



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# **GESTIÓN Y ACCESO A LOS SERVICIOS DE UNA ORGANIZACIÓN O EMPRESA MEDIANTE APLICACIÓN MÓVIL**

**Alumno: Francisco Javier Gallego Beltrán**

Tutor: Prof. D. Pedro José Sánchez Sánchez  
Dpto: Informática

**Septiembre, 2017**



Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Don Pedro José Sánchez Sánchez , tutor del Proyecto Fin de Carrera titulado: Gestión y Acceso a los Servicios de una Organización o Empresa mediante Aplicación móvil, que presenta Francisco Javier Gallego Beltrán, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2017

El alumno:

Los tutores:

Francisco Javier Gallego Beltrán

Pedro José Sánchez Sánchez

## **Agradecimientos**

En primer lugar quiero agradecer a mis padres todo el apoyo recibido cuando más lo he necesitado, sin ellos todo esto no sería posible.

A mi hermana, por siempre estar ahí y hacer que me sienta bien y ayudarme a no tirar la toalla.

En una mención especial, a mi hermano, que me ha ayudado en todo lo que he necesitado sin poner ninguna pega.

Y como no, a mi tita Agueda, que siempre se está preocupando por mi y siempre intenta que todo me vaya bien.

Y por último, a mis amigos, por ser la vía de escape de toda la presión que todo esto conlleva.

## Índice

CAPÍTULO 1: Introducción .....	7
1.1. Motivación .....	8
1.2. Objetivos .....	9
1.3. Resumen de la memoria .....	10
CAPÍTULO 2: Servicio Web.....	13
2.1. Introducción .....	14
2.2. Servicio Web .....	14
2.2.1. ¿Qué es un Servicio Web? .....	14
2.2.2. ¿Qué podemos hacer con los Servicios Web? .....	15
2.2.3. Ventajas de los Servicios Web.....	16
2.2.4. Desventajas de los Servicios Web .....	16
2.2.5. Por qué utilizar un Servicios Web .....	16
2.3. MySQL .....	17
2.3.1. ¿Qué es MySQL? .....	17
2.3.2. Aplicaciones .....	18
2.3.3. Ventajas de MySQL .....	18
2.3.4. Desventajas de MySQL .....	19
2.3.5. Por qué usar MySQL .....	19
2.4. Tomcat.....	20
2.4.1. ¿Qué es Tomcat? .....	20
2.4.2. Estado de su desarrollo .....	20
2.4.3. Entorno.....	21
2.4.4. Historia de Tomcat.....	21
2.5. Diferentes tecnologías para realizar Servicios Web .....	22
2.5.1. Tipos de tecnologías.....	22
2.5.1.1. SOAP .....	22
2.5.1.2. REST .....	23
2.5.2. ¿Qué tecnología vamos a utilizar? .....	23
CAPÍTULO 3: Android .....	25
3.1. ¿Qué es android? .....	26
3.2. Características .....	27
3.3. Arquitectura.....	31
3.4. Por qué usar Android .....	32

CAPÍTULO 4: Análisis .....	33
4.1. Especificación de Requisitos.....	34
4.2. Análisis del Sistema .....	41
4.2.1. Casos de uso.....	41
4.2.1.1. Actores.....	41
4.2.1.2. Caso de uso del actor Usuario .....	43
4.2.1.3. Caso de uso del actor Cliente .....	48
4.2.1.4. Caso de uso del actor Empleado .....	62
4.2.2. Diagramas de secuencia .....	71
CAPÍTULO 5: Diseño .....	85
5.1. Introducción .....	86
5.2. Diagrama de clases.....	86
5.3. Detalle de las clases .....	87
5.4. Diagrama de clases del Servicio Web .....	89
5.5. Diseño de la base de datos .....	91
5.5.1. Descripción de las Entidades.....	91
5.5.2. Diagrama Entidad-Relación .....	95
CAPÍTULO 6: Implementación .....	97
6.1. Lenguajes de programación utilizados .....	98
6.1.1. Java.....	98
6.1.2. JSON.....	99
6.2. Herramientas utilizadas para el desarrollo.....	101
6.2.1. Android studio.....	101
6.2.2. Netbeans .....	102
6.2.3. Phpmyadmin.....	104
6.3. Librerías auxiliares utilizadas.....	105
6.3.1. Hibernate.....	105
CAPÍTULO 7: Instalación y despliegue del Servidor Web.....	107
7.1. Introducción .....	108
7.2. Instalación y configuración de Apache2 .....	109
7.3. Instalación de Java .....	110
7.4. Instalación de MySQL-Server.....	110
7.5. Instalación y configuración de phpMyAdmin.....	110
7.6. Importación de las tablas de la base de datos.....	111
7.7. Instalación y configuración de Apache Tomcat 7.....	111
7.8. Instalación y configuración de Nginx .....	113

ANEXO 1: Manual de instalación de la aplicación móvil .....	115
ANEXO 2: Manual de usuario.....	119
ANEXO 3: Storyboard .....	123

# CAPÍTULO 1

---

Introducción

## 1.1. Motivación

Actualmente el desarrollo de aplicaciones móviles está teniendo una gran demanda gracias al gran número de dispositivos que existen en el mercado y a las nuevas necesidades que han surgido por su uso cotidiano.

Hoy en día, un gran número de personas disponen de un dispositivo móvil con la capacidad para conectarse a internet, ya sea para navegar, o para utilizar distintas aplicaciones.

Las previsiones indican que el crecimiento en el sector es y va a ser muy elevado. Crecerá cinco veces más rápido que la capacidad que tienen las organizaciones de TI para poder desarrollarlas. [1]

Este tipo de tecnología ofrece también un valor añadido a las organizaciones y empresas que lo utilizan.

Dado que este sector está en pleno crecimiento, en este Trabajo de Fin de Grado voy a desarrollar una aplicación móvil que permita a los usuarios/empleados poder acceder a los servicios que ofrece una organización o empresa.

En este trabajo, me centraré en crear una Aplicación Android para el acceso y gestión a los servicios de una empresa, la cual nos va a permitir comprar y gestionar todos los productos que dicha empresa ofrece. Además de esto, también voy a crear el Servicio Web que sustente la lógica de la aplicación.

Gracias al uso de estas tecnologías, vamos a facilitar a los usuarios/empleados el acceso a todos los productos de la empresa para poder comprar o gestionarlos, dependiendo del tipo de usuario que seas. Si eres un usuario(cliente) solo podrás comprarlos, pero si eres un empleado, a parte de poder comprarlos, también tendrá acceso a la gestión de todos los productos.

Además de todo esto, también se nos permitirá la gestión de comentarios, para mejorar la compra de los productos.



No solo podremos comprar un solo producto, sino también podremos realizar una lista de productos para poder hacer compras múltiples.

A parte de poder comprar, los usuarios también podrán dejar un comentario en el producto comprado, para así orientar a próximos compradores sobre este.

## **1.2. Objetivos**

Los objetivos generales del proyecto son los siguientes:

- Elegir un tipo de organización o empresa para la realización del trabajo.
- Definir un conjunto de servicios representativos y establecer sus necesidades de gestión y acceso por parte de los clientes/usuarios.
- Realizar un estudio comparativo de las diferentes tecnologías existentes y elegir la que mejor se adapte a la solución propuesta.
- Desarrollar los servicios Web asociados así como la definición e implementación de la Base de Datos para el almacenamiento de la información necesaria.
- Desarrollar un prototipo de la aplicación móvil para la gestión y uso de estos servicios.
- Redactar una memoria que recoja todo el trabajo desarrollado así como los manuales de instalación y usuario.

El objetivo de este trabajo es crear una aplicación para móvil que nos va a permitir a los distintos usuarios de dicha aplicación comprar todo tipo de productos que ofrece una empresa.

Además de los usuarios normales (clientes), vamos a tener los usuarios que se van a encargar de dar de alta productos, eliminarlos, de todo lo referente a la gestión de estos y de la gestión de las familias que contendrán a dichos productos (empleados).

Estas tareas, entre otras, se van a llevar a cabo mediante llamadas a servicios web que la empresa tendrá operativos para que la aplicación móvil pueda utilizarlos, y así poder llevar a cabo toda esta funcionalidad sin problema alguno.

### **1.3. Resumen de la memoria**

En los siguientes capítulos podremos encontrar las descripciones de las tecnologías y elementos que intervendrán en este proyecto.

En el Capítulo 2, extenderemos lo que es un Servicio Web dando su definición, además explicaremos que podemos hacer con ellos y mencionaremos las ventajas y desventajas que suponen el uso de estos. A parte de todo esto, también explicaremos donde vamos a albergar la base de datos, como es MySQL, junto con su definición, para qué se usa, sus aplicaciones y sus ventajas y desventajas. Una vez expuesto esto, se pasa a explicar tomcat, que es un contenedor web donde vamos a tener nuestros Servicios Web, junto con su definición, el estado de su desarrollo, su entorno y un poco de historia. Y por último, explicaremos las diferentes tecnologías que podemos utilizar para realizar los Servicios Web.

En el Capítulo 3, podremos conocer que es Android, sus características y finalmente su arquitectura.

En el Capítulo 4, encontraremos todo lo que engloba a la fase de análisis. Así especificaremos el análisis de requisitos, junto con los diagramas de casos de usos pertinentes y los diagramas de secuencia asociados a cada uno de estos.

En el Capítulo 5, desarrollaremos la parte de diseño, como viene a ser el diagrama de clases, el diagrama de clases del Servicio Web y el diseño de la base de datos.

En el Capítulo 6, explicaremos todo lo que necesitaremos para la implementación de la aplicación, como son los lenguajes de programación utilizados, las herramientas utilizadas para el desarrollo de la misma y las librerías auxiliares utilizadas.

En el Capítulo 7, encontraremos un manual para la instalación del Servicio Web y todas las herramientas necesarias en el Servidor.

En el Capítulo 8, encontraremos un manual para la instalación de la aplicación cualquier dispositivo móvil.

En el Capítulo 9, un breve guía de usuario, para que este sepa como funciona la aplicación.

En el Capítulo 10, encontraremos todos los storyboard de la aplicación.



# CAPÍTULO 2

---

Servicio Web

## 2.1. Introducción

En esta sección vamos a explicar qué son los servicios web y todo lo que necesitamos para poder utilizarlos, como son el sistema de gestión de bases de datos que vamos a usar para almacenar los datos, el contenedor que albergará nuestro servicio web para poder llamarlo desde la aplicación y las diferentes tecnologías con las que podemos realizar los servicios web.

## 2.2. Servicio Web

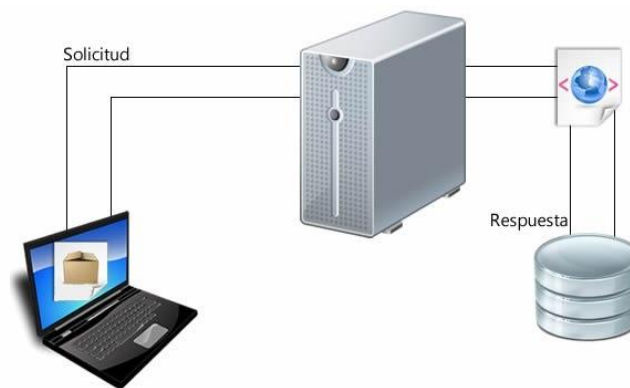


Ilustración 2.1: Servicio Web

### 2.2.1. ¿Qué es un Servicio Web?

El término “Servicio Web” [2] es una tecnología que consigue que las aplicaciones se comuniquen entre ellas sin depender de la plataforma ni del lenguaje de programación.

Un servicio web es una interfaz software que describe una serie de operaciones a las que podemos acceder por la red mediante mensajería XML estandarizada. Usa protocolos basados en el lenguaje XML buscando un objetivo que es describir una operación para poder ser ejecutada o datos para intercambiarlos con otro servicio web.

Un grupo de servicios web que interactúa de esa forma define la aplicación de un servicio web específico en una arquitectura orientada a servicios (SOA).

### **2.2.2. ¿Qué podemos hacer con los Servicios Web?**

En un principio, se empezó con la idea de tener una función a la cual se le proporcionase algunos parámetros, ejecutase alguna operación con dichos parámetros y retornase algún resultado basado en esos cálculos realizados. [2]

Con el paso del tiempo, ese primer concepto fue evolucionando de tal forma que cada objeto no sólo tenía varias funciones a realizar, sino también tenía sus propias variables de datos privados, en vez de basarse en variables externas, que llegaban a hacer mucho más complicado el desarrollo de aplicaciones.

Más recientemente, los servicios web se acercaron al concepto de interfaces y comunicaciones definidas en XML, para por fin finalmente unir aplicaciones de cualquier tipo, además de permitir la libertad de cambios y evoluciones con el paso del tiempo.

Lo que diferencia a los servicios web de las tecnologías que constituyen la generación anterior es la versatilidad de XML. Permite separar la estructura gramatical (sintaxis) del significado gramatical (semántica) y facilita la separación de como cada servicio del entorno procesa y entiende eso. Por tanto, en este momento, los objetos pueden ser definidos como servicios que se comunican con otros en la gramática definida por XML, donde cada servicio traduce y analiza el mensaje de acuerdo con la implementación local y el entorno.

Así, los servicios web permiten:

- Interacción entre servicios en cualquier plataforma y escritos en cualquier lenguaje.
- Permite que las interacciones entre aplicaciones de servicio no se rompan siempre que haya un cambio en la forma de diseño o implementación de un servicio.
- Adaptar las aplicaciones ya existentes a las cambiantes condiciones empresariales y necesidades de clientes.
- Proporcionar aplicaciones de software ya existentes o legadas con interfaces de servicio sin cambiar las aplicaciones originales, lo que permite operar totalmente en el entorno de servicios.

### **2.2.3. Ventajas de los Servicios Web**

- La interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se ejecuten. [3]
- Fomentar los estándares y protocolos basados en texto, haciendo mucho más sencillo acceder a su contenido y así poder comprender mejor su funcionamiento.
- Permiten que servicios y software creado por diferentes desarrolladores en cualquier parte del mundo puedan combinarse de manera sencilla para crear nuevas y potentes aplicaciones.

### **2.2.4. Desventajas de los Servicios Web**

- Su rendimiento es bajo comparado con otros sistemas como CORBA, DCOM o RMI, especialmente por el uso de protocolos y estándares basados en texto. [3]
- Para realizar transacciones no pueden compararse con los estándares abiertos de computación distribuida como CORBA.
- Al basarse en HTTP, pueden evitar ciertas medidas de seguridad basadas en firewall cuyas reglas consisten en bloquear la comunicación entre programas de ambos lados de la comunicación.

### **2.2.5. Por qué utilizar un Servicios Web**

Una de las razones más importantes para utilizar un servicio Web es que se pueden consumir desde cualquier ordenador o plataforma y eso facilita su uso.

Estos funcionan con HTTP por lo que no obtendremos bloqueos a la hora de comunicarnos en cualquier red.

Otra razón muy importante es que los servicios Web y las aplicaciones que lo utilizan son completamente independientes, por lo que los cambios que se producen en uno no afectan al otro.



## 2.3. MySQL



Ilustración 2.2: MySQL

### 2.3.1. ¿Qué es MySQL?

MySQL [4] (Ilustración 2.2) es un sistema de gestión de bases de datos relacional, multihilo y multiusuario mundialmente conocido y utilizado.

A diferencia de proyectos como puede ser Apache, donde el software es creado por una comunidad de desarrolladores pública y los derechos de autor del código realizado están en poder del creador de dicho código, MySQL pertenece a una empresa privada, que posee el copyright de la mayor parte del código.

La base de datos se distribuye en varias versiones:

- Una Community, distribuida bajo la Licencia pública general de GNU.
- Y varias versiones Enterprise, para aquellas empresas que quieran agregarlo en productos privativos.

Actualmente MySQL es utilizado por un gran número de sitios web famosos y grandes, como Wikipedia, Google, Facebook, Twitter y YouTube.

### 2.3.2. Aplicaciones

MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB y en diferentes plataformas como Linux, Windows, Apache, PHP, Perl y Python. Su gran aprobación en parte es porque está muy ligada a PHP, que frecuentemente aparece unido con MySQL. [4]

MySQL es mucho más rápido en la lectura cuando utiliza el motor no transaccional MyISAM.

En aplicaciones web se modifican muy poco los datos, sin embargo se leen de manera intensiva, lo que hace a MySQL como el sistema de gestión de base de datos relacional ideal para este tipo de aplicaciones.

Independientemente del entorno en el que se va a utilizar MySQL, es importante desde el comienzo monitorizar el rendimiento, para así, poder detectar y solucionar posibles errores tanto de SQL como de programación.

### 2.3.3. Ventajas de MySQL

- MySQL es Open Source. [5]
- Su gran velocidad al llevar a cabo las operaciones lo convierte en uno de los gestores con mejor rendimiento.
- Su bajo costo en requerimientos para la creación de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en máquinas con pocos recursos sin problema alguno.
- Configuración e instalación sencilla.
- Soporta una gran diversidad de Sistemas Operativos
- La probabilidad de que los datos se corrompan es muy baja.
- MySQL es apropiado para acceder a bases de datos en internet gracias a su conectividad, velocidad, y seguridad.

### **2.3.4. Desventajas de MySQL**

- No hay documentación de gran cantidad de las utilidades de MySQL.  
[\[5\]](#)
- A diferencia de otros programas, este no es intuitivo.

### **2.3.5. Por qué usar MySQL**

Como es un servidor de bases de datos muy rápido, seguro, y fácil de usar y como es esto lo que estamos buscando, le vamos a dar una oportunidad a MySQL.  
[\[6\]](#)

Originalmente MySQL fue creado para manipular mucho más rápido grandes bases de datos y exitosamente ha estado siendo usado por un tiempo en ambientes de producción muy exigentes.

Aunque está en constante desarrollo, el servidor MySQL hoy en día nos ofrece un conjunto muy rico y útil de funciones.

MySQL es un servidor muy recomendado para acceder a bases de datos en Internet gracias a su conectividad, velocidad y seguridad.

## 2.4. Tomcat



Ilustración 2.3: Tomcat

### 2.4.1. ¿Qué es Tomcat?

Tomcat [7] (Ilustración 2.3) es un contenedor web con un soporte para servlets y JSPs que son pequeños programas que se ejecutan en el contexto de un navegador web.

Tomcat desarrolla las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).

Este no es un servidor de aplicaciones, como si lo son JBoss o JOnAS. Contiene el compilador Jasper, que compila JSPs para así convertirlas en servlets.

El motor de servlets de Tomcat a veces puede presentarse unido al servidor web Apache.

Esta es la plataforma donde van a estar alojados todos nuestros servicios web.

### 2.4.2. Estado de su desarrollo

Mediante miembros de la Apache Software Foundation y voluntarios independientes Tomcat es desarrollado y actualizado. Los usuarios acceden libremente a su código fuente..

Tomcat se presenta desde las versiones 3.0.x hasta las versiones más recientes como son las 8.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2.

A partir de la versión 4.0, Jakarta Tomcat hace uso del contenedor de servlets Catalina.

### **2.4.3. Entorno**

Tomcat es un contenedor web con soporte de servlets y JSPs. Este no es un servidor de aplicaciones, como si lo son JBoss o JOnAS. Contiene el compilador Jasper, que compila JSPs para así convertirlas en servlets. El motor de servlets de Tomcat a veces puede presentarse unido al servidor web Apache.

Tomcat tiene la particularidad de que puede funcionar como servidor web por sí mismo. Inicialmente existía la percepción de que el uso de Tomcat de forma autónoma sólo era recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Pero hoy en día ya no existe esa percepción y es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Puesto que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que tenga instalada la máquina virtual Java.

### **2.4.4. Historia de Tomcat**

Tomcat comenzó siendo una implementación de la especificación de los servlets la cual fue empezada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun Microsystems y que al final ayudó a realizar el proyecto de código abierto el cual se donó a la Apache Software Foundation.

Duncan Davidson en un principio esperaba que el proyecto se convirtiese en software de código abierto y viendo que la gran mayoría de los proyectos de este tipo tienen libros cuya portada tiene un animal asociado, le puso al proyecto nombre de animal. Eligió Tomcat (gato), pretendiendo representar la capacidad de cuidarse por sí solo y de ser independiente.

## 2.5. Diferentes tecnologías para realizar Servicios Web



Ilustración 2.4: Diferentes tecnologías para realizar Servicios Web

### 2.5.1. Tipos de tecnologías

En cuanto a tecnologías [8] se refiere, hay dos esquemas relevantes, los cuales serán explicados a continuación. Estos son Soap y Rest (ilustración 2.4).

#### 2.5.1.1. SOAP

Este es el Protocolo Simple de Acceso a Objetos. Los mensajes que intercambian las aplicaciones los introduce en un “soap envelope” que es un paquete que se compone por la información referente al envío (por parte de quien consume el servicio) y por la respuesta (por parte de quien lo expone).

Para poder enviar la información a través de una red de datos como internet, dicho paquete se envuelve de nuevo, pero ahora utilizando el protocolo HTTP / HTTPS. Adicionalmente, también se formatea usando el lenguaje universal de etiquetas XML.

SOAP es un poco más lento que REST debido al doble empaquetamiento que es necesario para enviar mensajes.

SOAP UI es la aplicación más famosa que existe para consumir Servicios Web SOAP.

Su alta orientación a objetos es una de las ventajas de este protocolo, sobre todo cuando se quieren intercambiar mensajes que tienen gran cantidad de información o cuando la misma se encuentra estructurada.

### **2.5.1.2. REST**

Este protocolo mejora notablemente los tiempos de consumo y respuesta gracias a la eliminación de la capa XML y la envoltura adicional que existe en SOAP.

Los mensajes utilizan datos en formato JSON para intercambiar información.

Este protocolo facilita enormemente su consumo y pruebas debido a que es más simple y permite que el consumo de los servicios se haga usando como cliente un navegador web.

### **2.5.2. ¿Qué tecnología vamos a utilizar?**

Los servicios web SOAP, finalmente han acabado siendo una gran carga de múltiples capacidades, las cuales, la mayor parte del tiempo, no son necesarias.

Por el otro lado, Rest es mucho más sencilla, por lo que vamos a tener menos problemas a la hora de usarlo.

Una vez dicho todo esto y teniendo en cuenta que REST, como hemos comentado arriba, nos aporta mejoras notables en los tiempos de consumo y respuesta, he decidido usar esta tecnología (REST) a la hora de crear los servicios web los cuales serán llamados desde mi aplicación.





# CAPÍTULO 3

---

Android



Ilustración 3.1: Android

### 3.1. ¿Qué es android?

Android [9][10] (Ilustración 3.1) es un sistema operativo a parte de una plataforma de Software basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como:

- Teléfonos inteligentes
- Tablets
- Relojes inteligentes
- Televisores
- Automóviles

Android nos permite controlar dispositivos utilizando bibliotecas desarrolladas o adaptadas por Google usando el lenguaje de programación Java.

Es una plataforma de código abierto. Esto quiere decir, que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con diferentes lenguajes y compilarlas a código nativo de ARM (API de Android).

Inicialmente fue desarrollado por Android Inc., la cual fue respaldada económicamente por Google y que al final compró en 2005. Más tarde fue presentado en 2007 junto con la fundación del Open Handset Alliance, que es un consorcio de 48 compañías de hardware, software y telecomunicaciones, para avanzar en los estándares abiertos de los dispositivos móviles.

Google sin embargo, ha sido quien ha publicado la mayoría del código fuente de Android bajo la licencia de Software Apache, una licencia de software libre y de código abierto a cualquier desarrollador.

El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. En general los dispositivos Android se venden más que Windows Phone e IOS juntos.

Open Source Project (AOSP) es la versión básica de Android.

Google mostró una evolución de la marca Android , el 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, con el fin de unificar tanto el hardware como el software y ampliar mercados.

Más tarde, el 17 de mayo de 2017, se presentó Android Go. Esto es una versión más ligera del sistema operativo para así hacer que la mitad del mundo sin smartphone consiga uno en menos de cinco años. Incluye versiones especiales de sus aplicaciones donde el consumo de datos se reduce al máximo.

### 3.2. Características

Android presenta las siguientes características principales: [\[9\]](#)

- **Diseño de dispositivo**

La plataforma se puede adaptar para pantallas que cuentan con una resolución mayor, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.

- **Almacenamiento**

SQLite que es una base de datos liviana usada para el almacenamiento de datos, es el almacenamiento usado.

- **Conectividad**

Android soporta las siguientes tecnologías de conectividad:

- GSM/EDGE
- UMTS
- Bluetooth
- Wi-Fi
- LTE
- GPRS
- UMTS
- Entre otros.....

- **Mensajería**

SMS y MMS son las formas de mensajería soportadas.

- **Navegador web**

El navegador web incluido en Android se basa en el motor de renderizado WebKit.

- **Soporte de Java**

No hay una máquina virtual Java incluida en la plataforma a pesar de que la mayoría de las aplicaciones están desarrolladas en Java. Debido a esto, primeramente se compila en un ejecutable Dalvik para posteriormente ejecutarlo en una Máquina Virtual Dalvik.

Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART).

- **Soporte multimedia**

Android soporta los siguientes formatos multimedia:

- WebM
- MPEG-4 SP
- MP3
- MIDI
- WAV
- JPEG
- PNG
- GIF
- BMP
- Entre otros....

- **Soporte para Hardware adicional**

Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.

- **Entorno de desarrollo**

Inicialmente se utilizaba Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio.

- **Google Play**

Google Play es un catálogo de aplicaciones gratuitas o de pago en el que nos podemos descargar e instalar dichas aplicaciones en nuestros dispositivos Android sin la necesidad de un PC.

- **Multi-táctil**

Android soporta pantallas capacitivas con soporte multi-táctil. Estas aparecieron en dispositivos como el HTC Hero.

Originalmente esta funcionalidad fue desactivada a nivel de kernel pero más adelante, Google publicó una actualización para el Nexus One y el Motorola Droid que activaba el soporte multi-táctil de forma nativa.

- **Bluetooth**

El soporte para A2DP y AVRCP fue insertado en la versión 1.5. El envío de archivos y la exploración del directorio telefónico fueron agregados en la versión 2.0 y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.

- **Videollamada**

Desde la versión honeyComb, las videollamadas son soportadas por Android.

- **Multitarea**

Esta está disponible y gracias a ella las aplicaciones que estén ejecutandose en segundo plano recibirán ciclos de reloj.

