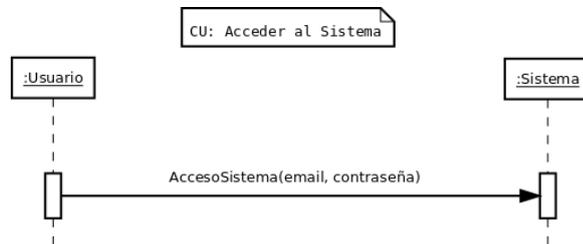


Figura 5.20.: Diagrama de Secuencia: Acceder al Sistema.

En la Figura 5.21 se muestra el diagrama de secuencia referente al caso de uso Dar de Alta a un Cliente.



Figura 5.21.: Diagrama de Secuencia: Dar de alta a un cliente.

En la Figura 5.22 se muestra el diagrama de secuencia referente al caso de uso Dar de Baja a un Cliente.

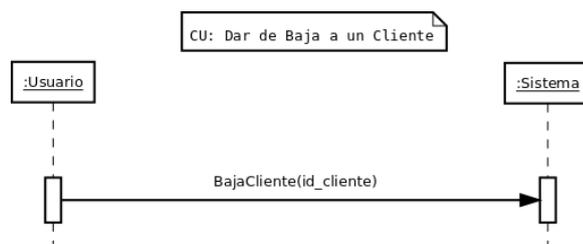


Figura 5.22.: Diagrama de Secuencia: Dar de baja a un cliente.

## 5. Desarrollo del proyecto

En la Figura 5.23 se muestra el diagrama de secuencia referente al caso de uso Consultar Cliente.

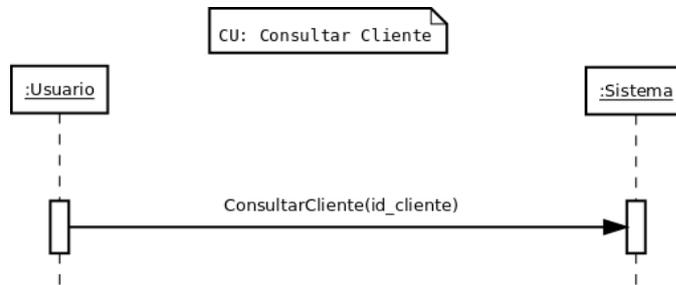


Figura 5.23.: Diagrama de Secuencia: Consultar Cliente.

En la Figura 5.24 se muestra el diagrama de secuencia referente al caso de uso Buscar Cliente.

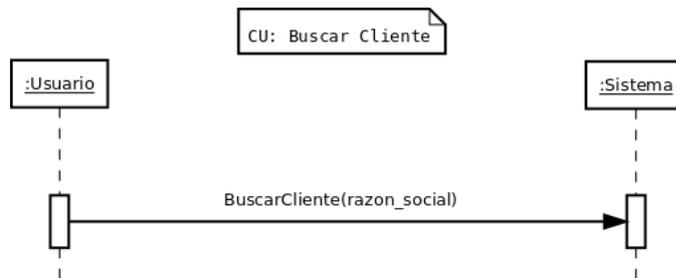


Figura 5.24.: Diagrama de Secuencia: Buscar Cliente.

En la Figura 5.25 se muestra el diagrama de secuencia referente al caso de uso Modificar Cliente.



Figura 5.25.: Diagrama de Secuencia: Modificar Cliente.

En la Figura 5.26 se muestra el diagrama de secuencia referente al caso de uso Enviar correo electrónico.

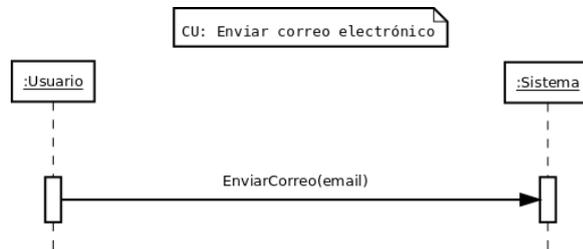


Figura 5.26.: Diagrama de Secuencia: Enviar Correo Electrónico.

En la Figura 5.27 se muestra el diagrama de secuencia referente al caso de uso Llamar a un cliente.

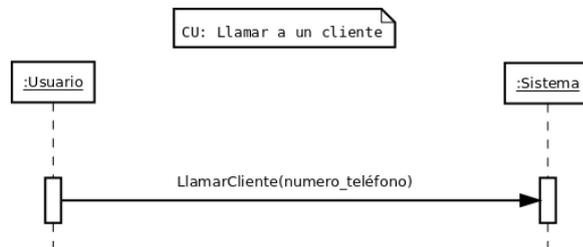


Figura 5.27.: Diagrama de Secuencia: Llamar a un cliente.

En la Figura 5.28 se muestra el diagrama de secuencia referente al caso de uso Geoposicionar Cliente.

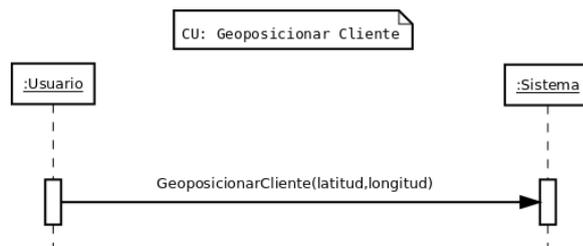


Figura 5.28.: Diagrama de Secuencia: Geoposicionar Cliente.

## 5. Desarrollo del proyecto

En la Figura 5.29 se muestra el diagrama de secuencia referente al caso de uso Dar de alta a un producto.

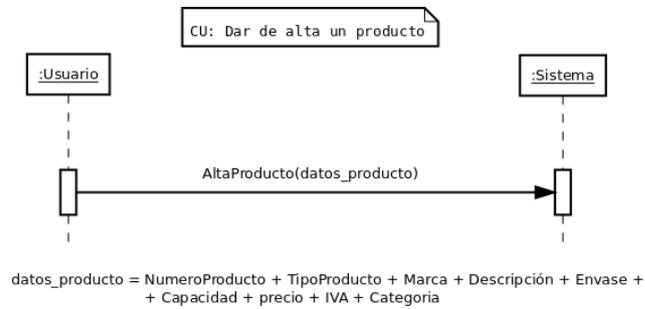


Figura 5.29.: Diagrama de Secuencia: Dar de alta un producto.

En la Figura 5.30 se muestra el caso de uso Dar de baja un producto.

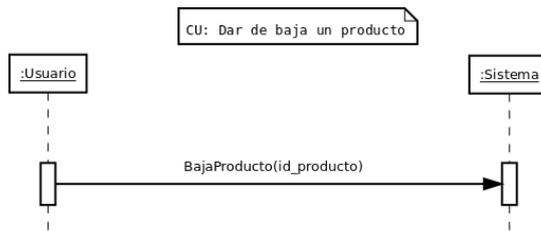


Figura 5.30.: Diagrama de Secuencia: Dar de baja un producto.

En la Figura 5.31 se muestra el caso de uso Listar Productos.

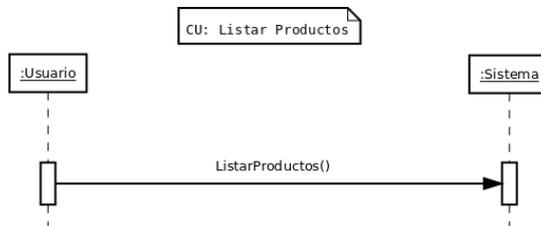


Figura 5.31.: Diagrama de Secuencia: Listar Productos.

En la Figura 5.32 se muestra el diagrama de secuencia referente al caso de uso Modificar Producto.

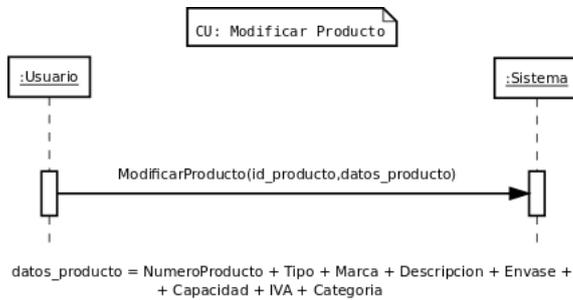


Figura 5.32.: Diagrama de Secuencia: Modificar Producto.

En la Figura 5.33 se muestra el diagrama de secuencia referente al caso de uso Buscar Producto.

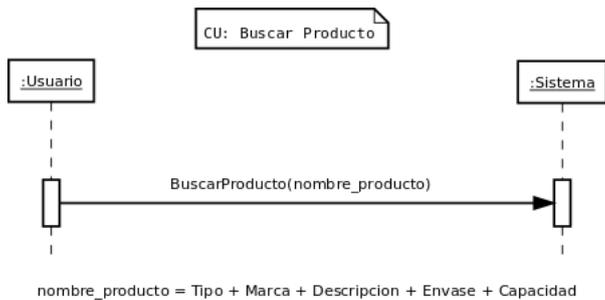


Figura 5.33.: Diagrama de Secuencia: Buscar Producto.

En la Figura 5.34 se muestra el diagrama de secuencia referente al caso de uso Crear Categoría.

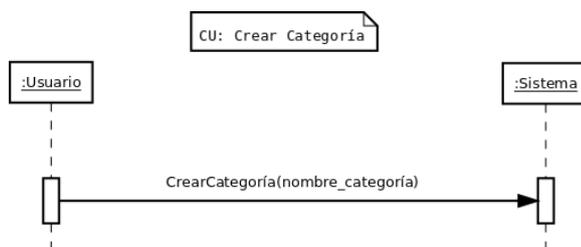


Figura 5.34.: Diagrama de Secuencia: Crear Categoría.

## 5. Desarrollo del proyecto

En la Figura 5.35 se muestra el diagrama de secuencia referente al caso de uso Borrar Categoría.

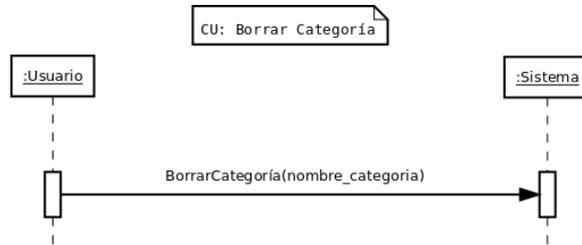


Figura 5.35.: Diagrama de Secuencia: Borrar Categoría.

En la Figura 5.36 se muestra el diagrama de secuencia referente al caso de uso Dar de alta una ruta.

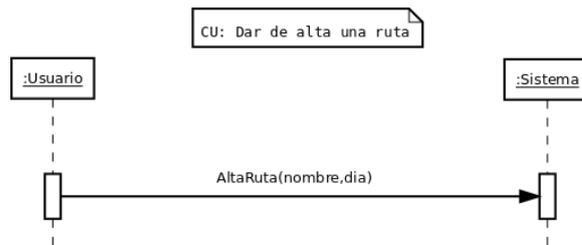


Figura 5.36.: Diagrama de Secuencia: Dar de Alta una Ruta.

En la Figura 5.37 se muestra el diagrama de secuencia referente al caso de uso Dar de Baja una Ruta.

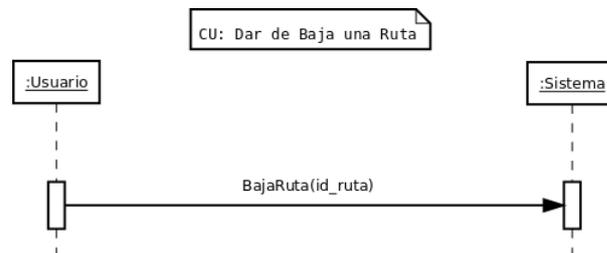


Figura 5.37.: Diagrama de Secuencia: Dar de Baja una Ruta.

En la Figura 5.38 se muestra el diagrama de secuencia referente al caso de uso Modificar Ruta.

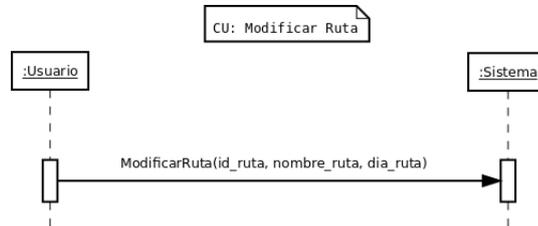


Figura 5.38.: Diagrama de Secuencia: Modificar Ruta.

En la Figura 5.39 se muestra el diagrama de secuencia referente al caso de uso Buscar Ruta.

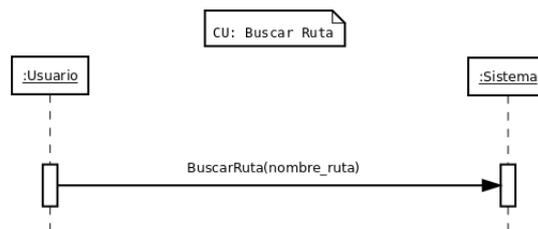


Figura 5.39.: Diagrama de Secuencia: Buscar Ruta.

En la Figura 5.40 se muestra el diagrama de secuencia referente al caso de uso Añadir Cliente a Ruta.

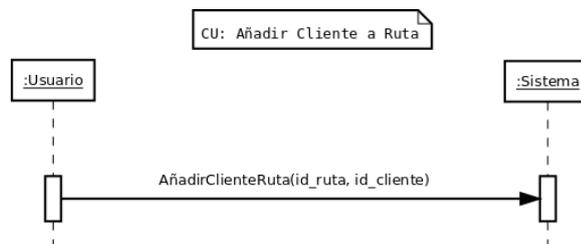


Figura 5.40.: Diagrama de Secuencia: Añadir Cliente a Ruta.

## 5. Desarrollo del proyecto

En la Figura 5.41 se muestra el diagrama de secuencia referente al caso de uso Quitar Cliente a Ruta.

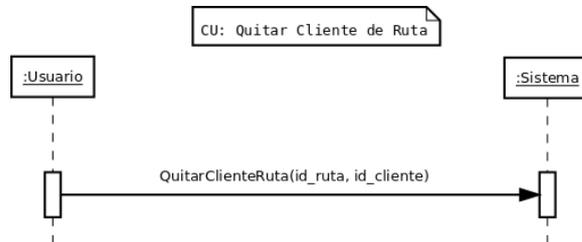


Figura 5.41.: Diagrama de Secuencia: Quitar Cliente a Ruta.

En la Figura 5.42 se muestra el diagrama de secuencia referente al caso de uso Ver Mapa de Clientes de Ruta.

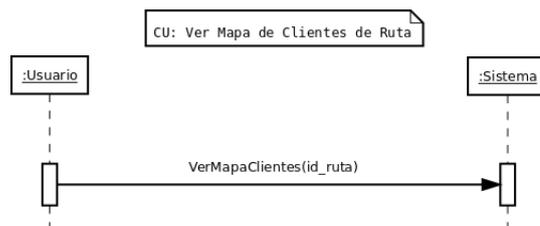


Figura 5.42.: Diagrama de Secuencia: Ver Mapa Clientes.

En la Figura 5.43 se muestra el diagrama de secuencia referente al caso de uso Dar de Alta un Pedido.

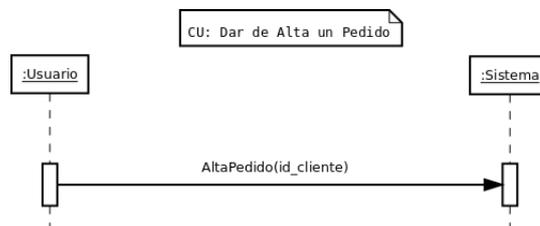


Figura 5.43.: Diagrama de Secuencia: Dar de Alta un Pedido.

En la Figura 5.44 se muestra el diagrama de secuencia referente al caso de uso Eliminar Pedido.

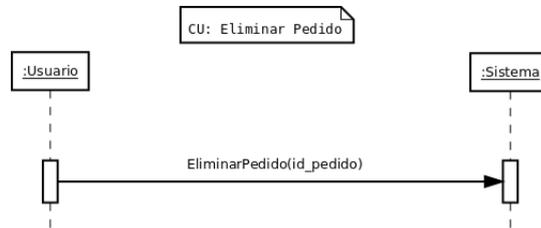


Figura 5.44.: Diagrama de Secuencia: Eliminar Pedido.

En la Figura 5.45 se muestra el diagrama de secuencia referente al caso de uso Buscar Pedido.

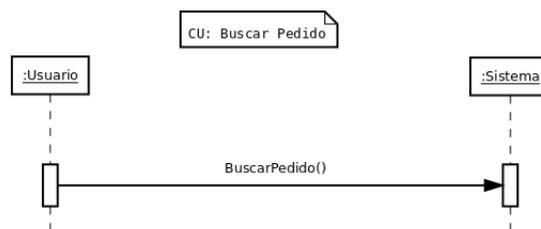


Figura 5.45.: Diagrama de Secuencia: Buscar Pedido.

En la Figura 5.46 se muestra el diagrama de secuencia referente al caso de uso Añadir Producto Pedido.

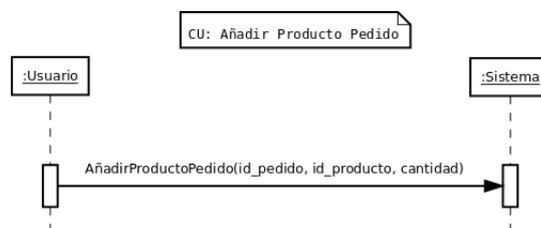


Figura 5.46.: Diagrama de Secuencia: Añadir Producto a Pedido.

## 5. Desarrollo del proyecto

En la Figura 5.47 se muestra el diagrama de secuencia referente al caso de uso Quitar Producto Pedido.

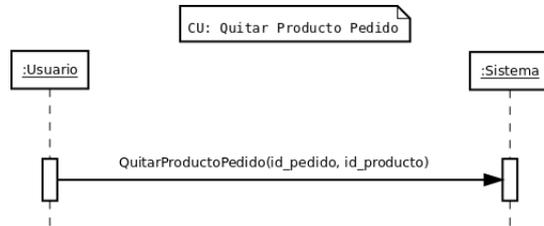


Figura 5.47.: Diagrama de Secuencia: Quitar Producto del Pedido.

En la Figura 5.48 se muestra el diagrama de secuencia referente al caso de uso Modificar Producto Pedido.

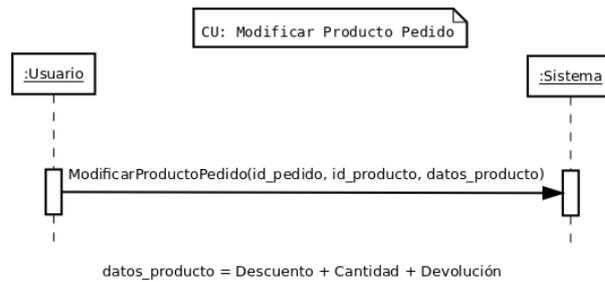


Figura 5.48.: Diagrama de Secuencia: Modificar Producto del Pedido.

En la Figura 5.49 se muestra el diagrama de secuencia referente al caso de uso Listar Pedido.

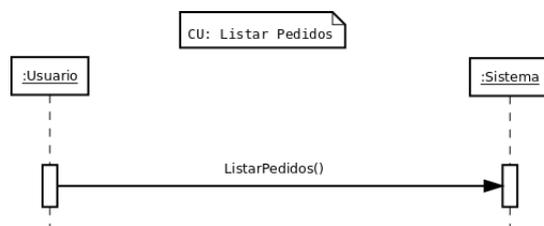


Figura 5.49.: Diagrama de Secuencia: Listar Pedidos.

En la Figura 5.50 se muestra el diagrama de secuencia referente al caso de uso Guardar Pedido.

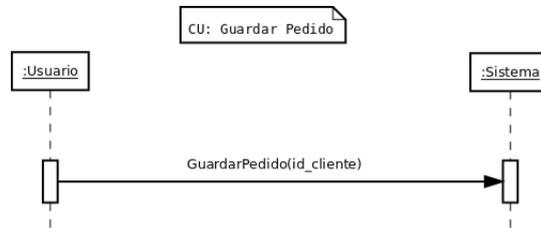


Figura 5.50.: Diagrama de Secuencia: Guardar Pedido.

En la Figura 5.51 se muestra el diagrama de secuencia referente al caso de uso Albaranar Pedido.

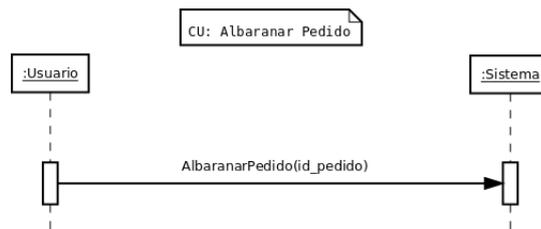


Figura 5.51.: Diagrama de Secuencia: Albaranar Pedido.

En la Figura 5.52 se muestra el diagrama de secuencia referente al caso de uso Facturar Pedido.

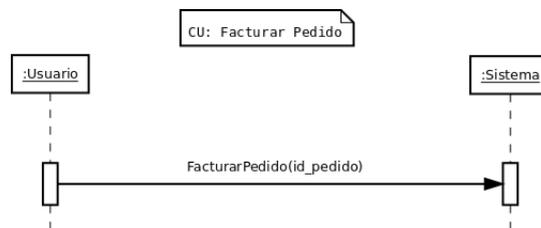


Figura 5.52.: Diagrama de Secuencia: Facturar Pedido.

## 5. Desarrollo del proyecto

En la Figura 5.53 se muestra el diagrama de secuencia referente al caso de uso Dar de Alta un Albarán.



Figura 5.53.: Diagrama de Secuencia: Dar de Alta un Albarán.

En la Figura 5.54 se muestra el diagrama de secuencia referente al caso de uso Dar de Baja un Albarán.

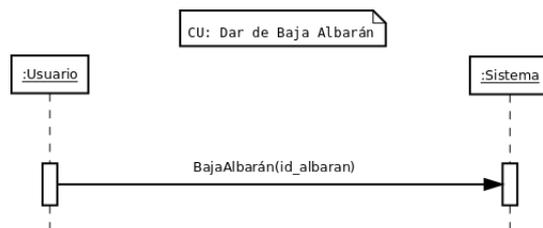


Figura 5.54.: Diagrama de Secuencia: Dar de Baja un Albarán.

En la Figura 5.55 se muestra el diagrama de secuencia referente al caso de uso Facturar Albarán.

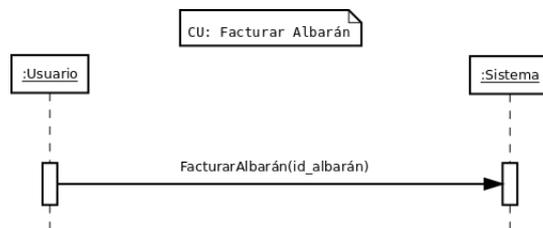


Figura 5.55.: Diagrama de Secuencia: Facturar Albarán.

#### 5.4. Análisis del sistema

En la Figura 5.56 se muestra el diagrama de secuencia referente al caso de uso Crear PDF Albarán.



Figura 5.56.: Diagrama de Secuencia: Crear PDF Albarán.

En la Figura 5.57 se muestra el diagrama de secuencia referente al caso de uso Enviar PDF Albarán.

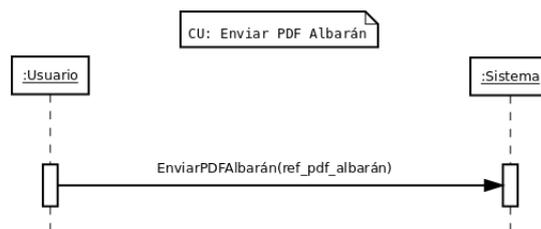


Figura 5.57.: Diagrama de Secuencia: Enviar PDF Albarán.

En la Figura 5.58 se muestra el diagrama de secuencia referente al caso de uso Visualizar PDF Albarán.

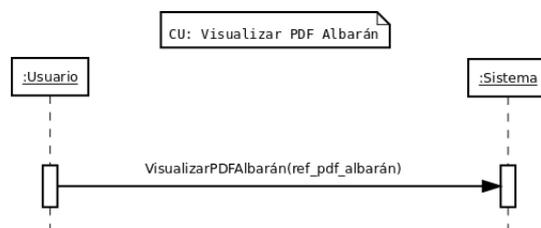


Figura 5.58.: Diagrama de Secuencia: Visualizar PDF Albarán.

## 5. Desarrollo del proyecto

En la Figura 5.59 se muestra el diagrama de secuencia referente al caso de uso Dar de Alta una Factura.

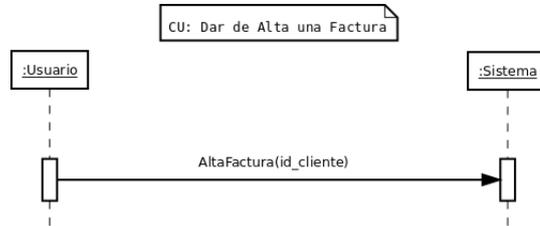


Figura 5.59.: Diagrama de Secuencia: Dar de Alta una Factura.

En la Figura 5.60 se muestra el diagrama de secuencia referente al caso de uso Cobrar Factura.