- **Características basadas en voz**

Desde la versión inicial del sistema tenemos disponible la búsqueda en Google usando la voz, como entrada de búsqueda.

- **Tethering**

Permite a un PC usar la conexión de datos del móvil ya que nos deja poner el teléfono como un punto de acceso alámbrico o inalámbrico. Para esto podría ser necesaria la instalación de software adicional.

### 3.3. Arquitectura

La arquitectura interna de la plataforma Android, está formada por estas 4 componentes [9]:



Ilustración 3.2: Arquitectura Android

- **Aplicaciones (Ilustración 3.2)**

Todas las aplicaciones creadas con la plataforma Android, incluirán como base un cliente de email (correo electrónico), calendario, programa de SMS, mapas, navegador, contactos, y algunos otros servicios mínimos. Todas estas aplicaciones estarán desarrolladas en el lenguaje de programación Java.

- **Framework de aplicaciones (Ilustración 3.2)**

Todos los desarrolladores de aplicaciones Android, tienen total acceso al código fuente usado en las aplicaciones base. Esto es así, para que no se creen gran cantidad de componentes de aplicaciones distintas que hagan lo mismo, dando la posibilidad de no tener que empezar a programar las aplicaciones desde el principio.

- **Librerías (Ilustración 3.2)**

A través del framework de las aplicaciones Android System C library, librerías de medios, librerías de gráficos, 3D, SQLite, etc, son expuestas a los desarrolladores un set de librerías C/C++ que se encuentran en la base de datos de Android.

- **Runtime de Android (Ilustración 3.2)**

Android consta de un set de bibliotecas base que ofrece la gran parte de las funciones disponibles en las bibliotecas base del lenguaje Java.

Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik nos permite tener funcionando varias máquinas virtuales de forma eficiente.

- **Núcleo Linux (Ilustración 3.2)**

Para los servicios base del sistema como seguridad, gestión de procesos, gestión de memoria, modelo de controladores y pila de red, Android, depende de Linux. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

### **3.4. Por qué usar Android**

El Sistema Operativo Android nos ofrece numerosas ventajas si hacemos una comparación con los sistemas competidores. Y hablando más concretamente de él para desarrollar aplicaciones móviles, Android se ha convertido, en el sistema favorito de muchos desarrolladores. [\[11\]](#)

También, en la carrera he tenido una asignatura que nos han enseñado a programar para el sistema operativo IOS, por lo que para la realización de este trabajo fin de grado he preferido el desarrollo en android para así aprender más cosas



# CAPÍTULO 4

---

Análisis

## 4.1. Especificación de Requisitos

El marco de trabajo donde nos encontramos es una empresa que se encarga de la venta de una amplia gama de productos.

La forma en la que la empresa actúa a la hora de trabajar es la siguiente:

- El cliente consulta el listado de productos.
- El cliente rellena su carrito con los productos que necesite.
- El cliente realiza el pedido.
- La empresa cobra el pedido realizado.
- La empresa prepara los productos para el envío.
- La empresa envía los productos pedidos.

### ➤ **Productos**

A la hora de comprar, se mira si la cantidad del producto seleccionada está disponible en stock en el almacén de la empresa, si es así, se podrá añadir el producto, con el número de unidades pertinente, al carrito, si no lo es, no se podrá añadir dicho producto ya que no está disponible.

### ➤ **Clientes**

Son los usuarios que se han registrado como tal en la aplicación, los cuales tendrán acceso a toda la información de los productos ofertados por la empresa y podrán realizar pedidos.

Todo usuario que no esté registrado en la aplicación, sólo podrá ver el listado de productos y la información de estos, pero no podrá realizar pedidos, para ello deberá estar registrado.

### ➤ **Lista de la compra (Carrito)**

El cliente puede ir añadiendo al carrito todo tipo de productos, siempre y cuando el producto no esté agotado. Esto quiere decir que no se pueden realizar pedidos de productos agotados o no se pueden añadir al carrito productos eligiendo una cantidad a comprar mayor que la que está disponible.

A la hora de hacer el pedido, se hará con todos los productos introducidos en este carrito hasta el momento. Antes de hacerlo, el cliente puede quitar productos del carrito o seguir añadiendo si se le ha olvidado alguno.

Una vez realizado el pedido, el carrito quedará vacío, por lo que para consultar los datos de compra se deberá acceder a la información del pedido realizado.

### ➤ **Pedidos**

A la hora de formalizar el pedido, el cliente debe pagarlo, además de proporcionarnos la información necesaria para poder enviarlo.

Una vez que el cliente pague el importe del pedido, este se formalizará y se pasaría a preparar los productos para el envío.

Un pedido no estará preparado para su envío hasta que no lo estén todos los productos.

Un pedido puede ser anulado desde que se realiza hasta el día siguiente a las 00:00. Después de ese tiempo el pedido pasaría a ser efectivo y ya no podría anularse.

Una vez expuesto todo esto, he diferenciado las siguientes funcionalidades básicas que son necesarias para la correcta utilización del sistema:

- Iniciar Sesión
- Registrarse
- Listar Productos
- Crear Productos
- Añadir Unidades
- Modificar Productos
- Eliminar Productos
- Añadir Productos al Carrito
- Quitar Productos al Carrito
- Consultar el Carrito
- Crear Pedido
- Anular Pedido
- Consultar Pedidos
- Crear Comentarios
- Modificar Comentarios
- Eliminar Comentarios
- Cerrar Sesión
- Crear Familias
- Modificar Familias
- Eliminar Familias
- Consultar Familias
- Modificar datos de registro

Todas estas no pueden ser realizadas por los clientes. Con esto quiero decir que un usuario, no sólo puede ser cliente de la empresa, sino también empleado. Ambos heredan funcionalidad de usuario, que sería un usuario el cual no está registrado y puede ver los productos y toda su información, buscar productos por familias, a parte de iniciar sesión o registrarse para poder realizar pedidos.

Un usuario cliente no puede ser empleado, pero un usuario empleado puede ser a su vez cliente.

Este usuario empleado será el que se encargue de la gestión de todos los productos que la empresa oferta.

Para que se vea más claro voy a diferenciar que funcionalidades pueden realizar cualquier usuario que no este registrado en la aplicación, cuales un usuario cliente y cuales un usuario empleado:

- **Para todos los usuarios**

- **Iniciar Sesión**

Esta etapa es clave a la hora de diferenciar que tipo de usuario accede a la aplicación y así poder ofrecerle todas las funcionalidades que dicho usuario pueda gestionar.

- **Registrarse**

Aquí permitimos que el usuario que aún no está registrado en la app lo haga de una manera sencilla y rápida.

- **Listar Productos**

Todos los tipos de usuarios, pueden listar los productos, uno (usuario sin registrar), para poder ver toda la información de los productos de la empresa, otro (cliente), para ver toda la información del producto y poder así comprarlo, y otro (empleado), para poder ver toda la información del producto y así poder modificarlo si hubiese alguna errata o también para poder comprarlo.

- **Consultar Familias**

Todo usuario podrá consultar el listado de familias, para así buscar de manera más fácil el producto deseado, ya que estarán ordenados por estas y la búsqueda quedará reducida casi al máximo.

- **Para clientes y empleados**

- **Modificar Datos de Registro**

A todo usuario se le permitirá cambiar sus datos de registro, por si alguno dato está erróneo o cambia a lo largo del tiempo.

- **Añadir y Quitar Productos al/del Carrito**

Teniendo en cuenta lo dicho anteriormente tanto el cliente como el empleado podrán añadir y quitar productos al/del carrito. Para así poder hacer compras múltiples de manera sencilla.

- **Consultar el Carrito**

Se podrá consultar el carrito para ver todos los productos insertados hasta el momento en él y así poder crear el pedido.

- **Crear y Anular Pedido**

Una vez teniendo el carrito preparado, pueden crear un pedido y por consiguiente, una vez con el pedido realizado, pueden anularlo, teniendo en cuenta que tenemos para ello desde el momento de su realización hasta el día siguiente a las 00:00. Una vez pasado ese plazo, el pedido se hará formal y ya no podríamos anularlo.

- **Consultar Pedidos**

También podrán consultar los pedidos que han realizado, para así ver toda la información más detallada del pedido seleccionado.

- **Crear Comentarios**

Pueden añadir un comentario a un producto para poder expresar su nivel de agrado o descontento con el producto comprado y así servir de guía a otros clientes a la hora de comprarlo.

- **Modificar Comentarios**

También podrán modificar los comentarios, siempre y cuando este haya sido realizado por él. Esto servirá para poder cambiarlo si has introducido un comentario erróneo o con alguna falta de ortografía.

- **Eliminar Comentarios**

Se permitirá eliminar los comentarios, siempre y cuando el usuario sea el propietario de este.

- **Cerrar Sesión**

Todo usuario que ha iniciado sesión en la aplicación podrá salir de ella cada vez que quiera gracias a esta funcionalidad.

- **Para empleados**

- **Crear Productos**

Cuando llega un producto nuevo a la empresa, por producto nuevo nos referimos a un producto que no esté siendo ofertado por la empresa actualmente, el usuario empleado es el encargado de introducirlo en el sistema, utilizando la aplicación.

Esta tarea está más enfocada para alguien que gestione la base de datos, pero como estamos haciendo una aplicación, estoy teniendo en cuenta que la gestión también se puede realizar desde la aplicación, por lo que el usuario empleado será el encargado de crear productos utilizando esta.

- **Añadir Unidades (Aumentar Stock)**

Por añadir unidades me refiero a la llegada al almacén de productos que actualmente están siendo ofertados por la empresa, por lo que el usuario empleado lo que debe de hacer es buscar el producto indicado y añadir el número de productos que han llegado al stock.

- **Modificar Productos**

Aquí lo que se hace es modificar la información de los productos, ya sea porque contengan alguna errata o porque el producto cambie algunos de sus campos de información.

- **Eliminar Productos**

Una vez que se agote un producto y este no vaya a ser ofertado más en la empresa, este producto será eliminado de la lista de productos.

- **Crear Familias**

El empleado podrá crear una familia de productos, para así poder agrupar los productos por cada una de ellas. Por familia me refiero a cada uno de los siguientes ejemplos:

- Videojuegos
- Motor
- Electrodomésticos
- Etc.

- **Modificar Familias**

Se podrá modificar cada una de las familias, por si a lo largo del tiempo pasa a llamarse de otra forma.

- **Eliminar Familias**

A la hora de ya no querer usar una familia de las existentes se podrá eliminar.

No podremos eliminar una familia que tenga asignados productos, esto quiere decir que para eliminarla, esta tiene que estar vacía, por lo que o se eliminan los productos que contiene o se le asignan a estos otra familia.



## 4.2. Análisis del Sistema

Una vez expuesta la especificación de requisitos que necesitamos saber a la hora de llevar a cabo nuestra aplicación, debemos de realizar un Análisis del Sistema.

En este análisis vamos a enseñar los diferentes casos de uso de nuestra aplicación y los diferentes diagramas de secuencia asociados a cada uno de ellos.

### 4.2.1. Casos de uso

En esta sección explicaremos y mostraremos todos los casos de uso de nuestra aplicación, junto con toda la información necesaria.

#### 4.2.1.1. Actores

Un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Dentro de nuestro caso de uso podemos diferenciar entre los siguientes actores:

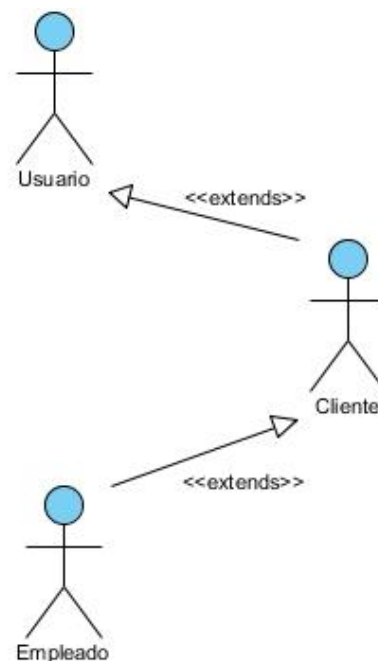


Ilustración 4.1: Actores

- **Usuario (Ilustración 4.1)**

Es todo aquel que entra en la aplicación y todavía no se ha identificado. Este usuario no queda registrado en ningún sitio en nuestro sistema ya que sólo lo utilizamos a la hora de entrar en la aplicación y no estamos registrados ni logeados todavía.

Este usuario, a parte de poder iniciar sesión o registrarse, también va a poder listar los productos de la empresa y tener acceso a toda la información sobre ellos, así como listar todas las familias para hacer una búsqueda más rápida, pero no podrá realizar pedidos, para ello deberá registrarse.

- **Cliente (Ilustración 4.1)**

Es un usuario que se ha identificado en la aplicación. Este, ahora sí, puede realizar compras. Contiene toda la funcionalidad referente a un cliente.

- **Empleado (Ilustración 4.1)**

Es el que entra en la aplicación para realizar las tareas correspondientes a un empleado de la empresa. El empleado también puede ejercer como cliente, de ahí su relación.

### 4.2.1.2. Caso de uso del actor Usuario

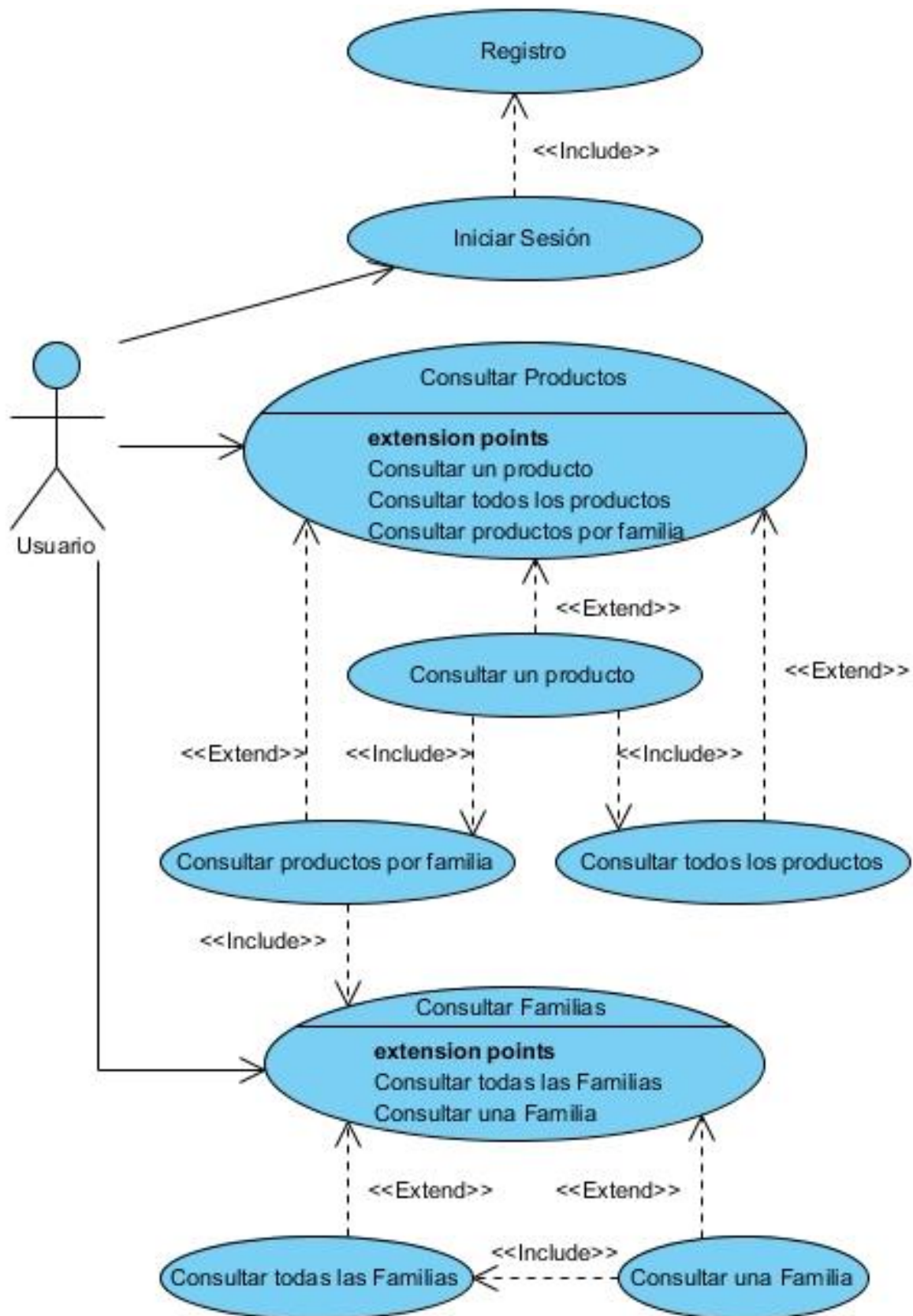


Ilustración 4.2: Diagrama de casos de uso del actor usuario

<b>CASO DE USO</b>		<b>Registro (Ilustración 4.2)</b>
<b>Actores</b>		Usuario
<b>Prerrequisitos</b>		El usuario aún no se ha autenticado.
<b>Propósito</b>		
<p>El usuario desea darse de alta como cliente o como empleado para poder comprar productos de la empresa o para realizar las tareas correspondientes a un empleado. Para darse de alta como empleado la aplicación le pedirá un número de empleado, que la empresa le proporcionaría.</p>		
<b>Resumen</b>		
<p>El sistema muestra al usuario una ventana de alta en la que este debe especificar un login, un password y una serie de datos necesarios para el sistema. El usuario envía todos estos datos ya rellenados. El sistema comprueba que los datos sean correctos y que el login no existía y nos da de alta en la aplicación.</p>		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de registrarse.</li> <li>2. El sistema muestra la ventana de alta.</li> <li>3. El usuario rellena todos los datos y los envía al sistema.</li> <li>4. El sistema comprueba que los datos recibidos sean correctos y que el login no existe.</li> <li>5. El sistema nos lleva a la página principal del cliente o del empleado dependiendo de como nos hayamos registrado.</li> </ol>		
<b>Curso alternativo</b>		
<ol style="list-style-type: none"> <li>5a. Si los datos no son correctos o el login ya existe, el sistema muestra un error dando la posibilidad de corregir los datos.</li> </ol>		

Tabla 4.1: Registro

<b>CASO DE USO</b>		<b>Iniciar Sesión (Ilustración 4.2)</b>
<b>Actores</b>		Usuario
<b>Prerrequisitos</b>		El usuario ya se ha registrado en el sistema pero aún no se ha autenticado.
<b>Propósito</b>		
El usuario debe autenticarse para poder acceder a todas las funcionalidades del cliente o del empleado.		
<b>Resumen</b>		
El sistema muestra al usuario una ventana para que escriba su login y password. Una vez enviados estos datos, el sistema los verificará y el usuario tendrá acceso a las opciones del sistema.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El sistema muestra una ventana donde el usuario debe escribir su login y su password.</li> <li>2. El usuario rellena los datos y los envía.</li> <li>3. El sistema verifica que los datos sean correctos.</li> <li>4. El sistema nos lleva a la página principal del cliente o del empleado dependiendo de que tipo de usuario seamos.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si el login no existe o el password no es correcto, el sistema muestra un error dando la posibilidad de especificar un nuevo login y password.		

Tabla 4.2: Iniciar Sesión

Los casos de uso Consultar Productos y Consultar Familias son abstractos, por lo que sólo especificaremos los casos de uso que heredan de ellos.

<b>CASO DE USO</b>		<b>Consultar todos los productos (Ilustración 4.2)</b>
<b>Actores</b>	Usuario	
<b>Prerrequisitos</b>	No hace falta estar autenticado.	
<b>Propósito</b>		
El usuario desea ver una lista de todos los productos que oferta la empresa.		
<b>Resumen</b>		
El usuario excoge la opción de ver todos los productos y el sistema muestra una lista con todos ellos.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario excoge la opción de ver todos los productos.</li> <li>2. El sistema muestra una ventana con la lista de todos ellos.</li> </ol>		

Tabla 4.3: Consultar todos los productos

<b>CASO DE USO</b>		<b>Consultar productos por familia( Ilustración 4.2)</b>
<b>Actores</b>	Usuario	
<b>Prerrequisitos</b>	No hace falta estar autenticado, pero si hay que consultar una familia para ver sus productos.	
<b>Propósito</b>		
El usuario desea ver la lista de todos los productos que pertenecen a una familia, por ejemplo: motor, electrodomésticos, etc.		
<b>Resumen</b>		
El usuario excoge la opción de ver productos por familia, el sistema muestra una ventana con todas las familias existentes, el usuario excoge una de ellas y el sistema muestra la lista con todos los productos pertenecientes a la familia seleccionada si esta tiene productos, si no tiene mostrará un mensaje de familia vacía.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario excoge la opción de ver productos por familia.</li> <li>2. El sistema proporciona una ventana con la lista de todas las familias.</li> <li>3. El usuario excoge una.</li> <li>4. El sistema muestra una ventana con la lista de todos los productos que pertenecen a ella.</li> </ol>		

Tabla 4.4: Consultar productos por familia

<b>CASO DE USO</b>		<b>Consultar un producto (Ilustración 4.2)</b>
<b>Actores</b>		Usuario
<b>Prerrequisitos</b>		No hace falta estar autenticado pero se han tenido que consultar todos los productos o los productos de una familia.
<b>Propósito</b>		
El usuario desea ver la información de un producto.		
<b>Resumen</b>		
El usuario ha debido de consultar antes o todos los productos o los productos de una familia, para así tener la lista de productos y poder escoger uno para ver su información más detallada.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta todos los productos o los productos de una familia.</li> <li>2. El sistema proporciona una ventana con la lista de los productos buscados.</li> <li>3. El usuario escoge uno.</li> <li>4. El sistema muestra una ventana con la información más detallada del producto seleccionado.</li> </ol>		

Tabla 4.5: Consultar un producto

<b>CASO DE USO</b>		<b>Consultar todas las familias (Ilustración 4.2)</b>
<b>Actores</b>		Usuario
<b>Prerrequisitos</b>		No hace falta estar autenticado.
<b>Propósito</b>		
El usuario desea ver la lista de todas las familias de las que se compone la empresa.		
<b>Resumen</b>		
El usuario escoge la opción de ver todas las familias y el sistema muestra una lista con todas ellas.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario escoge la opción de ver todas las familias.</li> <li>2. El sistema muestra una ventana con la lista de todas ellas.</li> </ol>		

Tabla 4.6: Consultar todas las familias

<b>CASO DE USO</b>		<b>Consultar una familia (Ilustración 4.2)</b>
<b>Actores</b>		Usuario
<b>Prerrequisitos</b>		No hay que estar autenticado pero se han tenido que consultar todas las familias.
<b>Propósito</b>		
El usuario desea ver la información de una familia.		
<b>Resumen</b>		
El usuario ha debido de consultar todas las familias para así tener la lista de estas y poder escoger una para ver su información detallada.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta todas las familias.</li> <li>2. El sistema muestra una ventana con la lista de las familias buscadas.</li> <li>3. El usuario escoge una.</li> <li>4. El sistema muestra una ventana con la información más detallada de la familia seleccionada.</li> </ol>		

Tabla 4.7: Consultar una familia

#### 4.2.1.3. Caso de uso del actor Cliente

En este caso de uso vamos a especificar los casos de uso que puede hacer específicamente el actor cliente, ya que este actor también puede realizar todos los anteriores.

Primeramente, al ser un diagrama muy grande, vamos a mostrar los casos de uso generales y posteriormente los ampliaremos en diagramas más detallados.



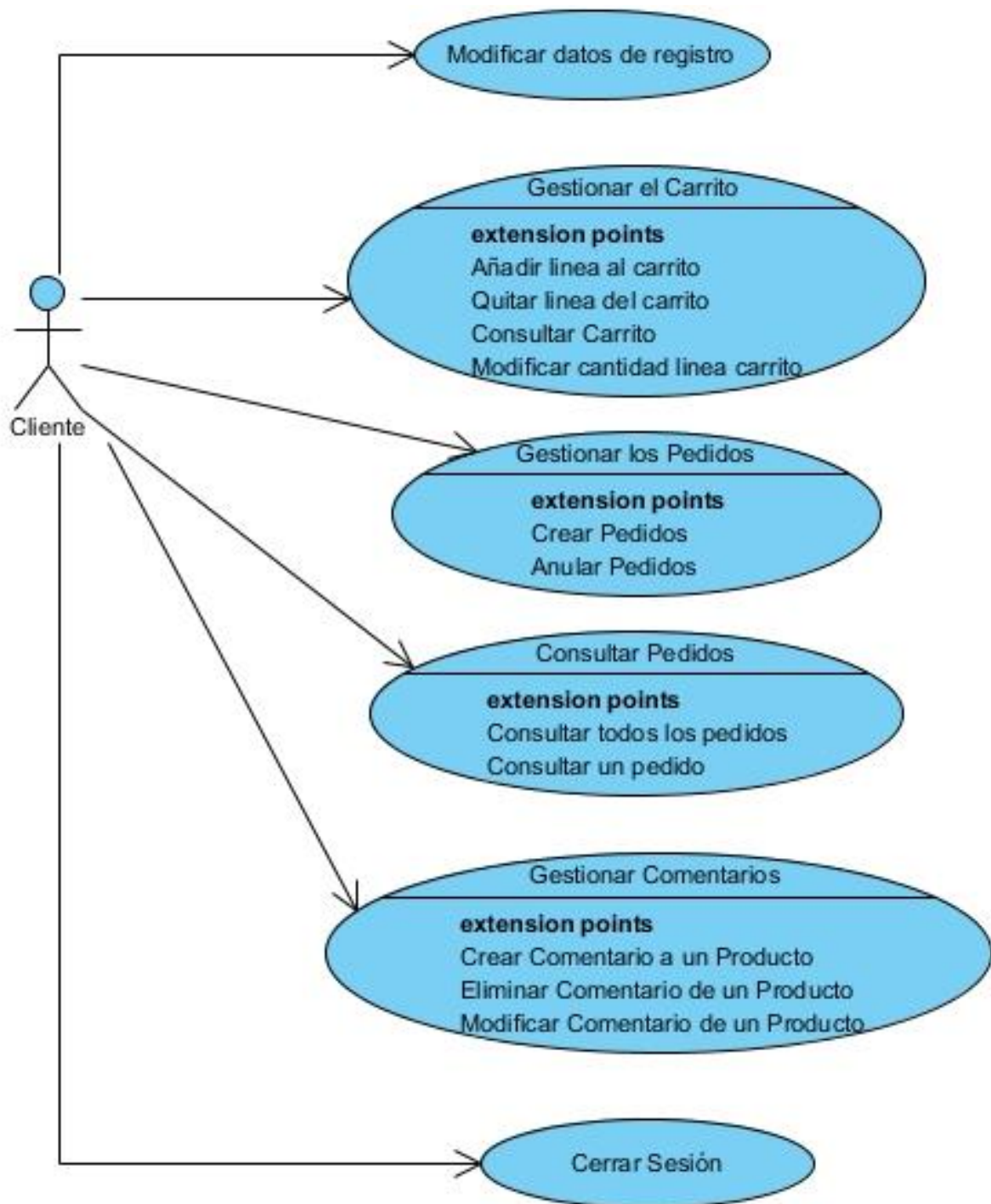


Ilustración 4.3: Diagrama de casos de uso generales del actor cliente

Antes de detallar los casos de uso generales del actor cliente, vamos a especificar los casos de uso “Modificar datos de registro” y “Cerrar Sesión”.