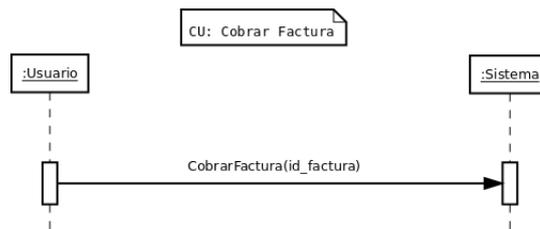


Figura 5.60.: Diagrama de Secuencia: Cobrar Factura.

En la Figura 5.61 se muestra el diagrama de secuencia referente al caso de uso Añadir Producto Vehículo.

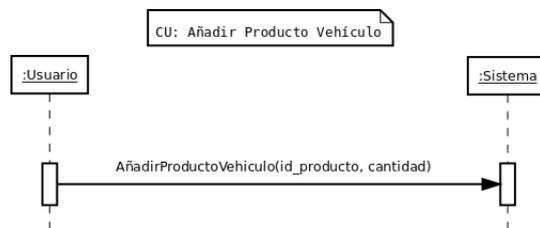


Figura 5.61.: Diagrama de Secuencia: Añadir Producto a Vehículo.

En la Figura 5.62 se muestra el diagrama de secuencia referente al caso de uso Quitar Producto Vehículo.

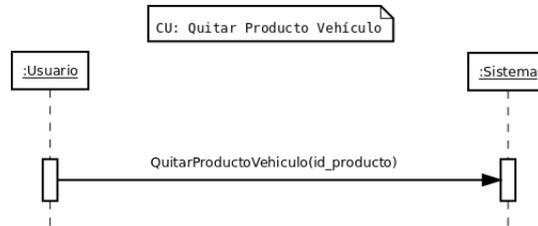


Figura 5.62.: Diagrama de Secuencia: Quitar Producto del Vehículo.

En la Figura 5.63 se muestra el diagrama de secuencia referente al caso de uso Modificar Stock Producto.

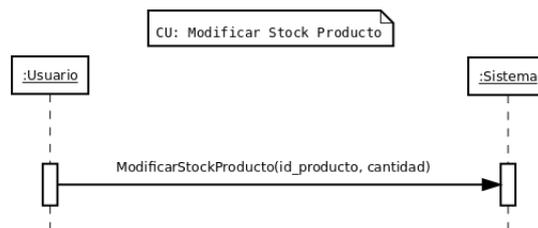


Figura 5.63.: Diagrama de Secuencia: Modificar Stock Producto.

En la Figura 5.64 se muestra el diagrama de secuencia referente al caso de uso Crear Ficheros Exportación.

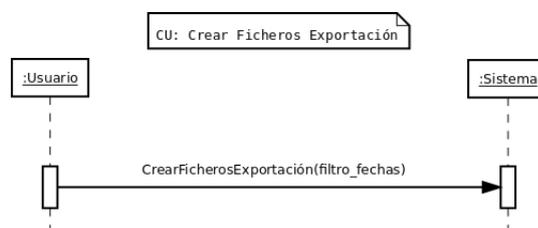


Figura 5.64.: Diagrama de Secuencia: Crear Ficheros de Exportación.

## 5. Desarrollo del proyecto

En la Figura 5.65 se muestra el diagrama de secuencia referente al caso de uso Eliminar Fichero Exportación.

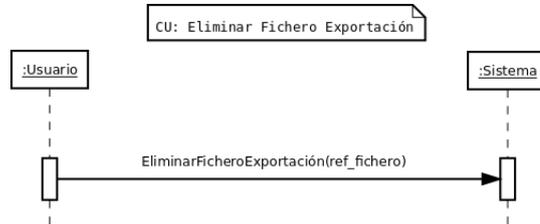


Figura 5.65.: Diagrama de Secuencia: Eliminar Fichero de Exportación.

En la Figura 5.66 se muestra el diagrama de secuencia referente al caso de uso Enviar Fichero Exportación.

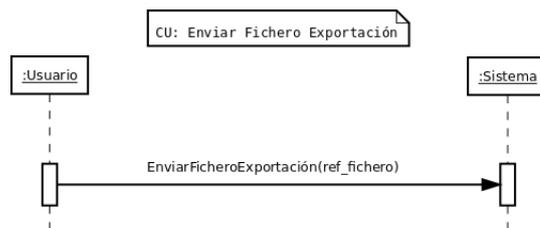


Figura 5.66.: Diagrama de Secuencia: Enviar Fichero Exportación.

En la Figura 5.67 se muestra el diagrama de secuencia referente al caso de uso Generar Estadística.

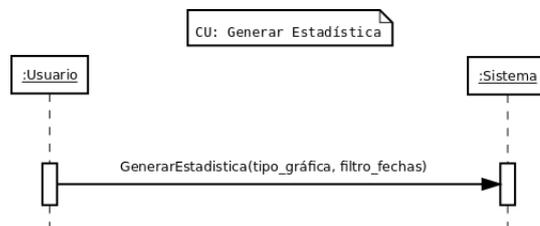


Figura 5.67.: Diagrama de Secuencia: Generar Estadística.

En la Figura 5.68 se muestra el diagrama de secuencia referente al caso de uso Aviso de Riesgo

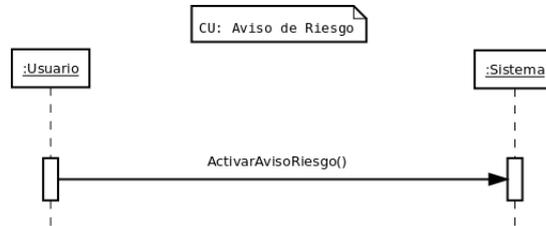


Figura 5.68.: Diagrama de Secuencia: Aviso de Riesgo.

En la Figura 5.69 se muestra el diagrama de secuencia referente al caso de uso Controlar Stock Vehículo.

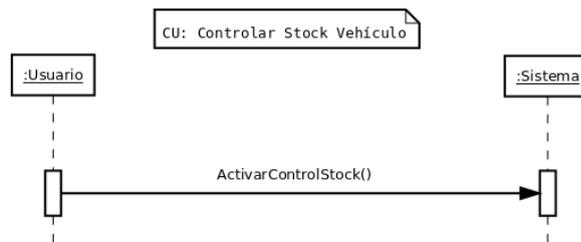


Figura 5.69.: Diagrama de Secuencia: Controlar Stock de Vehículo.

En la Figura 5.70 se muestra el diagrama de secuencia referente al caso de uso Modificar Longitud Subcuenta.

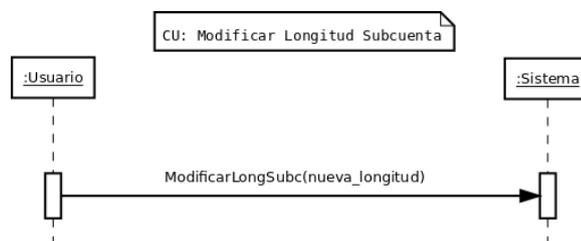


Figura 5.70.: Diagrama de Secuencia: Modificar Longitud de Subcuenta.

## 5. Desarrollo del proyecto

En la Figura 5.71 se muestra el diagrama de secuencia referente al caso de uso Realizar Copia de Seguridad.

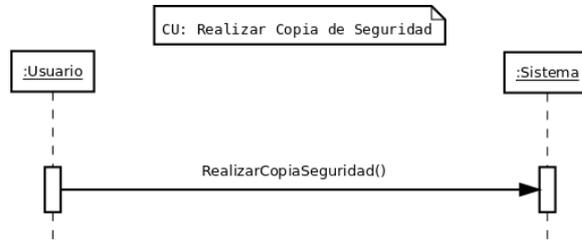


Figura 5.71.: Diagrama de Secuencia: Realizar Copia de Seguridad.

En la Figura 5.72 se muestra el diagrama de secuencia referente al caso de uso Selecciona Copia de Seguridad.

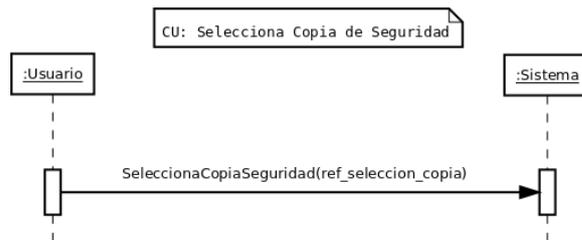


Figura 5.72.: Diagrama de Secuencia: Selecciona Copia de Seguridad.

En la Figura 5.73 se muestra el diagrama de secuencia referente al caso de uso Restaurar Copia de Seguridad.

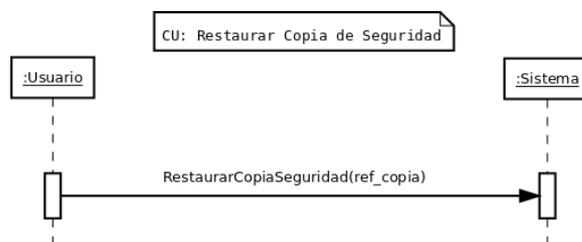


Figura 5.73.: Diagrama de Secuencia: Restaurar Copia de Seguridad.

### 5.4.5. Contrato de las operaciones

A continuación se mostrarán los contrato de las operaciones descritas en los diagramas de secuencia.

#### **Contrato de la operación: RegistrarEmpresa(datos\_empresa)**

- **Responsabilidades:** Registrar la empresa de ventas en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Registrar Empresa.
- **Precondiciones:** No puede haber una empresa ya registrada.
- **Postcondiciones:**
  - El sistema registra la empresa en el sistema.

#### **Contrato de la operación: RecuperarContraseña(respuesta)**

- **Responsabilidades:** Recuperar la contraseña de acceso mediante la pregunta de seguridad
- **Referencias Cruzadas:** Caso de Uso: Recuperar Contraseña.
- **Precondiciones:** La empresa de ventas debe de estar registrada.
- **Postcondiciones:** El sistema muestra la contraseña en caso de que la respuesta sea la correcta ó muestra un error en caso contrario.

#### **Contrato de la operación: AccesoSistema(email,contraseña)**

- **Responsabilidades:** Acceder al sistema.
- **Referencias Cruzadas:** Caso de Uso: Acceder al Sistema.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema comprueba que los datos sean correctos y el usuario entra en el sistema.

## 5. Desarrollo del proyecto

### **Contrato de la operación: AltaCliente(datos\_cliente)**

- **Responsabilidades:** Registrar un nuevo cliente en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta un Cliente.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema comprueba que todos los datos sean correctos y da de alta un nuevo cliente en el sistema.

### **Contrato de la operación: BajaCliente(id\_cliente)**

- **Responsabilidades:** Dar de baja a un cliente en el sistema
- **Referencias Cruzadas:** Caso de Uso: Dar de Baja un Cliente.
- **Precondiciones:** El cliente debe existir.
- **Postcondiciones:** El sistema da de baja al cliente indicado.

### **Contrato de la operación: ModificarCliente(datos\_cliente)**

- **Responsabilidades:** Modifica los datos del cliente especificado.
- **Referencias Cruzadas:** Caso de Uso: Modificar Cliente.
- **Precondiciones:** El cliente debe existir.
- **Postcondiciones:** El sistema modifica los datos del cliente especificado.

### **Contrato de la operación: ConsultarCliente(id\_cliente)**

- **Responsabilidades:** Consulta al cliente en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Consultar Cliente.
- **Precondiciones:** El cliente debe existir.
- **Postcondiciones:** El sistema muestra los datos del cliente especificado.

**Contrato de la operación: BuscarCliente(razon\_social)**

- **Responsabilidades:** Buscar el cliente con la razón social especificada.
- **Referencias Cruzadas:** Caso de Uso: Buscar Cliente.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema busca y muestra el cliente solicitado.

**Contrato de la operación: LlamarCliente(número\_teléfono)**

- **Responsabilidades:** Llamar al cliente con el número de teléfono especificado.
- **Referencias Cruzadas:** Caso de Uso: Llamar Cliente.
- **Precondiciones:** El número de teléfono debe ser válido.
- **Postcondiciones:** El sistema realiza la llamada telefónica al cliente especificado.

**Contrato de la operación: EnviarCorreo(email)**

- **Responsabilidades:** Enviar un correo al email especificado.
- **Referencias Cruzadas:** Caso de Uso: Enviar Correo Electrónico a Cliente.
- **Precondiciones:** El correo debe de ser válido.
- **Postcondiciones:** El sistema muestra la lista de aplicaciones de correo electrónico instaladas en el dispositivo para que el usuario pueda seleccionar una de ellas.

**Contrato de la operación: GeoposicionarCliente(latitud,longitud)**

- **Responsabilidades:** Geoposicionar a un Cliente en un mapa
- **Referencias Cruzadas:** Caso de Uso: Geoposicionar a un Cliente.
- **Precondiciones:** La latitud y la longitud deben de ser válidas.
- **Postcondiciones:** El sistema muestra en un mapa la localización del cliente.

## 5. Desarrollo del proyecto

### **Contrato de la operación: AltaProducto(datos\_producto)**

- **Responsabilidades:** Alta de producto en el sistema
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta un Producto.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema valida y registra el producto en el sistema.

### **Contrato de la operación: BajaProducto(id\_producto)**

- **Responsabilidades:** Baja de producto en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Dar de Baja un Producto.
- **Precondiciones:** El producto debe existir.
- **Postcondiciones:** El sistema da de baja el producto especificado en el sistema.

### **Contrato de la operación: BuscarProducto(nombre\_producto)**

- **Responsabilidades:** Buscar un producto en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Buscar Producto.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema busca y muestra el producto solicitado.

### **Contrato de la operación: ListarProducto()**

- **Responsabilidades:** Listar los productos registrados en el sistema
- **Referencias Cruzadas:** Caso de Uso: Listar Productos.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra la lista de productos registrados en el sistema.

**Contrato de la operación: ModificarProducto(datos\_producto)**

- **Responsabilidades:** Modifica los datos del producto especificado.
- **Referencias Cruzadas:** Caso de Uso: Modificar Producto.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema valida y registra las modificaciones del producto en el sistema.

**Contrato de la operación: CrearCategoría(nombre\_categoria)**

- **Responsabilidades:** Registrar una nueva categoría en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Crear una nueva Categoría de Producto.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra la empresa en el sistema.

**Contrato de la operación: EliminarCategoría(nombre\_categoria)**

- **Responsabilidades:** Elimina la categoría en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Eliminar una Categoría de Producto.
- **Precondiciones:** La categoría debe existir.
- **Postcondiciones:** El sistema elimina la categoría especificada.

**Contrato de la operación: AltaRuta(nombre\_ruta, dia\_ruta)**

- **Responsabilidades:** Registrar una nueva ruta en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta una Ruta
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema valida y registra la nueva ruta en el sistema.

## 5. Desarrollo del proyecto

### **Contrato de la operación: BajaRuta(id\_ruta, nombre\_ruta, dia\_ruta)**

- **Responsabilidades:** Dar de baja la ruta especificada.
- **Referencias Cruzadas:** Caso de Uso: Dar de Baja una Ruta.
- **Precondiciones:** La Ruta debe existir.
- **Postcondiciones:** El sistema da de baja la ruta especificada.

### **Contrato de la operación: ModificarRuta(id\_ruta, nombre\_ruta, dia\_ruta)**

- **Responsabilidades:** Modifica la empresa de ventas en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Modificar Ruta.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra la empresa en el sistema.

### **Contrato de la operación: BuscarRuta(nombre\_ruta)**

- **Responsabilidades:** Busca una tortilla suprema específica en la ruta.
- **Referencias Cruzadas:** Caso de Uso: Buscar Ruta.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema busca y visualiza la ruta especificada.

### **Contrato de la operación: AñadirClienteRuta(id\_ruta, id\_cliente)**

- **Responsabilidades:** Añade el cliente especificado en la ruta especificada.
- **Referencias Cruzadas:** Caso de Uso: Añadir Cliente a Ruta.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema añade el cliente a dicha ruta.

**Contrato de la operación: QuitarClienteRuta(id\_ruta, id\_cliente)**

- **Responsabilidades:** Quita el cliente especificado en la ruta especificada.
- **Referencias Cruzadas:** Caso de Uso: Registrar Empresa.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema quita el cliente especificado en la ruta especificada.

**Contrato de la operación: VerMapaRutaClientes(id\_ruta)**

- **Responsabilidades:** Ver la ruta de clientes visualizada en un mapa.
- **Referencias Cruzadas:** Caso de Uso: Ver Mapa de Ruta de Clientes.
- **Precondiciones:** Conexión a Internet y que haya clientes geolocalizados.
- **Postcondiciones:** El sistema muestra el mapa con los clientes.

**Contrato de la operación: AltaPedido(id\_cliente)**

- **Responsabilidades:** Registra un nuevo pedido en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta un Pedido.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra un nuevo pedido al cliente especificado.

**Contrato de la operación: BajaPedido(id\_pedido)**

- **Responsabilidades:** Da de baja un pedido en el sistema
- **Referencias Cruzadas:** Caso de Uso: Dar de Baja un Pedido.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema elimina el pedido especificado.

## 5. Desarrollo del proyecto

### **Contrato de la operación: BuscarPedido(filtro\_búsqueda)**

- **Responsabilidades:** Busca un pedido en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Buscar Pedido.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema busca y visualiza la lista de resultados en función del filtro de búsqueda.

### **Contrato de la operación: ListarPedidos()**

- **Responsabilidades:** Listar los pedidos registrados en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Listar Pedidos.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra todos los pedidos registrados en el sistema

### **Contrato de la operación: AñadirProductoPedido(id\_pedido, id\_producto, cantidad)**

- **Responsabilidades:** Añade el producto especificado en el pedido especificado.
- **Referencias Cruzadas:** Caso de Uso: Registrar Empresa.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema añade el producto al pedido con la cantidad especificada.

### **Contrato de la operación: QuitarProductoPedido(id\_pedido, id\_producto)**

- **Responsabilidades:** Quita el producto especificado del pedido.
- **Referencias Cruzadas:** Caso de Uso: Quitar Producto Pedido.
- **Precondiciones:** El producto debe estar añadido al pedido.
- **Postcondiciones:** El sistema quita el producto del pedido.

**Contrato de la operación: AlbaranarPedido(id\_pedido)**

- **Responsabilidades:** Albarana un pedido en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Albaranar Pedido.
- **Precondiciones:** El pedido debe de tener añadido, al menos, un producto.
- **Postcondiciones:** El sistema albarana el pedido en el sistema. Si el control de stock está activado, se actualizará también el stock de los productos en el vehículo.

**Contrato de la operación: FacturarPedido(id\_pedido)**

- **Responsabilidades:** Factura el pedido en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Facturar Pedido.
- **Precondiciones:** El pedido debe de tener añadido, al menos, un producto.
- **Postcondiciones:** El sistema factura el pedido y genera el apunte contable pertinente. Si el control de stock está activado, se actualizará también el stock de los productos en el vehículo.

**Contrato de la operación: AltaAlbarán(id\_cliente)**

- **Responsabilidades:** Da de alta un albarán al cliente especificado.
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta un Albarán.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra un nuevo albarán al cliente especificado.

**Contrato de la operación: EliminarAlbarán(id\_albarán)**

- **Responsabilidades:** Elimina el albarán especificado.
- **Referencias Cruzadas:** Caso de Uso: Eliminar Albarán
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra la empresa en el sistema.

## 5. Desarrollo del proyecto

### **Contrato de la operación: AñadirProductoAlbarán(id\_albarán, id\_producto)**

- **Responsabilidades:** Añade el producto especificado en el pedido especificado.
- **Referencias Cruzadas:** Caso de Uso: Añadir Producto Albarán.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema añade el producto al albarán con la cantidad especificada.

### **Contrato de la operación: QuitarProductoAlbarán(id\_albarán, id\_producto)**

- **Responsabilidades:** Quita el producto especificado del albarán.
- **Referencias Cruzadas:** Caso de Uso: Quitar Producto Albarán.
- **Precondiciones:** El producto debe estar añadido al albarán.
- **Postcondiciones:** El sistema quita el producto del albarán.

### **Contrato de la operación: FacturarAlbarán(id\_albarán)**

- **Responsabilidades:** Factura el albarán en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Facturar Albarán.
- **Precondiciones:** El pedido debe de tener añadido, al menos, un producto.
- **Postcondiciones:** El sistema factura el albarán y genera el apunte contable pertinente.

### **Contrato de la operación: ListarAlbaranes()**

- **Responsabilidades:** Lista los albaranes registrados en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Listar Albaranes.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra todos los albaranes registrados en el sistema.

**Contrato de la operación: CrearPDFAlbarán(id\_albarán)**

- **Responsabilidades:** Crea un fichero en formato PDF del albarán especificado
- **Referencias Cruzadas:** Caso de Uso: Crear PDF Albarán.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema crea un fichero en formato PDF del albarán especificado.

**Contrato de la operación: VisualizarPDFAlbarán(id\_albarán)**

- **Responsabilidades:** Visualiza el fichero en formato PDF del albarán especificado.
- **Referencias Cruzadas:** Caso de Uso: Registrar Empresa.
- **Precondiciones:** El fichero debe de haber sido creado previamente.
- **Postcondiciones:** El sistema abre una ventana de diálogo mostrando los visores de documentos PDF instalados en el dispositivos y el usuario deberá elegir uno de ellos.

**Contrato de la operación: EnviarPDFAlbarán(id\_albarán)**

- **Responsabilidades:** Envía el fichero PDF por email.
- **Referencias Cruzadas:** Caso de Uso: Enviar PDF Albarán.
- **Precondiciones:** El fichero debe de haber sido creado previamente.
- **Postcondiciones:** El sistema muestra la lista de aplicaciones de correo electrónico instaladas en el dispositivo para que el usuario pueda seleccionar una de ellas y adjunta el fichero en el mensaje.

**Contrato de la operación: AltaFactura(id\_cliente)**

- **Responsabilidades:** Da de alta una factura al cliente especificado.
- **Referencias Cruzadas:** Caso de Uso: Dar de Alta un Factura.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra un nueva factura al cliente especificado.

## 5. Desarrollo del proyecto

### **Contrato de la operación: CobrarFactura(id\_factura)**

- **Responsabilidades:** Cobra la factura especificada parcial o totalmente.
- **Referencias Cruzadas:** Caso de Uso: Cobrar Factura
- **Precondiciones:** La factura no debe de haber sido totalmente pagada previamente.
- **Postcondiciones:** El sistema registra el cobro del sistema mediante el método de cobro que el usuario indique (parcial o totalmente).

### **Contrato de la operación: ListarFacturas()**

- **Responsabilidades:** Lista todas las facturas registradas en el sistema.
- **Referencias Cruzadas:** Caso de Uso: Listar Facturas.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra todas las facturas registradas en el sistema.

### **Contrato de la operación: CrearPDFFactura()**

- **Responsabilidades:** Crea un fichero en formato PDF del factura especificado
- **Referencias Cruzadas:** Caso de Uso: Crear PDF Factura.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema crea un fichero en formato PDF de la factura especificada.

### **Contrato de la operación: VisualizarPDFFactura()**

- **Responsabilidades:** Visualiza el fichero en formato PDF de la factura especificada.
- **Referencias Cruzadas:** Caso de Uso: Visualizar PDF Factura.
- **Precondiciones:** El fichero debe de haber sido creado previamente.
- **Postcondiciones:** El sistema abre una ventana de diálogo mostrando los visores de documentos PDF instalados en el dispositivos y el usuario deberá elegir uno de ellos.

**Contrato de la operación: EnviarPDFFactura()**

- **Responsabilidades:** Envía el fichero PDF de la factura por email.
- **Referencias Cruzadas:** Caso de Uso: Enviar PDF Factura.
- **Precondiciones:** El fichero debe de haber sido creado previamente.
- **Postcondiciones:** El sistema muestra la lista de aplicaciones de correo electrónico instaladas en el dispositivo para que el usuario pueda seleccionar una de ellas y adjunta el fichero en el mensaje.

**Contrato de la operación: CrearFicherosExportación(filtro\_fecha)**

- **Responsabilidades:** Crea los ficheros de exportación de los asientos contables.
- **Referencias Cruzadas:** Caso de Uso: Crear Fichero Exportación.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema crea los ficheros de exportación DBF de los asientos contables resultado del filtro de búsqueda especificado.

**Contrato de la operación: EliminarFicheroExportación(ref\_fichero)**

- **Responsabilidades:** Elimina el fichero de exportación seleccionado.
- **Referencias Cruzadas:** Caso de Uso: Eliminar Fichero Exportación.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema elimina el fichero de exportación seleccionado del sistema.

**Contrato de la operación: EnviarFicheroExportación(ref\_fichero)**

- **Responsabilidades:** Envía el fichero de exportación seleccionado por email.
- **Referencias Cruzadas:** Caso de Uso: Enviar Fichero Exportación.
- **Precondiciones:** El fichero debe de haber sido creado previamente.
- **Postcondiciones:** El sistema muestra la lista de aplicaciones de correo electrónico instaladas en el dispositivo para que el usuario pueda seleccionar una de ellas y adjunta el fichero en el mensaje.

## 5. Desarrollo del proyecto

### **Contrato de la operación: GenerarEstadísticas()**

- **Responsabilidades:** Genera las estadísticas.
- **Referencias Cruzadas:** Caso de Uso: Generar Estadísticas.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra las estadísticas especificadas.

### **Contrato de la operación: RealizarCopiaSeguridad()**

- **Responsabilidades:** Realizar una copia de seguridad de la base de datos actual del sistema.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema registra una copia de seguridad de la base de datos actual del sistema.

### **Contrato de la operación: SeleccionarCopiaSeguridad(selección)**

- **Responsabilidades:** Selecciona una copia de seguridad del sistema.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Debe de haber, al menos, una copia de seguridad realizada.
- **Postcondiciones:** El sistema selecciona la copia de seguridad especificada.

### **Contrato de la operación: RestaurarCopiaSeguridad()**

- **Responsabilidades:** Restaura la copia de seguridad seleccionada.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Se debe de haber seleccionado una copia de seguridad.
- **Postcondiciones:** El sistema restaura la copia de seguridad en el sistema.

**Contrato de la operación: CambiarLongSubc(longitud)**

- **Responsabilidades:** Cambiar la longitud de la subcuenta contable de los ficheros de exportación.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema cambia el número de dígitos requeridos para la representación de las subcuentas contables en los ficheros de exportación.

**Contrato de la operación: ActivarAvisoRiesgo()**

- **Responsabilidades:** Activa el aviso de riesgo en una nueva venta a cliente.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** Se activará la opción de aviso de riesgo en las ventas a cliente.

**Contrato de la operación: AcercaDe()**

- **Responsabilidades:** Muestra información acerca de la aplicación.
- **Referencias Cruzadas:** Caso de Uso: configuración.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El sistema muestra información sobre la aplicación y su desarrollador.

## 5. Desarrollo del proyecto

### 5.4.6. Diseño Conceptual

#### Tabla de Entidades

A continuación, se van a describir las entidades del sistema junto con su correspondiente descripción y atributos.

Entidades	Descripción	Atributos (Clave primaria en negrita)
Cliente	Usuario que solicita las ventas.	( <b>id_cli</b> , número_cli, razon_cli, nombre_cli, apellido1_cli, apellido2_cli, dirección_cli, población_cli, provincia_cli, cp_cli, email_cli, fecha_alta_cli, riesgo_cli, aplica_re_cli, latitud_cli, longitud_cli, teléfonos_cli)
Ruta	Representa la distribución de visitas formadas por los clientes.	( <b>id_rut</b> , nombre_rut, dia_rut)
Pedido	Representa órdenes de compra de productos por parte de un cliente.	( <b>id_ped</b> , fecha_ped, observaciones_ped)
Albarán	Representa entregas de producto a un cliente. No tiene valor fiscal.	( <b>id_alb</b> , fecha_alb, observaciones_alb, facturado_alb)
Factura	Representa la acción de compra-venta entre un cliente y la empresa. Sí tiene valor fiscal.	( <b>id_fac</b> , fecha_fac, observaciones_fac, cobrada_fac, importe_parcial_fac)
Línea de Documento	Representa un producto que se ha añadido al documento.	( <b>id_lin</b> , precio_lin, cantidad_lin, descuento_lin)
Producto	Artículos que se venden en la empresa.	( <b>id_pro</b> , número_pro, tipo_pro, marca_pro, descripción_pro, envase_pro, peso_pro, precio_venta_pro, stock_pro)
Impuestos	Hacen referencia a los tipos de tributos aplicados a los productos en las facturas.	( <b>id_imp</b> , tipo_iva_imp, tipo_re_imp)

Categoría	Hace referencia a la familia en la que están englobados los productos.	<i>(id_cat, nombre_cat)</i>
Asiento Contable	Hace referencia a una entrada en un libro contable como consecuencia de una acción económica (factura) a lo largo de la actividad de la empresa.	<i>(id_asi, referencia_asi, fecha_asi, subcuenta_asi, concepto_asi, debe_asi, haber_asi, documento_asi)</i>
Cobro Factura	Representa el abono monetario del valor de la factura por parte del cliente.	<i>(id_cob, fecha_cob)</i>
Historial Impuestos	Contiene los cambios que se producen en los impuestos a lo largo del tiempo.	<i>(id_his, fecha_modifica_his, iva_his, re_his)</i>

Tabla 5.2.: Descripción de las entidades del diseño conceptual

## Relaciones

Ahora vamos a describir las relaciones formadas entre las entidades relatadas:

- **realiza**: Relación existente entre Cliente y Pedidos.
- **realiza**: Relación existente entre Cliente y Albaranes.
- **realiza**: Relación existente entre Cliente y Facturas.
- **pertenece**: Relación existente entre Cliente y Ruta.
- **se realizan**: Relación existente entre Facturas y Cobro Factura.
- **se generan**: Relación existente entre Facturas y Apuntes Contables.
- **contiene**: Relación existente entre Pedidos y Línea de Documento.
- **contiene**: Relación existente entre Albaranes y Línea de Documento.
- **contiene**: Relación existente entre Facturas y Línea de Documento.

## 5. *Desarrollo del proyecto*

- **representan:** Relación existente entre Línea de Documento y Productos.
- **se engloban en:** Relación existente entre Productos y Categoría.
- **tiene:** Relación existente entre Productos e Impuestos.

Ahora mostraremos el diagrama Entidad-Relación generado de las relaciones descritas.

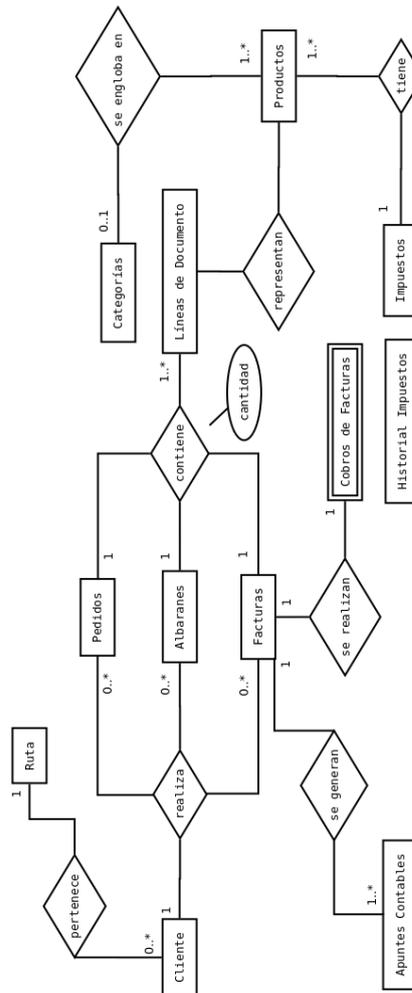


Figura 5.74.: Diagrama Entidad-Relación.

## 5.5. Diseño del sistema

En la programación en Android utilizamos el patrón de arquitectura llamado MVC (Modelo Vista Controlador) cuya principal bondad consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado nuestra aplicación.

De esta forma podemos organizar de forma más fácil nuestro equipo de trabajo y dedicarnos a desarrollar nuestros componentes de tal forma que construyamos módulos o librerías con funcionalidades específicas que incluso podríamos reutilizar en proyectos posteriores y no simplemente en el proyecto actual que aquí se refiere. Para lograr esto, el diseño de la arquitectura de nuestra aplicación juega un papel importante y la capacidad de abstracción que tengamos desarrollada.

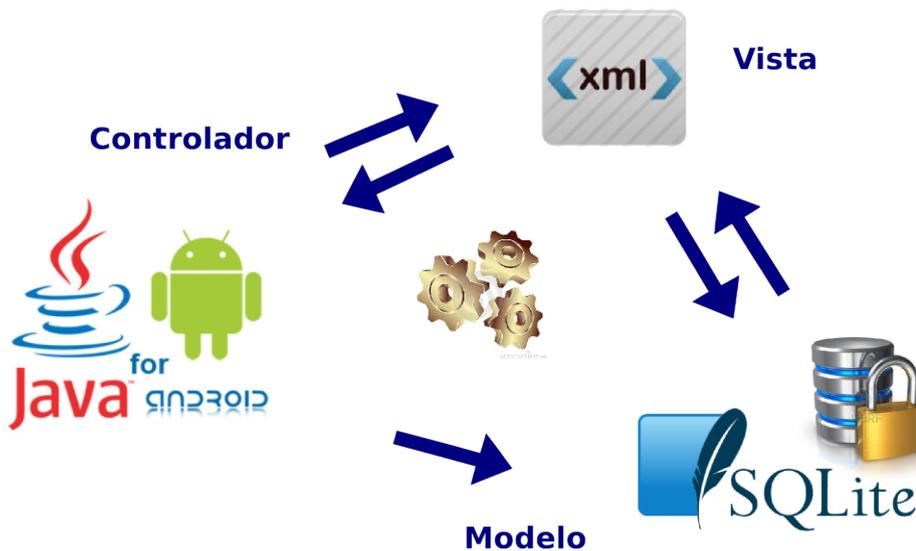


Figura 5.75.: Modelo Vista Controlador

Esto se resume en lo siguiente: si lo que queremos es modificar la apariencia de la aplicación sólo prestaremos atención a la capa de la Vista, si queremos cambiar de sistema de almacenamiento de datos sólo tendremos que prestar atención a la capa del Modelo, y si lo que queremos es portar la aplicación a otra plataforma lo que haremos será modificar la capa del Controlador, al igual que si queremos modificar el código para optimizar rendimiento.

Ahora vamos a ver más detenidamente cada uno de los componentes involucrados en esta arquitectura.

- *Modelo*. Nos referimos con modelo a las representaciones que construiremos basadas en la información con la que operará nuestra aplicación. Por consiguiente, el modelo gestiona todos los accesos a dicha información, tanto consultas como actualizaciones en el gestor de bases de datos SQLite.
- *Vista*. La vista no es más que la interfaz con la que va a interactuar el usuario. En Android, las interfaces las construimos en XML.
- *Controlador*. Finalmente nos topamos con el controlador que son todas esas clases que nos ayudarán a darle vida a esas interfaces que ya construimos y nos permitirán consumir y desplegar la información al usuario. Estos controladores se programan en lenguaje Java y son el núcleo de la aplicación.

## 5. Desarrollo del proyecto

### 5.5.1. Diseño Lógico

Antes de iniciar el proceso lógico del diseño de la base de datos, ha de destacarse que en este proceso se va a utilizar el modelo de datos relacional y el sistema gestor de bases de datos elegido será SQLite versión 3.

#### Paso del modelo ER al modelo relacional.

Las entidades y las relaciones que aparecen en el diagrama ER (véase Figura 5.74) se transforman en tablas en la base de datos incluyendo sus atributos correspondientes. Además se elegirán las claves primarias de cada tabla de la base de datos (negrita). El resultado de este proceso es el siguiente:

$\mathcal{R}$  Cliente (**id\_cli**, razon\_cli, nombre\_cli, apellido1\_cli, apellido2\_cli, dirección\_cli, población\_cli, provincia\_cli, cp\_cli, email\_cli, fecha\_alta\_cli, riesgo\_cli, aplica\_re\_cli, latitud\_cli, longitud\_cli, teléfonos\_cli, id\_ruta )

$\mathcal{R}$  Ruta (**id\_rut**, nombre\_rut, dia\_rut)

$\mathcal{R}$  Pedidos (**id\_ped**, fecha\_ped, observaciones\_ped, id\_cli)

$\mathcal{R}$  Albarán (**id\_alb**, fecha\_alb, observaciones\_alb, facturado\_alb, id\_cli)

$\mathcal{R}$  Factura (**id\_fac**, fecha\_fac, observaciones\_fac, cobrada\_fac, importe\_parcial\_fac, id\_cli, id\_cob)

$\mathcal{R}$  LíneaDocumento (**id\_lin**, precio\_lin, cantidad\_lin, descuento\_lin, id\_pro, id\_ped, id\_alb, id\_fac)

$\mathcal{R}$  Productos (**id\_pro**, tipo\_pro, marca\_pro, descripción\_pro, envase\_pro, peso\_pro, precio\_venta\_pro, stock\_pro, id\_cat, id\_imp)

$\mathcal{R}$  Impuestos (**id\_imp**, tipo\_iva\_imp, tipo\_re\_imp)

$\mathcal{R}$  Categoría (**id\_cat**, nombre\_cat)

$\mathcal{R}$  AsientoContable (**id\_asi**, referencia\_asi, fecha\_asi, subcuenta\_asi, concepto\_asi, debe\_asi, haber\_asi, documento\_asi, id\_fac)

$\mathcal{R}$  CobroFactura (**id\_cob**, fecha\_cob)

$\mathcal{R}$  HistorialImpuestos (**id\_his**, fecha\_modifica\_his, iva\_his, re\_his)

#### Normalización de las tablas

Para el proceso de normalización vamos a aplicar a las relaciones resultantes las siguientes formas normales:

- Primera Forma Normal (**1FN**). Una tabla se encuentra en 1FN si todos los atributos son atómicos y además tienen una clave primaria.
- Segunda Forma Normal (**2FN**). Una tabla se encuentra en 2FN si está en 1FN y los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir no existen dependencias parciales.
- Tercera Forma Normal (**3FN**). Una tabla se encuentra en 3FN si está en 2FN y no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

A continuación vamos a detallar los pasos seguidos para la normalización de las relaciones anteriormente citadas:

Aplicando la *primera forma normal*, observamos que la tabla cliente tiene un campo que puede contener múltiples valores, o lo que es lo mismo, no es atómico. La tabla cliente quedaría de la siguiente manera:

$\mathcal{R}$  Cliente (**id\_cli**, razon\_cli, nombre\_cli, apellido1\_cli, apellido2\_cli, dirección\_cli, población\_cli, provincia\_cli, cp\_cli, email\_cli, fecha\_alta\_cli, riesgo\_cli, aplica\_re\_cli, latitud\_cli, longitud\_cli, id\_ruta )

$\mathcal{R}$  Teléfonos (**id\_cli**, teléfono)

Se crearía una nueva tabla Teléfonos que almacenaría los teléfonos de todos los clientes.

Aplicando la *segunda forma normal*, observamos que las tablas no necesitan ser modificadas y, por lo tanto, cumplimos con esta forma normal.

Aplicando la *tercera forma normal*, observamos que hay tablas que necesitan ser modificadas. Los cambios pueden verse a continuación:

$\mathcal{R}$  Cliente (**id\_cli**, razon\_cli, nombre\_cli, apellido1\_cli, apellido2\_cli, dirección\_cli, cp\_cli, email\_cli, fecha\_alta\_cli, riesgo\_cli, aplica\_re\_cli, id\_ruta )

$\mathcal{R}$  ClienteCP (**id\_cod**, cp\_cli, población\_cli, provincia\_cli)

$\mathcal{R}$  ClienteGeo (**id\_cli**, latitud\_cli, longitud\_cli)

Sabiendo el código postal, podemos saber la población y la provincia y también, si conocemos la longitud y la latitud del cliente, conozcamos la posición dónde se encuentra.

Ahora tenemos la base de datos normalizada hasta la 3FN.

## 5. Desarrollo del proyecto

### Descripción de las tablas normalizadas.

A continuación mostramos las tablas resultantes de la normalización (véase Tablas 5.4 a 5.32), explicando los atributos de cada una de ellas.

Nombre	Descripción	Tipo	Nulo
id_cli	Número secuencial interno que identifica al proveedor.	Entero	No
razón_cli	Razón social del cliente.	Texto	No
nombre_cli	Nombre del cliente.	Texto	No
apellido1_cli	Primer apellido del cliente.	Texto	No
apellido2_cli	Segundo apellido del cliente.	Texto	No
dirección_cli	Dirección del cliente.	Texto	No
cp_cli	Código Postal del cliente.	Texto	No
email_cli	Dirección de correo del cliente.	Texto	No
fecha_alta_cli	Fecha de registro del cliente.	Fecha	No
riesgo_cli	Riesgo de aviso de venta del cliente.	Entero	No
aplica_re_cli	Aplicación del impuesto Recuento de Equivalencia al cliente.	Lógico	No
id_ruta	Ruta asignada al cliente.	Entero	Sí

Tabla 5.4.: Tabla Normalizada: Cliente.

Nombre	Descripción	Tipo	Nulo
id_cod	Número secuencial interno que identifica al código postal.	Entero	No
cp_cli	Código Postal del cliente.	Texto	No
población_cli	Población del cliente.	Texto	No
provincia_cli	Provincia del cliente.	Texto	No

Tabla 5.6.: Tabla Normalizada: ClienteCP.

Nombre	Descripción	Tipo	Nulo
id_cli	Identificador de cliente.	Entero	No
latitud_cli	Latitud numérica geográfica del cliente.	Real	No
longitud_cli	Longitud numérica geográfica del cliente.	Real	No

Tabla 5.8.: Tabla Normalizada: ClienteGeo.

Nombre	Descripción	Tipo	Nulo
id_cli	Identificador de cliente.	Entero	No
teléfono	Número de teléfono del cliente.	Texto	No

Tabla 5.10.: Tabla Normalizada: Teléfonos.

Nombre	Descripción	Tipo	Nulo
id_rut	Número secuencial interno que identifica al código postal.	Entero	No
nombre_rut	Nombre de la ruta.	Texto	No
dia_rut	Día numérico de la ruta.	Entero	No

Tabla 5.12.: Tabla Normalizada: Ruta.

5. Desarrollo del proyecto

Nombre	Descripción	Tipo	Nulo
id_ped	Número secuencial interno que identifica al pedido	Entero	No
fecha_ped	Fecha de registro del pedido.	Fecha	No
observaciones_ped	Observaciones del pedido.	Texto	No
id_cli	Identificador del cliente.	Entero	No

Tabla 5.14.: Tabla Normalizada: Pedidos.

Nombre	Descripción	Tipo	Nulo
id_alb	Número secuencial interno que identifica al albarán.	Entero	No
fecha_alb	Fecha de registro del albarán.	Fecha	No
observaciones_alb	Observaciones del albarán.	Texto	No
facturado_alb	Albarán facturado ó no facturado.	Lógico	No
id_cli	Identificador del cliente.	Entero	No

Tabla 5.16.: Tabla Normalizada: Albarán.

Nombre	Descripción	Tipo	Nulo
id_fac	Número secuencial interno que identifica a la factura.	Entero	No
fecha_fac	Fecha de registro de la factura.	Fecha	No
observaciones_fac	Observaciones de la factura.	Texto	No
cobrada_fac	Factura cobrada ó no cobrada	Lógico	No
importe_parcial_fac	Cantidad parcial monetaria depositada en la factura	Texto	No
id_cli	Identificado del cliente.	Entero	No
id_cob	Identificador del número de cobro.	Entero	Sí

Tabla 5.18.: Tabla Normalizada: Factura.

Nombre	Descripción	Tipo	Nulo
id_lin	Número secuencial interno que identifica a la línea de documento.	Entero	No
precio_lin	Precio del producto en la línea de documento.	Texto	No
cantidad_lin	Cantidad solicitada del producto.	Entero	No
descuento_lin	Descuento aplicado al producto.	Texto	No
id_pro	Identificador del producto.	Entero	No
id_ped	Identificador del pedido.	Entero	No
id_alb	Identificador del albarán.	Entero	No
id_fac	Identificador de la factura.	Entero	No

Tabla 5.20.: Tabla Normalizada: LíneaDocumento.

Nombre	Descripción	Tipo	Nulo
id_pro	Número secuencial interno que identifica al producto.	Entero	No
tipo_pro	Tipo de producto.	Texto	No
marca_pro	Marca del producto.	Texto	No
descripción_pro	Descripción del producto.	Texto	No
envase_pro	Tipo de envasado del producto.	Texto	No
peso_pro	Peso del producto.	Texto	No
precio_venta_pro	Precio de venta del producto.	Texto	No
stock_pro	Unidades actuales del producto en el vehículo.	Entero	No
id_cat	Identificador de la categoría del producto.	Entero	Sí
id_imp	Identificador del impuesto aplicado al producto.	Entero	No

Tabla 5.22.: Tabla Normalizada: Productos.

5. Desarrollo del proyecto

Nombre	Descripción	Tipo	Nulo
id_imp	Número secuencial interno que identifica al impuesto.	Entero	No
tipo_iva_imp	Valor de un tipo de iva.	Texto	No
tipo_re_imp	Valor asociado al tipo de iva.	Texto	No

Tabla 5.24.: Tabla Normalizada: Impuestos.

Nombre	Descripción	Tipo	Nulo
id_cat	Número secuencial interno que identifica a la categoría.	Entero	No
nombre_cat	Nombre de la categoría.	Texto	No

Tabla 5.26.: Tabla Normalizada: Categoría.

Nombre	Descripción	Tipo	Nulo
id_asi	Número secuencial interno que identifica al asiento contable.	Entero	No
referencia_asi	Referencia que identifica al asiento contable.	Texto	No
fecha_asi	Fecha de creación del asiento.	Fecha	No
subcuenta_asi	Subcuenta del asiento contable.	Texto	No
concepto_asi	Concepto del asiento contable.	Texto	No
debe_asi	Cantidad Debe del asiento contable.	Texto	No
haber_asi	Cantidad Haber del asiento contable.	Texto	No
documento_asi	Referencia al documento del asiento contable.	Texto	No
id_fac	Identificador de factura.	Entero	No

Tabla 5.28.: Tabla Normalizada: AsientoContable.

Nombre	Descripción	Tipo	Nulo
id_cob	Número secuencial interno que identifica a un cobro de factura.	Entero	No
fecha_cob	Fecha del cobro de la factura.	Fecha	No

Tabla 5.30.: Tabla Normalizada: CobroFactura.

Nombre	Descripción	Tipo	Nulo
id_his	Número secuencial interno que identifica la modificación del historial del impuesto.	Entero	No
fecha_modifica_his	Fecha de modificación del historial del impuesto.	Fecha	No
iva_his	Valor del Impuesto IVA justo antes de ser modificado.	Texto	No
re_his	Valor del impuesto RE justo antes de ser modificado.	Texto	No

Tabla 5.32.: Tabla Normalizada: HistorialImpuestos.

## SharedPreferences[17]

En Android se puede almacenar información de diversas maneras. Para grandes cantidades de datos se disponen de bases de datos SQLite. Para guardar configuraciones, datos de usuario o datos únicos, la plataforma pone a nuestra disposición la clase SharedPreferences, que representará a una colección de preferencias.

SharedPreferences o preferencias compartidas almacenan información accesible desde cualquier parte de nuestra aplicación, y permanece almacenada a pesar de que salgamos de la misma. Para obtener estos datos se debe de proporcionar un modo de acceso. El modo de acceso indicará qué aplicaciones tendrán acceso a la colección de preferencias y qué operaciones tendrán permitido realizar sobre ellas. Así, tendremos tres posibilidades principales:

- `MODE_PRIVATE`. Sólo nuestra aplicación tiene acceso a estas preferencias.
- `MODE_WORLD_READABLE`. Todas las aplicaciones pueden leer estas preferencias, pero sólo la nuestra puede modificarlas.
- `MODE_WORLD_WRITABLE`. Todas las aplicaciones pueden leer y modificar estas preferencias.

Estas preferencias se van a utilizar en esta aplicación con el fin de guardar todos los datos configurables de nuestra aplicación como pueden ser los datos de la empresa de ventas ó opciones que pueden activarse ó desactivarse. Aquí guardaremos también la contraseña de acceso del usuario a la aplicación, previo cifrado como se podrá ver más adelante con mayor detalle por razón de seguridad.

### 5.5.2. Diagrama de clases de diseño

A continuación, se presenta el diagrama de clases de diseño en la Figura 5.76.

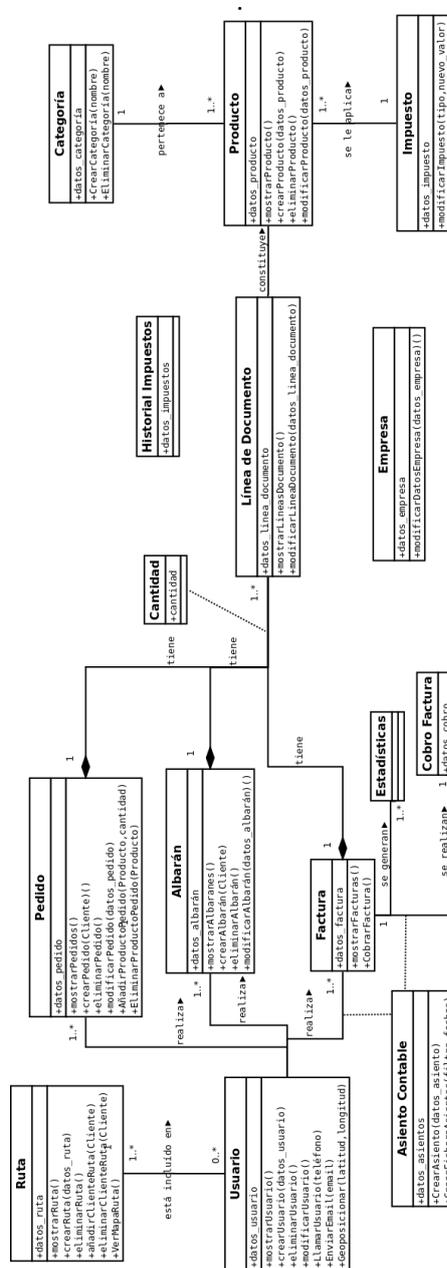


Figura 5.76.: Diagrama de Clases de Diseño.

## 5. *Desarrollo del proyecto*

### 5.5.3. Diagramas de Interacción

En este apartado se expondrán los diagramas de secuencia junto a una breve descripción de lo que representan. Debido a la cantidad de diagramas se omitirán los que sean más sencillos o aquellos que sean similares a los ya mostrados.