<b>CASO DE USO</b>		<b>Modificar datos de registro (Ilustración 4.3)</b>
<b>Actores</b>	Cliente	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado.	
<b>Propósito</b>		
El cliente desea modificar algunos de los datos de registro que introdujo.		
<b>Resumen</b>		
El sistema muestra al usuario una ventana con los datos que tiene registrados. El cliente modifica los datos correspondientes y los envía. El sistema comprueba que los datos sean correctos y notifica que los datos han sido modificados.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de modificar sus datos de registro.</li> <li>2. El sistema muestra una ventana con los datos registrados.</li> <li>3. El usuario modifica los datos y los envía al sistema.</li> <li>4. El sistema comprueba que los datos recibidos sean correctos.</li> <li>5. El sistema notifica al usuario que los datos han sido modificados.</li> </ol>		
<b>Curso alternativo</b>		
5a. Si los datos no son correctos, el sistema muestra un error dando la posibilidad de modificarlos.		

Tabla 4.8: Modificar datos de registro

<b>CASO DE USO</b>		<b>Cerrar Sesión (Ilustración 4.3)</b>
<b>Actores</b>	Cliente	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado.	
<b>Propósito</b>		
El cliente desea cerrar la sesión abierta en nuestra aplicación.		
<b>Resumen</b>		
Una vez que el usuario está dentro de la aplicación, este puede cerrar sesión cuando quiera solo pulsando el botón de cerrar sesión.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona el botón de cerrar sesión.</li> <li>2. El sistema desconecta al usuario de la aplicación.</li> </ol>		

Tabla 4.9: Cerrar sesión

Ahora si, vamos a proceder a extender cada uno de los casos de uso generales expuestos en el diagrama anterior (Ilustración 9).

- **Gestionar el Carrito**

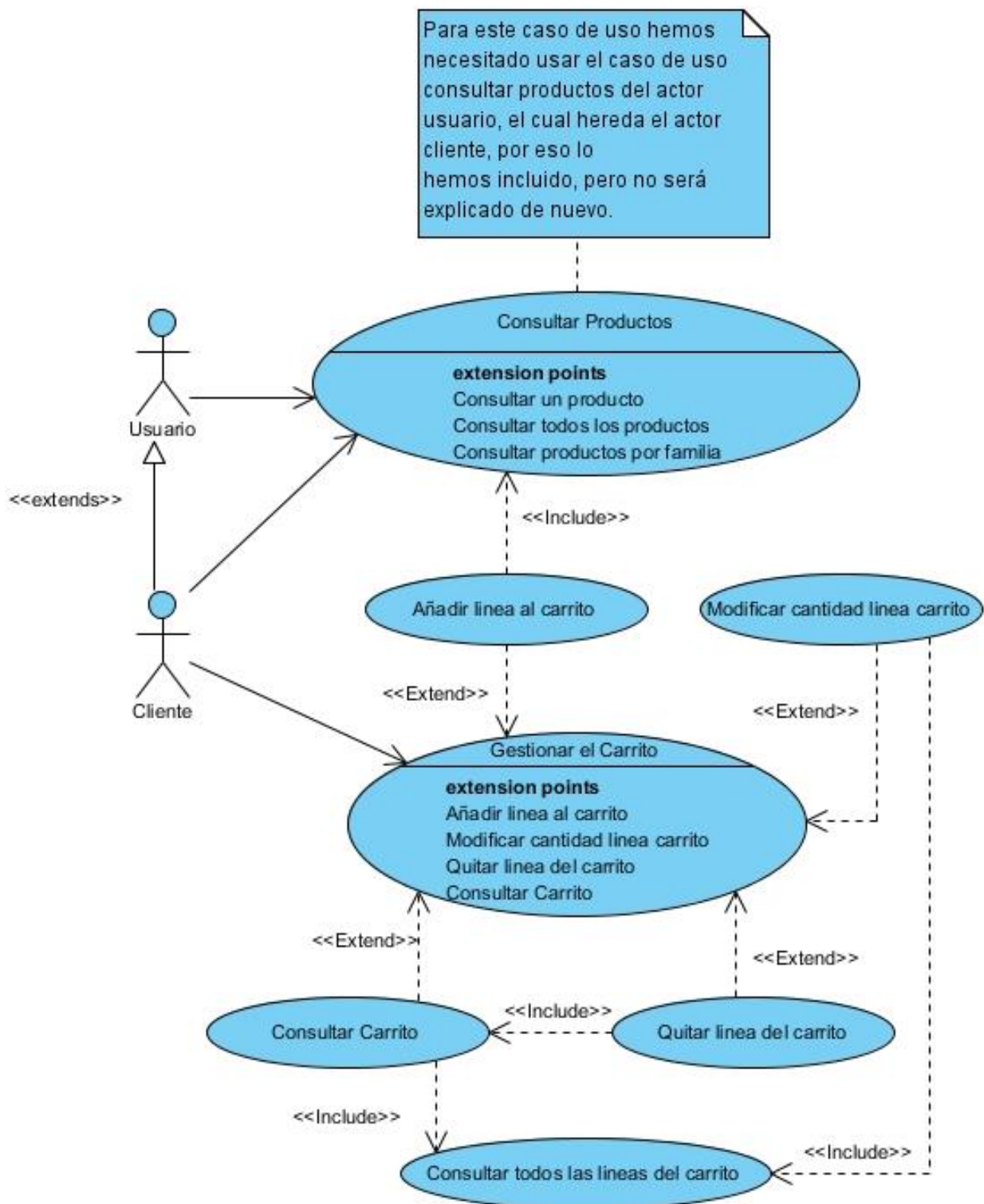


Ilustración 4.4: Diagrama de casos de uso “Gestionar el carrito”

<b>CASO DE USO</b>		<b>Añadir línea al Carrito (Ilustración 4.4)</b>
<b>Actores</b>	Cliente	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha consultado algún producto para poder añadirlo.	
<b>Propósito</b>		
El cliente añade una línea al carrito, con la cantidad de unidades que desea comprar de ese producto, para así, al final, poder ver cuanto le costaría toda la compra y poder realizar el pedido.		
<b>Resumen</b>		
Una vez que el usuario ha consultado algún producto, este puede ser añadido al carrito para poder ser comprado posteriormente.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona un producto de la lista de productos consultada.</li> <li>2. El sistema le muestra una ventana con la información detallada de este producto.</li> <li>3. El usuario selecciona la opción de añadir dicho producto al carrito junto con la cantidad de unidades que desee.</li> <li>4. El sistema lo añade.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si las unidades excogidas exceden el stock disponible, el sistema mostrará un mensaje de error y nos permitirá cambiarlas.		

Tabla 4.10: Añadir línea al carrito

<b>CASO DE USO</b>		<b>Modificar cantidad línea de carrito (Ilustración 4.4)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y ha consultado alguna línea de carrito.
<b>Propósito</b>		
El cliente desea añadir o quitar unidades de algún producto que ha añadido al carrito.		
<b>Resumen</b>		
Una vez que el usuario ha consultado las líneas de carrito, este puede modificar la cantidad de unidades que una línea tiene asignada.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta las líneas de carrito.</li> <li>2. El sistema le muestra una ventana con la lista de estas.</li> <li>3. El usuario excoge una.</li> <li>4. El sistema muestra una ventana para cambiar la cantidad.</li> <li>5. El usuario modifica la cantidad y la envía.</li> <li>6. El sistema la modifica.</li> </ol>		
<b>Curso alternativo</b>		
6a. Si las unidades excogidas exceden el stock disponible, el sistema mostrará un mensaje de error y nos permitirá cambiarlas.		

Tabla 4.11: Modificar cantidad línea de carrito

<b>CASO DE USO</b>		<b>Quitar línea del Carrito (Ilustración 4.4)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y ha consultado el carrito.
<b>Propósito</b>		
El cliente desea quitar una línea del carrito porque ya no desee comprar ese producto.		
<b>Resumen</b>		
Una vez que el usuario ha consultado el carrito, puede quitar productos (líneas de carrito) de este.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta el carrito para ver que productos contiene.</li> <li>2. El sistema muestra una ventana con todas las líneas que forman el carrito.</li> <li>3. El usuario selecciona una y le da a eliminar.</li> <li>4. El sistema la quita.</li> </ol>		

Tabla 4.12: Quitar línea del carrito

<b>CASO DE USO</b>		<b>Consultar Carrito (Ilustración 4.4)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado.
<b>Propósito</b>		
El cliente desea consultar todo lo que él mismo ha ido introduciendo en el carrito.		
<b>Resumen</b>		
El usuario selecciona la opción de consultar carrito y el sistema mostrará un listado con todas las líneas de carrito que han sido insertadas por él mismo en este, si no hay líneas no mostrará nada.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de consultar carrito.</li> <li>2. El sistema muestra una ventana con la lista de líneas de carrito que conforman a este, si no hay líneas no mostrará nada.</li> </ol>		

Tabla 4.13: Consultar carrito

<b>CASO DE USO</b>		<b>Consultar todas las líneas de carrito (Ilustración 4.4)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado.
<b>Propósito</b>		
El cliente desea consultar los productos que tiene actualmente insertados en el carrito, para ello consulta todas las líneas que conforman el carrito.		
<b>Resumen</b>		
El usuario escoge esta opción y el sistema le mostrará una lista con todas las líneas que tiene el carrito. Así se pueden ver todos los productos, cantidades y precio de cada una de ellas.		
<b>Curso normal</b>		
<ol style="list-style-type: none"><li>1. El usuario selecciona esta opción.</li><li>2. El sistema muestra una ventana con la lista de todas las líneas de carrito que componen a este.</li></ol>		

Tabla 4.14: Consultar todas las líneas de carrito

- **Gestionar los Pedidos**

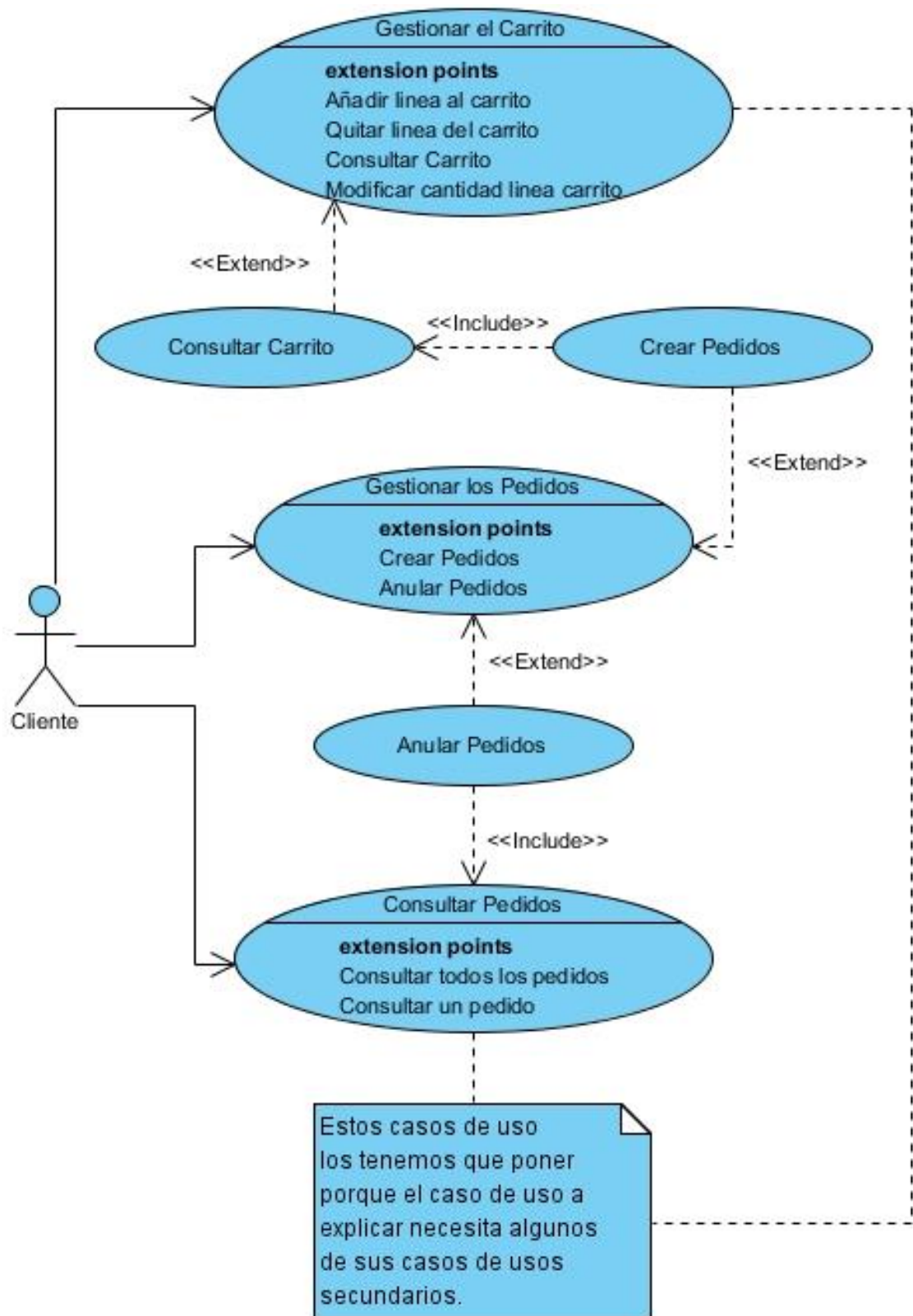


Ilustración 4.5: Diagrama de casos de uso "Gestionar los pedidos"



<b>CASO DE USO</b>		<b>Crear Pedidos (Ilustración 4.5)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y debe de haber consultado el carrito que tendrá la compra deseada.
<b>Propósito</b>		
El cliente desea crear un pedido.		
<b>Resumen</b>		
El usuario selecciona la opción de crear un pedido y el sistema nos llevará a una para introducir los datos del pedido y así poder crearlo.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de crear pedido.</li> <li>2. El sistema muestra una ventana para que el cliente rellene todos los datos necesarios para realizar el envío y el pago.</li> <li>3. El sistema comprueba que se ha realizado el pago y que los datos sean correctos.</li> <li>4. El sistema almacena el pedido y este ya sería preparado y enviado por la empresa. Se comunicaría al usuario que el pedido ha sido realizado con éxito.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si no se ha podido hacer el pago, porque los datos de pago sean incorrectos o los otros datos introducidos sean incorrectos, el sistema mostrará un error y dará la posibilidad de cambiar los datos introducidos.		

Tabla 4.15: Crear pedidos

<b>CASO DE USO</b>		<b>Anular Pedidos (Ilustración 4.5)</b>
<b>Actores</b>	Cliente	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado, ha debido de consultar los pedidos y no se ha debido de pasar el plazo para anular el pedido.	
<b>Propósito</b>		
El cliente desea anular un pedido.		
<b>Resumen</b>		
El usuario selecciona la opción consultar pedidos y el sistema le muestra todos los pedidos que ha realizado, selecciona el que desee y si aún no se ha pasado el plazo de anulación, este se podrá anular.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de consultar pedidos.</li> <li>2. El sistema muestra una ventana con la lista de pedidos realizados por el usuario.</li> <li>3. El usuario excoge uno de los pedidos de la lista y le da a anular.</li> <li>4. El sistema comunicaría al usuario que el pedido ha sido anulado con éxito.</li> </ol>		
<b>Curso alternativo</b>		
3a. Si se ha pasado el plazo de anulación, el usuario no tendrá acceso a la anulación del pedido.		

Tabla 4.16: Anular pedidos

- Consultar los Pedidos

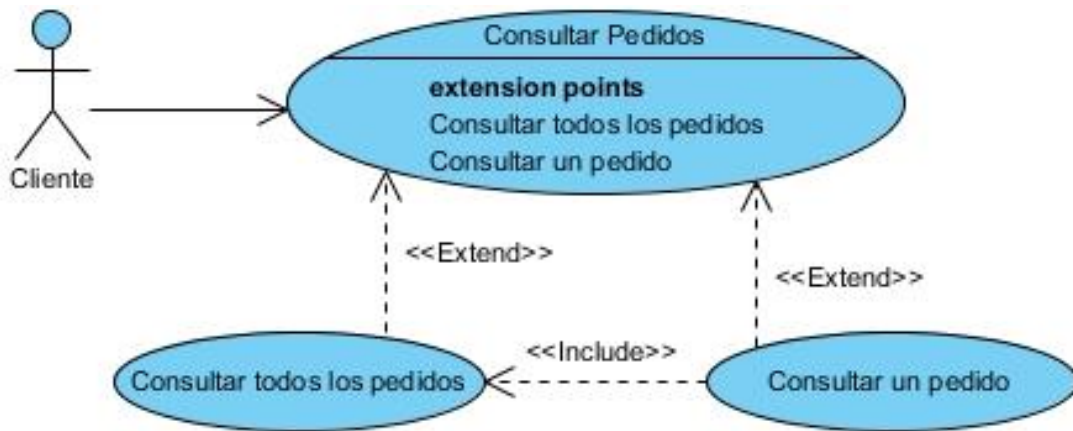


Ilustración 4.6: Diagrama de casos de uso “Consultar los pedidos”

CASO DE USO	Consultar todos los pedidos (Ilustración 4.6)
<b>Actores</b>	Cliente
<b>Prerrequisitos</b>	El usuario ya se ha autenticado.
<b>Propósito</b>	El cliente desea consultar todos los pedidos que ha realizado.
<b>Resumen</b>	El usuario selecciona la opción de consultar todos los pedidos y el sistema le mostrará un listado con todos los pedidos realizados por él mismo.
<b>Curso normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de consultar todos los pedidos.</li> <li>2. El sistema muestra una lista con todos los pedidos realizados por él.</li> </ol>

Tabla 4.17: Consultar todos los pedidos

CASO DE USO Consultar un pedido (Ilustración 4.6)	
<b>Actores</b>	Cliente
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha tenido que consultar la lista de pedidos.
<b>Propósito</b>	
El cliente desea consultar la información más detallada de uno de los pedidos.	
<b>Resumen</b>	
Una vez que el usuario a consultado todos los pedidos, este seleccionará uno de ellos para que el sistema le proporcione la información detallada de dicho pedido.	
<b>Curso normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona un pedido de la lista.</li> <li>2. El sistema muestra una ventana con la información detallada de este.</li> </ol>	

Tabla 4.18: Consultar un pedido

- **Gestionar Comentarios**

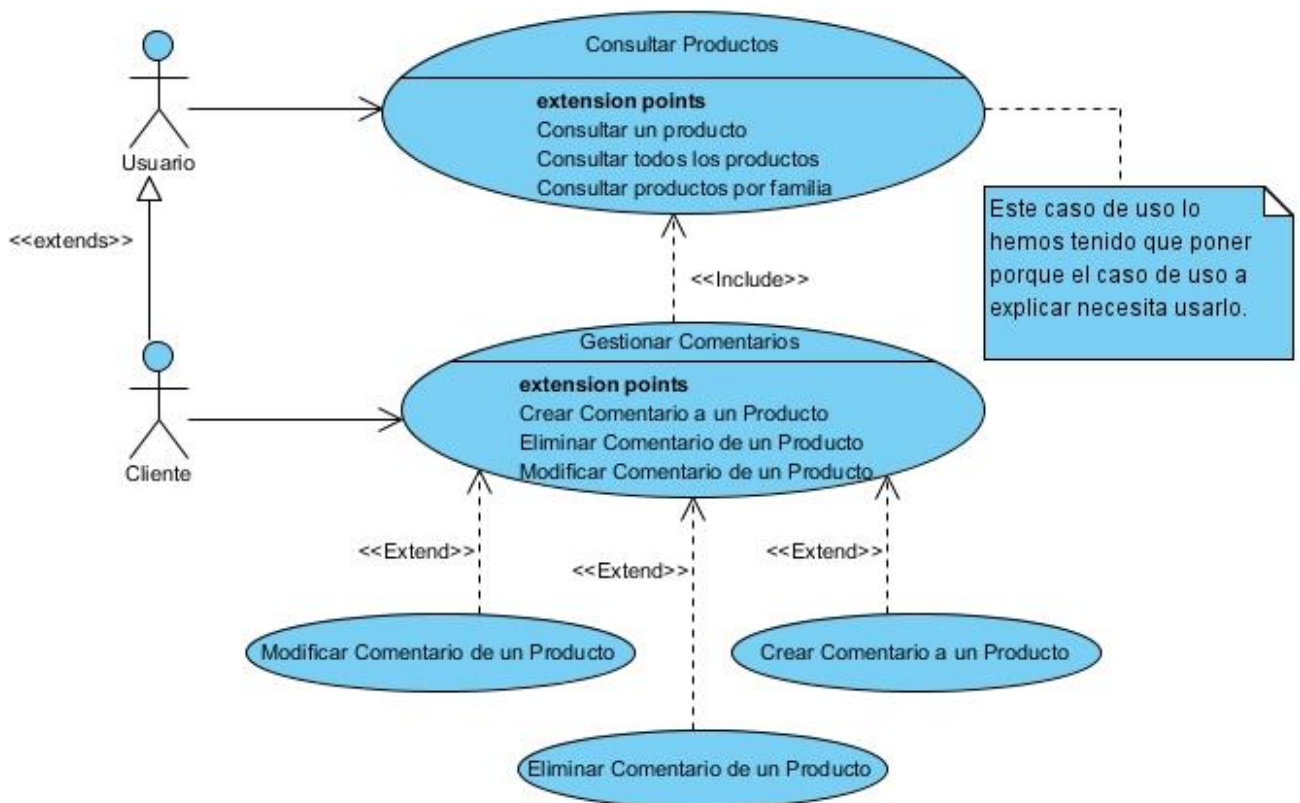


Ilustración 4.7: Diagrama de casos de uso "Gestionar comentarios"

<b>CASO DE USO</b>		<b>Crear comentario a un Producto(Ilustración 4.7)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y ha consultado el producto.
<b>Propósito</b>		
El cliente desea añadir un comentario a uno de los productos.		
<b>Resumen</b>		
Una vez que el usuario ha buscado el producto, este podrá añadir un comentario sobre ese producto.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona el producto deseado.</li> <li>2. El sistema muestra la ventana con la información del producto.</li> <li>3. El usuario introduce un comentario y lo envía.</li> <li>4. El sistema lo almacena.</li> </ol>		

Tabla 4.19: Crear comentario a un producto

<b>CASO DE USO</b>		<b>Eliminar comentario a un Producto (Ilustración 4.7)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y ha consultado el producto.
<b>Propósito</b>		
El cliente desea eliminar un comentario que haya realizado.		
<b>Resumen</b>		
Una vez que el usuario ha buscado el producto, este podrá eliminar un comentario que haya hecho él sobre ese producto.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona el producto deseado.</li> <li>2. El sistema muestra la ventana con la información del producto.</li> <li>3. El usuario localiza el comentario y le da a eliminar.</li> <li>4. El sistema lo elimina.</li> </ol>		

Tabla 4.20: Eliminar comentario a un producto

<b>CASO DE USO</b>		<b>Modificar comentario de un Producto (Ilustración 4.7)</b>
<b>Actores</b>		Cliente
<b>Prerrequisitos</b>		El usuario ya se ha autenticado y ha consultado el producto.
<b>Propósito</b>		
El cliente desea modificar un comentario que ha hecho.		
<b>Resumen</b>		
Una vez que el usuario ha buscado el producto, este podrá modificar un comentario que haya hecho él sobre ese producto.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona el producto deseado.</li> <li>2. El sistema muestra la ventana con la información del producto.</li> <li>3. El usuario localiza el comentario y le da a modificar.</li> <li>4. El sistema muestra una ventana para modificar el comentario.</li> <li>5. El usuario lo modifica y lo envía.</li> <li>6. El sistema lo almacena.</li> </ol>		

Tabla 4.21: Modificar comentario de un producto

#### 4.2.1.4. Caso de uso del actor Empleado

En este caso de uso vamos a especificar los casos de uso que puede hacer específicamente el actor empleado, ya que este actor también puede realizar todos los anteriores. De esta manera nos quedará un caso de uso más fácil de entender y de ver.