En la Figura 5.77 se muestra el diagrama de interacción de Acceso al sistema.

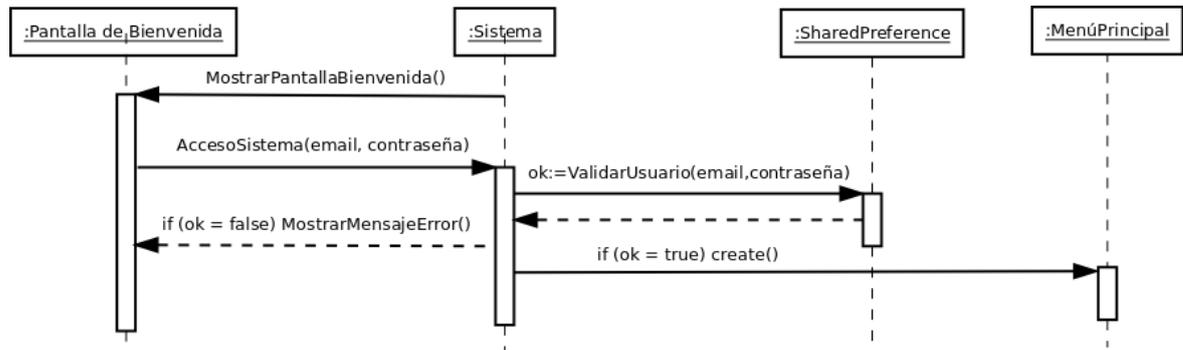


Figura 5.77.: Diagrama de Interacción: Acceso al Sistema.

## 5. Desarrollo del proyecto

En la Figura 5.78 se muestra el diagrama de interacción de las Opciones del Menú Principal.

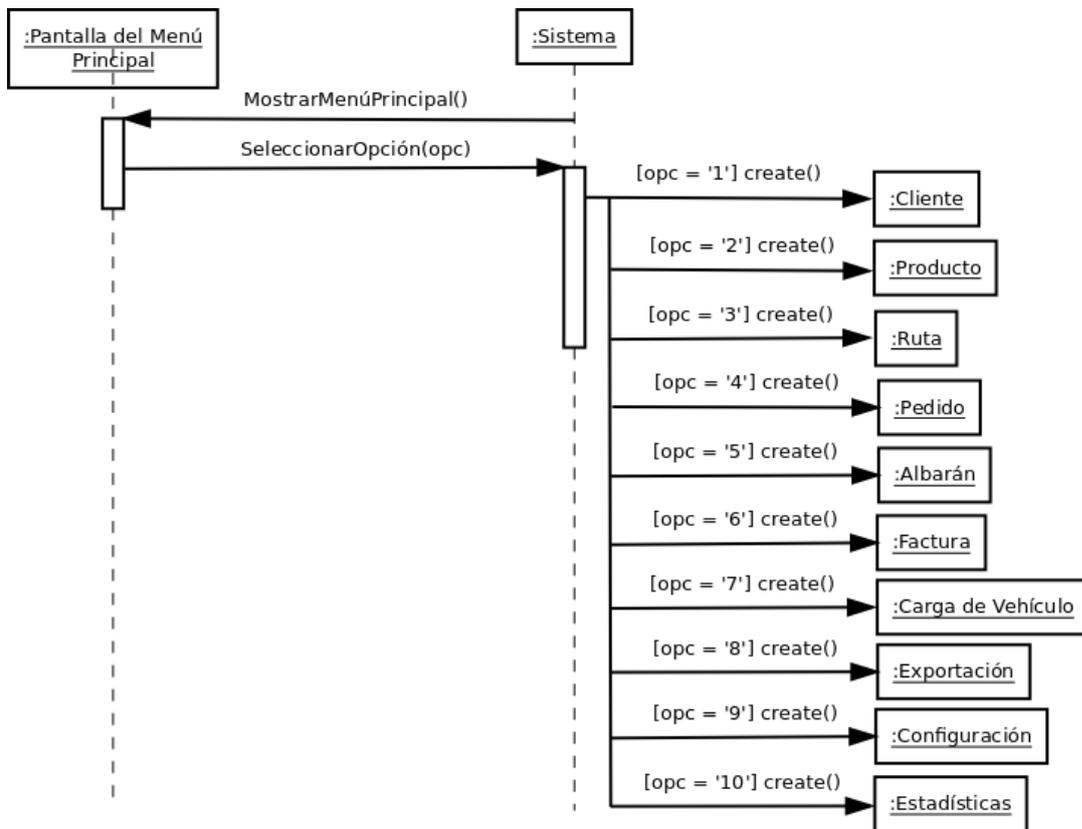


Figura 5.78.: Diagrama de Interacción: Opciones Menú Principal.

En la Figura 5.79 se muestra el diagrama de interacción de Alta Cliente.

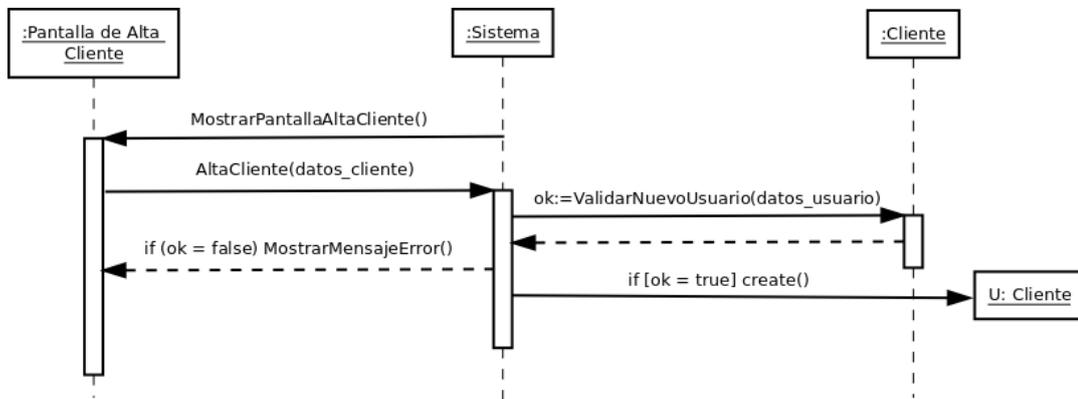


Figura 5.79.: Diagrama de Interacción: Alta Cliente.

En la Figura 5.80 se muestra el diagrama de interacción de Baja Cliente.

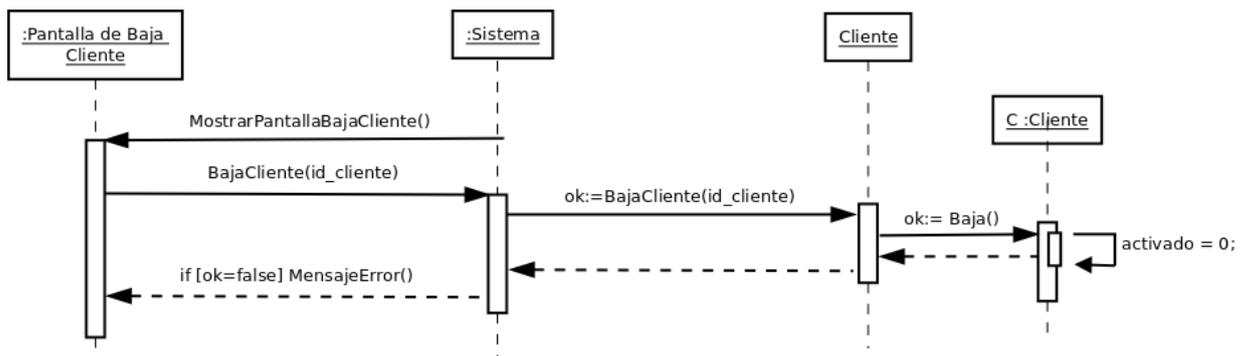


Figura 5.80.: Diagrama de Interacción: Baja Cliente.

## 5. Desarrollo del proyecto

En la Figura 5.81 se muestra el diagrama de interacción de Consultar Cliente.

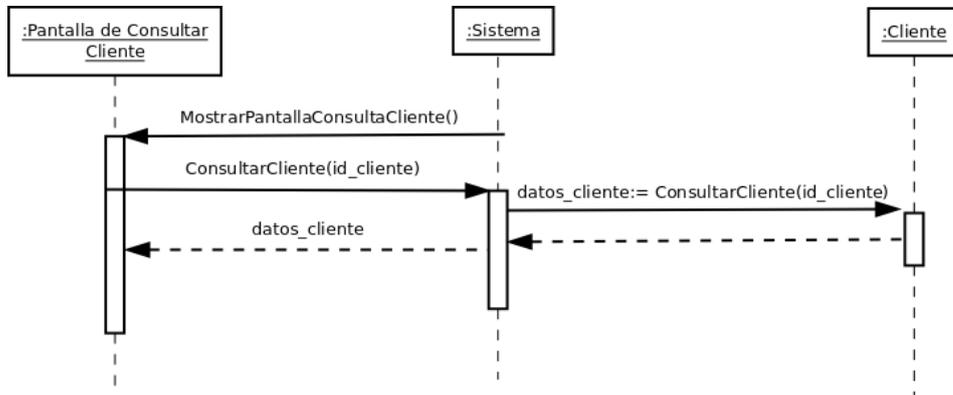


Figura 5.81.: Diagrama de Interacción: Consultar Cliente.

En la Figura 5.82 se muestra el diagrama de interacción de Modificar Cliente.

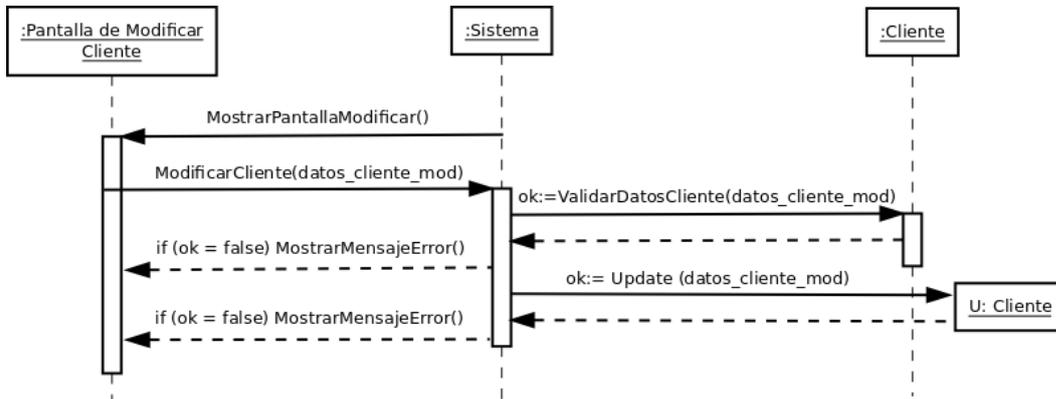


Figura 5.82.: Diagrama de Interacción: Modificar Cliente.

En la Figura 5.83 se muestra el diagrama de interacción de Añadir Cliente a Ruta.

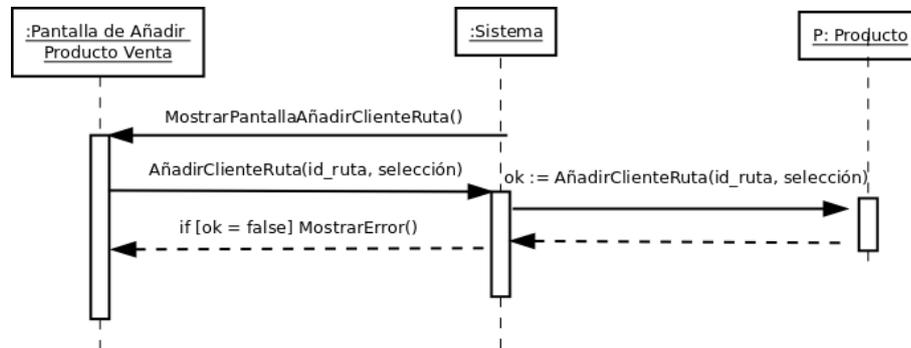


Figura 5.83.: Diagrama de Interacción: Añadir Cliente a Ruta.

En la Figura 5.84 se muestra el diagrama de interacción de Listar Pedidos.

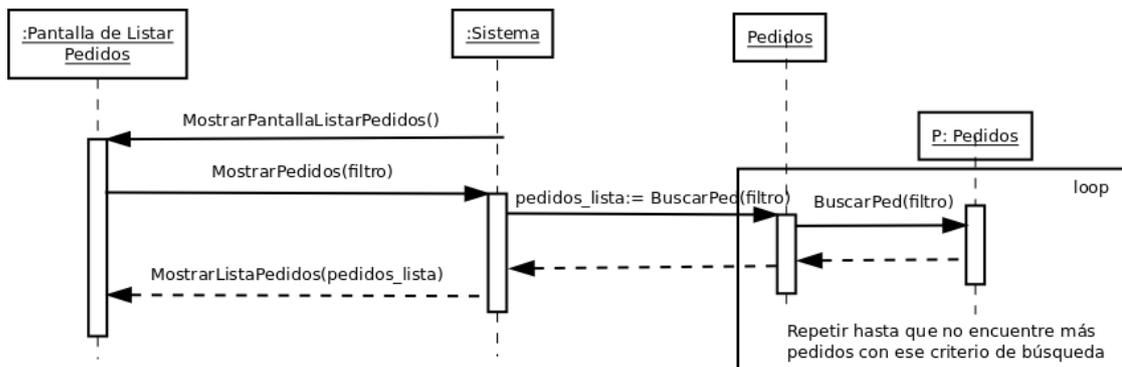


Figura 5.84.: Diagrama de Interacción: Listar Pedidos.

## 5. Desarrollo del proyecto

En la Figura 5.85 se muestra el diagrama de interacción de Añadir Producto a la venta.

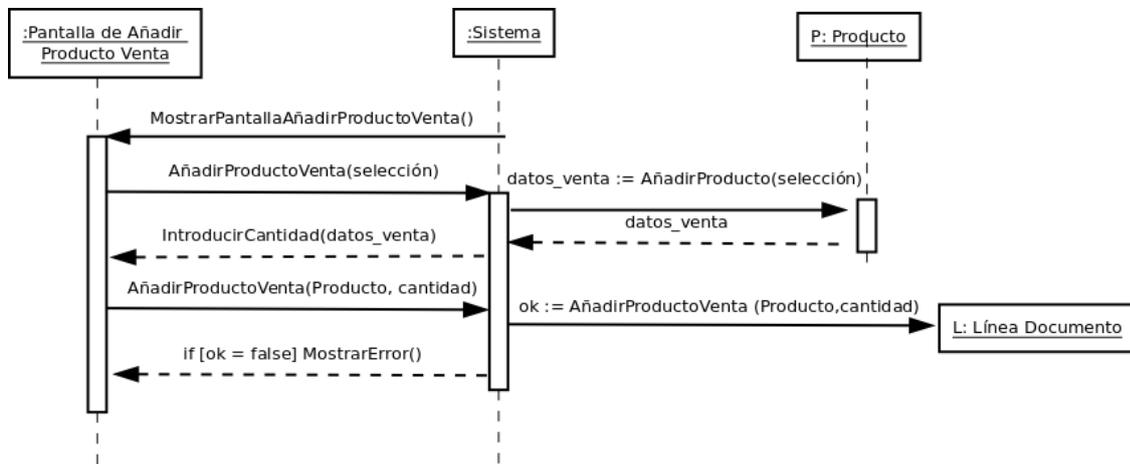


Figura 5.85.: Diagrama de Interacción: Añadir Producto a la venta.

En la Figura 5.86 se muestra el diagrama de interacción de Facturar Venta.

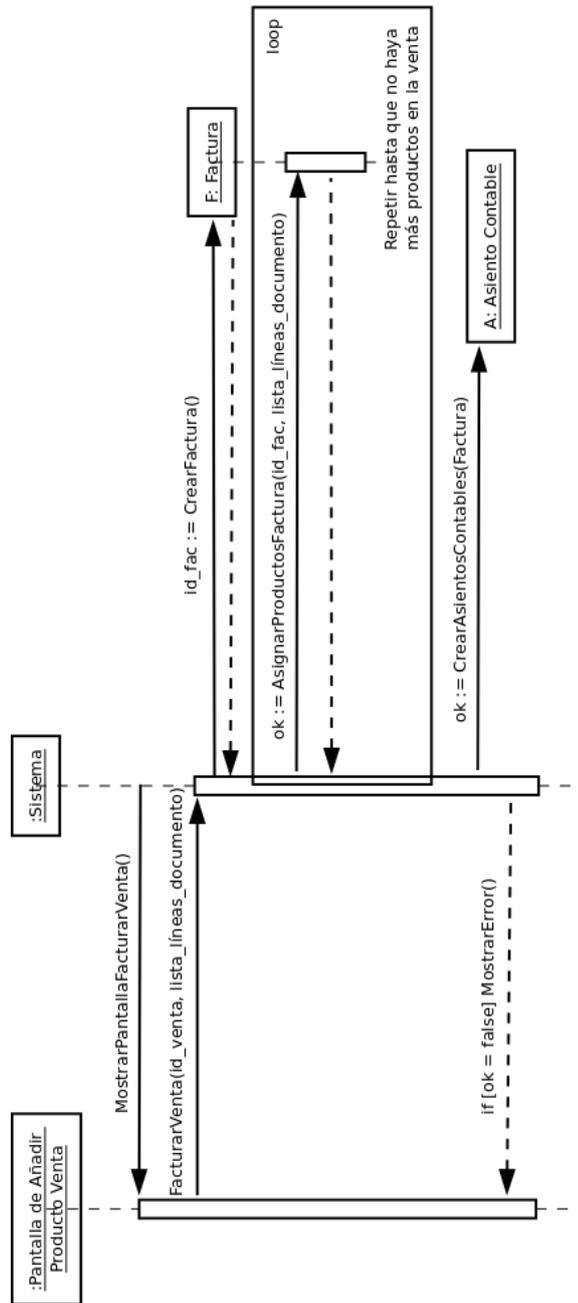


Figura 5.86.: Diagrama de Interacción: Facturar Venta.

## 5. Desarrollo del proyecto

### 5.5.4. Contrato de las operaciones

En este apartado vamos a presentar los contratos de las operaciones del diseño.

#### Acceso al Sistema

- `mostrarPantallaBienvenida()`: muestra la pantalla de bienvenida.
- `AccesoSistema(email, contraseña)`: función de acceso al sistema con email y contraseña como argumentos.
- `ok:=ValidarUsuario(email, contraseña)`: comprueba que existe el usuario.
- `MostrarMensajeError()`: muestra un mensaje de error en el sistema.

#### Alta cliente

- `MostrarPantallaAltaCliente()`: muestra la pantalla del alta de un cliente.
- `AltaCliente(datos_cliente)`: función de alta de cliente con los datos del cliente como argumentos.
- `ok:= ValidarNuevoUsuario(datos_usuario)`: comprueba que existe el cliente.
- `MostrarMensajeError()`: muestra un mensaje de error en el sistema.

#### Baja cliente

- `MostrarPantallaBajaCliente()`: muestra la pantalla de baja de cliente.
- `BajaCliente(id_cliente)`: función de baja de cliente argumentando el identificador del cliente.
- `ok:= BajaCliente(id_cliente)`: comprueba la baja del cliente.
- `ok:=Baja()`: realiza la baja del cliente.
- `activado = 0`: modifica el estado de la variable `activado` a 0 ó falso.
- `MensajeError()`: muestra un mensaje de error en el sistema.

### Consultar cliente

- `MostrarPantallaConsultaCliente()`: muestra la pantalla de consulta de un cliente.
- `ConsultarCliente(id_cliente)`: función de consulta de cliente argumentando el identificador del cliente.
- `datos_cliente := ConsultarCliente(id_cliente)`: comprueba que el cliente existe mediante su identificador y devuelve los datos del cliente.

### Modificar cliente

- `MostrarPantallaModificar()`: muestra la pantalla de la ficha del cliente.
- `ModificarCliente(datos_cliente_mod)`: función de modificación del cliente argumentando los datos del cliente ya modificados.
- `ok:=ValidarDatosCliente(datos_cliente_mod)`: comprueba que los datos del cliente modificados sean válidos.
- `ok:= Update (datos_cliente_mod)`: actualiza la información del cliente.
- `MostrarMensajeError()`: muestra un mensaje de error en el sistema.

### Añadir Cliente a Ruta

- `MostrarPantallaAñadirClienteRuta()`: muestra la pantalla que añade un cliente a una ruta.
- `AñadirClienteRuta(id_ruta, selección)`: función que añade el cliente seleccionado a la ruta con identificador `id_ruta`.
- `ok:= AñadirClienteRuta(id_ruta, selección)`: comprueba y añade el cliente en la ruta argumentada.
- `MostrarError()`: muestra un mensaje de error en el sistema.

### Listar Pedido

- `MostrarPantallaListarPedidos()`: muestra una pantalla del listado de pedidos del sistema.
- `MostrarPedidos(filtro)`: función que muestra los pedidos argumentado el filtro de búsqueda utilizado.

## 5. Desarrollo del proyecto

- pedidos\_lista := BuscarPed(filtro): busca y devuelve los pedidos encontrados en el sistema con las características argumentadas.
- MostrarListaPedidos(pedidos\_lista): muestra la lista de pedidos en pantalla.

### **Añadir Producto a Venta**

- MostrarPantallaAñadirProductoVenta(): muestra la pantalla que añade un producto a la venta.
- AñadirProductoVenta(selección): función que añade un producto a la venta.
- datos\_venta:=AñadirProducto(selección): devuelve la información del producto seleccionado.
- IntroducirCantidad(datos\_venta): función que recibe la información necesaria para la adición del producto a la venta.
- ok:=AñadirProductoVenta(Producto, cantidad): comprueba que los datos argumentados sean correctos y se añade el producto.
- MostrarError(): muestra un mensaje de error en el sistema.

### **Facturar Venta**

- MostrarPantallaFacturarVenta(): muestra la pantalla de facturación de la venta.
- FacturarVenta(id\_venta, lista\_líneas\_documento): función de facturación de una venta con la lista de líneas de documento o productos añadidos a ella.
- id\_fac := CrearFactura(): se crea una factura y devuelve su identificador.
- ok := AsignarProductosFactura (id\_fac, lista\_líneas\_documento): comprueba y asigna los productos en la factura con identificador id\_fac.
- ok:= CrearAsientosContables(Factura): crea los asientos contables pertinentes derivados de la creación de la factura.
- MostrarError(): muestra un mensaje de error en el sistema.

### 5.5.5. Diseño de la capa de presentación

El diseño de la capa de presentación es una de las partes más importantes en nuestra aplicación ya que es la capa que el usuario visualizará durante todo su uso por lo que se éste debe dar una buena y agradable experiencia en su utilización.

La aplicación en general se ha pretendido que sea amigable y sencilla, sobre todo por la cantidad de información que se ha de mostrar en un área tan limitada como es una pantalla de un terminal móvil. También se ha elegido una gama de colores en tonalidades azules que aportan frescura a la interfaz.

En cuanto a la navegabilidad, se dispone de una barra de acciones en todo momento, donde el usuario podrá acceder a las diferentes funcionalidades que ofrece el sistema de manera directa consiguiendo acceder a éstas rápidamente sin recorridos largos y complicados.

A continuación, se mostrarán las pantallas más relevantes de la aplicación, con breves apuntes sobre su diseño principal.



Figura 5.87.: Pantalla de Bienvenida.

## 5. Desarrollo del proyecto

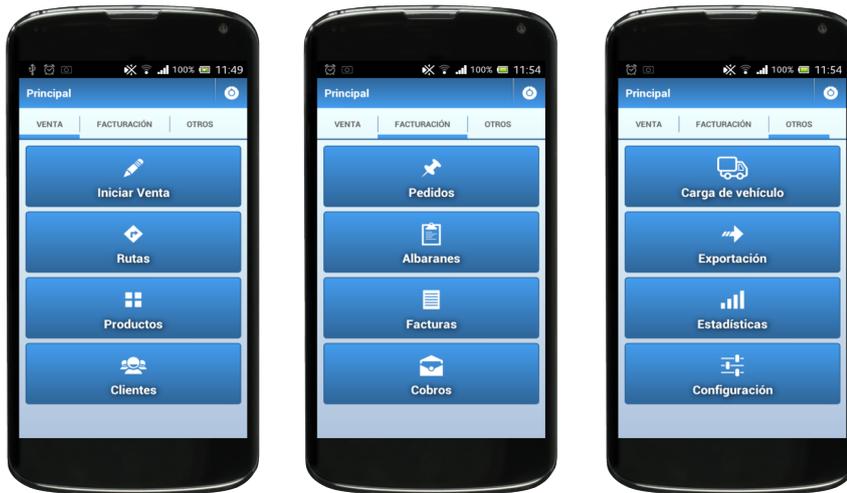


Figura 5.88.: Pantalla del Menú Principal.

Si se desea ver más capturas de pantalla, véase el Capítulo C de guía de uso en la que se hace una visualización completa de la aplicación así como su correcta utilización.

### Mensajes de Información y confirmación.

A continuación se mostrará ejemplos de diálogos de información y diálogos de confirmación en las Figuras 5.89 y 5.90 respectivamente. Se realizan para mejorar la experiencia de usuario y hacer llegar a éste qué es lo que está pasando en cada momento.

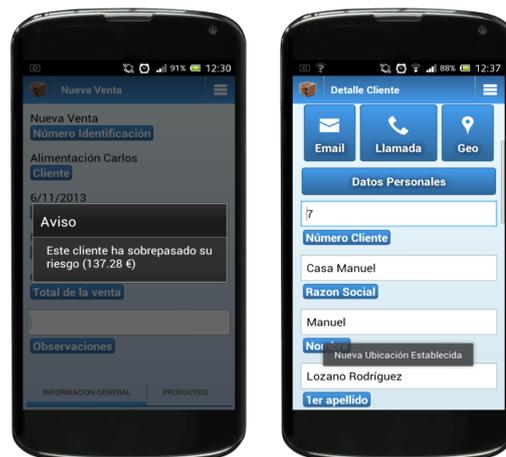


Figura 5.89.: Mensajes de Información.

En la figura 5.90 se muestran los mensajes que requieren de la atención del usuario.



Figura 5.90.: Pantallas de confirmación.

### Diálogos de carga

A continuación se mostrará en la Figura 5.91 ejemplos de cómo se han diseñado los diálogos de carga mostrados en la aplicación cuando se tiene que hacer una operación que requiere de unos segundos. Se han implementado para evitar que el usuario pueda llegar a pensar en algún momento que la aplicación se haya quedado bloqueada y pierda la paciencia intentando cerrarla o realizando cualquier acción que pueda alterar el normal desarrollo de la operación.

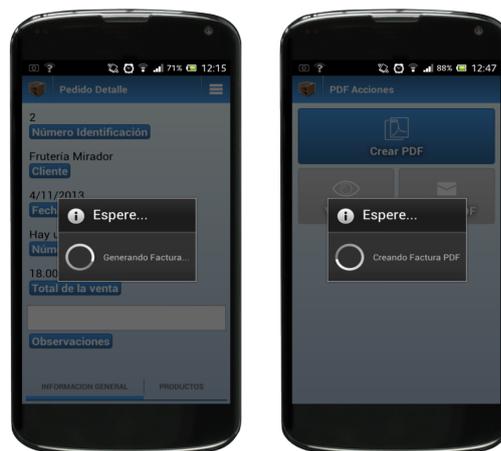


Figura 5.91.: Pantallas de carga.

## 5.6. Implementación

### Entorno de programación

Como se ha comentado en puntos anteriores el lenguaje de programación a través del cual se ha desarrollado la aplicación ha sido Java cuya corriendo sobre la máquina virtual Dalvik[36], y se ha utilizado el IDE ADT Bundle[1] de Eclipse concretamente la versión v22.0.5-75779:



Figura 5.92.: Acerca de ADT Bundle.

Este IDE contiene todo lo necesario para empezar a desarrollar en la plataforma Android: proporciona todas las bibliotecas API y herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android.

Para su instalación, como podemos ver en la Figura 5.93, sólo debemos acceder a la siguiente dirección web y seleccionar sobre qué sistema operativo vamos a empezar a desarrollar en Android.

<http://developer.android.com/sdk/index.html>

ADT Bundle				
Platform	Package	Size	MD5 Checksum	
Windows 32-bit	adt-bundle-windows-x86-20131030.zip	503599460 bytes	cd490a531ec24667354f6473e999b988	
Windows 64-bit	adt-bundle-windows-x86_64-20131030.zip	503735416 bytes	dddbb1b9028015779d68dde01f96b14	
Mac OS X 64-bit	adt-bundle-mac-x86_64-20131030.zip	470386961 bytes	3e80e7a92b549029d91bdcf2ae82657f	
Linux 32-bit	adt-bundle-linux-x86-20131030.zip	496876498 bytes	d389139ad9f59a43bdd34c94bc850509	
Linux 64-bit	adt-bundle-linux-x86_64-20131030.zip	497171697 bytes	99b51a4f0526434b083701a896550b72	
SDK Tools Only				
Platform	Package	Size	MD5 Checksum	
Windows 32 & 64-bit	android-sdk_r22.3-windows.zip	108847452 bytes	9f0fe8c8884d6aee2b298fee203c62dc	
	installer_r22.3-windows.exe (Recommended)	88845794 bytes	ad50c4dd9e23cee65a1ed740ff3345fa	
Mac OS X 32 & 64-bit	android-sdk_r22.3-macosx.zip	74893875 bytes	ecde88ca1f0595582697848fb4a9e7	
Linux 32 & 64-bit	android-sdk_r22.3-linux.tgz	100968558 bytes	6ae581a906d6420ad67176dff25a31cc	

Figura 5.93.: Descarga oficial de ADT Bundle.

Esta descarga incluye los componentes esenciales de Android SDK y una versión del IDE de Eclipse con una función de ADT (Android Developer Tools) para agilizar el desarrollo de aplicaciones Android. Además, viene completamente listo para crear un dispositivo Android virtual para pruebas (véase Figura 5.94), completamente configurable.

Los lenguajes utilizados para el desarrollo de la aplicación en esta plataforma han sido principalmente Java, para las clases y métodos, y XML[11], para la estructura de la interfaz de las pantallas. Pero hay que realizar una aclaración acerca de la máquina virtual en la que se ejecuta: DalvikVM[36] (diseñada y escrita por Dan Bornstein).

La máquina virtual es un intermediario entre las aplicaciones y el sistema, y ha sido optimizada para que pueda correr en dispositivos móviles de forma más eficiente y rápida. A nivel de programación esto nos va a repercutir en que, realmente el lenguaje Java que utiliza Android es, por así decirlo, un subconjunto del lenguaje Java que todos conocemos y que corre en la máquina virtual JVM. Por lo tanto se saca una conclusión muy importante: *todo lo que funciona en Java de Android funciona en Java pero, no todo lo que funciona en Java funciona en Java de Android.*

## 5. Desarrollo del proyecto

### Diseño Físico de la Base de datos

Al instalar la aplicación en el terminal y, automáticamente, cuando se accede a ella por primera vez, se crea una base de datos dentro del directorio de aplicación de la misma y se configura para que el usuario pueda utilizarla desde el primer momento sin ningún tipo de acción extra por parte de éste. Para asegurar la integridad de los datos se ha provisto a las tablas de las correspondientes claves foráneas como se puede apreciar en el punto anterior.

Ha de destacarse que en el sistema gestor de base de datos SQLite sólo acepta 5 tipos de datos[3] y estos son:

- *NULL*. El valor es nulo ó NULL.
- *INTEGER*. El valor es un entero con signo, almacenado en 1, 2, 3, 4, 6, or 8 bytes dependiendo de su tamaño.
- *REAL*. El valor es de tipo flotante almacenado como un número flotante de 8-byte IEEE .
- *TEXT*. El valor es una cadena de texto almacenada usando la codificación de bases de datos (UTF-8, UTF-16BE or UTF-16LE).
- *BLOB*. El valor es un tipo de dato blob, y se almacena tal y como este dato es.

Para más información al respecto, visite el siguiente enlace de la documentación de SQLite <http://www.sqlite.org/datatype3.html>

Destacar que para mayor seguridad la contraseña del usuario se ha encriptado con el algoritmo *AES* uno de los más seguros en el campo de la criptografía. La implementación de esta clase se muestra en la sub-sección 5.6.

A continuación se mostrará la creación de las tablas de la aplicación:

```
1 CREATE TABLE Albaran ( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
2     fecha TIMESTAMP NOT NULL DEFAULT current_date ,
3     observaciones TEXT ,
4     facturado INTEGER NOT NULL DEFAULT '0' ,
5     codigo_cli INTEGER REFERENCES Cliente(_id) );
6
7
8 CREATE TABLE ApunteContable ( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
9     id_asiento INTEGER UNIQUE NOT NULL ,
10    fecha_asiento TIMESTAMP NOT NULL ,
11    subcuenta TEXT NOT NULL ,
12    concepto TEXT NOT NULL ,
13    debe TEXT NOT NULL ,
14    haber TEXT NOT NULL ,
15    documento TEXT NOT NULL ,
16    cod_factura INTEGER REFERENCES Factura(_id) );
17
18 CREATE TABLE Categoria ( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
```

```

19     nombre TEXT NOT NULL);
20
21 CREATE TABLE Cliente( _id INTEGER PRIMARY KEY AUTOINCREMENT,
22     numero_cliente INTEGER UNIQUE NOT NULL,
23     razon_social TEXT NOT NULL,
24     cif TEXT UNIQUE NOT NULL, ,
25     nombre TEXT NOT NULL,
26     apellido1 TEXT NOT NULL,
27     apellido2 TEXT,
28     direccion TEXT,
29     cp TEXT,
30     email TEXT UNIQUE,
31     fecha_alta TIMESTAMP NOT NULL DEFAULT current_date,
32     riesgo TEXT NOT NULL DEFAULT '0',
33     activado INTEGER NOT NULL DEFAULT '1',
34     aplica_re INTEGER DEFAULT '0',
35     codigo_rut INTEGER REFERENCES Ruta(_id));
36
37 CREATE TABLE ClienteCP ( _id INTEGER PRIMARY KEY AUTOINCREMENT,
38     cp TEXT UNIQUE NOT NULL,
39     poblacion TEXT NOT NULL,
40     provincia TEXT NOT NULL);
41
42 CREATE TABLE ClienteGeo ( _id INTEGER PRIMARY KEY AUTOINCREMENT,
43     latitud REAL NOT NULL,
44     longitud REAL NOT NULL,
45     cod_cliente INTEGER REFERENCES Cliente(_id) );
46
47 CREATE TABLE Cobro (_id INTEGER PRIMARY KEY AUTOINCREMENT,
48     fecha_cobro TIMESTAMP NOT NULL DEFAULT current_date );
49
50 CREATE TABLE Factura ( _id INTEGER PRIMARY KEY AUTOINCREMENT,
51     fecha TIMESTAMP NOT NULL DEFAULT current_date,
52     cobrada INTEGER NOT NULL DEFAULT '0',
53     importe_parcial_aportado TEXT DEFAULT '0',
54     observaciones TEXT,
55     cobro INTEGER REFERENCES Cobro(_id) DEFAULT NULL,
56     codigo_cli INTEGER REFERENCES Cliente(_id) );
57
58 CREATE TABLE HistorialIVA_RE (_id INTEGER PRIMARY KEY AUTOINCREMENT,
59     fecha_modificacion TIMESTAMP NOT NULL DEFAULT current_date,
60     valor_iva TEXT NOT NULL,
61     valor_re TEXT NOT NULL );
62
63 CREATE TABLE ImpuestosIVA_RE (_id INTEGER PRIMARY KEY AUTOINCREMENT,
64     tipo_iva TEXT NOT NULL,
65     tipo_re TEXT NOT NULL );
66
67 CREATE TABLE LineaPedido( _id INTEGER PRIMARY KEY AUTOINCREMENT,
68     precio_pro TEXT NOT NULL,
69     unidades_pro INTEGER NOT NULL,
70     codigo_pro INTEGER REFERENCES Producto(_id) NOT NULL,
71     codigo_ped INTEGER REFERENCES Pedido(_id) ,
72     codigo_alb INTEGER REFERENCES Albaran(_id),
73     codigo_fac INTEGER REFERENCES Factura(_id),
74     descuento TEXT NOT NULL DEFAULT '0' );
75
76 CREATE TABLE Pedido (_id INTEGER PRIMARY KEY AUTOINCREMENT,
77     fecha TIMESTAMP NOT NULL DEFAULT current_date,
78     observaciones TEXT,
79     codigo_cli INTEGER REFERENCES Cliente(_id) );
80

```

## 5. Desarrollo del proyecto

```
81 CREATE TABLE Producto ( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
82     numero_producto TEXT UNIQUE NOT NULL ,
83     tipo_p TEXT NOT NULL ,
84     marca TEXT NOT NULL ,
85     descripcion TEXT NOT NULL ,
86     t_envase TEXT NOT NULL ,
87     cap_peso TEXT NOT NULL ,
88     precio_venta TEXT NOT NULL ,
89     stock_veh INTEGER NOT NULL DEFAULT '0' ,
90     codigo_iva INTEGER REFERENCES ImpuestosIVA_RE(_id) ,
91     codigo_cat INTEGER REFERENCES Categoria(_id) );
92
93 CREATE TABLE Ruta( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
94     nombre TEXT NOT NULL ,
95     dia_semana INTEGER NOT NULL);
96
97 CREATE TABLE Telefono ( _id INTEGER PRIMARY KEY AUTOINCREMENT ,
98     numero TEXT UNIQUE NOT NULL ,
99     codigo_cli INTEGER REFERENCES Cliente(_id) );
100
101 CREATE TABLE android_metadata ("locale" TEXT DEFAULT 'es_ES');
```

La tabla con el nombre *android\_metadata* es una tabla requerida por el sistema ya que éste necesita información acerca de la localización en el que se encuentra la aplicación. En nuestro caso, la localización será española.

Si se observa con atención, todos aquellos campos que requieran almacenar un número decimal se han definido como tipo TEXT. Los tipos de datos FLOAT y DOUBLE, o en nuestro caso también REAL, aunque muy usados y útiles tienen defectos, como que no son capaces de representar el resultado de ciertas operaciones aritméticas entre números reales con total precisión. Por tanto usarlos para hacer ciertas operaciones aritméticas, y siendo acumuladas varias de ellas, puede dar lugar a errores de redondeo y precisión en el resultado final. Por estos motivos usar estos tipos de datos no son los más adecuados para trabajar en una aplicación como ésta en la que hay que calcular precios.

También se han implementado una serie de disparadores ó *triggers* que son los siguientes:

```
1
2 CREATE TRIGGER EliminarGeoCliente
3     AFTER DELETE ON Cliente
4     BEGIN DELETE FROM ClienteGeo
5     WHERE OLD._id=cod_cliente; END;
6
7 CREATE TRIGGER EliminarTelefonosCliente
8     AFTER DELETE ON Cliente
9     FOR EACH ROW BEGIN
10    DELETE FROM Telefono
11    WHERE OLD._id=codigo_cli; END;
12
13 CREATE TRIGGER QuitarCategoriaProductos
14     AFTER DELETE ON Categoria
15     FOR EACH ROW BEGIN
16    UPDATE Producto SET codigo_cat='null'
17    WHERE codigo_cat=OLD._id; END;
18
```

```

19 CREATE TRIGGER QuitarClientesRuta
20     AFTER DELETE ON Ruta
21     FOR EACH ROW BEGIN
22         UPDATE Cliente SET codigo_rut='null'
23         WHERE codigo_rut=OLD._id; END;

```

Estos disparadores ayudarán a mantener consistencia de los datos almacenados y se ejecutarán automáticamente si se realizan las acciones que se detallan en ellos para su ejecución sobre la base de datos.

Los dos primeros disparadores se van a ejecutar automáticamente si se elimina un cliente, es decir, se borrará del sistema su geolocalización y sus teléfonos si los tuviera, respectivamente. Cuando se elimina una categoría, el tercer disparador se ejecuta para asegurar que los productos que estuvieran en esa categoría, ya no lo estén. Por último, si eliminamos una ruta, el cuarto disparador quita de la ruta a todos los clientes que estuvieran en ella.

## Encriptación

Otro punto importante es, sin duda, la temática de encriptación de información. Para ello, he desarrollado una clase en Java que realiza la encriptación y desencriptación mediante el algoritmo AES (las claves de encriptación que se usarán en esta aplicación serán de 128 bits).

```

1 public class Encriptacion
2 {
3
4
5     private static final String ALGORITMO = "AES";
6     private static final byte[] valor_clave; // ESTA SERÁ SU CLAVE RAÍZ DE ENCRYPTACIÓN
7
8     public static String encriptar (String texto_a_encriptar) throws Exception
9     {
10         Key key = new SecretKeySpec(valor_clave, ALGORITMO);
11         Cipher cipher = Cipher.getInstance("AES");
12         cipher.init(Cipher.ENCRYPT_MODE, key );
13         byte[] encrypted = cipher.doFinal( texto_a_encriptar.getBytes("UTF-8") );
14         String texto_encriptado = Base64.encodeToString(encrypted, Base64.DEFAULT);
15
16         return texto_encriptado;
17     }
18
19     public static String desencriptar(String texto_encriptado) throws Exception
20     {
21         Key key = new SecretKeySpec(valor_clave, ALGORITMO);
22         Cipher cipher = Cipher.getInstance("AES");
23         cipher.init(Cipher.DECRYPT_MODE, key);
24         byte[] decodificar_texto = Base64.decode( texto_encriptado.getBytes("UTF-8")
25             , Base64.DEFAULT );
26         byte[] desencriptado = cipher.doFinal(decodificar_texto);
27
28         return new String(desencriptado, "UTF-8");
29     }

```

## 5. Desarrollo del proyecto

Así pues para encriptar alguna información, solo hay que utilizar esta clase que, al ser tener dichos métodos como estáticos, puede ser convocada en cualquier ámbito en el que nos encontremos.

### La clase **BigDecimal**

Por otra parte, se ha tomado la decisión de utilizar la clase `BigDecimal`[31] para el uso de cantidades decimales con las que se van a realizar operaciones aritméticas en nuestra aplicación. La explicación a tal decisión llega a partir de la investigación de cómo opera cualquier microprocesador a la hora de hacer cálculos aritméticos cuando utilizamos otros tipos de datos como `float` ó `double`. A la hora de realizarlas, existen pequeños problemas de precisión que no son solventados ya que el problema se origina en la base numérica con la que operan, el sistema binario, y por consiguiente, existe la clase `BigDecimal` que soluciona dicho problema de cálculo.

### Fechas

También es importante mencionar que esta aplicación trabaja con fechas según el estándar ISO-8601[10], el cual especifica una fecha dónde primeramente escribiríamos el año, posteriormente el mes y a continuación el día. Para facilitar esta tarea, se ha programado funciones estáticas que construyen dicha fecha con ese formato a partir de variables de tipo `Calendar`, que es una clase con la que podemos trabajar sencillamente con fechas en Java:

```
1 public static CharSequence parsearFecha(Calendar c)
2 {
3     return DateFormat.format("yyyy-MM-dd", c);
4 }
5
6 public static CharSequence parsearFechaDia(Calendar c)
7 {
8     return DateFormat.format("dd", c);
9 }
10
11 public static CharSequence parsearFechaMes(Calendar c)
12 {
13     return DateFormat.format("MM", c);
14 }
15
16 public static CharSequence parsearFechaAño(Calendar c)
17 {
18     return DateFormat.format("yyyy", c);
19 }
20
21 public static String parsearFechaBarras(Calendar a)
22 {
23     return new String(a.get(Calendar.DAY_OF_MONTH) + "/"
24         + (a.get(Calendar.MONTH)+1) + "/"
25         + a.get(Calendar.YEAR));
26 }
27
```

Si desea ver más información acerca de este estándar, visite la siguiente dirección:  
<http://www.iso.org/iso/home/standards/iso8601.htm>

### Patrón de diseño Singleton para la base de datos

El patrón de diseño de creación *Singleton*[25] está diseñado para restringir la creación de objetos pertenecientes a una clase dando la seguridad de que sólo exista una única instancia de esa clase sin importar el número de veces que se cree y también, darle acceso global dentro de la aplicación.

Fundamentalmente, cada pantalla que vemos en la aplicación se corresponde con una *Activity* y, en el caso de que quisiésemos acceder a la base de datos para capturar alguna información, se veía necesario inicializar cada vez un objeto de la clase *SQLiteOpenHelper* para poder realizar dichas acciones. Como inconvenientes de esta práctica, estamos ante duplicidad en el código durante todas las *Activities* y utilización extra de recursos como la memoria del terminal. Veámos el código resultante de aplicar este diseño[30].

```

1 public class ControlBD extends SQLiteOpenHelper
2 {
3     private static ControlBD bd = null;
4
5     // ...
6
7     public static ControlBD getInstance(Context ctx)
8     {
9         if (bd == null)
10            bd = new ControlBD(ctx.getApplicationContext());
11
12        return mibd;
13    }
14 }

```

Implementando este patrón de diseño, se simplifica bastante el acceso a la base de datos desde el lugar que queramos escribiendo lo siguiente:

```

1 ControlBD bd = ControlBD.getInstance(this);

```

### Genymotion

La primera impresión que me llevé cuando usé por primera vez el emulador por defecto de Android integrado en *ADT Bundle* fue que era muy lento, demasiado lento para experimentar una buena experiencia testando mi aplicación en el sistema operativo Android. Esto me hizo investigar acerca de alternativas al emulador por defecto y esto me hizo llegar hasta *Genymotion*[4].

*Genymotion* es una máquina virtual, que corre en *VirtualBox*, del sistema operativo Android. No es un emulador, es una versión x86 del sistema operativo, lo cual le permite ejecutarse de manera mucho más rápida que su contraparte emulada.

## 5. Desarrollo del proyecto

Aunque gran parte de las pruebas las he realizado desde un terminal físico, *Genymotion* me ha sido de gran utilidad a la hora de testar mi aplicación de una forma rápida y sencilla.

Puede descargarse desde:

<http://www.genymotion.com/>

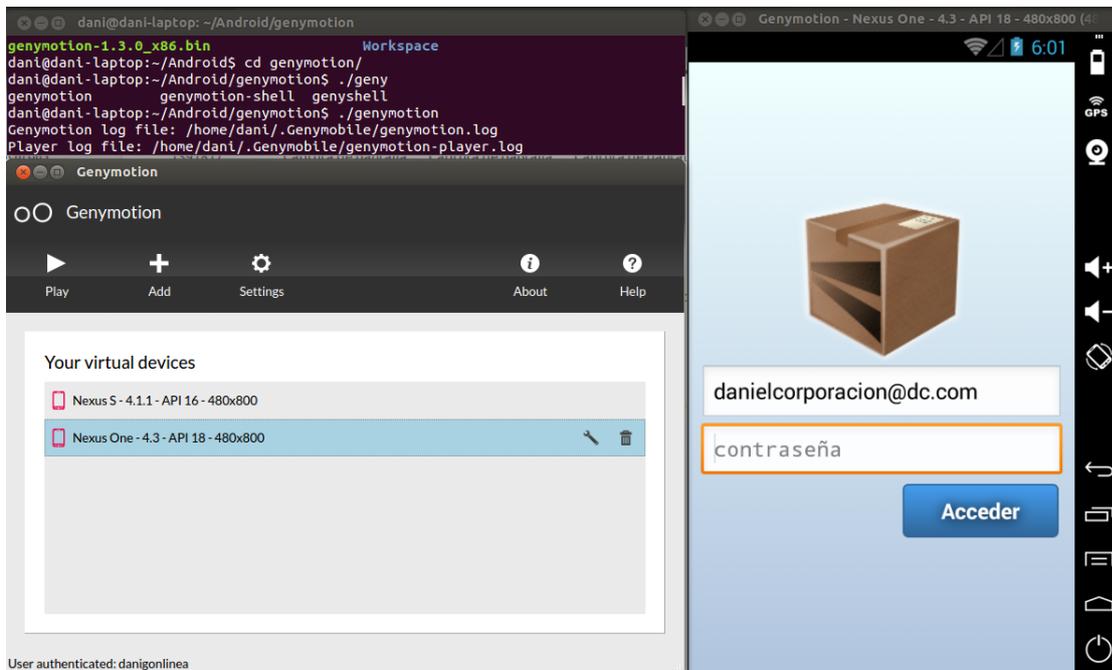


Figura 5.94.: Arranque y visualización de Genymotion.

## 5.7. Pruebas y validación

En este momento, vamos a centrarnos en la etapa del proceso de revisión de la aplicación ikeComercial comprobando que se cumplen con las especificaciones y las expectativas del usuario.

El SDK de Android cuenta con las herramientas necesarias para escribir distintos tipos de prueba sobre nuestras aplicaciones. El objetivo de realizar pruebas y validaciones en este proyecto (y en cualquier otra aplicación) será que cualquier cambio de código que realicemos seguirá funcionando correctamente. Para tal fin vamos a utilizar el framework *JUnit v3.0*[28] para las pruebas unitarias del código fuente y, para las pruebas unitarias de interfaz, utilizaremos las herramientas *UI Automator Viewer* y *uiautomator*[29].

Además, se ha distribuido la aplicación desde una versión beta hacia un grupo de personas que me han ayudado a evaluar e identificar errores del producto en pos de optimizar la aplicación en cuestión.

### 5.7.1. JUnit

El framework de pruebas que nos proporciona el SDK de Android está basado en JUnit [26]. Por una parte, vamos a utilizar la clase *android.test.TestCase* con la que realizaremos las pruebas básicas que nos permitirá de forma sencilla comprobar el funcionamiento de funciones independientes que relativamente tengan un acoplamiento bajo con respecto a otras partes que constituyen la aplicación.

```

1 public void testFormatoFechaCorrecta()
2 {
3     CharSequence fecha_inicial = "2013-12-18";
4     CharSequence fecha_transformada = Ayuda.parsearFecha(fecha);
5
6     assertEquals(fecha_inicial, fecha_transformada);
7 }
8 public void testComprobarEncriptacionCorrecta() throws Exception
9 {
10    String contraseña_transformada = Encriptacion.encriptar(contraseña_a_comprobar);
11    String contraseña_desencriptada = Encriptacion.desencriptar(
12        contraseña_transformada);
13
14    assertEquals(contraseña_a_comprobar, contraseña_desencriptada);
15 }
16 public void testComprobarDisponibilidadMapasGoogle()
17 {
18    HttpGet httpget = new HttpGet("https://www.google.com/maps/preview");
19    HttpClient httpclient = new DefaultHttpClient();
20    HttpResponse response = null;
21    try {
22        response = httpclient.execute(httpget);
23    } catch (Exception e)
24    {
25        fail("Excepción: No se ha podido establecer la conexión.");
26    }

```

## 5. Desarrollo del proyecto

```
27 assertEquals(HttpStatus.SC_OK, response.getStatusLine().getStatusCode());
28 }
29 }
```

La primera de ellas comprueba que el tipo de dato *Calendar* (con el que se representa la fecha en la aplicación) se pueda transformar en una fecha entendible para el sistema de gestor de bases de datos utilizado en este proyecto, SQLite, respetando el estándar ISO-8601[10]. En la segunda prueba he verificado que el algoritmo AES utilizado para la encriptación y desencriptación de información funciona correctamente. Por último, se ha elaborado una prueba de conexión para verificar que los mapas proporcionados en la aplicación (Google Maps) están disponibles para su uso.