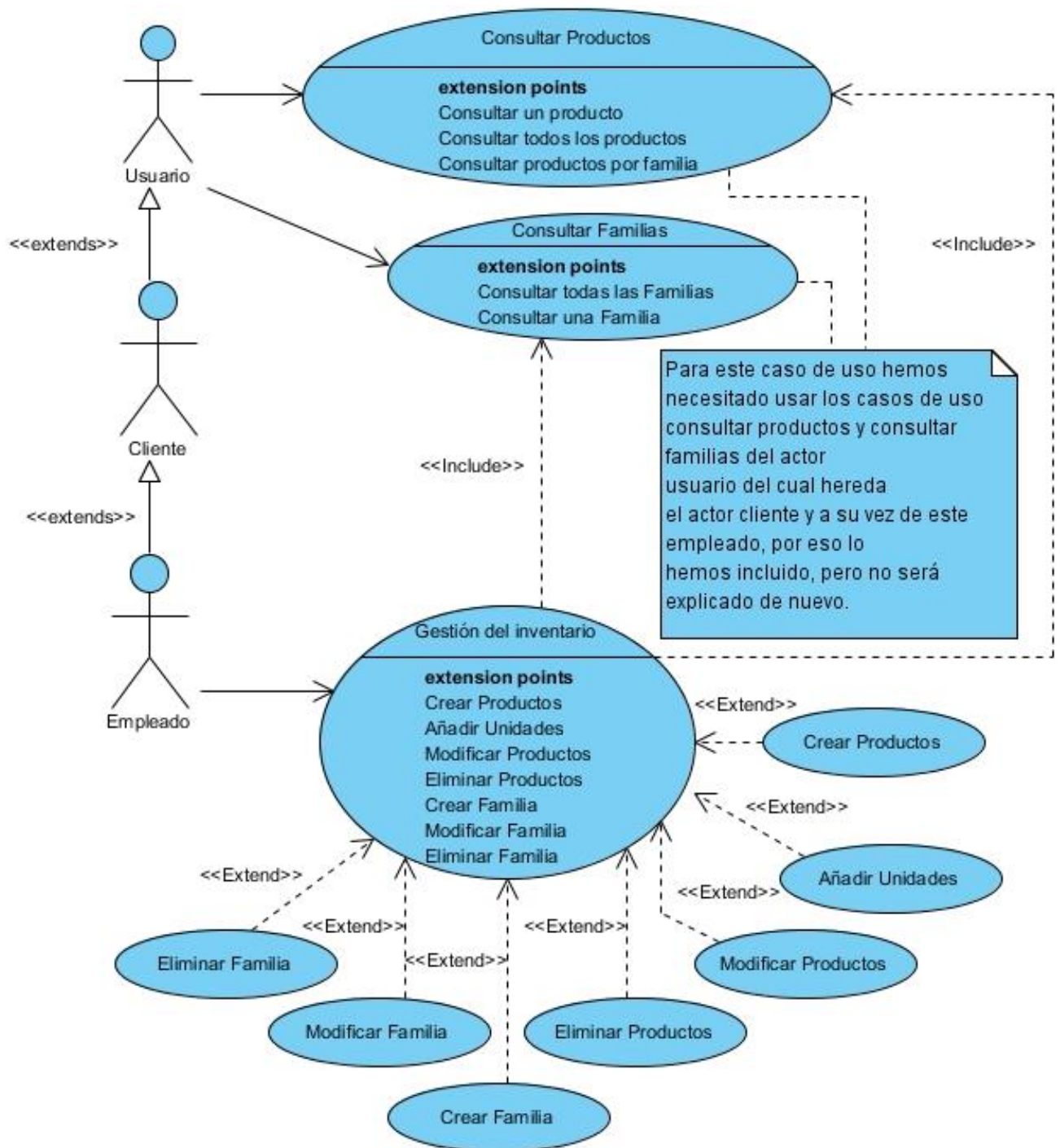


Ilustración 4.8: Diagrama de casos de uso del actor empleado

El caso de uso Gestión del inventario es abstracto, por lo que sólo especificaremos los casos de uso que heredan de él.

<b>CASO DE USO</b>		<b>Crear Productos (Ilustración 4.8)</b>
<b>Actores</b>		Empleado
<b>Prerrequisitos</b>		El usuario ya se ha autenticado.
<b>Propósito</b>		
El empleado desea introducir un nuevo producto al sistema, para que sea visible en la aplicación.		
<b>Resumen</b>		
El empleado selecciona la opción de crear producto y le aparecerá una ventana para que rellene toda la información que se necesita a la hora de crear un nuevo producto.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de crear producto.</li> <li>2. El sistema muestra una ventana para que el empleado rellene toda la información necesaria sobre el producto.</li> <li>3. El usuario rellena dichos datos y los envía.</li> <li>4. El sistema comprueba que no exista un producto igual y comunica al usuario que el producto ha sido creado con éxito.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si existe un producto igual ya insertado, el sistema muestra al usuario un error y le permitirá cambiar los datos del producto, para que no se oferten 2 productos iguales en la aplicación.		

Tabla 4.22: Crear producto



<b>CASO DE USO</b>		<b>Añadir Unidades (Ilustración 4.8)</b>
<b>Actores</b>	Empleado	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha buscado el producto.	
<b>Propósito</b>		
El empleado desea añadir unidades a los productos que ya existen en el sistema. Esta funcionalidad sirve para aumentar el stock de un producto.		
<b>Resumen</b>		
El empleado selecciona la opción de añadir productos y el sistema le mostrará una ventana con todos los productos de la empresa. Una vez encontrado el producto deseado se seleccionará y se dará la opción de cuantas unidades quiere añadir. Si no lo encuentra debería de crearlo con la opción de arriba.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de añadir unidades al producto.</li> <li>2. El sistema muestra una ventana con todos los productos.</li> <li>3. El usuario selecciona el producto deseado.</li> <li>4. El sistema muestra una ventana con el producto seleccionado y da la opción de insertar nuevas unidades.</li> <li>5. El usuario introduce el número de unidades que quiere añadir y las envía.</li> <li>6. El sistema las guarda.</li> </ol>		
<b>Curso alternativo</b>		
6a. Si el número de unidades no es correcto, es decir, está fuera de unos límites, el sistema mostraría un error y nos permitiría cambiarlo.		

Tabla 4.23: Añadir unidades

<b>CASO DE USO</b>		<b>Modificar Productos (Ilustración 4.8)</b>
<b>Actores</b>	Empleado	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha buscado el producto.	
<b>Propósito</b>		
El empleado desea modificar los datos de algún producto del sistema.		
<b>Resumen</b>		
El empleado selecciona la opción de modificar productos y el sistema le mostrará una ventana con todos los productos. Una vez encontrado el que deseamos modificar, se selecciona y el sistema le mostrará otra ventana con todos los datos del producto para que pueda cambiarlos.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de modificar producto.</li> <li>2. El sistema muestra una ventana con todos los productos.</li> <li>3. El usuario selecciona el producto a modificar.</li> <li>4. El sistema muestra una ventana con todos los datos del producto buscado.</li> <li>5. El usuario cambiará los datos que sean necesarios y los enviará.</li> <li>6. El sistema actualiza el producto.</li> </ol>		
<b>Curso alternativo</b>		
6a. Si algún dato no es correcto, el sistema mostrará un error y nos permitirá cambiar los datos que son erróneos.		

Tabla 4.24: Modificar productos

<b>CASO DE USO</b>		<b>Eliminar Productos (Ilustración 4.8)</b>
<b>Actores</b>	Empleado	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha buscado el producto.	
<b>Propósito</b>		
El empleado desea eliminar un producto por completo de la aplicación, porque deje de ofertarse en la empresa.		
<b>Resumen</b>		
El empleado selecciona la opción de eliminar productos y el sistema le mostrará una ventana con todos los productos y dará la opción de eliminarlos.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción de eliminar productos.</li> <li>2. El sistema muestra una ventana con todos los productos.</li> <li>3. El usuario elige el que desea eliminar y le da a eliminar.</li> <li>4. El sistema mostrará una advertencia de si en realidad quiere eliminarlo.</li> <li>5. El usuario seleccionará que "sí".</li> <li>6. El sistema eliminará el producto.</li> </ol>		
<b>Curso alternativo</b>		
6a. Si el usuario selecciona "no" el sistema no eliminará el producto y nos volvería a mostrar toda la lista de productos.		

Tabla 4.25: Eliminar productos

<b>CASO DE USO</b>		<b>Crear Familia (Ilustración 4.8)</b>
<b>Actores</b>		Empleado
<b>Prerrequisitos</b>		El usuario ya se ha autenticado.
<b>Propósito</b>		
El empleado desea crear una familia de productos para ordenarlos más fácilmente de esta manera.		
<b>Resumen</b>		
El empleado selecciona la opción de crear familia, rellena todos los datos pertinentes y el sistema la crea.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario selecciona esta opción.</li> <li>2. El sistema muestra una ventana para rellenar los datos de la familia.</li> <li>3. El usuario los rellena y los envía.</li> <li>4. El sistema la crea.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si los datos no son correctos o la familia ya existe, el sistema nos permitirá cambiarlos y enviarlos de nuevo.		

Tabla 4.26: Crear familia

<b>CASO DE USO</b>		<b>Modificar Familia (Ilustración 4.8)</b>
<b>Actores</b>	Empleado	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha consultado las familias.	
<b>Propósito</b>		
El empleado desea modificar los datos de una familia de productos.		
<b>Resumen</b>		
El empleado selecciona una familia y le da a modificar, el sistema nos muestra una ventana con los datos de la familia, el usuario rellena todos los datos pertinentes y el sistema la modifica.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta las familias</li> <li>2. El sistema muestra una ventana con la lista de familias</li> <li>3. El usuario selecciona una y le da a modificar.</li> <li>4. El sistema muestra una ventana con la información de dicha familia.</li> <li>5. El usuario modifica los datos que quiera y los envía.</li> <li>6. El sistema la modifica.</li> </ol>		
<b>Curso alternativo</b>		
4a. Si los datos no son correctos o, al cambiar el nombre de la familia, esta existe, el sistema nos permitirá cambiarlos y enviarlos de nuevo.		

Tabla 4.27: Modificar familia

<b>CASO DE USO</b>		<b>Eliminar Familia (Ilustración 4.8)</b>
<b>Actores</b>	Empleado	
<b>Prerrequisitos</b>	El usuario ya se ha autenticado y ha consultado la lista de familias.	
<b>Propósito</b>		
El empleado desea eliminar una familia de productos, porque ya no contenga productos o porque ya no se vaya a usar más.		
<b>Resumen</b>		
El empleado consulta la lista de familias, el sistema se las muestra y el empleado podrá eliminar la familia que quiera.		
<b>Curso normal</b>		
<ol style="list-style-type: none"> <li>1. El usuario consulta las familias</li> <li>2. El sistema muestra la lista de familias</li> <li>3. El usuario selecciona una y le da a eliminar.</li> <li>4. El sistema muestra una advertencia de si en realidad quiere eliminar la familia.</li> <li>5. El usuario selecciona que “sí”.</li> <li>6. El sistema la elimina.</li> </ol>		
<b>Curso alternativo</b>		
<p>3a. Si la familia tiene asignados productos el sistema nos mostraria un error, porque una familia con productos asignados no se puede eliminar.</p> <p>6a. Si el usuario selecciona “no”, el sistema no eliminará nada y este nos volverá a mostrar toda la lista de familias.</p>		

Tabla 4.28: Eliminar familia

Como ya dijimos anteriormente, los demás casos de uso que son heredados de usuario y de cliente no se van a volver a especificar puesto que no vamos a repetir lo mismo dos veces. Solo hay que tener claro que el usuario empleado puede realizar también todos los casos de uso de ambos.

### 4.2.2. Diagramas de secuencia

A continuación se muestran los diagramas de secuencia asociados a los caso de uso expuesto anteriormente. No los vamos a exponer todos para no extender demasiado la documentación, sólo los diagramas más complejos e interesantes para el uso de nuestra aplicación.

- **Registro**

Como podemos observar en la ilustración 4.9, una vez rellenos los datos de registro, se comprueban, y si son erróneos devuelve un error, permitiendonos cambiarlos, mientras que si son correctos, se registrará al usuario.

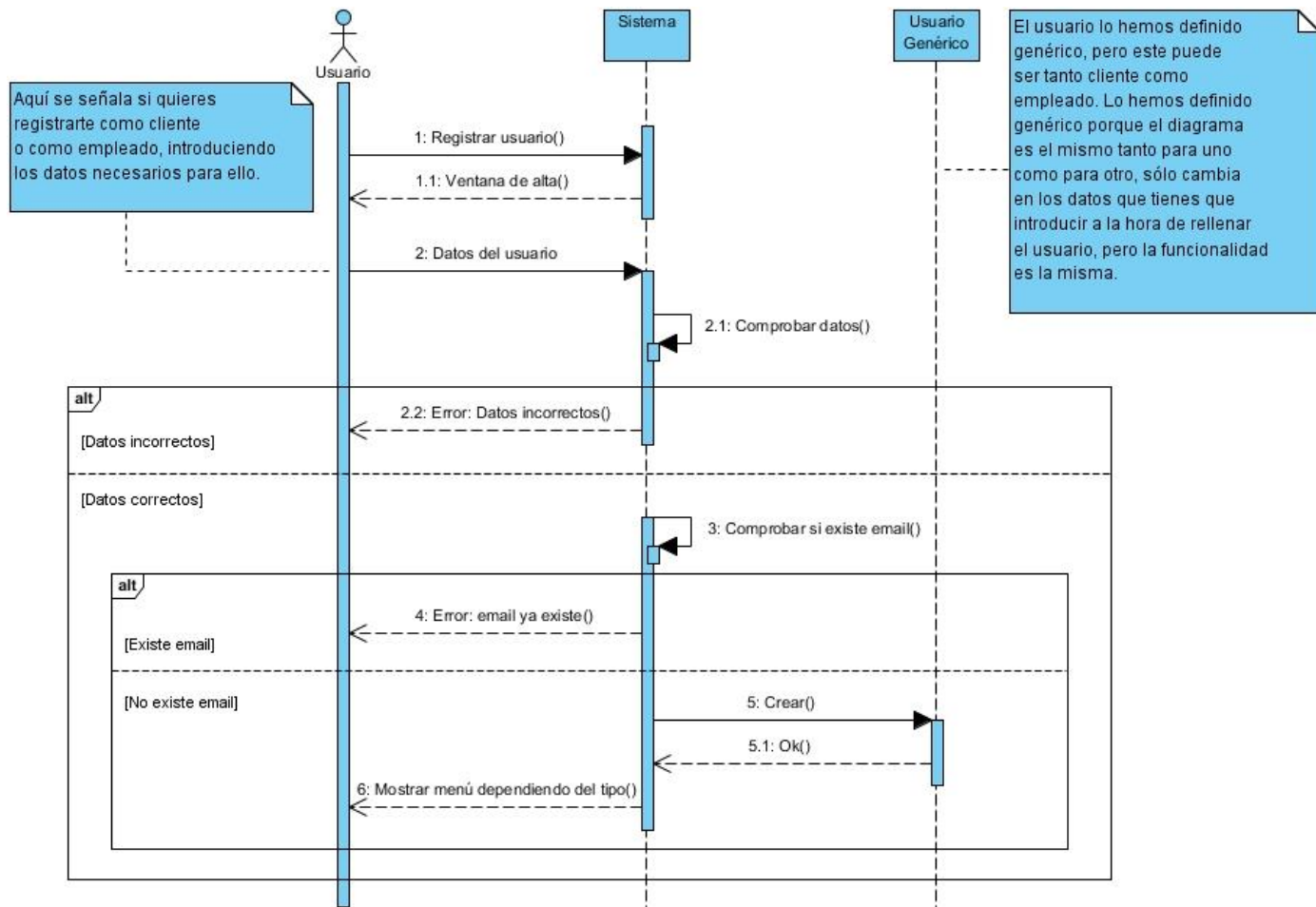
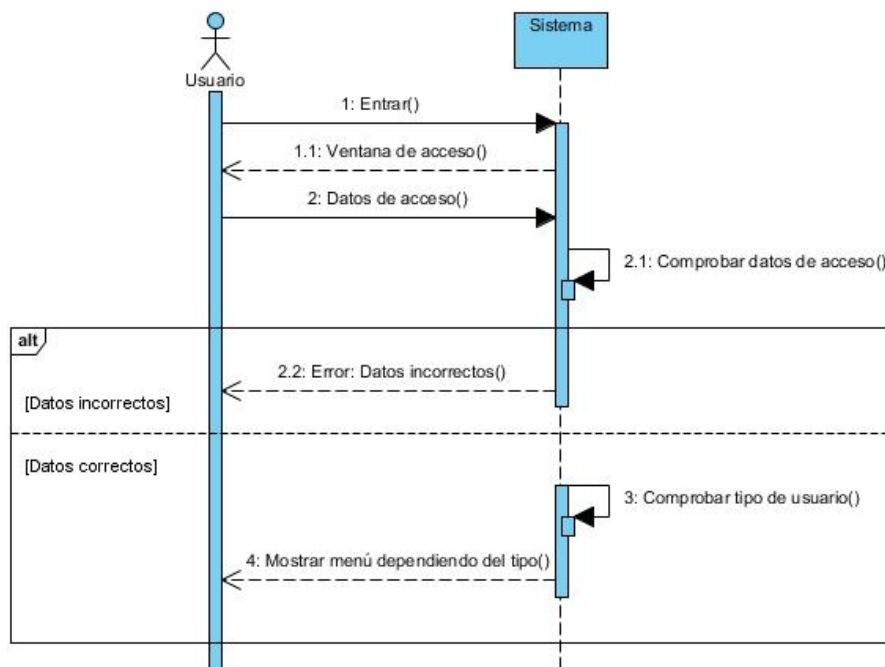


Ilustración 4.9: Diagrama de secuencia “Registro”

- **Iniciar Sesión**

En la ilustración 4.10 podemos apreciar como, una vez introducidos los datos de acceso, estos se comprueban y si son incorrectos nos muestra un error y nos permite volverlos a introducir, mientras que si son correctos, accederemos al sistema.



**Ilustración 4.10: Diagrama de secuencia “Iniciar sesión”**



- **Consultar un producto**

Para consultar un producto, como vemos en la ilustración 4.11, es necesario consultar antes o todos los productos o todos los productos por familias. Una vez hecho esto, ya podríamos consultar un producto seleccionandolo en dichas listas.

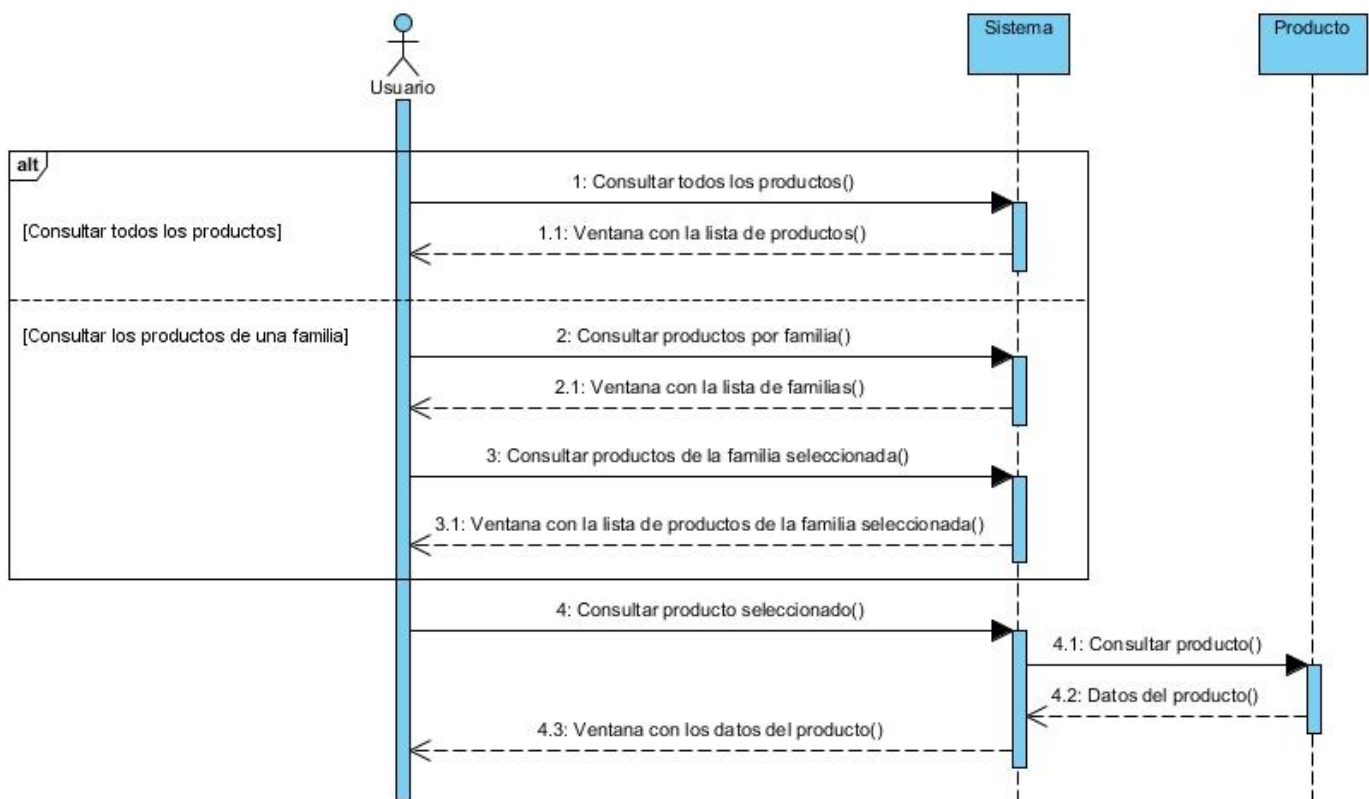


Ilustración 4.11: Diagrama de secuencia “Consultar un producto”

- **Modificar datos de registro**

Para modificar los datos, primeramente hay que consultarlos. Una vez que los tenemos, podemos modificarlos y enviarlos y el sistema se encarga de comprobar si los datos son correctos, si es así los modifica, o si no lo son, que en ese caso mostraría un error y nos permitiría cambiarlos.

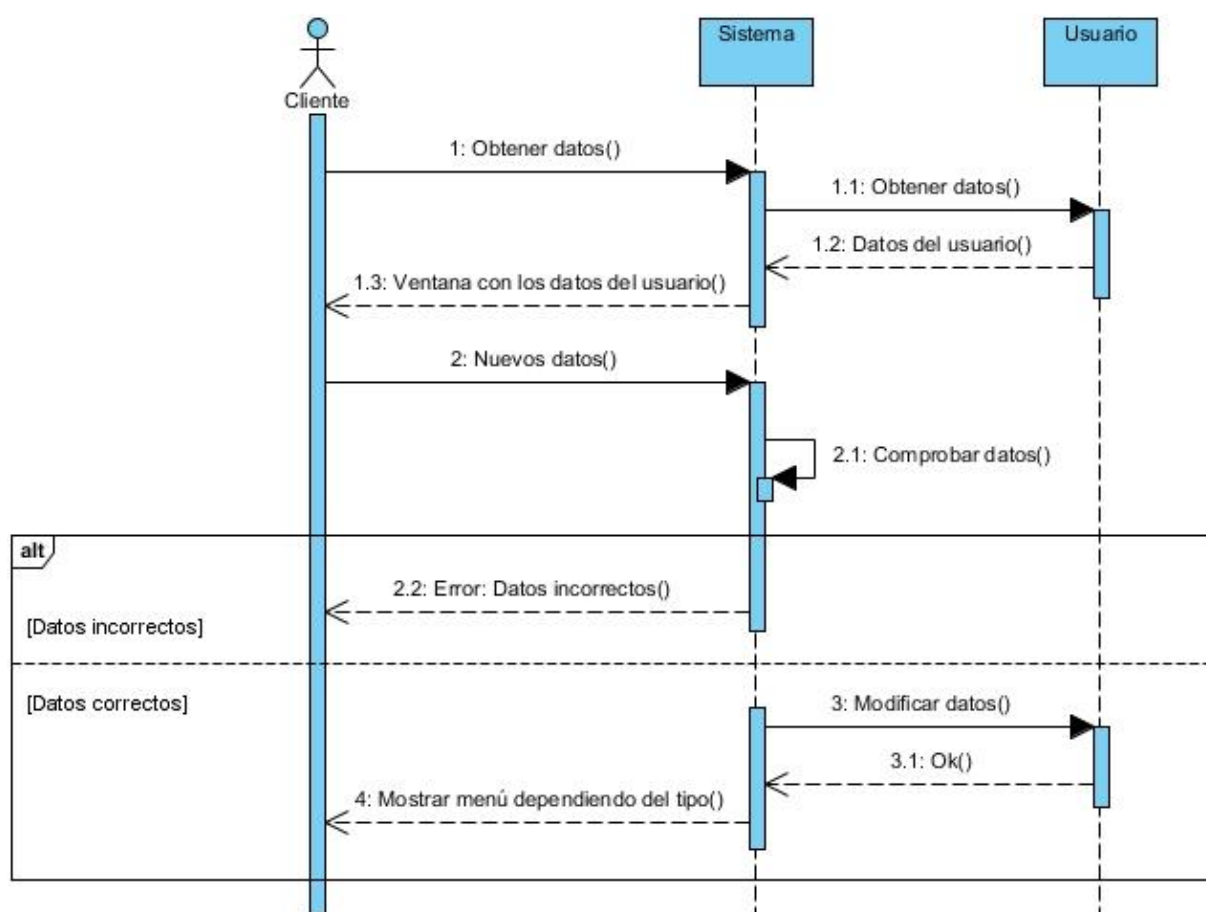


Ilustración 4.12: Diagrama de secuencia “Modificar datos de registro”

○ **Añadir línea al carrito**

Para añadir líneas al carrito, lo primero que hay que hacer es consultar un listado cualquiera de productos y cuando lo tengamos, para cada producto añadir se selecciona y se le da a añadir. Una vez hecho esto nos pedirá que cuantas unidades queremos y al ingresarlas, si es una cantidad válida, creará la línea, de lo contrario, mostraría un error y nos permitiría cambiarla.