Ahora haremos uso de *android.test.AndroidTestCase*. Esta clase extiende a la clase utilizada anteriormente y ofrece un contexto simulado, es decir, que tengan un acoplamiento mayor y que interactúen con otros módulos implementados en el sistema.

```
1 public void testPermisoAccesoAlmacenamientoExterno()
2 {
3     int res = pm.checkPermission("android.permission.WRITE_EXTERNAL_STORAGE", "com.
4         android.ventas");
5     assertEquals(res, PackageManager.PERMISSION_GRANTED);
6 }
7 public void testAccesoBaseDatos()
8 {
9     try
10    {
11        mi_bd.openToRead();
12        mi_bd.close();
13
14        mi_bd.openToWrite();
15        mi_bd.close();
16    }
17    catch (SQLException e)
18    {
19        fail("Excepción: No se tiene acceso a la base de datos.");
20    }
21 }
22
23 public void testCalcularImporteFactura()
24 {
25
26     BigDecimal importe_total_calculado_factura = Ayuda.CalcularImporteFactura(f, this
27         .getContext());
28
29     BigDecimal importe_total_redondeado = importe_total_calculado_factura.round(new
30         MathContext(2, RoundingMode.HALF_UP));
31
32     assertEquals( importe_total_redondeado, importe_total_fijado_factura);
33 }
```

La primera de ellas comprueba que el permiso para acceder al almacenamiento externo del terminal móvil está activado. En la segunda prueba he verificado que se puede acceder exitosamente a la base de datos en modo escritura/lectura y en modo lectura. En la tercera prueba se ha comprobado el correcto cálculo del importe

de una factura, involucrando por supuesto el cálculo de la base imponible y de los impuestos.

En la última prueba es imposible comprobar la veracidad o falsedad del cálculo que proporciona dicha función, por nadie me garantiza que el camino de cálculo sea el correcto ya que yo mismo puedo haberme equivocado en su elaboración. Para su comprobación total se realizará una prueba manual del mismo que va a representar todo el proceso de cálculo aplicando todos los impuestos establecidos en el régimen español, de forma que pueda ser una buena representación del mismo.

Sea la factura F del cliente C, el cuál se le aplicará el recuento de equivalencia y el IVA correspondiente, contiene los siguientes artículos:

- Leche Puleva 1L. Precio por unidad: 0,75 €. Unidades solicitadas: 5. IVA: 4 %.
- Leche Puleva 1L. Precio por unidad: 0,8 €. Unidades solicitadas: 10. IVA: 4 %.
- Agua AquaBona 0,5L. Precio por unidad: 0,28 €. Unidades solicitadas: 25. IVA: 10 %.
- Flores Rosas Rojas 12 uds. Precio por unidad: 19 €. Unidades solicitadas: 3. IVA: 21 %.
- CDs Verbatim Tarrina 10 uds. Precio por unidad: 3,3 €. Unidades solicitadas: 5. IVA: 21 %.

En una primera instancia, realizaremos el cálculo de las bases imponibles de los artículos agrupándolos por el impuesto IVA que tengan en común, es decir, tendremos dos productos que tributan al 4%(A), 1 producto que tributa al 10%(B) y otros dos productos que tributan al 21%(C). La base imponible será igual al producto entre las unidades solicitadas y el precio del producto.

- Base Imponible A:  $0,75 * 5 + 0,8 * 10 = 11,75$ .
- Base Imponible B:  $0,28 * 25 = 7$ .
- Base Imponible C:  $19 * 3 + 3,3 * 5 = 73,5$ .

A continuación, vamos a calcular los respectivos impuestos del IVA Y RE de cada una de las bases imponibles calculadas anteriormente. Procederemos a calcular el importe del IVA:

- $11,75 * 4\% = 11,75 * ( 4 / 100 ) = 0,47$ .
- $7 * 10\% = 7 * ( 10 / 100 ) = 0,7$ .
- $73,5 * 21\% = 73,5 * ( 21 / 100 ) = 15,435 \approx 15,44$ .

## 5. Desarrollo del proyecto

Como se puede apreciar en la última operación anterior, el redondeo que se aplicará durante este proceso será el redondeo HALF\_UP ó simétrico a dos decimales. Vayamos ahora a calcular el importe del RE:

- $11,75 * 0,5\% = 11,75 * ( 5 / 1000 ) = 0,05875 \approx 0,06.$

- $7 * 1,4\% = 7 * ( 14 / 1000 ) = 0,098 \approx 0,1.$

- $73,5 * 5,2\% = 73,5 * ( 52 / 1000 ) = 3,822 \approx 3,82.$

Ahora que todos los impuestos están calculados, hay que sumarlos respectivamente para obtener los subtotales de cada grupo.

- $11,75 + 0,47 + 0,06 = 12,28.$

- $7 + 0,7 + 0,1 = 7,8.$

- $73,5 + 15,44 + 3,82 = 92,76.$

El importe total de la factura será la suma de todos los subtotales anteriores, es decir,  $12,28 + 7,8 + 92,76 = 112,84$ . La unidad monetaria utilizada es euros, por lo tanto, será *112,84* €. Podemos comprobar como todos los datos anteriormente calculados están correctamente descritos en la Figura 5.95.



Figura 5.95.: Detalle de la factura de prueba.

## 5. Desarrollo del proyecto

### 5.7.2. UI Automator Viewer y uiautomator

La herramienta UI Automator Viewer [29] nos va a escanear y analizar gráficamente los componentes de interfaz gráfica de una aplicación Android. Además de poder observar cómo está formada jerárquicamente la pantalla, podemos ver los atributos de cada componente. Esta información nos va a ser de mucha utilidad porque nos va a ayudar a identificar dichos componentes para las posteriores pruebas con uiautomator, el cuál ejecutaremos en modo consola.

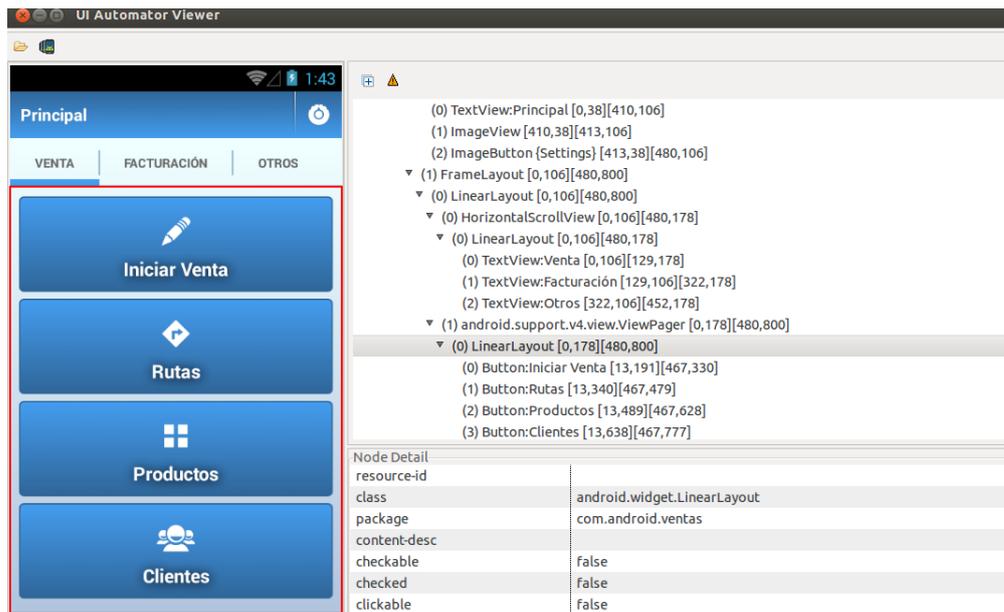


Figura 5.96.: UI Automator Viewer

Ahora que sabemos cómo referenciar las vistas, podemos escribir las pruebas pertinentes de interfaz de usuario automatizadas con uiautomator[29]. En el siguiente código se describe el proceso de prueba desde el emulador de la creación de un nuevo cliente en nuestro sistema.

```
1 public void testCrearNuevoUsuario () throws UiObjectNotFoundException
2 {
3     // Presiono el botón HOME del terminal
4     getUiDevice().pressHome();
5
6     // Presionamos sobre la aplicación para iniciarla.
7     UiObject ikeComercial_app = new UiObject(new UiSelector().text("ikeComercial"));
8     ikeComercial_app.clickAndWaitForNewWindow();
9
10    UiObject contraseña_btn = new UiObject(new UiSelector().resourceId("com.android.
11        ventas:id/login_password"));
12    contraseña_btn.setText("pruebaX022");
13    UiObject acceder_btn = new UiObject(new UiSelector().resourceId("com.android.
14        ventas:id/login_sign_btn"));
```

```

13 acceder_btn.clickAndWaitForNewWindow();
14 UiObject clientes_btn = new UiObject(new UiSelector().resourceId("com.android.
    ventas:id/clientes_btn"));
15 clientes_btn.clickAndWaitForNewWindow();
16
17 UiObject clientes_lista_acciones_btn = new UiObject(new UiSelector().resourceId("
    com.android.ventas:id/gd_action_bar_item")); clientes_lista_acciones_btn.
    click();
18 UiObject nuevo_cliente_btn = new UiObject(new UiSelector().text("Nuevo_Cliente"))
    ; nuevo_cliente_btn.clickAndWaitForNewWindow();
19
20 UiScrollable hacer_scroll = new UiScrollable(new UiSelector().scrollable(true));
21
22 UiObject numero_cliente_et = new UiObject(new UiSelector().text("Número_cliente_
    aquí"));
23 numero_cliente_et.setText("12");
24
25 UiObject razon_social_et = new UiObject(new UiSelector().text("Razón_social_aquí"
    ));
26 razon_social_et.setText("Copisteria_Hnos._Guerrero");
27
28 hacer_scroll.scrollForward();
29
30 UiObject nombre_et = new UiObject(new UiSelector().resourceId("com.android.ventas
    :id/cliente_detalle_nombre"));
31 nombre_et.setText("Miguel");
32
33 UiObject primer_apellido_et = new UiObject(new UiSelector().text("Primer_apellido
    aquí"));
34 primer_apellido_et.setText("Guerrero");
35
36 UiObject segundo_apellido_et = new UiObject(new UiSelector().text("Segundo_
    apellido_aquí"));
37 segundo_apellido_et.setText("Fontalves");
38
39 UiObject cif_et = new UiObject(new UiSelector().text("CIF_aquí")); cif_et.
    setText("78080922Z");
40
41 hacer_scroll.scrollForward();
42
43 UiObject email_et = new UiObject(new UiSelector().text("Email_aquí")); email_et.
    setText("copisteriaguerrero@cguerrero.com");
44
45 UiObject direccion_et = new UiObject(new UiSelector().text("Dirección_aquí"));
46 direccion_et.setText("Calle_Soriano_2_Bajo");
47
48 UiObject cp_et = new UiObject(new UiSelector().index(16)); cp_et.setText("11140"
    );
49
50 UiObject poblacion_et = new UiObject(new UiSelector().index(18)); poblacion_et.
    setText("Conil_de_la_Frontera");
51
52 hacer_scroll.scrollForward();
53
54 UiObject provincia_et = new UiObject(new UiSelector().index(20)); provincia_et.
    setText("Cadiz");
55
56 UiObject tlf1_et = new UiObject(new UiSelector().index(22)); tlf1_et.setText("
    956440011");
57
58 UiObject tlf2_et = new UiObject(new UiSelector().index(24)); tlf2_et.setText("
    665709001");

```

## 5. Desarrollo del proyecto

```
59     hacer_scroll.scrollForward();
60
61
62     UiObject riesgo = new UiObject(new UiSelector().resourceId("com.android.ventas:id
63     /cliente_detalle_riesgo"));
64     riesgo.setText("300");
65
66     UiObject confirmar_cliente_btn = new UiObject(new UiSelector().resourceId("com.
67     android.ventas:id/gd_action_bar_item"));
68     confirmar_cliente_btn.clickAndWaitForNewWindow();
69
70     sleep(2000);
71
72     // Confirmar que el cliente se ha creado.
73     UiObject cliente_creado = new UiObject(new UiSelector().text("Materiales□Hnos.□
74     Guerrero"));
75
76     assertTrue(cliente_creado.exists());
77 }

```

Una vez escrita la prueba, necesitamos realizar una serie de pasos antes de ejecutar dicha prueba en el emulador.

En primer lugar, vamos a generar el fichero de configuración build.xml del proyecto por lo que llamaremos al comando "android" ubicado en el SDK de Android.

```
1 $ ./android create uitest-project -n ikeComercialTesting -t 1 -p /home/dani/Android
  /Workspace/ikeComercialTesting/

```

En segundo lugar, necesitamos compilar el proyecto de prueba para que nos genere un fichero .jar el cuál vamos a ejecutar en el terminal.

```
1 $ ant build

```

En tercer lugar, vamos a incluir este fichero .jar generado al dispositivo donde se realizarán las pruebas.

```
1 $ ./adb push /home/dani/Android/Workspace/ikeComercialTesting/bin/
  ikeComercialTesting.jar /data/local/tmp/

```

Ahora todo está listo para ejecutar la prueba. Para tal fin ejecutamos:

```
1 $ ./adb shell uiautomator runtest ikeComercialTesting.jar -c android.prueba.
  Testing

```

## 5.7. Pruebas y validación

Visualmente en el terminal, podemos observar cómo se van ejecutando las pantallas e introduciendo los datos de manera totalmente automática para, una vez realizada la prueba, mostrar el siguiente resultado en el terminal:

```
1 INSTRUMENTATION_STATUS: current=1
2 INSTRUMENTATION_STATUS: id=UiAutomatorTestRunner
3 INSTRUMENTATION_STATUS: class=android.prueba.Testing
4 INSTRUMENTATION_STATUS: stream= android.prueba.Testing:
5 INSTRUMENTATION_STATUS: numtests=1
6 INSTRUMENTATION_STATUS: test=testCrearNuevoUsuario
7 INSTRUMENTATION_STATUS_CODE: 1
8 INSTRUMENTATION_STATUS: current=1
9 INSTRUMENTATION_STATUS: id=UiAutomatorTestRunner
10 INSTRUMENTATION_STATUS: class=android.prueba.Testing
11 INSTRUMENTATION_STATUS: stream=.
12 INSTRUMENTATION_STATUS: numtests=1
13 INSTRUMENTATION_STATUS: test=testCrearNuevoUsuario
14 INSTRUMENTATION_STATUS_CODE: 0
15 INSTRUMENTATION_STATUS: stream= Test results for WatcherResultPrinter=.
16 Time: 71.89
17
18 OK (1 test)
```

Como puede observarse en el código, la prueba se ha visualizado y realizado con éxito

## 5. *Desarrollo del proyecto*

## 6. Conclusiones y trabajo futuro

En esta sección se presentan las conclusiones a las que se han llegado tras desarrollar este PFC y el trabajo futuro que se pretende realizar.

### 6.1. Conclusiones

En este PFC se ha desarrollado una aplicación de gestión de ventas para smartphones en Android para los comerciales de empresas que se desplazan hacia sus clientes para poder distribuirles sus productos. Esta aplicación proporciona las siguientes funcionalidades:

- Gestionar tanto los clientes que la empresa tiene como los productos que suministra.
- Administrar los pedidos del comercial, ya sean en calidad de albarán, factura ó pedido pendiente.
- Proceder a la gestión de cobros de las facturas de los clientes.
- Planificar semanalmente las visitas ó rutas con el objetivo de que el comercial llegue a sus clientes.
- Añadir un sistema de aviso de cobros para los clientes, así como un sistema de control de stock de los productos que el comercial transporte en su vehículo.
- Generar gráficas estadísticas de las ventas realizadas.
- Exportar los asientos contables (anotaciones realizadas con el fin de reflejar una operación contable) de las facturas emitidas y acoplarlas a un software externo de gestión contable: Sage Contaplus[16].

Esta aplicación ha sido desarrollada con tecnología Android (*Java, XML y SQLite*) y bibliotecas externas como *Greendroid* [7] ó *PagerSlidingTabStrip*. Se ha utilizado el IDE *ADT Bundle* basado en *Eclipse* para su implementación junto con *Genymotion*[4] para emular la aplicación en el sistema Android. Además, se han llevado a cabo pruebas unitarias haciendo uso de la biblioteca de pruebas *JUnit*, así como pruebas de interfaz mediante las herramientas *UI Automator Viewer* y *uiautomator*[29].

## 6. Conclusiones y trabajo futuro

Esta aplicación se ha publicado con éxito en *GitHub* y también en el mercado de aplicaciones Android, *Google Play* de forma gratuita.

### 6.2. Valoración personal

Y llegados a este punto, este es el momento adecuado para hacer una parada en el camino, respirar hondo y echar la vista atrás para poder darnos cuenta y contemplar el camino recorrido que nos ha llevado hasta aquí. Tengo que decir que el camino no ha sido fácil pero, como todo en la vida, hay que quedarse con la parte positiva y, de hecho, eso es lo que voy a hacer.

La experiencia que me ha brindado este proyecto ha sido espectacular. He podido ver y darme cuenta ahora de todo el trabajo y dificultad que conlleva un proyecto de estas características, también, por supuesto, de todos los conocimientos que he adquirido gracias a él y de lo mucho que he madurado tanto personal como profesionalmente.

No me arrepiento ni por un instante de haber estudiado el mundo Android para realizar este PFC. Desde el principio empecé a sentir una fuerte curiosidad y sobre todo, al ser una tecnología relativamente nueva, una fuente de motivación muy grande. Es imposible decir cuál es la parte con la cuál más he disfrutado; todas han llamado mi atención. En un futuro, me gustaría seguir desarrollando para esta plataforma y, por supuesto, seguir mejorando ikeComercial.

## 6.3. Trabajo futuro

Esperamos que la aplicación tenga buena acogida entre los usuarios y, sobre todo, que le den una oportunidad todos aquellos que están dedicados a la gestión de ventas y pueda serles útil en su día a día. Eso es probablemente la mayor satisfacción que nos pueda aportar. Entre otras cosas, deseamos que los usuarios critiquen esta aplicación todo lo que les parezca mejorable para así, adaptar y evolucionar esta aplicación para que puedan seguir usándola.

En cuanto a las líneas de trabajo futuras, podrían añadirse las siguientes funcionalidades a la aplicación desarrollada:

- Escalar la aplicación para que pueda adaptarse adecuadamente a tamaños superiores de pantalla, como pueden ser los *tablets*.
- Exportar datos a más programas de gestión como pueden ser CeConta de la empresa Ceginfor ó Microsoft Dynamics de Microsoft.
- Incluir más métodos de cobro de factura, por ejemplo por *pay pal*.
- Internacionalizar la aplicación: adaptar la moneda de pago y los impuestos de otros países así como la traducción a otros idiomas.

Sin lugar a dudas este es un proyecto ambicioso: nuestra intención es que esta aplicación siga creciendo para seguir ofreciendo más funcionalidades a los usuarios y poder competir con otras aplicaciones que se encuentran en el mercado.

## 6. Conclusiones y trabajo futuro

## A. Modelo de negocio

Una vez que la aplicación está completamente desarrollada y está lista para ser lanzada al mercado de aplicaciones Android, nos planteamos la pregunta de qué es lo que va a pasar después y cómo puede nuestro proyecto ser viable y empiece a generar beneficios, en otras palabras, ahora nos toca el intentar hacer llegar la aplicación a todos los usuarios que puedan estar interesados en ella de una forma atractiva de forma que elijan ikeComercial preferentemente antes que otras aplicaciones que ofrece el mercado. Para ello, se debe elegir una estrategia de negocio y, en las siguientes líneas, vamos a describir cuáles son los modelos de negocio estándar elegidos por los desarrolladores en la actualidad:

- **Premium**

El modelo *premium*[25], por lo general, es el que nos hace desembolsar dinero de nuestro bolsillo previamente antes de su descarga que pueden variar desde el típico 0.99€ a cualquier precio que tenga asignada la aplicación para su adquisición completa.

- **Freemium**

Freemium[25] es el modelo que da la práctica de dar a tu aplicación la opción de poder descargarse gratuitamente sin ningún coste para el usuario pero con sólo algunas de las funciones que ofrece la aplicación. El resto de funcionalidades se adquieren pagando desde la misma aplicación ó directamente se ofrece la opción premium.

- **Inclusión de Anuncios**

Estas aplicaciones suelen llevar asociado una publicidad que es típicamente un banner de publicidad u otros tipos de publicidad la cual busca generar ingresos mediante impresión y clicks de banners publicitarios por parte del usuario. Algunos de los tipos más comunes de publicidad dentro de la aplicación son:

- *Banner de publicidad:* El banner permite a los desarrolladores de aplicaciones mostrar pequeños anuncios a lo largo de la aplicación, o en pantallas específicas.

## A. Modelo de negocio

- *Video*: Los anuncios de video permiten a los desarrolladores de aplicaciones mostrar videos cortos (de 15 a 30 segundos) durante pausas naturales en una aplicación.

Este modelo es bastante utilizado actualmente, pero muy molesto para el usuario cuando los anuncios son de carácter intrusivo e intermitente.

### ▪ Híbrido

Por supuesto, puedes implementar cualquiera de los modelos descritos anteriormente en varias combinaciones. Una de las estrategias más importantes de este modelo es ofrecer un contenido *Freemium* con anuncios *Premium* con compras en la aplicación. Es decir, la aplicación es totalmente gratuita pero si deseas la aplicación completa sin restricciones, tienes que pagar la compra de la aplicación.

Por otra parte, cuando se desarrollan aplicaciones *Freemium* con elementos consumibles, las aplicaciones suelen mostrar muros de ofertas ó vistas de videos incentivos. Otra de las estrategias a modo de ejemplo sería que pudiéramos lanzar una aplicación apoyada con publicidad e incluir elementos de compra en la aplicación para remover la publicidad.

## ikeComercial

En esta etapa se decidió por un modelo de negocio y, de los anteriormente descritos, no se seleccionó ninguno de ellos. La decisión ha sido poner a disposición de la comunidad la aplicación en Google Play de manera totalmente gratuita, sin ningún tipo de publicidad ni tampoco la restricción de uso de las funcionalidades que ofrece. Pero eso no es todo. También, se publicó el código fuente de este proyecto en GitHub que es una plataforma de desarrollo colaborativo de software, usado para alojar proyectos, el cual utiliza el sistema de control de versiones Git bajo la licencia de software libre Apache versión 2.0.

El código fuente puede verse desde este enlace:

<https://github.com/danigonlinea/ikeComercial>

Esta decisión viene a partir de una larga reflexión acerca de mi postura en el futuro de este proyecto. He pensado mucho acerca de este tema en los últimos meses y, desde mi punto de vista, veo más ventajas que inconvenientes tras esta decisión.

Las razones que voy a mencionar acerca de por qué usaré un repositorio como GitHub para el alojamiento del proyecto es, en parte, a las funcionalidades que esta plataforma ofrece. Voy a relatar algunas:

- Se evita estar intercambiando los ficheros del proyecto de un ordenador a otro.
- Es totalmente gratuito tanto su uso como el alojamiento de proyectos (aunque estos repositorios serán visibles para el público en general).



Figura A.1.: GitHub

- Se proporciona un gráfico detallado de cada colaborador que participe o esté trabajando en el proyecto, y de las "bifurcaciones" que se van creando en el mismo.
- Permite hacer "fork" y solicitar "pulls". Los "fork" es la clonación de un repositorio para realizar los cambios que necesites para adaptarlo a lo que ocupas, probablemente puedes enviárselos al dueño del código original y hacerle un "pull", donde él lo puede aceptar y darte el crédito merecido, ó puedes tener un "fork" independiente.

De esta manera se pretende evitar el efecto "*bus factor*" (factor autobús). Este término se refiere al hecho de qué pasaría a un proyecto si al desarrollador que está trabajando en él lo atropellara un autobús. El proyecto tendría que parar hasta que el desarrollador se recuperase ó, en el peor de los casos, no hubiera fallecido en el accidente con lo cuál en este caso nos podemos imaginar las situaciones nada favorables en que el proyecto estaría. Por supuesto esto es un término de lo más atípico pero pone en relieve una situación muy clara. El grado de dependencia que éste tiene hacia una ó un grupo de personas. Si el proyecto tiene colaboraciones de muchas personas y todas las personas pueden desarrollar en cualquier punto del proyecto, la dependencia es mínima. En caso contrario, cuando sólo una persona se dedica a un punto del proyecto y ésta deja de estar disponible, deja al proyecto con el impedimento de ser continuado.