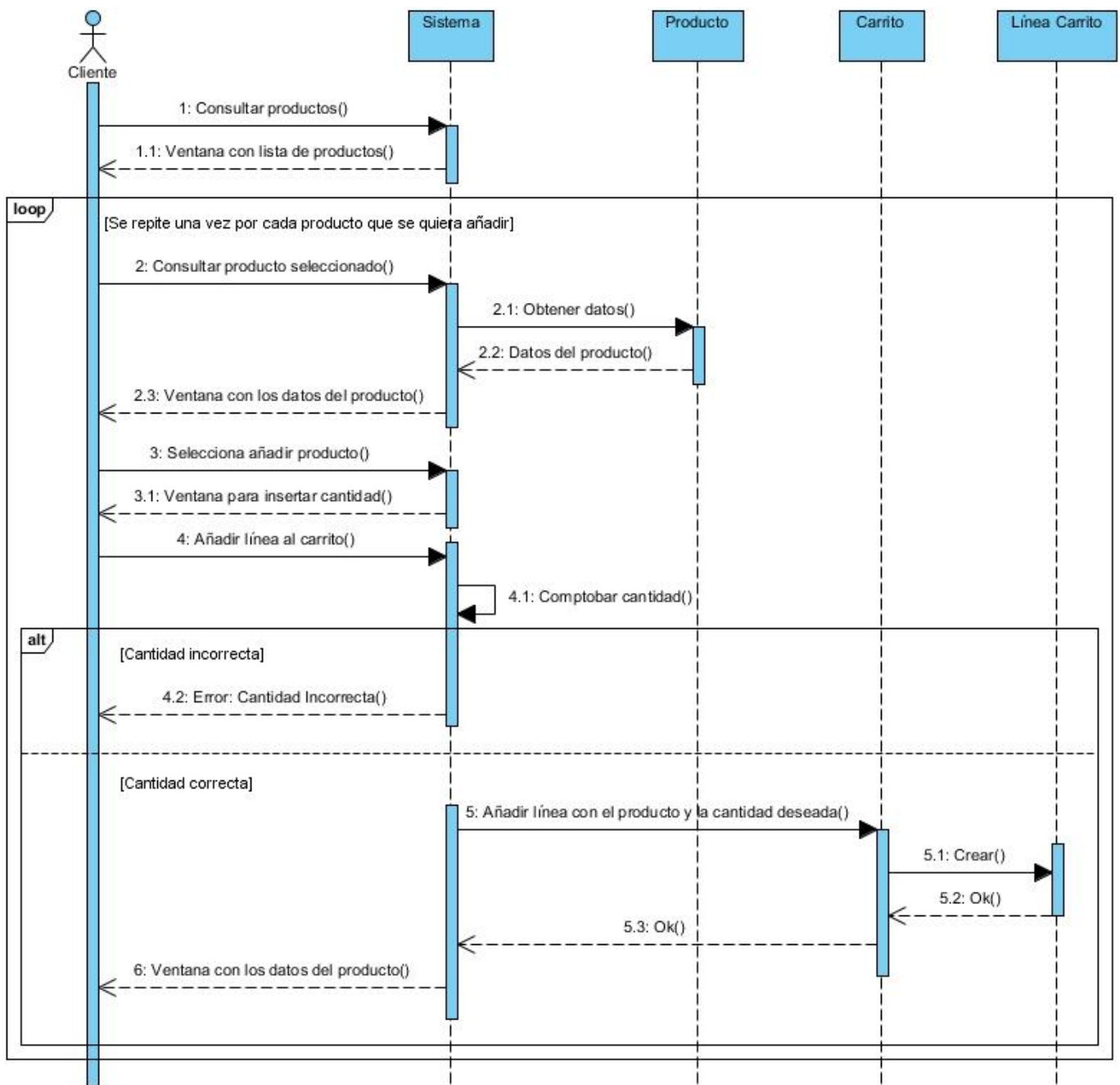


Ilustración 4.13: Diagrama de secuencia “Añadir línea al carrito”

○ **Modificar cantidad línea de carrito**

Una vez consultado el carrito u seleccionada la línea deseada, se introduce la cantidad nueva, y si esta cantidad es válida, se modificará la línea, de lo contrario, se nos mostrará un error y nos permitirá cambiarla.

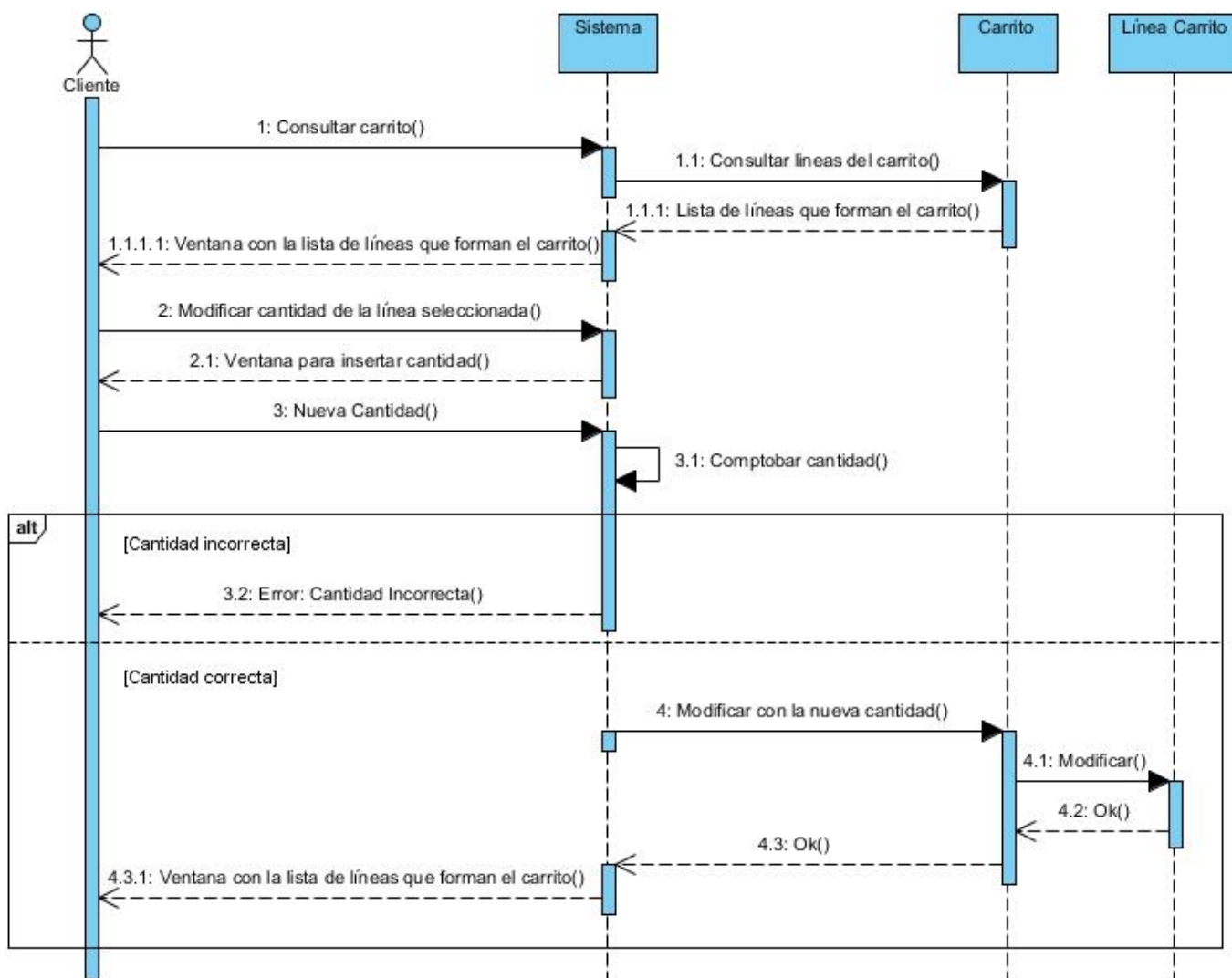


Ilustración 4.14: Diagrama de secuencia “Modificar cantidad línea de carrito”

○ **Quitar línea del carrito**

Una vez consultado el carrito, se selecciona la línea que se desea eliminar y esta es eliminada del sistema.

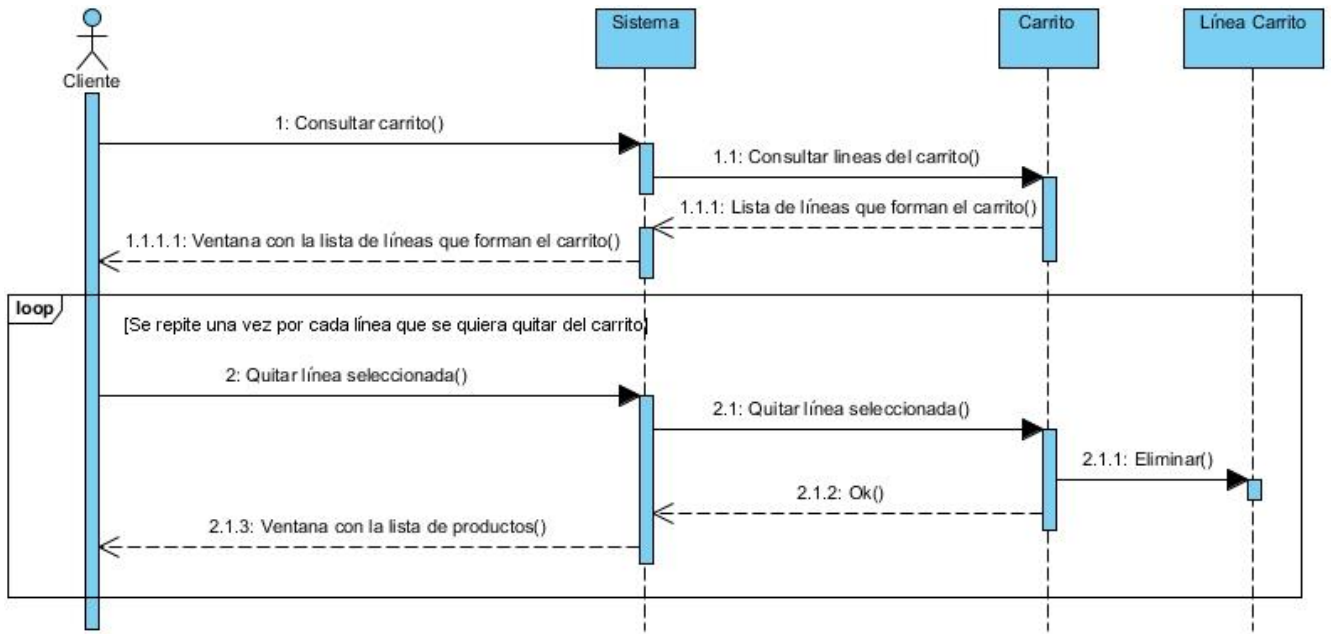


Ilustración 4.15: Diagrama de secuencia “Quitar línea del carrito”

○ **Consultar un pedido**

Una vez consultado el listado de pedidos, seleccionamos uno cualquiera de dicho listado y se nos mostrará toda la información del pedido seleccionado.

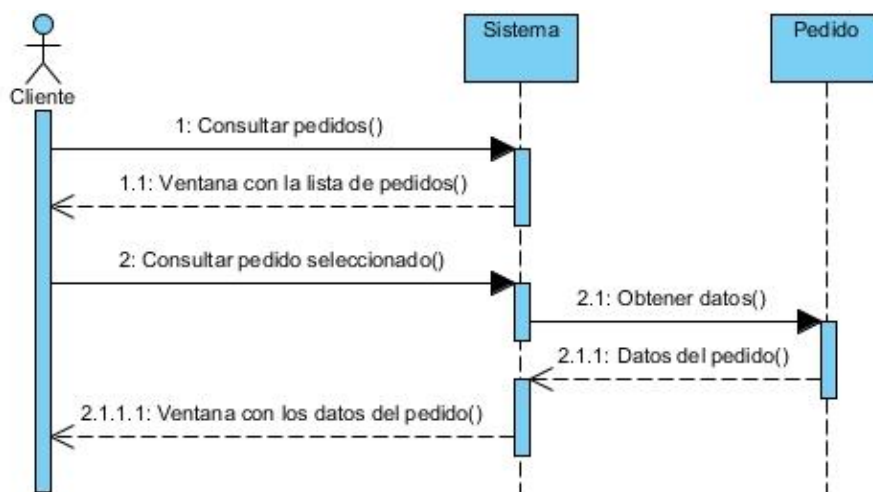


Ilustración 4.16: Diagrama de secuencia “Consultar un pedido”

○ **Crear pedidos**

Para poder crear pedidos, hay que consultar antes el carrito y una vez consultado, se pasaría a crear el pedido con los productos que lo componen. Antes de crearlo, hay que introducir los datos de envío. Una vez hecho esto, el sistema comprobará si son correctos, si lo son, se creará el pedido, y si no lo son, se nos mostrará un error y nos permitirá cambiarlos.

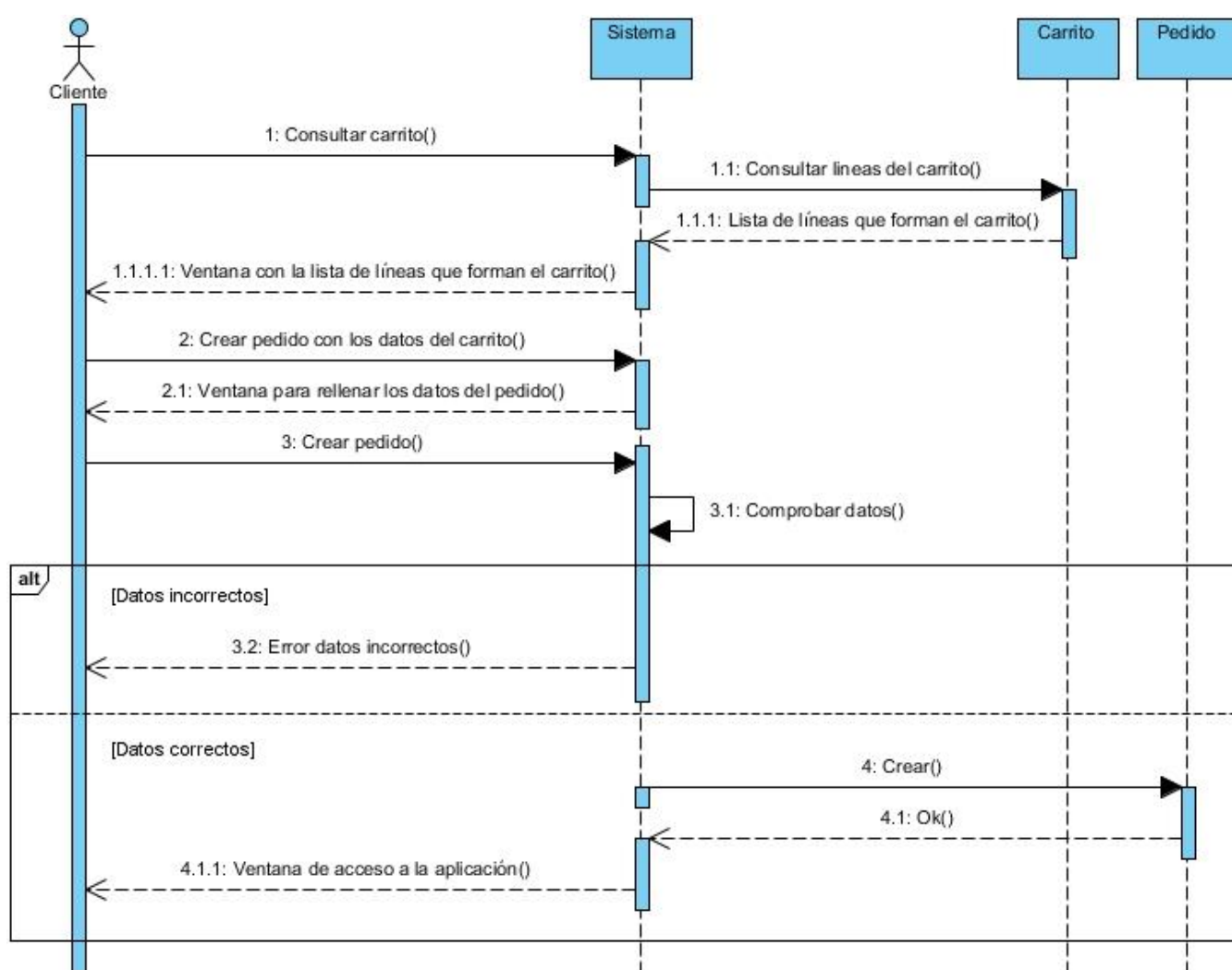


Ilustración 4.17: Diagrama de secuencia “Crear pedidos”

○ **Anular pedidos**

Para anular los pedidos hay que consultar el pedido que se desea anular y si no se ha pasado el período de anulación se podrá anular.

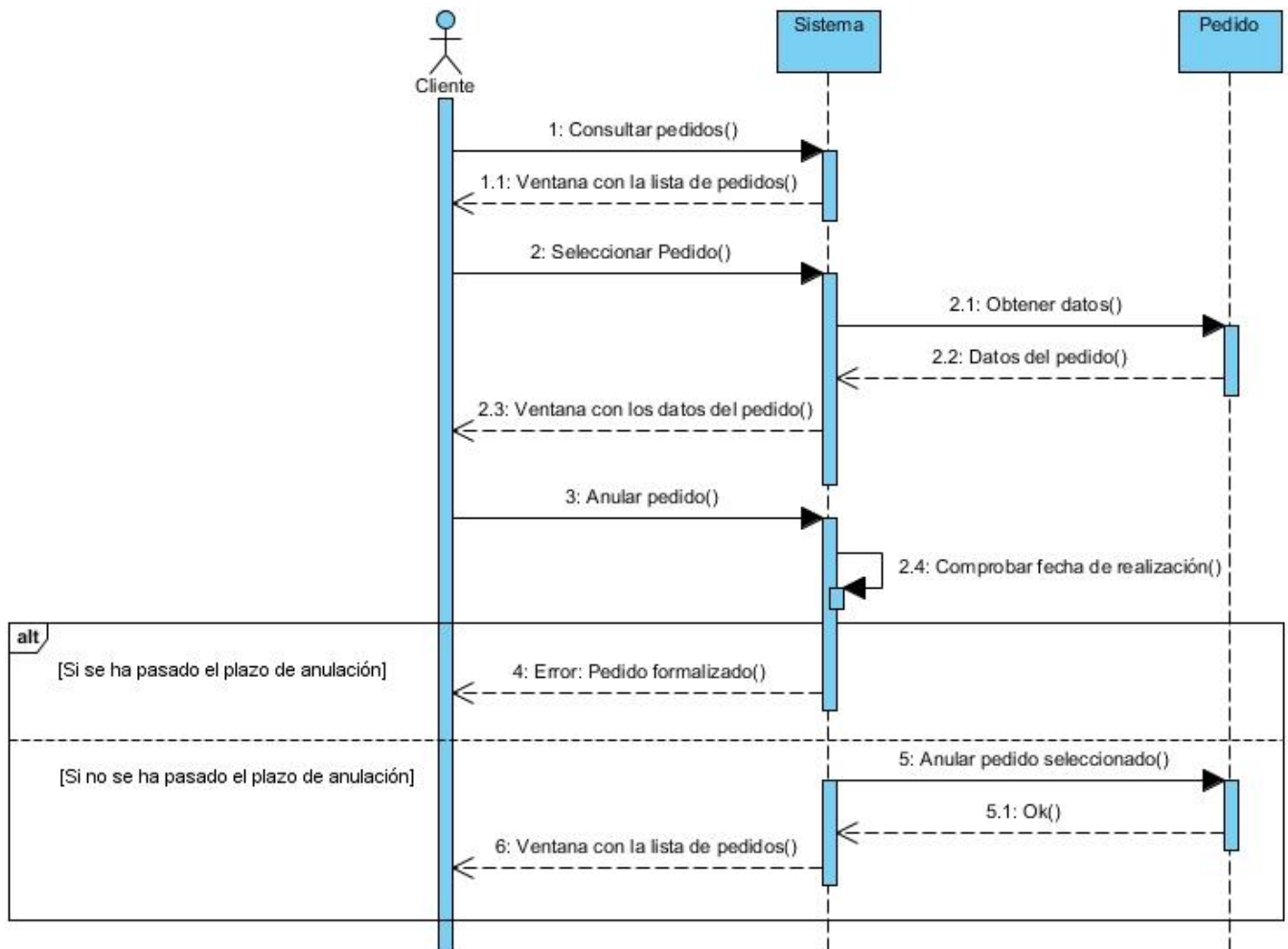


Ilustración 4.18: Diagrama de secuencia “Anular pedidos”

- **Eliminar productos**

Para la eliminación de estos, hay que consultar cualquier lista de productos y una vez hecho esto, se seleccionará el producto a eliminar y éste será eliminado.

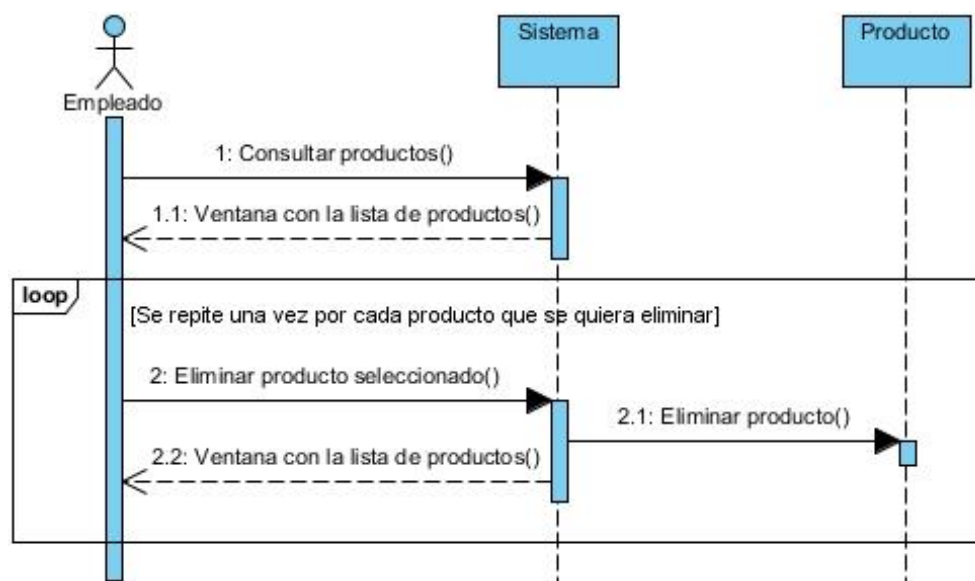


Ilustración 4.19: Diagrama de secuencia "Eliminar productos"



- **Crear productos**

Para crear un producto, se ha de rellenar toda su información. Si los datos introducidos no son correctos, se nos mostrará un error y nos permitirá cambiarlos, si estos son correctos, se comprobará si el producto ya existe y si es así nos mostrará otro error y nos dejaría modificar los datos introducidos, de lo contrario, el producto se crearía con éxito.

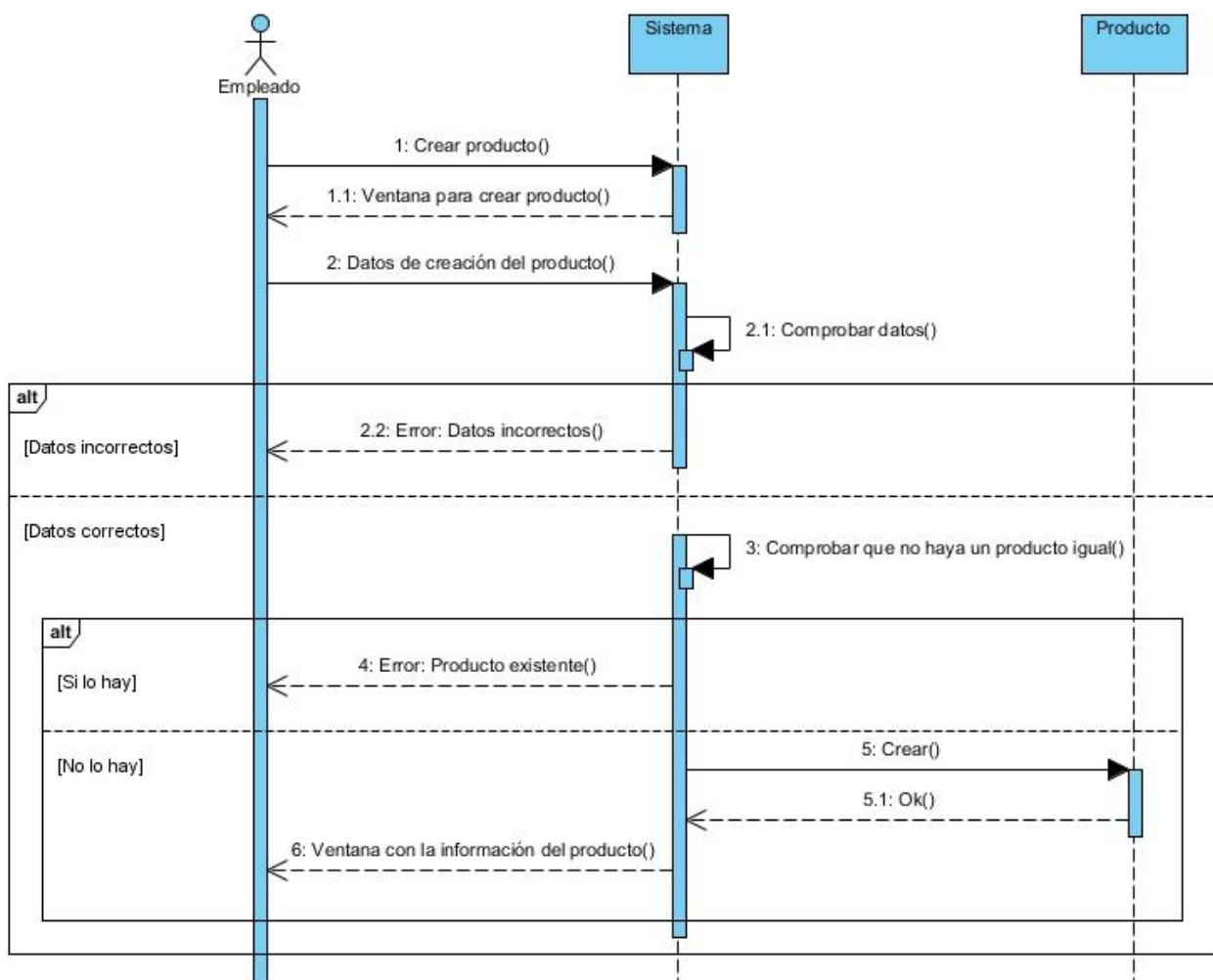


Ilustración 4.20: Diagrama de secuencia “Crear productos”

- **Añadir Unidades**

Para añadir unidades, hay que consultar el producto deseado e introducir la cantidad a añadir. Si la cantidad es incorrecta, se nos mostraría un error y nos permitiría cambiarla, de lo contrario se añadiría la cantidad insertada.

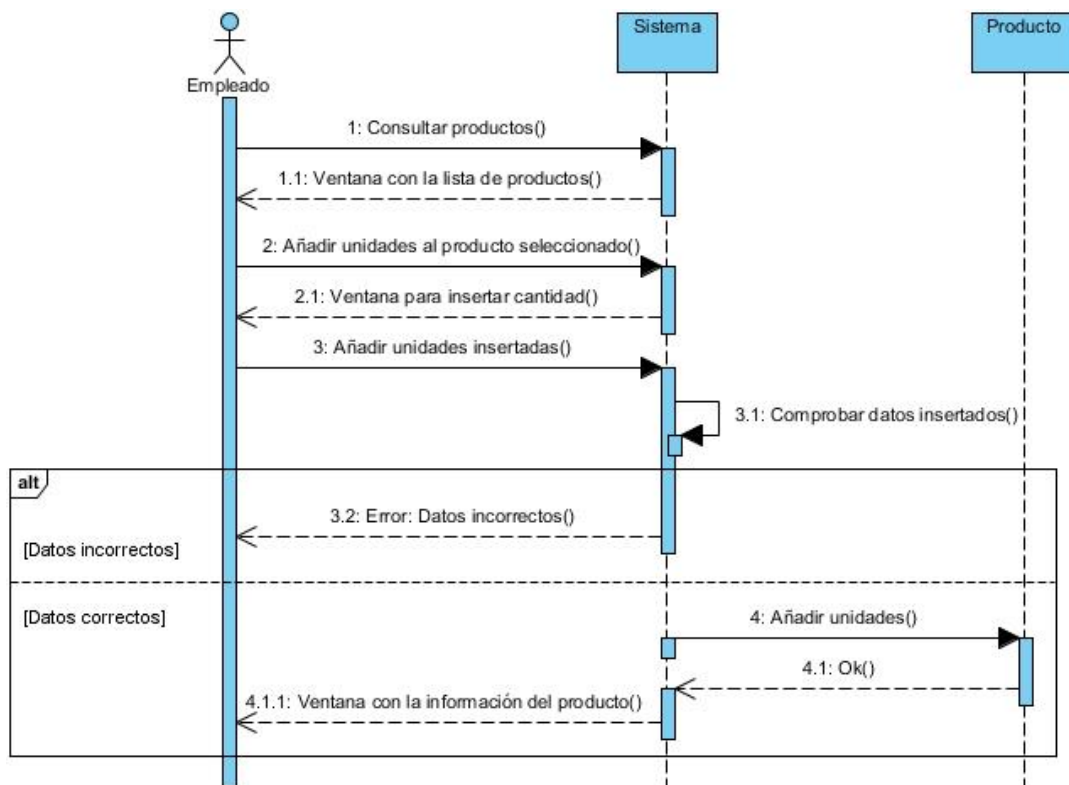


Ilustración 4.21: Diagrama de secuencia "Añadir unidades"

- **Modificar productos**

Hay que consultar el producto deseado para poder modificarlo. Una vez introducidos los datos, si estos son incorrectos, se nos mostraría un error y nos permitiría cambiar los datos introducidos, de lo contrario el producto se modificaría con éxito.

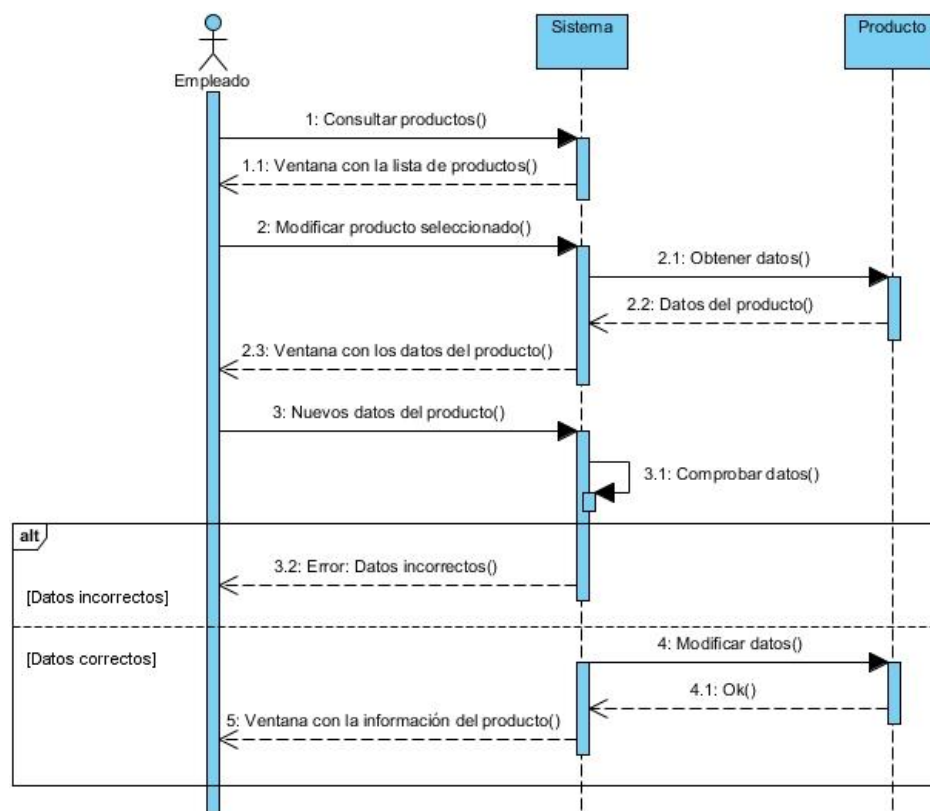


Ilustración 4.22: Diagrama de secuencia "Modificar productos"



# CAPÍTULO 5

---

Diseño

## 5.1. Introducción

En este capítulo veremos todo lo referente al diseño, donde expondremos el diagrama de clases de nuestra aplicación y explicaremos para que sirve cada una, también mostraremos el diagrama de clases de nuestro Servicio Web y por último enseñaremos el diseño de la base de datos.

## 5.2. Diagrama de clases

En este apartado se mostrará el diagrama de clases de nuestra aplicación. Este aparecerá sin atributos, para hacer más visible el diagrama.

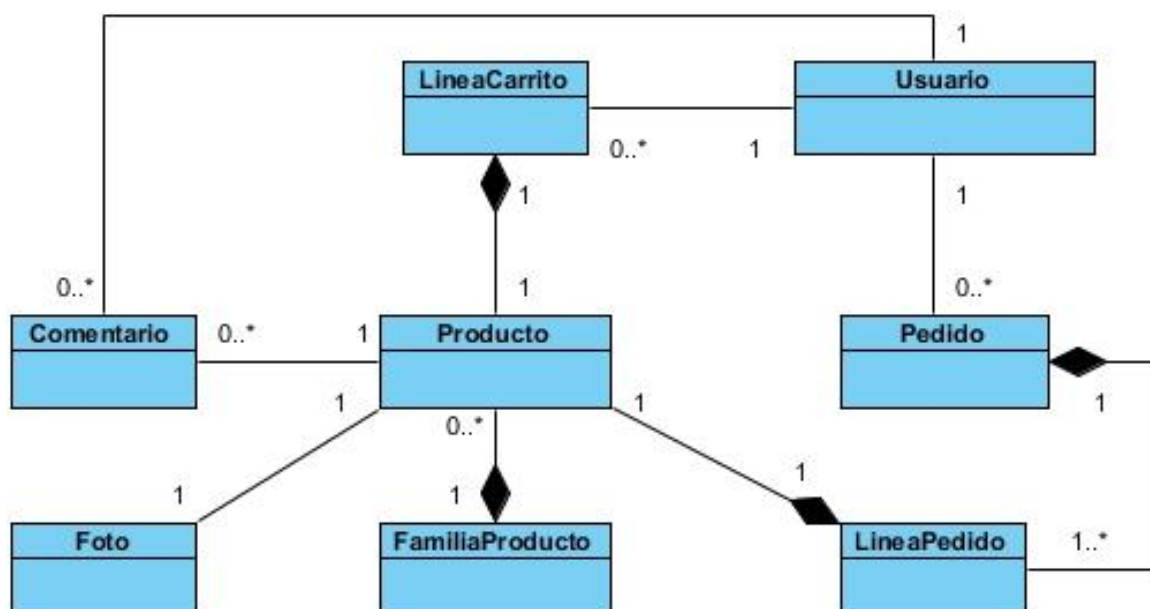


Ilustración 5.1: Diagrama de clases

En el diagrama (ilustración 29) podemos ver como un usuario tiene asociado de cero a muchas líneas de carrito y estas a su vez se componen de un producto.

Un producto a parte de todos sus atributos tiene asociado una fotos y cero o muchos comentarios y cada comentario tendrá asociado un usuario, lo mismo que cada usuario tendrá asociados de cero a muchos comentarios.

También podemos ver como un usuario esta asociado a cero o muchos pedidos que están compuestos de una a muchas líneas de pedido y esta a su vez se compone de un producto.

Por último vemos como una familia esta compuesta de cero a muchos productos.

### 5.3. Detalle de las clases

En el diagrama anterior (Ilustración 5.1) se muestran las distintas clases y todas las relaciones existentes entre ellas. En este apartado se explicará cada clase un poco más en detalle.

- **Usuario**

Esta clase va a contener todos los atributos necesarios para definir un Usuario.

- **Línea Carrito**

Esta clase nos va a permitir realizar la gestión de cada una de las líneas que un usuario añade al carrito, como va a ser la cantidad de dicho producto, como el precio que cuesta, etc.

- **Pedido**

Esta clase nos permitirá almacenar los datos de un pedido, para cuando el usuario desee consultarlos o por si este desea anularlo.

- **Línea Pedido**

Esta clase almacenará la información de cada una de las líneas que conforman el pedido, para cuando se muestra la información detallada de este, poder ofrecerle al usuario también una información detallada del producto al que se refiere dicha línea.

- **Producto**

Esta clase contendrá toda la información referente a un producto, para así, a la hora de consultarlo, poder dar una información detallada del mismo.

- **Comentario**

Esta clase va a contener un comentario que un usuario habrá dejado respecto a un producto.

- **Foto**

Esta clase contendrá una foto, la cual estará asociada a un producto, para así favorecer la visión de este a la hora de comprarlo.

- **FamiliaProducto**

Esta clase va a contener toda la información de una familia, para así poder ordenar los productos por estas y hacer su búsqueda más sencilla.



### 5.4. Diagrama de clases del Servicio Web

En este apartado se mostrará el diagrama de clases que forma nuestro servicio web.

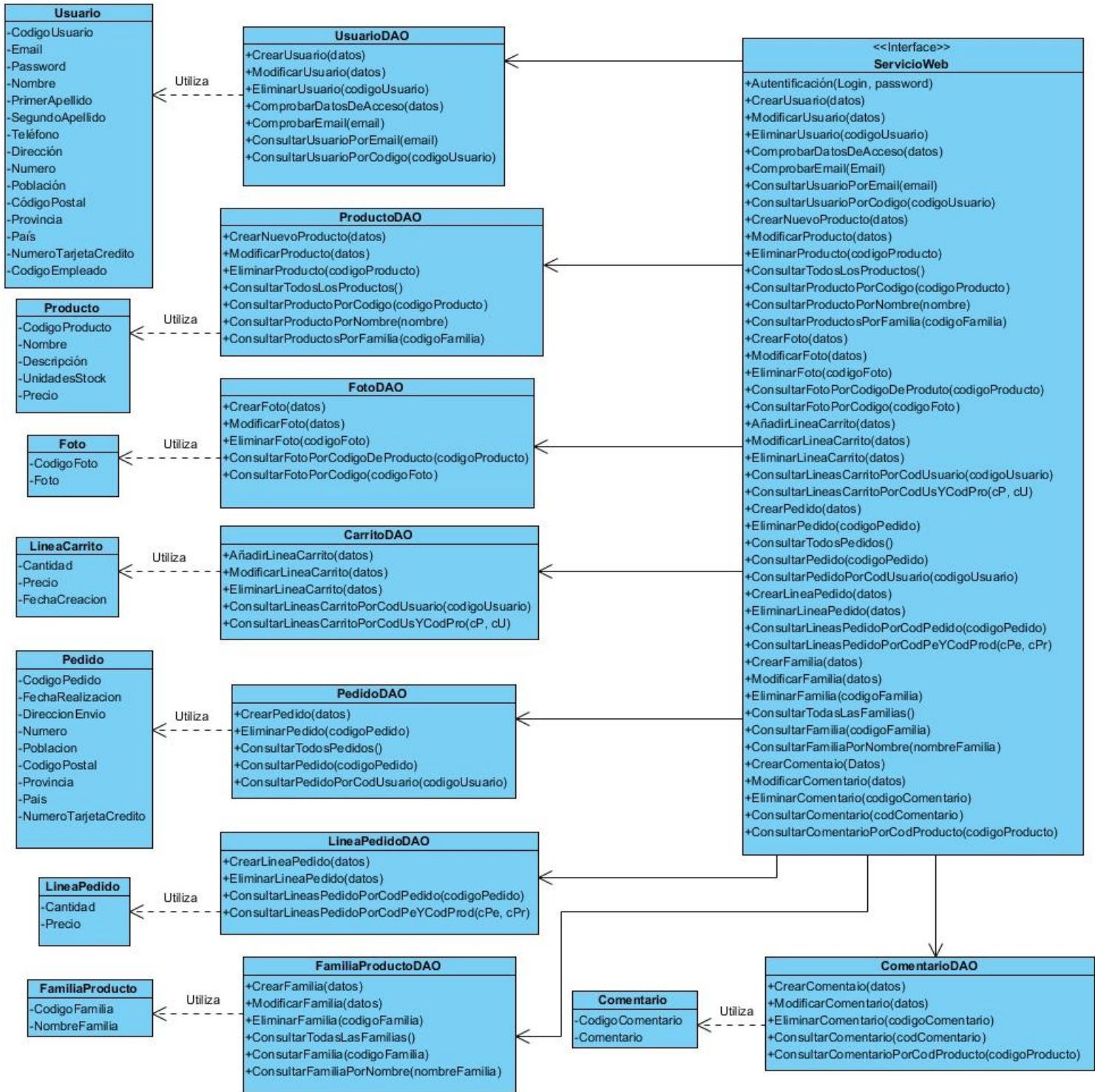


Ilustración 5.2: Diagrama de clases del Servicio Web

Como podemos observar (ilustración 5.2), para poder acceder a los datos a través de nuestro servicio web necesitamos crearnos una serie de clases a las que denominaremos DAO (Data Access Object).

La conexión se realizaría de la siguiente manera:

- La aplicación llama al servicio web.
- El servicio web accede a los métodos del DAO.
- El DAO se encarga de acceder a la base de datos y devolver los datos al servicio web. Este, a su vez, los devuelve a la aplicación.

En el diagrama tenemos todas las clases que son necesarias para que todas las funcionalidades de nuestro Servicio Web trabajen correctamente.

- En primer lugar, tenemos la clase ServicioWeb que viene a ser la interfaz principal del Servicio Web y dentro de ella podemos observar sus funcionalidades. Estas serán accesibles mediante la utilización de los métodos expuestos dentro de dicha clase.
- En segundo lugar, tenemos las clases que se van a encargar de la interacción con la base de datos. Estas son todas las que acaban en DAO (Data Access Object). Todas ellas utilizan los modelos como tipo de objetos y sus funcionalidades actúan directamente sobre la base de datos insertando, actualizando, consultando o eliminando tuplas.
- Por último, tenemos las ocho clases, que son los modelos que definen los tipos de objetos utilizados por el servicio.

## 5.5. Diseño de la base de datos

### 5.5.1. Descripción de las Entidades

Antes de enseñar el diagrama Entidad-Relación (Ilustración 5.3), vamos a describir más a fondo cada una de las entidades que lo conforman.

- **Usuario**

Esta clase va a contener todos los atributos necesarios para definir un Usuario.

Atributos:

- CódigoUsuario: (clave primaria) es el identificador del usuario.
- Email: es el email del usuario. Este nos va a hacer de login.
- Password: es la contraseña necesaria para la autenticación.
- Nombre: es el nombre del usuario.
- PrimerApellido: es el primer apellido del usuario.
- SegundoApellido: es el segundo apellido del usuario.
- Teléfono: es el teléfono del usuario.
- Dirección: es la dirección del usuario.
- Número: es el número que corresponde a la casa del usuario.
- Población: es la población del usuario.
- CódigoPostal: es el código postal del usuario.
- Provincia: es la provincia del usuario.
- País: es el país del usuario.
- NúmeroTarjetaCrédito: es el número de la tarjeta de crédito con la que el usuario va a realizar los pagos.
- CódigoDeEmpleado: Es el código que tendrá el usuario si es empleado de la empresa.

- **LíneaCarrito**

Almacenará un producto deseado por el usuario, junto con la cantidad que este quiera y el precio.

Atributos:

- CódigoUsuario: (clave primaria de esta Entidad y a parte clave foránea de la Entidad Usuario) es el código del usuario asignado a esta línea de carrito.
- CódigoProducto: (clave primaria de esta Entidad y a parte clave foránea de la Entidad Producto) es el código del producto asignado a esta línea de carrito.
- Cantidad: es el número de unidades que el usuario desea comprar de dicho producto.
- Precio: es el precio total de la línea de carrito.
- FechaCreación: es la fecha de creación de la línea.

- **Pedido**

Almacenará los datos de un pedido realizado por el usuario.

Atributos:

- CódigoPedido: (clave primaria) es el código asociado a un pedido.
- FechaRealización: es la fecha en la que se realizó el pedido.
- DirecciónEnvío: es la dirección donde el pedido se mandará.
- Número: es el número correspondiente a la casa donde se mandará el pedido.
- CódigoUsuario: (clave foránea de la Entidad Usuario) es el código del usuario asociado a este pedido.
- Población: es la población donde se mandará el pedido.
- Provincia: es la provincia donde se mandará el pedido.
- CódigoPosta: es el código postal correspondiente a la población donde se enviará el pedido.

- País: es el país donde se enviará el pedido.
- NúmeroTarjetaCrédito: es el número de la tarjeta con la que se pagará el pedido.

- **LíneaPedido**

Es muy parecida a la línea de carrito y contiene el producto pedido por el usuario, junto con la cantidad de unidades pedidas y su precio.

Atributos:

- CódigoPedido: (clave primaria de esta Entidad y a parte clave foránea de la Entidad Pedido) es el código del pedido asignado a esta línea de pedido.
- CódigoProducto: (clave primaria de esta Entidad y a parte clave foránea de la Entidad Producto) es el código del producto asignado a esta línea de pedido.
- Cantidad: es la cantidad de unidades que el usuario ha pedido de dicho producto.
- Precio: es el precio total de la línea.

- **Producto**

Almacenará los datos referentes a un producto.

Atributos:

- CódigoProducto: (clave primaria) es el código del producto.
- Nombre: es el nombre del producto.
- Descripción: es una breve descripción del producto.
- UnidadesStock: es la cantidad de unidades que hay en stock en el almacén.
- Precio: es el precio del producto.
- CódigoFamilia: (clave foránea de la Entidad FamiliaProducto) es el código de la familia asignada a este producto.

- **Foto**

Almacenará la foto asignada a un producto.

Atributos:

- CódigoFoto: (clave primaria) es el código de la foto.
- Foto: es el nombre de la foto.
- CódigoProducto: (clave foránea de la Entidad Producto) es el código del producto asociado a esta foto.

- **Comentario**

Almacenará un comentario realizado sobre un producto.

Atributos:

- CódigoComentario: (clave primaria) es el código del comentario.
- Comentario: es el comentario realizado.
- CódigoProducto: (clave foránea de la Entidad Producto) es el código del producto asociado a este comentario.
- CódigoUsuario: (clave foránea de la Entidad Usuario) es el código del usuario asociado a este comentario.

- **FamiliaProducto**

Almacenará los datos de una familia de productos.

Atributos:

- CódigoFamilia: (clave primaria) es el código de la familia.
- NombreFamilia: es el nombre de la familia.

### 5.5.2. Diagrama Entidad-Relación

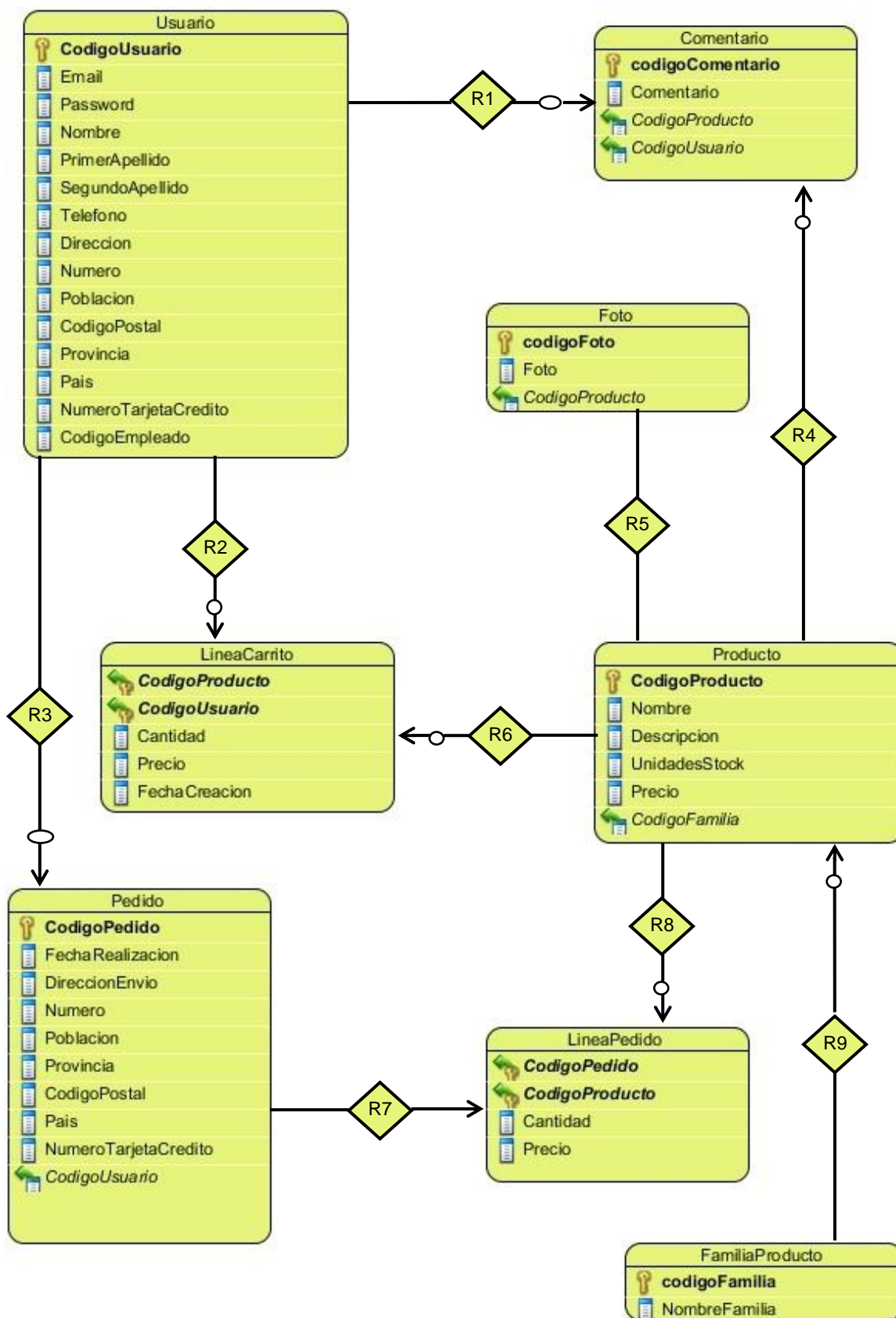


Ilustración 5.3: Diagrama Entidad-Relación

Una vez visto el diagrama entidad relación (Ilustración 5.3), vamos a especificar cada una de las relaciones:

- **R1:** Está asociado.
- **R2:** Tiene asociado.
- **R3:** Tiene asociado.
- **R4:** Está asociado.
- **R5:** Está asociado.
- **R6:** Está asociado.
- **R7:** Tiene asociado.
- **R8:** Está asociado.
- **R9:** Tiene asignado.



# CAPÍTULO 6

---

Implementación

## 6.1. Lenguajes de programación utilizados

### 6.1.1. Java



Ilustración 6.1: Java

Tanto para realizar la aplicación como para realizar los servicios web vamos a utilizar Java (Ilustración 6.1).

Java [\[12\]](#) es un lenguaje de programación de propósito general, concurrente y orientado a objetos que se diseñó para tener las mínimas dependencias de implementación posibles.

Lo que busca es permitir que los creadores de aplicaciones desarrollen el programa solo una vez y se pueda ejecutar en cualquier dispositivo sin necesidad de recompilar el código cada vez que se quiera correr en otra plataforma.

Java se ha convertido en uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva de C y C++, pero las utilidades de bajo nivel que tiene son menores que cualquiera de ellos.

El lenguaje Java se compone de cinco objetivos principales:

1. Se debería usar el paradigma de la programación orientada a objetos.
2. Debería permitir que un programa se ejecutara en múltiples sistemas operativos.
3. Debería incluir soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

### 6.1.2. JSON



Ilustración 6.2: JSON

Para realizar el paso de mensajes entre el servicio web y la aplicación vamos a utilizar el formato JSON [\[13\]](#).

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Su lectura y su escritura es simple para humanos, mientras que para las máquinas es simple su interpretación y su generación.

Se basa en un subconjunto del Lenguaje de Programación JavaScript.

JSON es un formato de texto que es totalmente independiente del lenguaje, pero utiliza convenciones que son muy conocidas por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Todas estas propiedades convierten a JSON en un lenguaje ideal para el intercambio de datos.

JSON se conforma de dos estructuras:

1. Una colección de pares de nombre/valor. En varios lenguajes se conoce como un objeto, registro, estructura, diccionario, tabla hash o lista de claves.
2. Una lista ordenada de valores. En la mayor parte de los lenguajes, esto se crea como vectores, listas o secuencias.

Un JSON puede ser representado de diferentes maneras:

- Un objeto (Ilustración 6.3) es un conjunto desordenado de pares nombre/valor. Un objeto empieza con { (llave de apertura) y termine con } (llave de cierre). Cada nombre le sigue : (dos puntos) y los pares nombre/valor se separan por , (coma).