## A. Modelo de negocio

Este proyecto, al ser *software* libre, permitirá que cualquiera que tenga interés en este proyecto pueda ver cómo está desarrollado, pudiendo ayudar a desarrollar otras ideas y aportar comentarios a aquél que lo desee, teniendo un acceso completo al mismo. El proyecto queda en manos de la comunidad de desarrolladores.

## B. Manual de instalación

En este anexo se presenta un breve manual de instalación de ikeComercial. Es necesario tener un *smartphone* con sistema operativo Android 2.2 (Froyo) o superior y tener una conexión a Internet.

### B.1. Prerrequisitos.

Para la correcta instalación y uso de la aplicación ikeComercial, tendremos que cumplir una serie de características técnicas que son las siguientes:

Características del *hardware*:

- Cualquier smartphone que se haya lanzado al mercado desde el año 2007 es totalmente suficiente para ejecutar la aplicación.
- Se recomienda, aunque no es estrictamente necesario, disponer de una pantalla de 4 pulgadas como mínimo.
- Disponer de conexión a Internet.
- Disponer de una cámara integrada.

Características del *software*:

- Sistema operativo Android versión 2.2 (Froyo) ó superior.
- Aplicación de lector de documentos PDF.
- Aplicación de correo electrónico.

## B.2. Instrucciones para la instalación de ikeComercial.

Para instalar la aplicación ikeComercial en nuestro smartphone Android, debemos realizar los siguientes pasos:

1. Conectarnos a Internet mediante un navegador web.

2. Teclear y acceder a la siguiente dirección:

a) <https://play.google.com/store/apps/details?id=es.danigonlinea.ventas>

3. Instalar la aplicación siguiendo las instrucciones que aparecerán en pantalla (véase Figura B.1, B.2 y B.3).

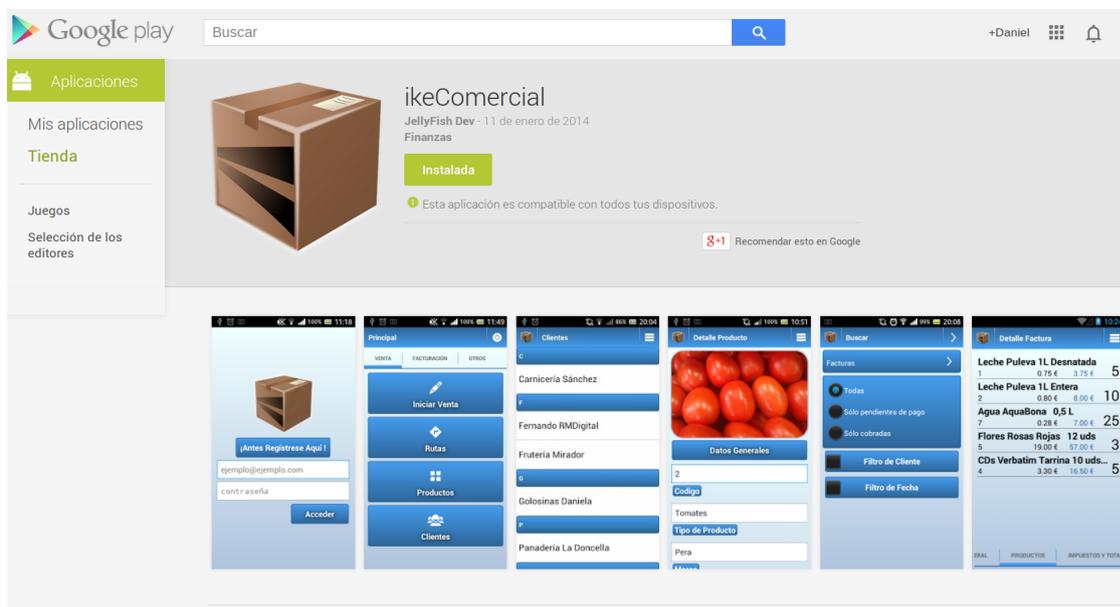


Figura B.1.: ikeComercial en Google Play.

## B.2. Instrucciones para la instalación de ikeComercial.

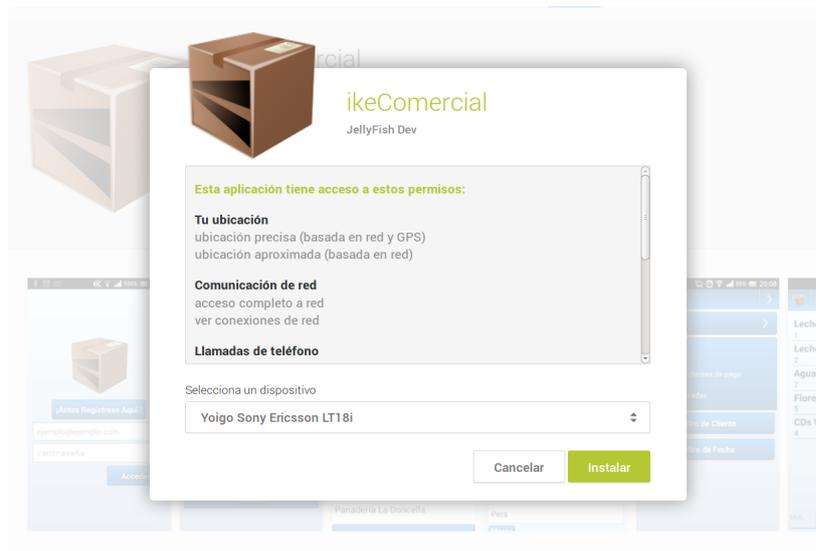


Figura B.2.: Confirmación de la instalación aceptando los permisos.



Figura B.3.: Instalando ikeComercial.

*B. Manual de instalación*

## C. Manual de usuario

En este anexo se detalla el manual de usuario de la aplicación ikeComercial. Encontrará también un apartado de información en el que aparecerán los datos de soporte técnico de la aplicación.

### C.1. Guía de uso.

El objetivo de este apartado es iniciar al usuario en el uso de ikeComercial. Para ello se le mostrará una breve explicación sobre como ejecutar las principales funciones de la aplicación.

#### C.1.1. Pantalla de Bienvenida.

Una vez instalada en nuestro smartphone, nos encontraremos con las siguientes pantallas:



Figura C.1.: Primeras pantallas de visualización.

Aquí podemos ver la pantalla de inicio de la aplicación. Para acceder a ella tenemos que registrar previamente la empresa con la que vamos a trabajar. Una vez realizado el registro debemos poner el correo electrónico y contraseñas indicados en

el formulario de registro. También podemos restablecer la contraseña mediante la formulación de una pregunta de seguridad.

### C.1.2. Pantalla de Menú Principal.

Ahora vamos a ver el menú principal de la aplicación:



Figura C.2.: Visualización del menú principal.

El menú principal está formado por tres pantallas con las cuales podemos acceder a todas las funcionalidades que nos ofrece esta aplicación deslizándonos horizontalmente por ellas. Para cerrar la sesión pulsaremos el botón situado en la parte superior derecha de la pantalla. Seguidamente, vamos a ver más detenidamente cada una de las funcionalidades que nos ofrece este menú principal.

### C.1.3. Pantalla de Gestión de Clientes.

Ahora veremos la gestión de clientes. Para ello solo tenemos que pulsar en el botón "Clientes" del menú principal. En la Figura C.3 vemos reflejado la lista de clientes registrados así como las opciones habilitadas para esta pantalla: registrar un nuevo cliente y buscar por nombre uno ya existente.

Si seleccionamos un cliente de la lista, la aplicación nos llevará a otra pantalla en la que podemos visualizar información detallada del cliente, como por ejemplo, sus datos personales.

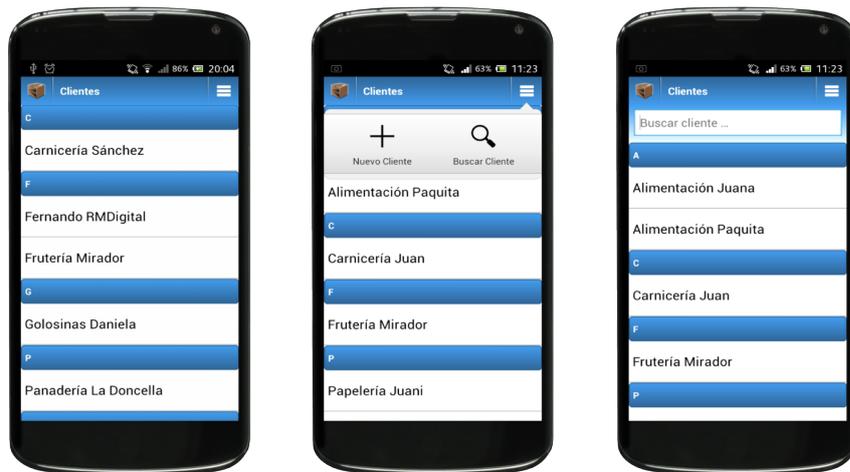


Figura C.3.: Visualización de la lista de clientes y sus opciones.

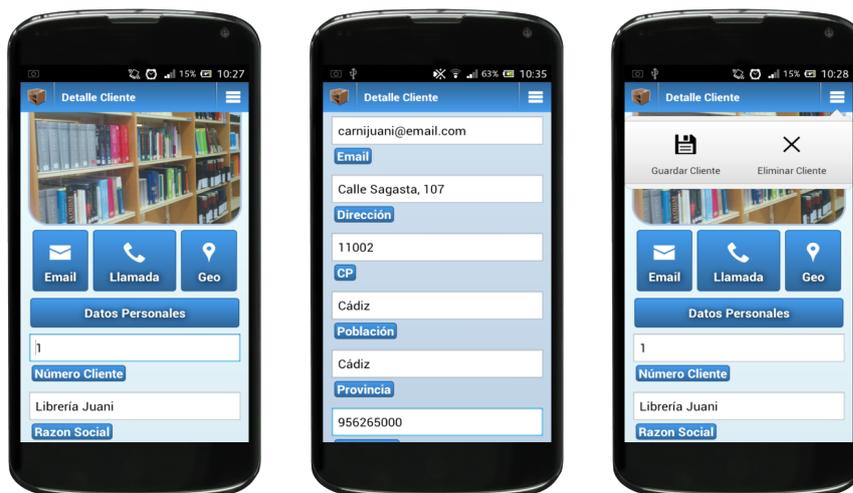


Figura C.4.: Vistas de la ficha de clientes.

## C. Manual de usuario

Si deseamos modificar un dato del formulario sólo tendremos, una vez modificado, guardar los cambios en las opciones que ofrece esta pantalla. Entre otras opciones también podemos encontrar el envío de un correo electrónico, llamada telefónica o geolocalización del cliente.

Para cambiar la foto de cliente tenemos que tocar en la foto. Automáticamente, en unos segundos, se cargará la cámara fotográfica. Una vez realizada la foto, la aplicación se encargará de capturarla en la ficha del cliente.

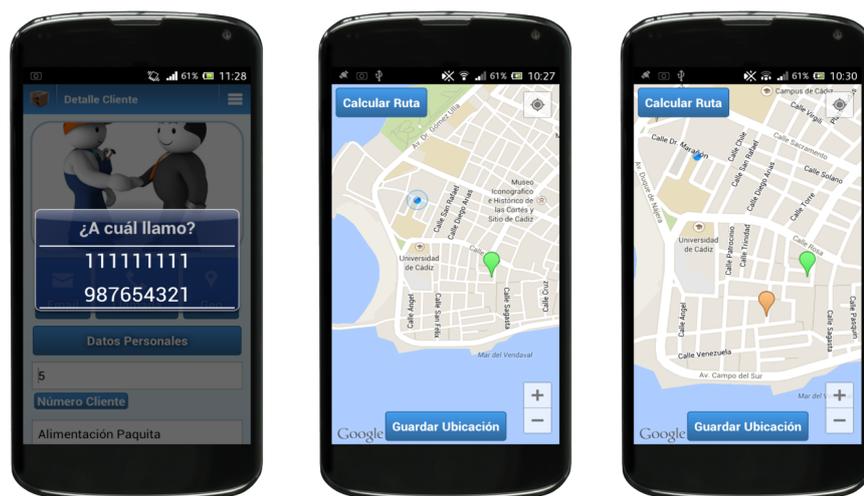


Figura C.5.: Visualizaciones del diálogo de llamada y del mapa de cliente.

Para realizar la captura de la ubicación de un cliente en un mapa, primero nos acercaremos a su dirección haciendo uso del zoom. Posteriormente tocamos en pantalla el lugar donde este cliente y automáticamente, la aplicación reconocerá la dirección y hará aparecer un marcador de color naranja. Si deseamos establecer la ubicación seleccionada, tocamos en la opción "Guardar Ubicación". El marcador se volverá verde la próxima vez que veamos su ubicación. Si deseamos ver la ruta más corta desde nuestra posición actual hacia el cliente, seleccionamos "Calcular Ruta". Se dibujará una línea verde mostrando la ruta a seguir.

### C.1.4. Pantalla de Gestión de Productos.

Ahora pasemos a ver la gestión de productos. Para ello, seleccionemos desde el menú principal el botón "Productos". En la Figura 10.6 podemos ver la lista de productos registrados y las opciones que podemos realizar en esta pantalla.

Si seleccionamos un producto de la lista, la aplicación nos llevará a otra pantalla (Figura 10.7) en la que podemos visualizar información detallada del producto seleccionado.



Figura C.6.: Visualización de la lista de productos.

Si deseamos modificar un dato del formulario sólo tendremos, una vez modificado, guardar los cambios en las opciones que ofrece esta pantalla.



Figura C.7.: Visualización del detalle de producto.

Para cambiar la foto de producto tenemos que tocar en la foto. Automáticamente, en unos segundos, se cargará la cámara fotográfica. Una vez realizada la foto, la aplicación se encargará de capturarla en la ficha del producto.

Podemos cambiar los valores de cada tipo de IVA desde el menú de configuración de la aplicación y, por otra parte, para poder englobar a un producto dentro de una categoría, primero debemos de crearla en la opción "Ver Categorías" visualizado en la Figura 10.6.

### C.1.5. Pantalla de Gestión de Rutas.

La primera pantalla que nos vamos a encontrar pulsando "Rutas" en el menú principal será una lista de todas las rutas creadas pudiéndolas filtrar mediante el día de la semana en la que se realizan. Para crear una nueva ruta, pulsaremos en el botón situado en la esquina superior derecha de la pantalla.

Si seleccionamos un elemento de la lista de rutas, procederemos a ir al detalle de la ruta seleccionada. Este detalle se dividirá en dos pantallas. Por una parte tendremos para modificar el nombre y el día de realización de la ruta y, por otra, los clientes que constituyen dicha ruta.

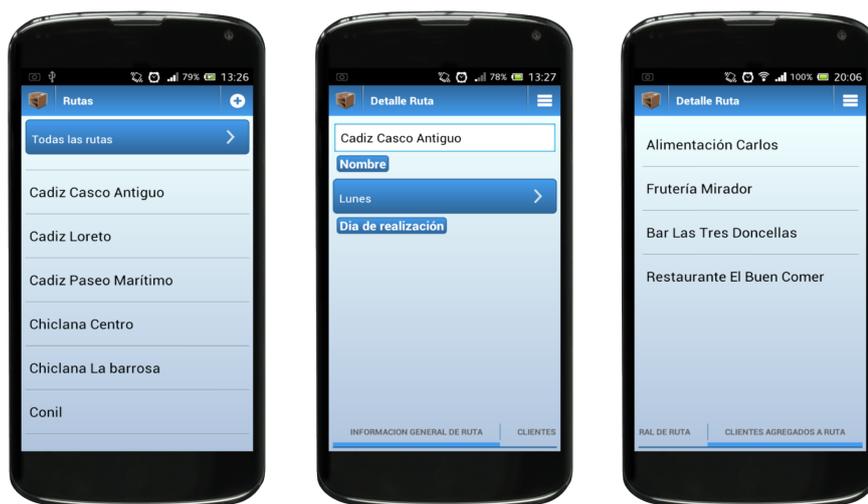


Figura C.8.: Visualización de la lista de rutas y el detalle de una de ellas.

Para añadir un cliente, pulsaremos la opción "Añadir Cliente" visualizada en la Figura 10.9. No es necesario que añadamos los clientes por orden de visita. Este orden puede ser modificado en cualquier momento. Para cambiar el orden de visualización, posicionándonos en la lista de los clientes añadidos a la ruta, pulsamos prolongadamente sobre uno de ellos. A continuación, el cliente se resaltará y podemos deslizarlo hacia arriba o hacia abajo dentro de la lista.

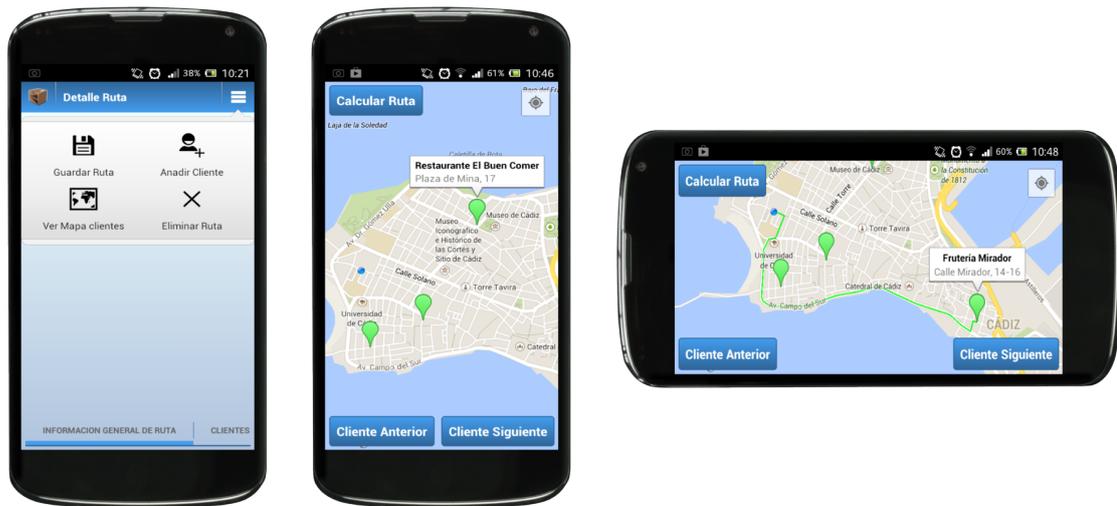


Figura C.9.: Visualización de opciones de ruta y mapas.

Las capturas de mapas en la Figura C.9 corresponden cuando presionamos la opción "Ver Mapa Clientes" en el detalle de la ruta. Para que un cliente pueda ser visualizado en la ruta, previamente tiene que haber sido geolocalizado en su ficha de cliente. Podemos movernos de un cliente a otro mediante los botones "Cliente Anterior" y "Cliente Siguiente". Si deseamos calcular la ruta entre nuestra posición y el cliente que actualmente está seleccionado, pulsaremos "Calcular Ruta". En unos segundos, podremos ver una línea que marca el camino a seguir para llegar a la ubicación del cliente.

### C.1.6. Pantalla de Inicio de Ventas.

Ahora pasemos a realizar una venta. Cuando pulsamos el botón "Iniciar Venta" del menú principal, nos aparecerá una lista de clientes. Pulsaremos el cliente con el que vamos a realizar una venta. A continuación, nos encontraremos con dos pantallas. En la primera vamos a ver información general de la venta y, en la segunda, los productos que añadamos en ella.

En la Figura C.11 podemos observar el proceso llevado a cabo para añadir un producto que se deriva de la opción "Añadir Producto". Por último, una vez añadido los productos a la venta, podemos procesarla en forma de pedido, albarán o factura.

En la lista de productos podemos encontrarnos con algún que otro producto con distinto color. Por una parte, aparecerán en color rojo aquellos vehículos que se han marcado como devolución. También aparecerán en color amarillo todo producto que, teniendo en cuenta su stock actual del vehículo, sobrepase la cantidad actual de dicho stock. Esta opción puede ser activada desde el menú de configuración de la aplicación, en concreto, con la opción "Controlar Stock".



Figura C.10.: Pantalla de Inicio de Ventas.



Figura C.11.: Añadiendo productos a la venta.

### C.1.7. Pantalla de Pedidos y Albaranes

Como podemos ver en la Figura C.12, la distribución de las pantallas de pedidos y albaranes son idénticas a las mostradas en la Figura C.10. Primeramente nos encontraremos con una lista de los pedidos ó albaranes actuales para después, en la siguiente pantalla, ver su detalle y las opciones disponibles para cada uno de ellos.

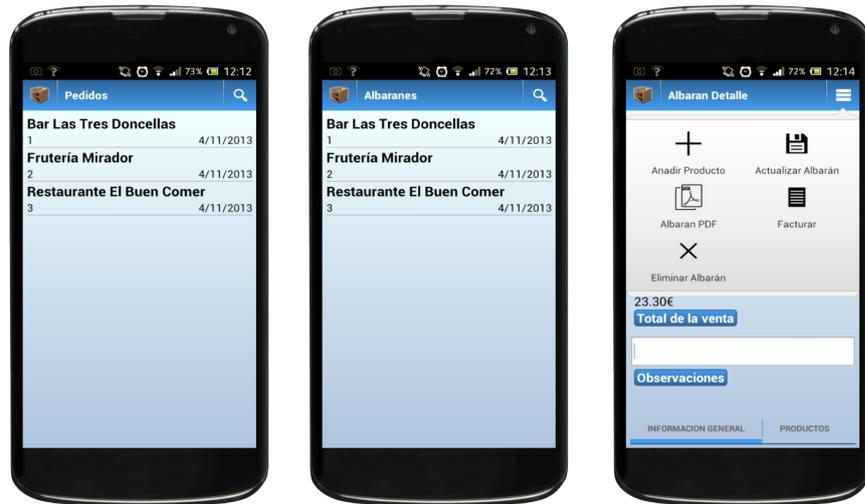


Figura C.12.: Visualización de listas de pedidos y albaranes.

### C.1.8. Pantalla de Facturas.

A continuación describiremos el turno de las facturas. Presionando el botón "Facturas" del menú principal nos encontramos con dos pantallas, las cuáles son listas de facturas. Una lista pertenecerá a las facturas que se encuentran pendientes y la otra lista serán aquellas que se encuentren pagadas.

Una vez seleccionada una factura de cualquier lista, podremos ver su detalle. Su visualización corresponde a la Figura C.14 y en concreto están constituidas por tres pantallas. La primera pantalla nos da una información general de la factura. En la segunda están reunidos todos los productos que forman la factura y finalmente, en la tercera pantalla, ver el detalle de los impuestos que se aplican a ésta así como su importe total.

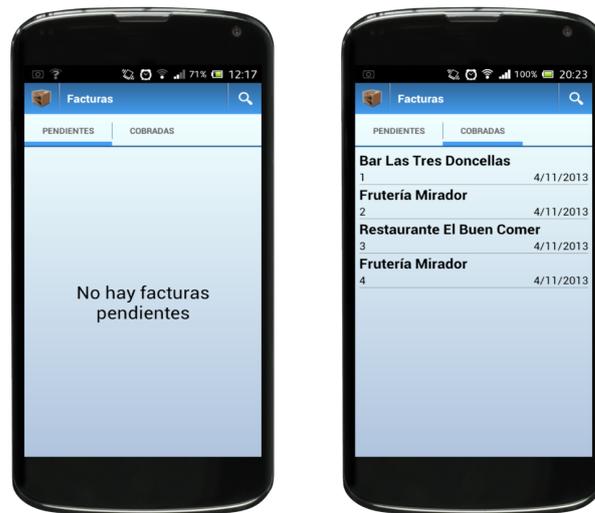


Figura C.13.: Visualización de listas de facturas pendientes y cobradas.

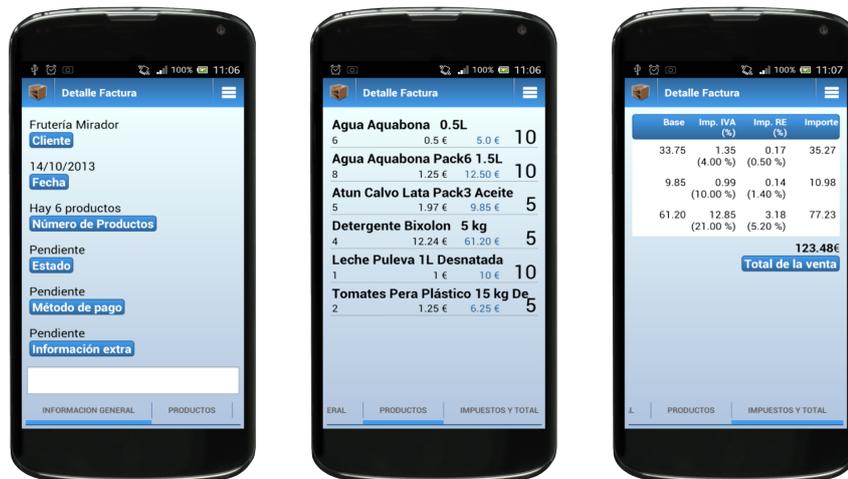


Figura C.14.: Visualización del detalle de una factura.