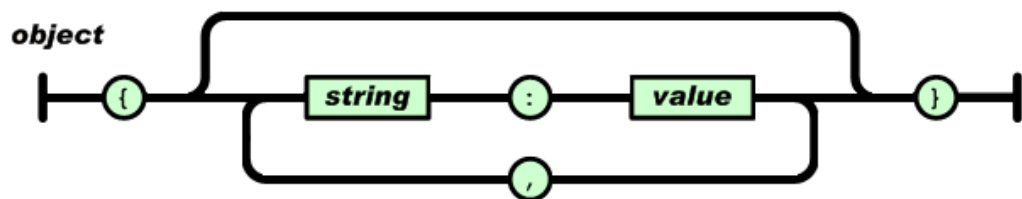


Ilustración 6.3: Objeto

- Un array (Ilustración 6.4) es una colección de valores. Un array empieza con [ (corchete izquierdo) y acaba con ] (corchete derecho). Los valores son separados por , (coma).

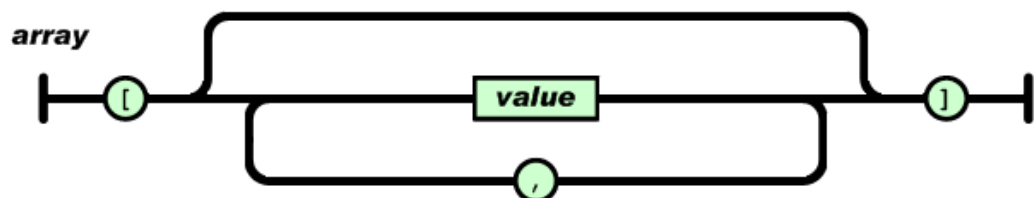


Ilustración 6.4: Array

## 6.2. Herramientas utilizadas para el desarrollo

### 6.2.1. Android studio



Ilustración 6.5: Android Studio

Android studio [15] (Ilustración 6.5) va a ser la herramienta que utilizaremos para crear nuestra aplicación.

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y Eclipse fue reemplazado como el IDE oficial para el desarrollo de aplicaciones para Android.

Se basa en el software IntelliJ IDEA de JetBrains y fue publicado de forma gratuita utilizando la Licencia Apache 2.0. Se encuentra disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido creado específicamente para el desarrollo de Android.

Actualmente android studio nos ofrece las siguientes características:[14][15][16]

1. Integración de ProGuard y funciones de firma de aplicaciones.
2. Renderizado en tiempo real.
3. Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
4. Un sistema de compilación basado en Gradle flexible.
5. Refactorización específica de Android y arreglos rápidos.
6. Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario
7. Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.

8. Plantillas para crear diseños comunes de Android y otros componentes.
9. Soporte para programar aplicaciones para Android Wear.
10. Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
11. Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.
12. Un emulador rápido con varias funciones.
13. Compatibilidad con C++ y NDK.
14. Gran cantidad de herramientas y frameworks de prueba.

### 6.2.2. Netbeans



**Ilustración 6.6: Netbeans**

Para crear el servicio web vamos a utilizar la herramienta NetBeans [\[17\]](#) (Ilustración 6.6) usando programación Java.

NetBeans es un entorno de desarrollo libre, creado principalmente para el lenguaje de programación Java. Se puede extender con numerosos módulos. NetBeans IDE2 es un producto libre y gratuito que no tiene restricciones de uso.

NetBeans es un proyecto de código abierto que tiene muchísimo éxito con una gran base de usuarios y una comunidad en crecimiento constante.

El proyecto de código abierto NetBeans fue creado por Sun Microsystems en junio de 2000 y sigue siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de módulos. Un módulo es un archivo Java que se compone de clases java que son escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans IDE soporta la creación de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características encontramos un sistema de proyectos basado en Ant, control de versiones y refactoring.

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans se conforma de todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario empezar a trabajar inmediatamente.

La plataforma nos aporta servicios reusables comunes para las aplicaciones de escritorio, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. Algunas de las características de la aplicación son:

1. Gestión de la interfaz de usuario.
2. Gestión de configuración de usuario.
3. Gestión de almacenamiento.
4. Gestión de ventana.
5. Marco Asistente.
6. Librería visual de Netbeans.
7. Herramientas de desarrollo integrado

NetBeans IDE es libre, código abierto, multiplataforma con soporte integrado para el lenguaje de programación Java.

### 6.2.3. Phpmyadmin



Ilustración 6.7: MySQL

phpMyAdmin [19] (Ilustración 6.7) es una herramienta que fue escrita en PHP con la intención de manipular la administración de MySQL a través de páginas web, usando Internet.

Soporta una amplia gama de operaciones, siendo efectuadas de modo gráfico a través de una interfaz gráfica. Las más recientes que realiza son el manejo de bases datos, tablas, campos, relaciones, índices, usuarios, permisos, etc. También proporciona la capacidad de ejecutar directamente sentencias SQL. [18]

phpMyAdmin tiene las siguientes características:

1. Interfaz sobre web intuitiva.
2. Proporciona herramientas de gestión de la base de datos: edición, creación, modificación y eliminación de bases de datos, tablas, vistas, campos, relaciones e índices.
3. Mantenimiento de usuarios y sus privilegios.
4. Mantenimiento de procedimientos almacenados.
5. Importación de datos desde CSV y SQL.
6. Exportación a varios formatos: CSV, SQL, XML, PDF, SO/IEC 26300 - OpenDocument Text y Spreadsheet, Word, LATEX y otros.
7. Administración de múltiples servidores.
8. Creación del despliegue de la base de datos en un gráfico exportado a PDF.
9. Creación de consultas complejas haciendo uso QBE (Query By Example).



El software, hoy en día se encuentra disponible en 72 idiomas, y se está mantenido por el proyecto phpMyAdmin bajo responsabilidad de Olivier Müller, Marc Delisle, Alexander M. Turek, Michal Čihař y Garvin Hicking.

También soporta el motor MariaDB y a partir de la versión 3.5 es soportado el motor Drizzle.

## 6.3. Librerías auxiliares utilizadas

### 6.3.1. Hibernate



Ilustración 6.8: Hibernate

Cuando comenzamos a estudiar Java escuchamos nombrar este Framework (Ilustración 6.8) y en en muchas ocasiones nos preguntamos para qué sirve o por qué utilizarlo.

“Hibernate [20] es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (disponible también para .Net usando el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.”

En otras palabras, Hibernate es un Framework que acelera la relación entre la aplicación y la base de datos. Para poder aprender a utilizarlo es necesario contar con los conocimientos básicos de base de datos y SQL así como manejar el lenguaje Java.

A la hora de desarrollar aplicaciones en muchos casos todo acaba siendo un conjunto de ABM (alta, baja y modificaciones de datos) que luego consultamos.

Para ello se utiliza una base de datos donde hay muchas tareas repetidas: por cada objeto que quiero persistir debo crear una clase que me permita insertarlo, eliminarlo, modificarlo y consultarlo. Con excepción de consultas especiales, el resto es siempre lo mismo.

Este es el momento dónde un ORM coge una importancia fundamental. Con solo configurarlo todas estas tareas se ejecutan automáticamente y solo tendrás que preocuparte por las consultas especiales.

El desarrollador deberá configurar en un archivo XML o mediante annotations donde corresponde un atributo de una clase, con una columna de una tabla. Es una tarea simple donde existen herramientas que lo hacen por nosotros

# CAPÍTULO 7

---

Instalación y despliegue del  
servidor web

## 7.1. Introducción

En este capítulo explicaremos como instalar los componentes necesarios para poder ejecutar el servicio Web en el servidor con todas sus funcionalidades.

Para todo esto necesitaremos instalar una serie de tecnologías y herramientas para que nuestro servicio Web funcione correctamente.

Estas son:

- Apache2
- Java
- MySQL-Server
- phpMyAdmin
- Nginx
- Apache Tomcat 7

Hay que tener ciertas consideraciones previas especificadas en anteriores capítulos, como por ejemplo que el sistema operativo de nuestro servidor será el Ubuntu.

Todo lo necesario a instalar será instalado a través del terminal o por descarga directa e instalación en su lugar correspondiente.

Antes de comenzar a explicarlo todo, hay que dejar claro porqué se cambian los puertos por defecto de Apache2 y de Tomcat. Bien, se debe a que tanto Apache2 como Tomcat trabajan por defecto en el puerto 80 así como Nginx, por lo que para que todo funcione bien, se cambian los puertos de Apache2 y Tomcat a 81 y 82 respectivamente y se deja Nginx en el 80 para redireccionar los puertos (de Apache2 y Tomcat).

## 7.2. Instalación y configuración de Apache2

La instalación se llevará a cabo con el terminal de Ubuntu. Para ello primeramente ejecutaremos el terminal y una vez que este esté abierto escribiremos y ejecutaremos el siguiente comando:

```
sudo apt-get install apache2
```

Cuando se vaya a instalar te solicitará una contraseña que es la del usuario administrador.

Ahora vamos a configurar Apache2 para que no funcione en el puerto 80 sino en el 81 y así dejar Tomcat en el puerto 82. Todo esto toma forma después con la configuración de Nginx.

Para esto tenemos que editar dos ficheros: "/etc/apache2/ports.conf" y "/etc/apache2/sites-available/default".

Para el primer fichero ejecutaremos el siguiente comando:

```
sudo gedit /etc/apache2/ports.conf
```

En este fichero modificaremos las siguientes líneas:

```
NameVirtualHost *:80      a      NameVirtualHost *:81  
  
Listen 80                      a      Listen 81
```

Y para el otro ejecutaremos el siguiente comando:

```
sudo gedit /etc/apache2/sites-available/default
```

En este fichero modificaremos las siguientes líneas:

```
<VirtualHost *:80>              a      <VirtualHost *:81>
```

### 7.3. Instalación de Java

Para llevar a cabo la instalación de Java tendremos que ejecutar las siguientes órdenes en el terminal:

```
sudo apt-get install python-software-properties  
  
sudo add-apt-repository ppa:webupd8team/java  
  
sudo apt-get update  
  
sudo apt-get install oracle-java7-installer
```

### 7.4. Instalación de MySQL-Server

Para realizar esta instalación solo necesitaremos ejecutar una sola orden:

```
sudo apt-get install mysql-server
```

### 7.5. Instalación y configuración de phpMyAdmin

Para empezar, hay que ejecutar el siguiente comando:

```
sudo apt-get install phpmyadmin
```

Nos pedirá que donde queremos instalarlo y le diremos que Apache2. Después de la instalación tenemos que ejecutar lo siguiente:

```
sudo ln -s /usr/share/phpmyadmin/ /var/www/phpmyadmin
```

Ahora tenemos que reiniciar apache2 para que cambie el puerto correctamente al 81 y podamos acceder a phpMyadmin:

```
sudo /etc/init.d/apache2 restart
```

## 7.6. Importación de las tablas de la base de datos

En este apartado explicaremos como importar las tablas de nuestra base de datos para dejarla en funcionamiento.

Una vez que tenemos todos los archivos .sql, para importarlos tenemos que crear una base de datos nueva, le ponemos de nombre "datos" y excogemos de cotejamiento "utf8\_general\_ci". Con esto abremo creado la base de datos.

Ahora, para importar los ficheros .sql seleccionamos nuestra base de datos recién creada y después nos vamos a la pestaña "Importar".

Una vez hecho todo esto tenemos que pinchar en el botón "Examinar..." y buscamos los ficheros .sql y pulsamos en el botón "Continuar", para importarlos de uno en uno.

Así habremos creado todas las tablas con todas sus relaciones pertinentes.

## 7.7. Instalación y configuración de Apache Tomcat 7

Para la instalación de Tomcat 7 necesitaremos ejecutar la siguiente orden en el terminal:

```
sudo apt-get install tomcat7
```

Para configurarlo necesitaremos seguir una serie de pasos:

- Editamos el siguiente fichero con la siguiente orden:

```
sudo gedit /etc/tomcat7/server.xml
```

- Ahora tenemos que modificar la siguiente línea:

```
<Connector port="8080" protocol="HTTP/1.1"
```

```
    a
```

```
<Connector port="8082" protocol="HTTP/1.1"
```

- También tendremos que editar este fichero:

```
sudo gedit /etc/tomcat7/tomcat-users.xml
```

- para poder entrar en el panel de administración de Tomcat tenemos que añadir al final del fichero las siguientes líneas:

```
<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<user username="tomcat" password="tomcat"

roles="manager-gui, admin-gui"/>

</tomcat-users>
```

Nótese que el usuario y la contraseña son a gusto de cada administrador por lo que se pueden cambiar.

Con esta configuración ya podemos entrar en el panel de administración de Tomcat y cuando nos pida el usuario y la contraseña la introducimos.

Una vez que nos ha mostrado el panel principal, bajamos para ver las demás funcionalidades.

Ahora añadiremos nuestro servicio Web al servidor Tomcat para que pueda estar operativo. Para ello buscaremos un botón que diga "Examinar..." en la sección de "Archivo WAR a desplegar". Buscamos nuestro fichero .war de nuestro servicio web dentro del CD proporcionado y le damos al botón "Desplegar" que está debajo del botón de examinar que utilizamos anteriormente.



## 7.8. Instalación y configuración de Nginx

Cuando tengamos instalados y configurados Apache y Tomcat, pasaremos a instalar y configurar Nginx para poder hacer accesible Tomcat desde Internet para que el cliente Android se conecte con el servicio Web.

Para instalar Nginx debemos de ejecutar la siguiente orden en el terminal:

```
sudo apt-get install nginx
```

Para configurar Nginx correctamente hay que seguir los siguientes pasos:

- Tenemos que modificar el siguiente fichero con la siguiente orden:

```
sudo gedit /etc/nginx/nginx.conf
```

- Nginx no permite manejar ficheros muy grandes en peticiones. Esto es un problema porque el .war de nuestro servicio Web es más grande y a la hora de desplegarlo nos dará un error. Para solucionarlo, debemos modificar la siguiente línea:

```
client_max_body_size 2M;  
  
a  
  
client_max_body_size 200M; // La cantidad que necesites
```

- Ahora hay que pasar a configurar las redirecciones de los puertos en direcciones de acceso por rutas en la URL:

```
sudo gedit /etc/nginx/sites-available/default
```

- Hay que insertar las siguientes líneas dentro de "server {":

```
Listen:80; // Si no está en 80, cambiarlo a 80

Client_max_body_size 50M; // Aumentarlo según se necesite

// Apache accesible desde /localhost/a/

location /a/ {

    proxy_pass http://localhost:81/;

}

// Tomcat accesible desde /localhost/t/

location /t/ {

    proxy_pass http://localhost:8082/;

}
```

Para acabar tendremos que reiniciar Nginx para que todos los cambios se apliquen:

```
sudo /etc/init.d/nginx restart
```

# ANEXO 1

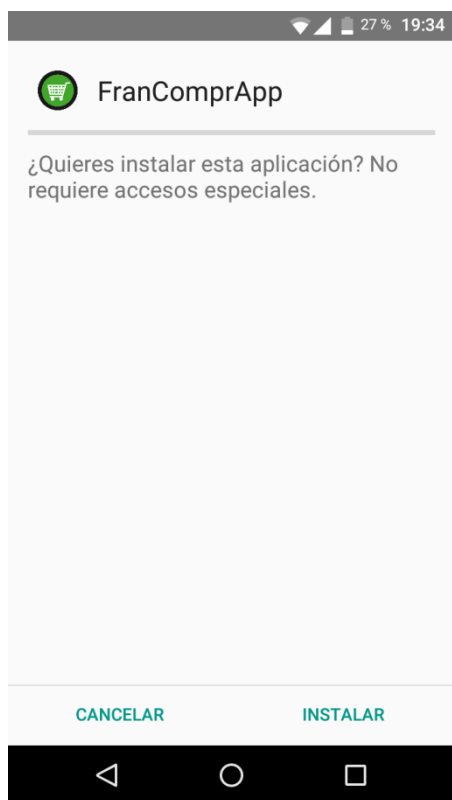
---

Manual de Instalación de la  
aplicación móvil

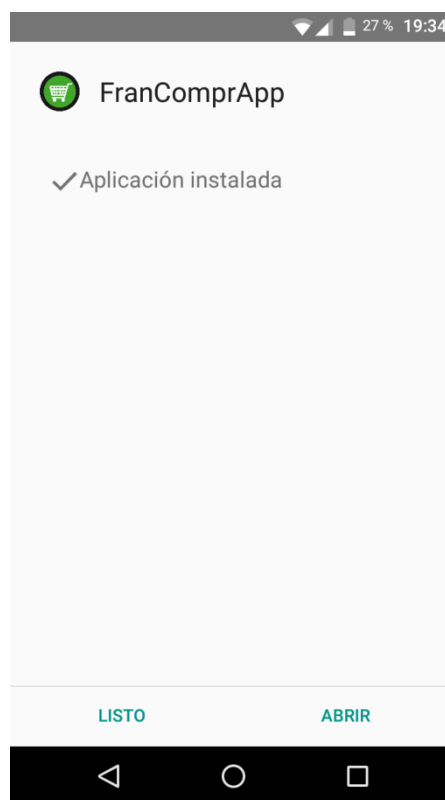
El proceso para la instalación de la aplicación en un dispositivo móvil cualquiera es muy sencillo y no supondrá problema alguno.

Primeramente, debemos de buscar el .apk de nuestra aplicación (franComprApp.apk), que se encuentra en el interior del CD entregado. Una vez encontrado, lo copiamos en el dispositivo móvil en el que se desee instalar.

Una vez hecho esto, con el gestor de archivos que tengamos instalado en el dispositivo, buscamos el archivo recién copiado y lo ejecutamos.



**Ilustración Anexo 1.1: Instalar aplicación**



**Ilustración Anexo 1.2: Aplicación instalada**

Al ejecutar nos aparecerá una pantalla igual a la de la Ilustración Anexo 1.1, en la que tendremos que darle al botón “INSTALAR”. Esperamos hasta que se instale y una vez instalada nos aparecerá una pantalla igual que la de la Ilustración Anexo 1.2, en la cual pulsaremos el botón “abrir”.

Cuando pulsamos “ABRIR”, la aplicación nos pedirá que habilitemos los permisos para acceder a fotos, contenido multimedia y archivos de nuestro dispositivo (Ilustración Anexo 1.3)



Ilustración Anexo 1.3: Habilitar permisos

Hay que darle a “PERMITIR”, para que la aplicación funcione correctamente.

Por último, una vez que le damos a “PERMITIR”, ya se nos abrirá la aplicación y podremos pasar a usarla (Ilustración Anexo 1.4).

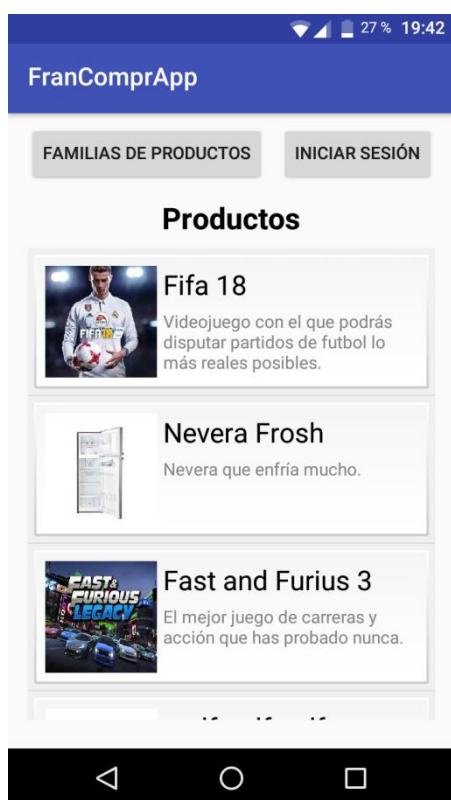


Ilustración Anexo 1.4: Página principal



# ANEXO 2

---

Manual de Usuario

Esta guía de usuario la hago para ayudar a entender al usuario como debe interactuar con la aplicación Android.

En nuestra aplicación dispondremos de diferentes maneras de interacción en forma de:

- Botones
- Campos de texto
- Listas
- Menús de opciones

### **Botones**

Podremos encontrar botones en toda la aplicación Android para poder llevar a cabo las diferentes funcionalidades de las que se compone.

Los botones que podemos encontrar en la aplicación tendrán este aspecto:



Ilustración Anexo 2.1: Botones

### **Campos de texto**

Los campos de texto son una parte muy importante de la aplicación, ya que nos sirven para recoger todo tipo de información, ya sea de inicio de sesión, como diferentes datos. De esta manera podremos llevar a cabo todas las funcionalidades que proporciona la aplicación pasándoles los datos necesarios.



Los campos de texto de la aplicación se encontraran con la siguiente apariencia:

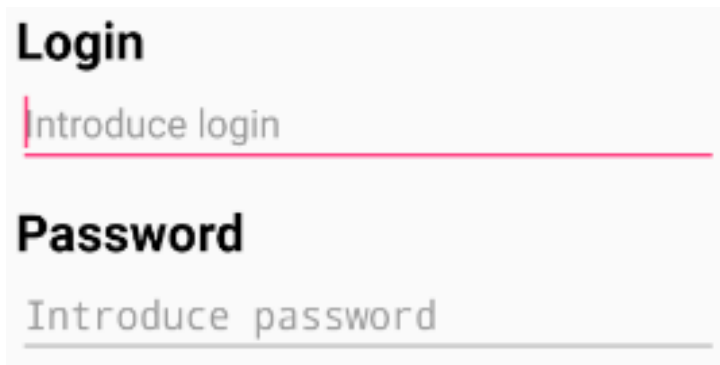


Ilustración Anexo 2.2: Campos de texto

Los campos de texto son los que se encuentran con la letra en gris.

## Listas

Las listas son un elemento muy útil que nos van a ayudar a mostrar los diferentes datos como pueden ser listas de productos, de líneas de carrito/pedido, listas de familias o listas de comentarios entre otras.

Algunas listas de nuestra aplicación tendrán el siguiente aspecto:

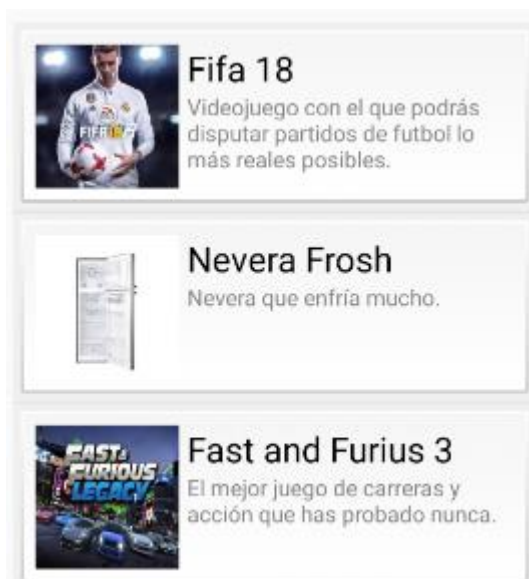


Ilustración Anexo 2.3: Lista de productos

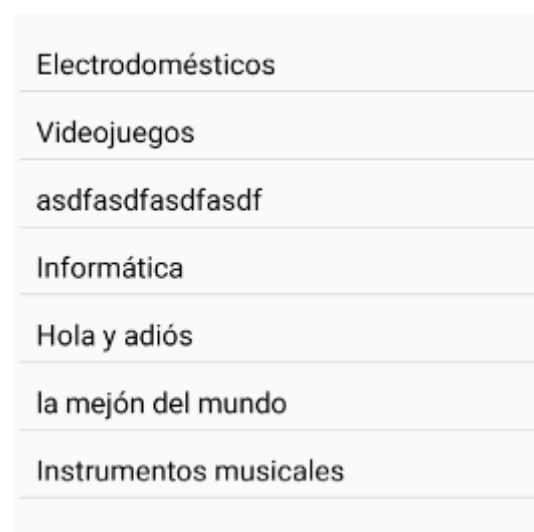


Ilustración Anexo 2.4: Lista de familias

Las listas mostradas en las ilustraciones Anexo 2.3 y Anexo 2.4 son algunas de las diferentes listas que encontrará en la aplicación.

### **Menús de opciones**

Estos menús nos dejarán elegir diferentes opciones dentro de una misma vista. Estos menús los encontraremos de la siguiente manera:



**Ilustración Anexo 2.5: Menús de opciones**

# ANEXO 3

---

Storyboards

El storyboard es una herramienta muy útil ya que nos permite mostrar con imágenes como funciona la interfaz de la aplicación.

Muestran las diferentes pantallas a seguir para realizar una acción. El paso de una pantalla a otra se hará mediante números. Cada pantalla tendrá asignado un número, por lo que, por ejemplo, en un botón, si aparece el número 5, quiere decir que al pulsar ese botón la aplicación se va a la pantalla número 5. Si aparecen 2 números, quiere decir, que dependiendo de lo que ocurra puede ir a una pantalla u a otra.

Los storyboards que se realizarán en función de las funcionalidades que hemos creído más convenientes para mostrar son los siguientes:

- **Storyboard de consultar producto, familias, iniciar sesión y registrarse**

Conforme arrancamos la aplicación se nos muestra la vista 1. Si seleccionamos un producto de la lista, la aplicación nos llevará hasta la vista 9 y estando en esta vista si pulsamos el iniciar sesión, se nos llevará hasta la vista 2.

Volviendo a la vista principal, si pulsamos en el botón “Familias de Productos”, se nos llevará hasta la vista 6. Estando en esta, si pulsamos una familia que no tiene productos para mostrar, iremos hasta la vista 7. De lo contrario, si pulsamos una familia que tiene productos para mostrar, nos llevará hasta la vista 8. Estando en esta vista 8, si pulsamos en algún producto de la lista nos llevará hasta la vista 9, donde se mostrará la información del producto.

Volviendo a la vista principal, si pulsamos en el botón “Iniciar Sesión”, nos iremos hasta la vista 2. Desde aquí, al pulsar en iniciar sesión, nos llevará o a la vista 3 o la 10, dependiendo de si se han introducido datos o si los datos introducidos son correctos.

Ahora si le damos a registrar, nos llevará hasta la vista 4 y dependiendo si no se rellenan todos los datos o si los datos son correctos, nos iremos hasta la vista 5 o 10.