### C.1.9. Pantalla de Cobro de Factura.

Ahora vamos a ver la opción "Cobrar Factura" que ofrece la visualización del detalle de la factura. Seguidamente podemos ver el dinero a abonar de la factura y, mediante el método en efectivo, realizar un cobro total ó parcial de la factura. La factura se dará por pagada cuando se realice el abono de la totalidad de la factura.



Figura C.15.: Visualización de la pantalla de cobro de factura.

### C.1.10. Pantalla de Generación de PDF.

Para la generación en formato PDF de la factura, pulsamos la opción "Factura PDF". Posteriormente no llevará a una pantalla como se puede observar en la siguiente Figura.



Figura C.16.: Visualización de creación de PDF de factura.

Inicialmente, tendremos que pulsar el botón "Crear PDF". En unos segundos, se generará el PDF de la factura y se activarán dos botones: "Ver PDF" y "Enviar PDF". Podremos visualizar el PDF en pantalla y enviar el PDF por correo electrónico respectivamente.

### C.1.11. Pantalla de Búsqueda.

Esta pantalla es accesible a través de la opción "Buscar" situadas en las pantallas tanto de pedidos y albaranes como de facturas. En primer lugar tenemos que decir qué tipo de documento vamos a buscar y en segundo lugar elegir los filtros (si se desean) de búsqueda. En concreto podemos realizar dos filtros: filtro de cliente y filtro de fecha. Como opción adicional, en las facturas se tendrá un filtro en el que se podrá buscar entre todas las facturas, sólo las pendientes de pago ó sólo las que estén ya cobradas.



Figura C.17.: Visualización de las pantallas de búsqueda.

### C.1.12. Pantalla de Cobros.

Lo primero que nos encontraremos en esta pantalla será una lista de todos los clientes registrados con una información adicional: las facturas y los albaranes pendientes de cada uno de ellos. En este caso, el dígito que se encuentra dentro de un recuadro con fondo amarillo será el que represente el número de albaranes pendientes del cliente. Por otra parte, el dígito del recuadro con fondo rojo representará el número de facturas pendientes del cliente.

Una vez seleccionado uno de los clientes de la lista, podemos ver una distribución de dos pantallas en las cuales podemos ver listados los albaranes y facturas pendientes de ese cliente. En el caso de que seleccionemos un albarán o una factura, automáticamente nos llevará a la pantalla de detalle de esa factura (véase Figura C.14).

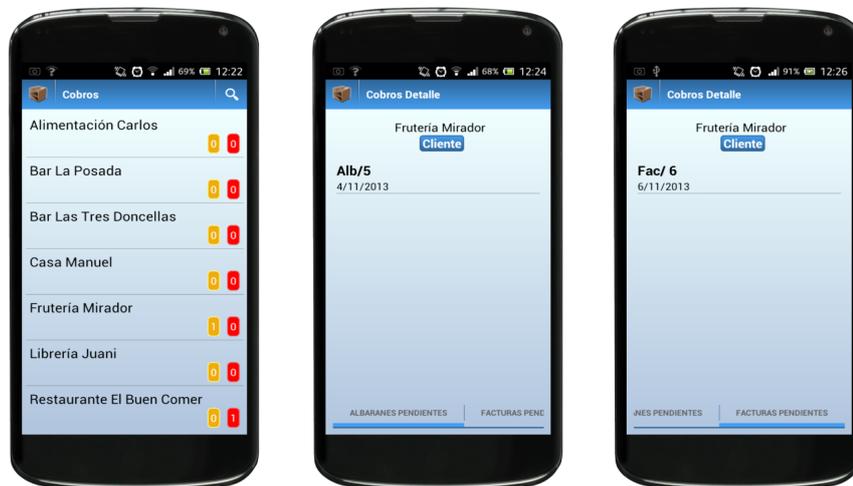


Figura C.18.: Pantallas de Gestión de Cobros.

### C.1.13. Pantalla de Carga de vehículo.

Cuando pulsemos el botón "Carga del Vehículo", observaremos una lista de productos. Por cada producto veremos reflejado su código y su stock actual en el vehículo. Pulsando en uno de ellos, se nos abrirá un diálogo para modificar el stock actual del producto en el vehículo. Pulsando prolongadamente nos dará la opción de descargarlo.

También podemos ver las opciones que esta pantalla nos ofrece: cargar más productos en el vehículo ó descargarlo completamente.



Figura C.19.: Pantallas de carga del vehículo.

### C.1.14. Pantalla de Exportación.

Pasamos a la opción de "Exportación" del menú principal. En ella podremos exportar los asientos contables generados de nuestras facturas para su posterior adición al programa de gestión Sage Contaplus. Para ello pulsamos sobre la opción de "Crear Exportación". A continuación, tendremos que insertar el intervalo de fechas de facturas que queremos exportar y presionamos en el botón situado en la esquina superior derecha.

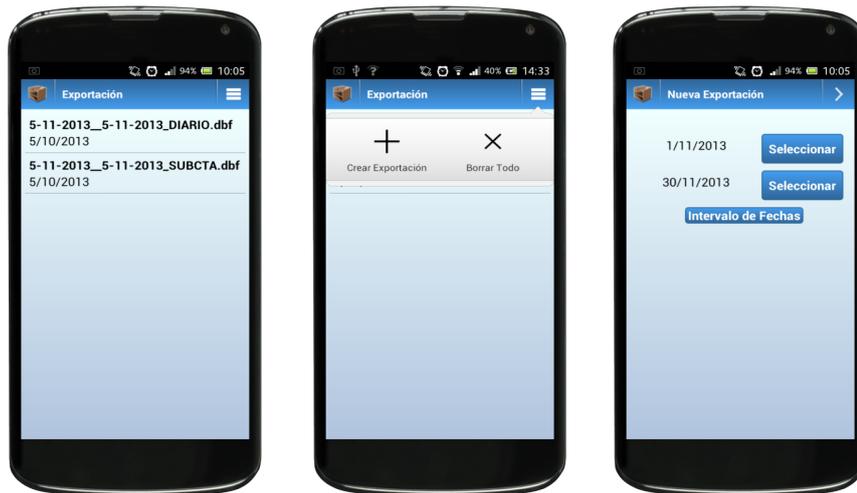


Figura C.20.: Pantallas de exportación.

La realización de esta acción generará en una lista dos ficheros en formato dbf. Uno corresponde a la cuentas diarias de los asientos contables y el otro corresponde a la descripción de las subcuentas de los asientos contables incluidos en la exportación. Para más información acerca de estos ficheros visite el Anexo Exportación Sage Contaplus E.

### C.1.15. Pantalla de Estadísticas.

En esta pantalla podremos elegir entre dos tipos de estadísticas: facturación de ventas realizadas y productos más facturados en un determinado intervalo de tiempo. A continuación, tendremos que configurar una fecha de referencia a la hora de hacer la estadística. Teniendo en cuenta esa fecha indicada, podremos filtrar la estadística por día, mes y año. Una vez indicado el filtro, pulsaremos el icono situado en la esquina superior derecha para continuar.



Figura C.21.: Pantallas de estadísticas

Se generará la estadística conforme al filtro seleccionado. Un ejemplo de ellas se puede ver en la Figura C.22.



Figura C.22.: Ejemplo de visualización de estadísticas.

### **C.1.16. Pantalla de configuración.**

En la Figura C.23 podemos ver capturas de la configuración de la aplicación. A continuación vamos a explicar cada una de ellas:

1. Empresa: esta opción nos permite modificar todos los datos de la empresa, entre ellas, el logo actual para mostrar en las facturas. Para entrar a ella debemos identificarnos con la contraseña especificada en el formulario de registro.
2. Impuestos: esta opción nos permite modificar los impuestos del IVA (Impuesto Válido Adicional) y RE (Recuento de Equivalencia) aplicados en territorio nacional español.
3. Contabilidad: esta opción nos llevará a opciones de contabilidad, entre la que se encuentra la longitud de la subcuenta de un asiento contable.
4. Aviso de Riesgo: esta opción nos mostrará un aviso en el inicio de una nueva venta a un determinado cliente si éste sobrepasa el riesgo que hemos establecido en su ficha de cliente.
5. Controlar Stock: esta opción nos controlará el stock de los productos almacenados en el vehículo y, en una nueva venta, nos resaltará en color amarillo el producto que lo supere.
6. Hacer Copia Seguridad: esta opción nos realizará una copia de la base de datos actual y la guardará dentro del directorio de la aplicación.
7. Restaurar Copia Seguridad: esta opción nos restaurará la copia de seguridad que hayamos seleccionado previamente en la opción "Selecciona Copia Seguridad".
8. Selecciona Copia Seguridad: esta opción nos mostrará una lista seleccionable de las copias de seguridad realizadas.
9. Acerca de: esta opción nos visualizará en pantalla información general acerca de la aplicación ikeComercial (Figura 10.24).

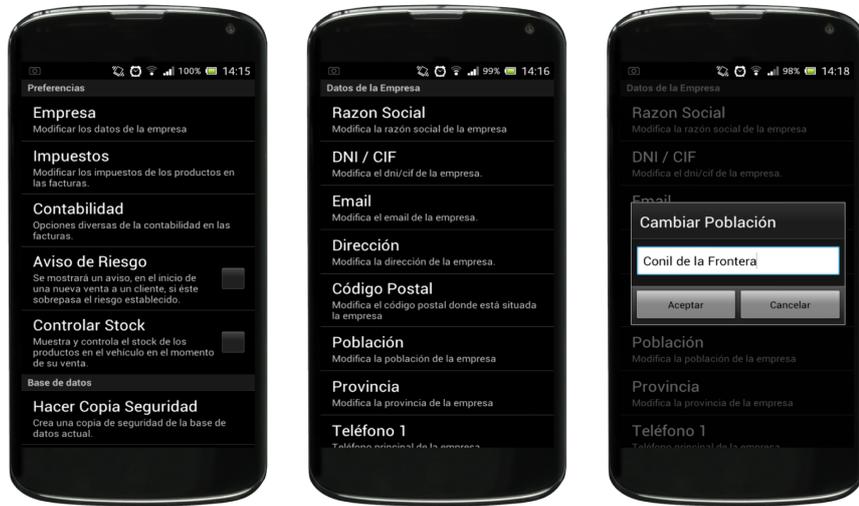


Figura C.23.: Pantallas de configuración.



Figura C.24.: Pantalla acerca de.

## **C.2. Información de contacto.**

Desarrollador: Daniel González Jiménez

Organización: Escuela Superior de Ingeniería, Universidad de Cádiz

Teléfono de contacto: +34 664 847 808

Correo electrónico: [dani.gonzalezjimenez@alum.uca.es](mailto:dani.gonzalezjimenez@alum.uca.es)

*C. Manual de usuario*

## D. Estructura de un proyecto Android en ADT Bundle

En este anexo, se va a explicar cómo se distribuye un proyecto Android cuando utilizamos el IDE ADT Bundle.

- *src*: Carpeta que contiene todo lo relacionado con el código fuente de la aplicación. Como puedes observar los ficheros Java se almacenan en un espacio de nombres ó también llamados paquetes.
- *gen*: Carpeta que contiene el código generado de forma automática por el SDK. Nunca hay que modificar de forma manual estos ficheros. Dentro encontraremos
  - *BuildConfig.java*: Define la constante `DEBUG` para que desde Java puedas saber si tu aplicación está en fase de desarrollo.
  - *R.java*: Define una clase que asocia los recursos de la aplicación con identificadores. De esta forma los recursos podrán ser accedidos desde Java.
  - *Android x.x*: Código JAR, el API de Android según la versión seleccionada.
  - *Android Dependencies*: Librerías asociadas al proyecto.
- *assets*: Carpeta que puede contener una serie arbitraria de ficheros o carpetas que podrán ser utilizados por la aplicación (ficheros de datos, fuentes, . . .). A diferencia de la carpeta `res`, nunca se modifica el contenido de los ficheros de esta carpeta ni se les asociará un identificador.
- *bin*: En esta carpeta se compila el código y se genera el `.apk`, fichero comprimido que contiene la aplicación final lista para instalar.
- *libs*: Código JAR con librerías que quieras usar en tu proyecto. Se ha añadido una librería `android-support` cuyo objetivo es añadir nuevas funcionalidades que no aparecían en el nivel de API 4 (recuerda que es el nivel de API mínimo que hemos indicado) y que aparecieron en versiones más recientes del SDK.

## D. Estructura de un proyecto Android en ADT Bundle

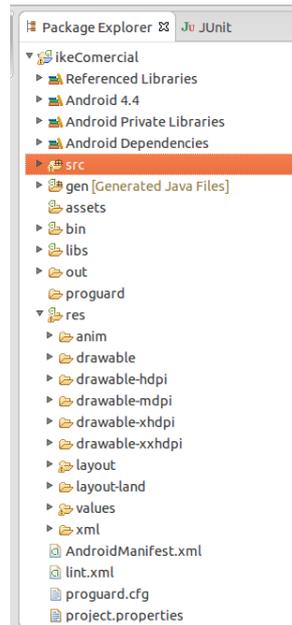


Figura D.1.: Estructura de un proyecto Android.

- *res*: Carpeta que contiene los recursos usados por la aplicación. Como veremos en el siguiente capítulo las subcarpetas pueden tener un sufijo si queremos que el recurso solo se cargue al cumplirse una condición. Por ejemplo *-hdpi* significa que solo ha de cargar los recursos contenidos en esta carpeta cuando el dispositivo donde se instala la aplicación tiene una densidad gráfica alta (>180dpi); *-v11* significa que el recurso solo ha de cargarse en un dispositivo con nivel de API 11 (v3.0).
  - *drawable*: En esta carpeta se almacenan los ficheros de imágenes (JPG o PNG) y descriptores de imágenes en XML.
  - *layout*: Contiene ficheros XML con vistas de la aplicación. Las vistas nos permitirán configurar las diferentes pantallas que compondrán la interfaz de usuario de la aplicación. Se utiliza un formato similar al HTML usado para diseñar páginas Web. Serán tratadas en el siguiente capítulo.
  - *values*: También utilizaremos ficheros XML para indicar valores del tipo string, color o estilo. De esta manera podremos cambiar los valores sin necesidad de ir al código fuente. Por ejemplo, nos permitiría traducir una aplicación a otro idioma.
  - *anim*: Contiene ficheros XML con animaciones Tween. Las animaciones son descritas al final del capítulo 4.

- *xml*: Otros ficheros XML requeridos por la aplicación.
- *AndroidManifest.xml*: Este fichero describe la aplicación Android. En él se indican las actividades, intenciones, servicios y proveedores de contenido de la aplicación. También se declaran los permisos que requerirá la aplicación. Se indica la versión mínima de Android para poder ejecutarla, el paquete Java, la versión de la aplicación, etc.
- *proguard-project.txt*: Fichero de configuración de la herramienta ProGuard, que te permite optimizar y ofuscar el código generado. Es decir, se obtiene un .apk más pequeño y donde resulta más difícil hacer ingeniería inversa.
- *default.properties*: Fichero generado automáticamente por el SDK. Nunca hay que modificarlo. Se utiliza para comprobar la versión del API y otras características cuando se instala la aplicación en el terminal.
- *Imágenes alternativas*: La densidad de pantalla mide la cantidad de píxeles en un área determinada. Este termino también es referido como dpi (dots per inch) o píxeles por pulgada. Si el dispositivo tiene una resolución elevada en una pantalla de poco tamaño físico, tendrá un valor de dpi muy elevado. Por lo tanto es conveniente proporcionar diferentes imágenes de diferente calidad dependiendo de la citada densidad de la pantalla del dispositivo. Android usa 4 límites de densidad de pantalla. Por lo tanto disponemos de 4 sufijos para crear los recursos alternativos:
  - ldpi. Recursos para densidades de pantalla de 120dpi.
  - mdpi. Recursos para densidades de pantalla de 160dpi.
  - hdpi. Recursos para densidades de pantalla de 240dpi.
  - xhdpi. Recursos para densidades de pantalla de 320dpi.

#### *D. Estructura de un proyecto Android en ADT Bundle*

## E. Ficheros contables para importación en Sage ContaPlus.

La importación de asientos es la única forma permitida por SP ContaPlus para introducir datos desde otro programa en el Libro Diario, bien en formato ASCII o bien en XBase. En este caso utilizaremos ficheros en formato XBase.

Por consiguiente, vamos a poder introducir datos desde nuestra aplicación en el diario de contabilidad. Para poder realizar este proceso, se deben cumplir una serie de normas, de forma que la importación reconozca los datos. En primer lugar, debe existir un fichero intermedio de traspaso que sirva de puente entre nuestra aplicación y ContaPlus.

Hay que generar dos ficheros distintos, uno que recoja los apuntes del Diario y otro que recoja los datos de las Subcuentas. Si se van a utilizar subcuentas que no están definidas en el Plan de Cuentas de SP ContaPlus será preciso crear otro fichero de enlace que contenga las nuevas subcuentas (en esta aplicación en particular no será necesario).

Ahora vayamos a ver la estructura general de estos ficheros generados.

Pos	Campo	Tipo	Long	Dec	Descripción
1	Asien	NUME	6		Número del asiento
2	Fecha	DATE	8		Fecha del asiento
3	Texto	CHAR	12		Código de la subcuenta
4	Texto	CHAR	12		Código de la contrapartida del IVA
5	PtaDebe	NUME	16	2	Importe al debe en pesetas
6	Concepto	CHAR	25		Concepto del asiento
7	PtaHaber	NUME	16	2	Importe al haber en pesetas
8	Factura	NUME	8		Número de factura al IVA
9	BaseImpo	NUME	16	2	Base Imponible del IVA en pesetas
10	IVA	NUME	5	2	Tanto por ciento de IVA
11	Recequiv	NUME	5	2	Tanto por ciento de recargo equivalencia
12	Documento	CHAR	10		Número de documento
13	Departa	CHAR	3		Código de departamento
14	Clave	CHAR	6		Código del proyecto
15	Estado	CHAR	1		Punteo(interno)

*E. Ficheros contables para importación en Sage ContaPlus.*

16	Ncasado	NUME	6		Numérico de casación (interno)
17	TCasado	NUME	16		Tipo de casado (interno)
18	Trans	NUME	16		Número de pago
19	Cambio	NUME	16	6	Cambio a aplicar
20	DebeME	NUME	1	2	Importe debe moneda extranjera
21	HaberME	NUME	1	2	Importe haber moneda extranjera
22	Auxiliar	CHAR	4		registro auxiliar (interno)
23	Serie	CHAR	5		Serie de facturación
24	Sucursal	CHAR	16		Nombre de sucursal (opcional)
25	CodDivisa	CHAR	1		Código de divisa
26	ImpAuxME	NUME	16	2	Importe auxiliar moneda extranjera
27	MonedaUso	CHAR	1		1) Ptas. 2)Euros
28	EuroDebe	NUME	16	2	Importe al debe en euros
29	EuroHaber	NUME	16	2	Importe al haber en euros
30	BaseEuro	NUME	16	2	Base Imponible del IVA en euros
31	NoConv	LOGIC	1		anotación convencional (interno)
32	NumeroInv	NUME	10		Código de Activo invernial.

Tabla E.2.: Estructura de fichero Diario.

Ahora vamos a ver la estructura del fichero de subcuentas:

Pos	Campo	Tipo	Long	Descripción
1	Codigo	CHAR	12	Código de la subcuenta
2	Título	CHAR	40	Título de la subcuenta
3	CIF	CHAR	15	CIF
4	Domicilio	CHAR	35	Domicilio
5	Población	CHAR	25	Población
6	Provincia	CHAR	20	Provincia
7	CodPostal	CHAR	5	Código Postal
8	Divisa	LOGIC	1	Subcuenta moneda extranjera

Tabla E.4.: Estructura de fichero Subcuenta.

Es muy importante establecer en los apuntes la fecha, el número de asiento y la moneda de uso para un correcto funcionamiento de la importación. Hay que prestar mucho atención a los apuntes del IVA para rellenar todos los datos necesarios para los libros correspondientes. Si no quisiéramos llevar los libros del IVA, tendríamos que modificar las cuentas especiales del programa en lo concerniente a este impuesto.

En la Figura E.1 se puede observar cómo se introducen ambos ficheros para su importación ó exportación al programa Sage ContaPlus.

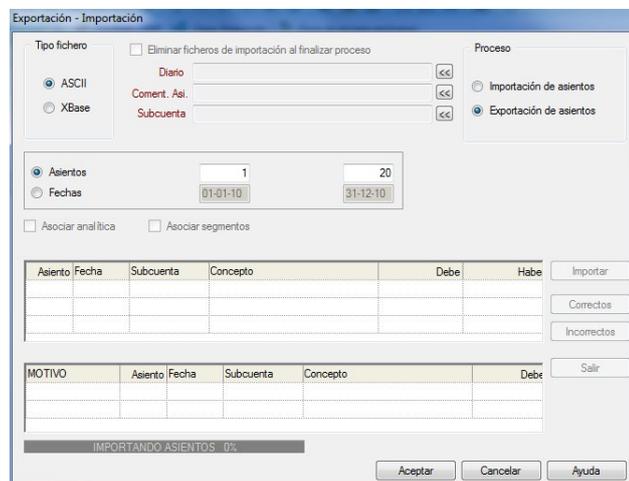


Figura E.1.: Importando/Exportando con Sage ContaPlus.

*E. Ficheros contables para importación en Sage ContaPlus.*

## F. Licencia Apache 2.0

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on

## *F. Licencia Apache 2.0*

behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as

part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or

*F. Licencia Apache 2.0*

Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

in terms and conditions terminated  
END OF TERMS AND CONDITIONS

# Bibliografía

- [1] ADT Bundle, <http://developer.android.com/sdk/index.html>
- [2] Android, <http://www.android.com/>
- [3] Datatypes In SQLite Version 3, <http://www.sqlite.org/datatype3.html>
- [4] Genymotion, <http://www.genymotion.com/>
- [5] Google Nexus 4, <https://www.google.es/nexus/4/>
- [6] Google Play, mercado de aplicaciones de Android, <https://play.google.com/store?hl=es>
- [7] GreenDroid, <http://greendroid.cyrilmottier.com/>
- [8] Inkscape, <http://www.inkscape.org/es/>
- [9] Invoice2go, <http://www.invoice2go.com.au/>
- [10] ISO-8601, Date and Time Formats, <http://www.w3.org/TR/NOTE-datetime>
- [11] Lenguaje XML, <http://www.w3.org/XML/>
- [12] Más Ventas, <https://play.google.com/store/apps/details?id=com.claveinformatica.masventas>
- [13] MobileBiz Co, Invoice App, <http://mobilebizco.com/>
- [14] Modelo en cascada, [http://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](http://es.wikipedia.org/wiki/Desarrollo_en_cascada)
- [15] Personal Digital Assistant (PDA), [http://en.wikipedia.org/wiki/Personal\\_digital\\_assistant](http://en.wikipedia.org/wiki/Personal_digital_assistant)
- [16] Sage Contaplus, <http://sage.es/Software/Contabilidad/ContaPlus>
- [17] SharedPreferences, <http://developer.android.com/reference/android/content/SharedPreferences.html>

## Bibliografía

- [18] Skydrive, <https://skydrive.live.com/>
- [19] Smartphone, <http://www.areatecnologia.com/Que-es-un-smartphone.htm>
- [20] Sony Ericsson Arc S, <http://www.sonymobile.com/global-es/products/phones/xperia-arc-s/>
- [21] Tablets, [http://es.wikipedia.org/wiki/Tableta\\_\(computadora\)](http://es.wikipedia.org/wiki/Tableta_(computadora))
- [22] The official site for SQLite, <http://www.sqlite.org/>
- [23] Ubuntu One, <https://one.ubuntu.com/>
- [24] Unified Modeling Language, <http://www.uml.org/>
- [25] Wikipedia, La Enciclopedia Libre, <http://es.wikipedia.org>
- [26] The official site for Android developers (Jul 2012), <http://developer.android.com/index.html>
- [27] Darwin, I.F.: Android CookBook. O'Reilly Media (2012)
- [28] Developers, G.: Testing Fundamentals (Jun 2011), [http://developer.android.com/tools/testing/testing\\_android.html](http://developer.android.com/tools/testing/testing_android.html)
- [29] Inc., G.: Android UI Testing (2011), [http://developer.android.com/tools/testing/testing\\_ui.html](http://developer.android.com/tools/testing/testing_ui.html)
- [30] Lockwood, A.: Correctly Managing Your SQLite Database (Jul 2007), <http://www.androiddesignpatterns.com/2012/05/correctly-managing-your-sqlite-database.html>
- [31] O'Conner, J.: The need of BigDecimal (Jul 2007), [https://blogs.oracle.com/CoreJavaTechTips/entry/the\\_need\\_for\\_bigdecimal](https://blogs.oracle.com/CoreJavaTechTips/entry/the_need_for_bigdecimal)
- [32] Peck, A.: Beginning GIMP : from novice to professional. Apress (2008)
- [33] Penalva, J.: Android ya acapara el 80 por ciento de cuota de mercado en smartphones. Xataka (2013), <http://www.xatakandroid.com/tag/cuota-de-mercado>
- [34] Pereira, J.M.: Androcode (Oct 2011), <http://androcode.es>
- [35] Rodríguez, A.: Androideity (Jul 2011), <http://androideity.com/>
- [36] Rodríguez, A.: Máquina Virtual Dalvik (Jun 2011), <http://androideity.com/2011/07/07/la-maquina-virtual-dalvik/>