- **Storyboard de modificar datos de registro y cerrar sesión**

Nos encontramos en la vista principal, una vez iniciada sesión. Para poder desplegar el menú tendremos que pulsar las barricas de arriba a la izquierda. Una vez pulsadas se nos llevará a la vista 2.

Al seleccionar “Modificar datos de registro” se nos redirigirá hasta la vista 3 y dependiendo de si se han introducido todos los datos o si los datos son correctos, nos llevará a la vista 4 o 2 respectivamente.

En la vista 2, si pulsamos en “Cerrar Sesión”, nos llevará hasta la vista 5.

- **Storyboard de añadir producto al carrito, modificarlo y eliminarlo**

Para añadir un producto al carrito hay que consultarlo primero pulsando en una la fila de la lista deseada y nos llevará hasta la vista 2. Ahora pulsamos en añadir al carrito y nos llevará hasta la vista 3 y por último si al pulsar “Añadir al carrito” no se ha introducido ningún valor nos llevará a la vista 4, de lo contrario nos devolverá a 2 pero con la línea de carrito ya creada.

Ahora para modificarla o eliminarla, hay que consultar el carrito pulsando, en la vista 5, “Consultar carrito”, esto nos llevará a la vista 6, si no hay líneas de carrito creadas o a la 7 si las hay.

Si pulsamos en la línea deseada se nos llevará a la vista 8 y se nos dará a elegir entre modificarla o eliminarla. Si pulsamos “Eliminarla” se nos llevará hasta la vista 7 y si pulsamos “Modificarla” se nos llevará a la 9.

Una vez en la vista 9, al pulsar el “Modificar línea”, si no se ha introducido ninguna cantidad se nos llevará hasta la vista 10, de lo contrario nos devolvería a la 7.

- **Storyboard de crear pedido y anularlo**

Una vez consultado el carrito, pulsamos en el botón “Crear pedido” y nos llevará a la vista 2. En esta vista cuando pulsamos en el botón “Realiza pedido” se nos redigira a la vista 3 o 4, dependiendo si no se han introducido todos los dato o si se ha creado el pedido con éxito, respectivamente.

Ahora para anularlo hay que consultar los pedidos para después seleccionar el pedido deseado. Partimos de la vista 5. En esta vista al pulsar el “Consultar pedidos” se nos llevará a la vista 7 o 6 dependiendo de si el usuario ha realizado algún pedido o no, respectivamente.

Una vez en la vista 7, hay que seleccionar el pedido deseado para anularlo y esto nos lleva a la vista 8. Una vez allí, al pulsar en “Anular pedido” se nos llevará a la vista 9, donde nos preguntarán si queremos anularlo. Si le damos a que “No” se nos llevará a la vista 8 y si le damos a que “Sí” a la 9.

- **Storyboard de crear, modificar, eliminar y añadir unidades a un producto**

Primeramente hay que pulsar en “Gestiones de empleado” y esto nos llevará a la vista 2. Aquí, si queremos crear un producto, pulsamos dicho botón y nos llevará a la vista 3. Dentro de esta, al pulsar en el botón “Crear Producto” nos puede llevar a la vista 2, si se ha creado con éxito, o a la vista 4, si no se han introducido todos los datos.

Para modificar un producto, en la vista 2 pulsamos dicho botón y nos llevará a la vista 5 y una vez allí seleccionamos un producto de la lista y nos llevará a la vista 6. En esta vista, al pulsar en el botón “Modificar producto” nos llevará a la vista 5 si se ha modificado correctamente, o a la 7 si no se han rellenado todos los datos.

Para añadir undidades a un producto, en la vista 2 pulsamos dicho botón y nos llevará a la vista 8, allí seleccionamos el producto deseado y nos llevará a la vista 9. En esta, cuando pulsamos en el botón “Añadir unidades” nos podrá llevar a la vista 8, si se ha introducido una cantidad correcta o a la 10 si no se ha introducido ninguna cantidad.

Por último para eliminar un producto, en la vista 2 pulsaremos dicho botón y nos llevará a la vista 11, en esta seleccionamos el producto deseado y nos llevará a la vista 12, donde se nos dará a elegir si queremos eliminarlo. Si pulsamos que no, nos devolverá a la vista 11 y si pulsamos que si, también nos llevará a la vista 11, pero en esta ya no aparecerá el producto eliminado.

- **Storyboard de crear, modificar y eliminar familias**

Partiendo de la vista de gestiones de empleado anterior, vista 1, para crear una familia pulsaremos dicho botón y nos llevará a la vista 2 donde se introducirá el nombre de la familia y al pulsar el botón “Crear familia”, nos llevará a la vista 1, si se ha creado con éxito, o a la vista 3, si no se ha introducido ningún nombre.

Para modificarla, en la vista 1 pulsaremos dicho botón y nos llevará a la vista 4, donde seleccionaremos una familia del listado y nos llevará a la vista 5, en la cual al pulsar en el botón “Modifiacar familia” nos llevará a la vista 4, si se ha modificado correctamente, o a la vista 6, si no se ha introducido ningún nombre.

Para eliminarla, en la vista 1 pulsaremos dicho botón y nos llevará a la vista 7, donde si seleccionamos una familia que tiene asignados productos, nos llevará a la vista 9 y si seleccionamos una familia vacía, nos llevará a la vista 8.

- **Storyboard de crear, modificar y elmimananr comentarios**

Para esa funcionalidad partimos de la vista de un producto, esta va a ser la vista 1. Si pulsamos el botón “Añadir”, si no se ha introducido ningún comentario, nos llevará a la vista 2, de lo contrario, si se ha rellenado este se añadirá a la lista de arriba.

Si pulsamos en uno de los comentarios realizados por el usuario, nos llevará a la vista 3 y si pulsamos en “Eliminarlo” volvemos a la vista 1 y si pulsamos en “Modificarlo” nos llevará a la vista 4. Si al modificar no se introduce ningún comentario, nos llevará a la vista 5, de lo contrario nos llevará de vuelta a la 1.

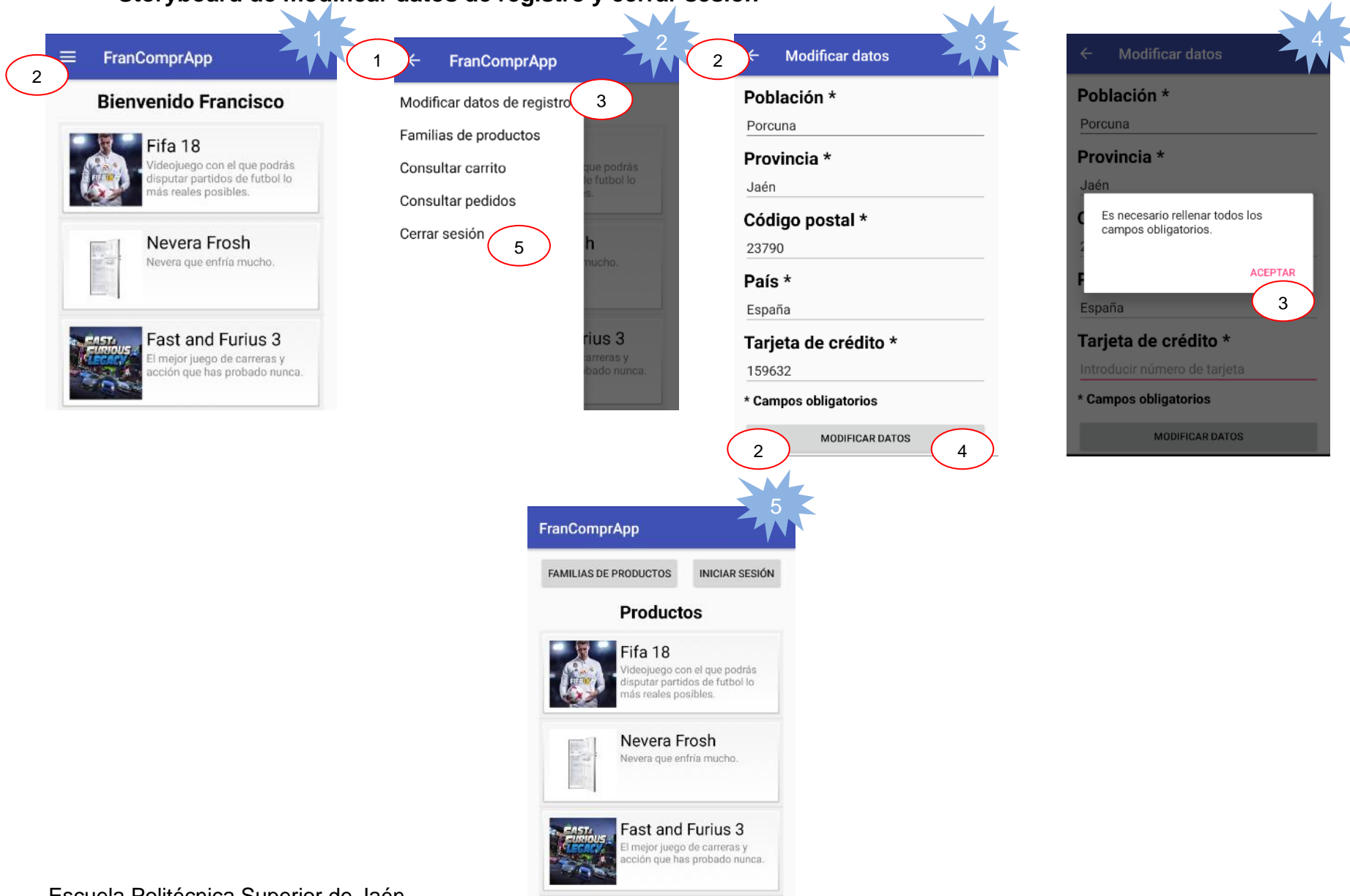
### Storyboard de consultar producto, familias, iniciar sesión y registrarse

The storyboard consists of the following screens and their key elements:

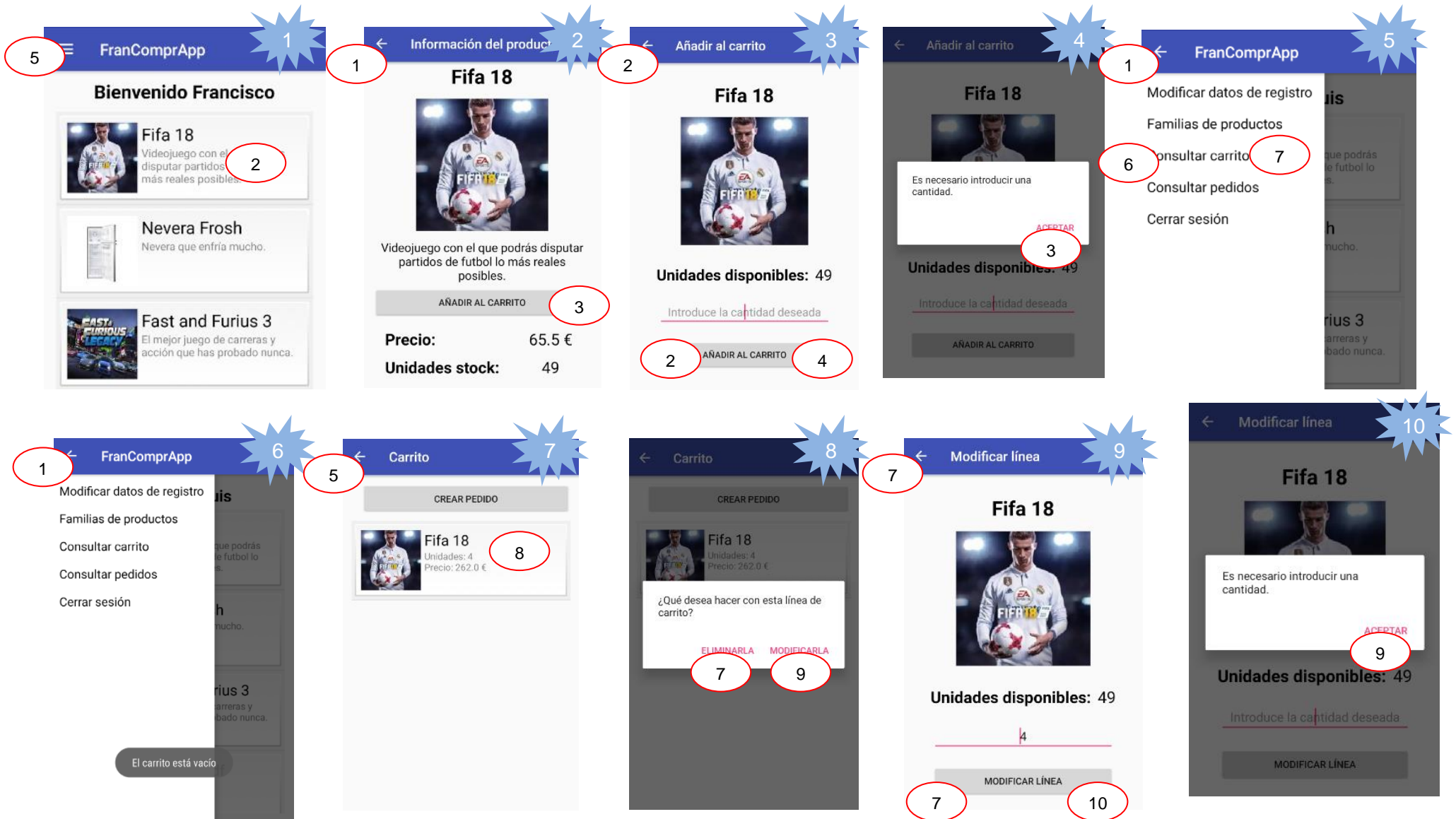
- Home Screen:** Shows navigation options: FAMILIAS DE PRODUCTOS (6), INICIAR SESIÓN (2), and a list of products including **Fifa 18** (9), **Nevera Frosh**, and **Fast and Furious 3**.
- Acceso (Access) Screen:** Contains Login and Password fields, and buttons for REGISTRARSE (4), INICIAR SESIÓN (3), and ACEPTAR (2).
- Registro (Registration) Screen:** Includes fields for Provincia \*, Código postal \*, País \*, Tarjeta de crédito \*, and Código de empleado, with a REGISTRARSE button (5) and a validation message (4).
- Familias (Families) Screen 1:** A list of product families: Electrodomésticos, Videojuegos (8), Lectura, Informática, Móviles, Hogar (7), and Instrumentos musicales.
- Familias (Families) Screen 2:** Shows a message: "La familia seleccionada no tiene asignado ningún producto" (1).
- Videojuegos (Video Games) Screen:** Displays a list of video games: **Fifa 18** (9) and **Fast and Furious 3**.
- Información del producto (Product Information) Screen:** Shows details for **Fifa 18**, including a price of 65.5 € and 49 units in stock. It includes a button for INICIAR SESIÓN (2) and a note that users must log in to purchase (8).
- Bienvenido Francisco (Welcome Francisco) Screen:** Shows a personalized welcome message and a list of recommended products: **Fifa 18**, **Nevera Frosh**, and **Fast and Furious 3**.



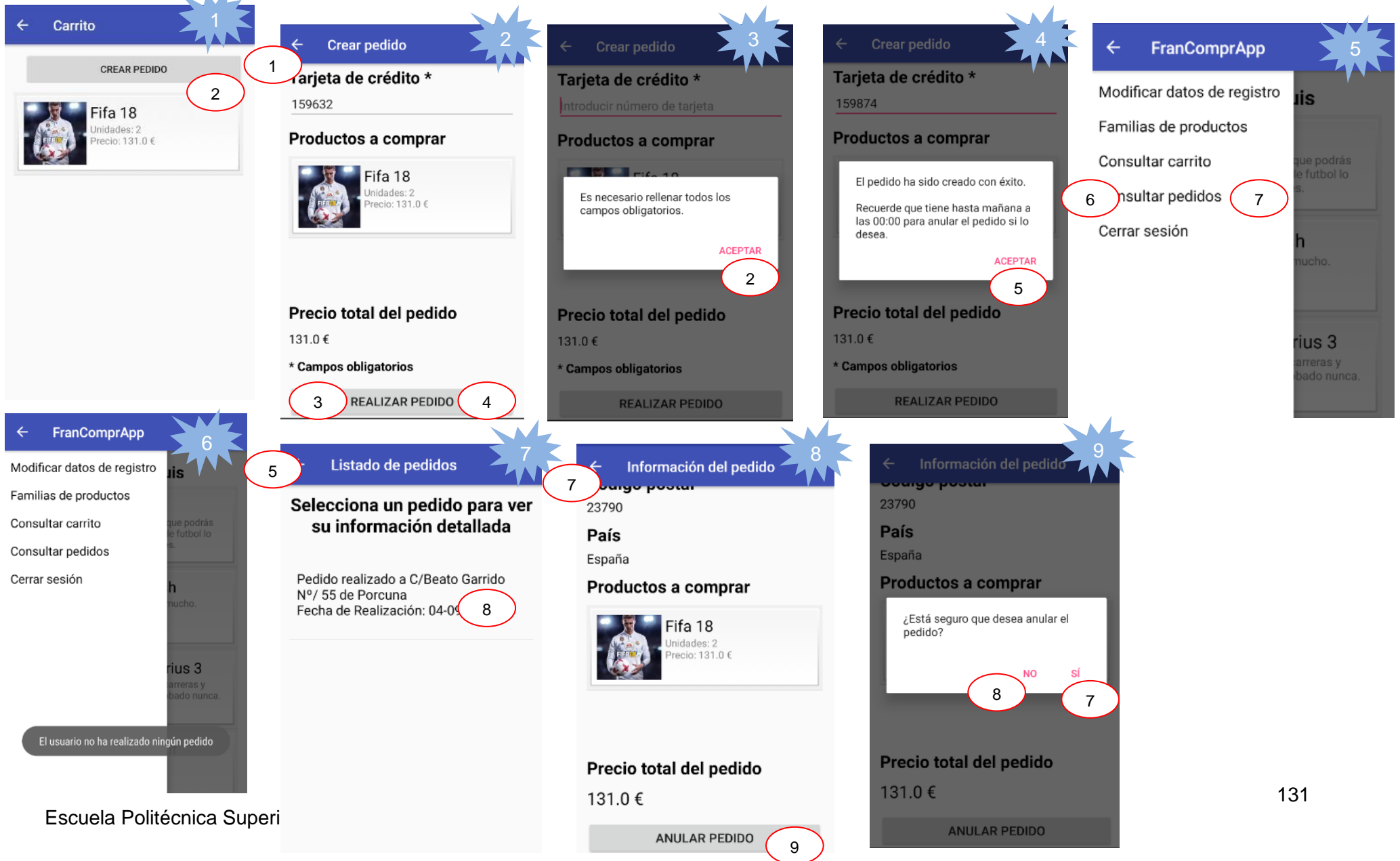
### Storyboard de modificar datos de registro y cerrar sesión



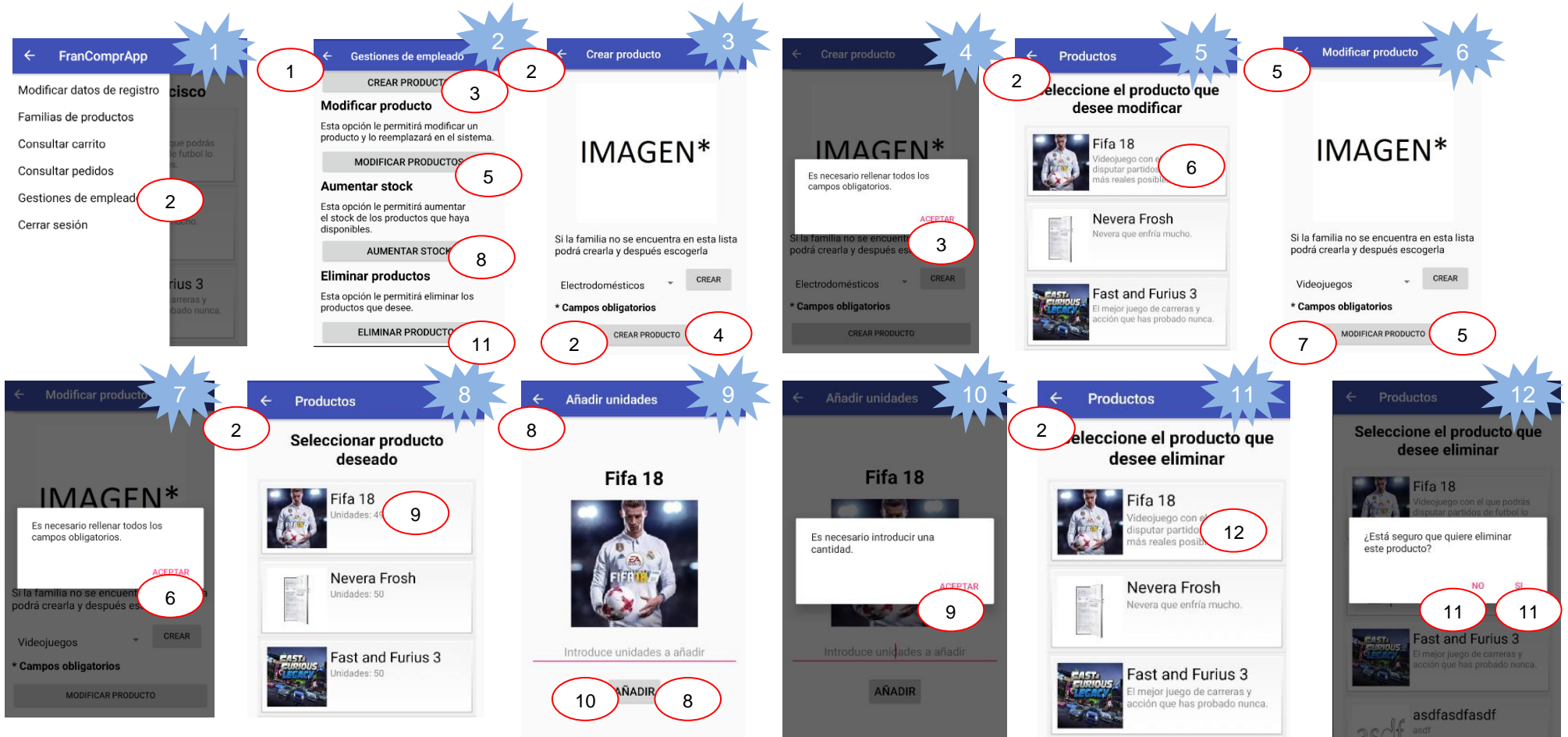
### Storyboard de añadir producto al carrito, modificarlo y eliminarlo



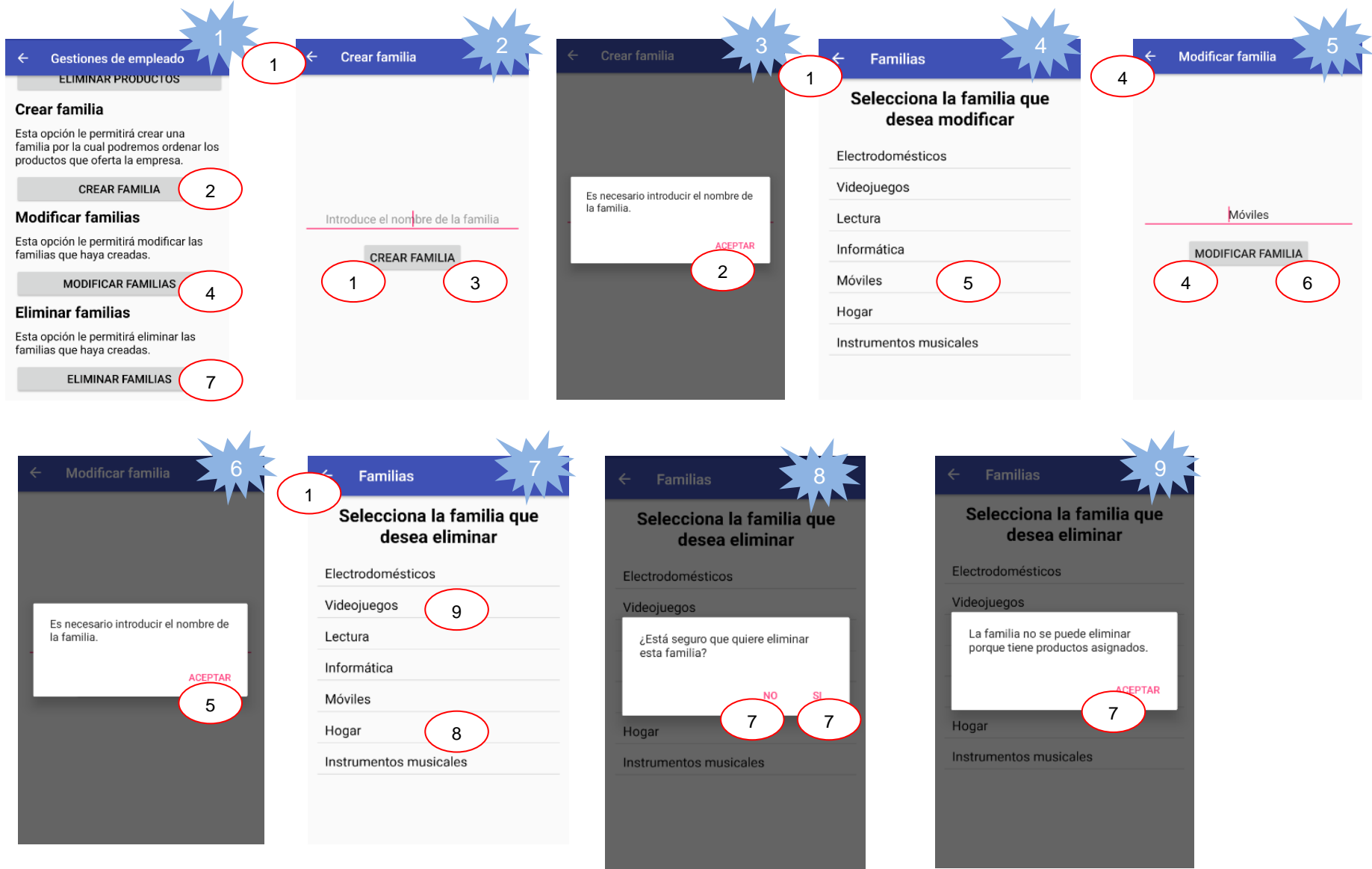
### Storyboard de crear pedido y anularlo



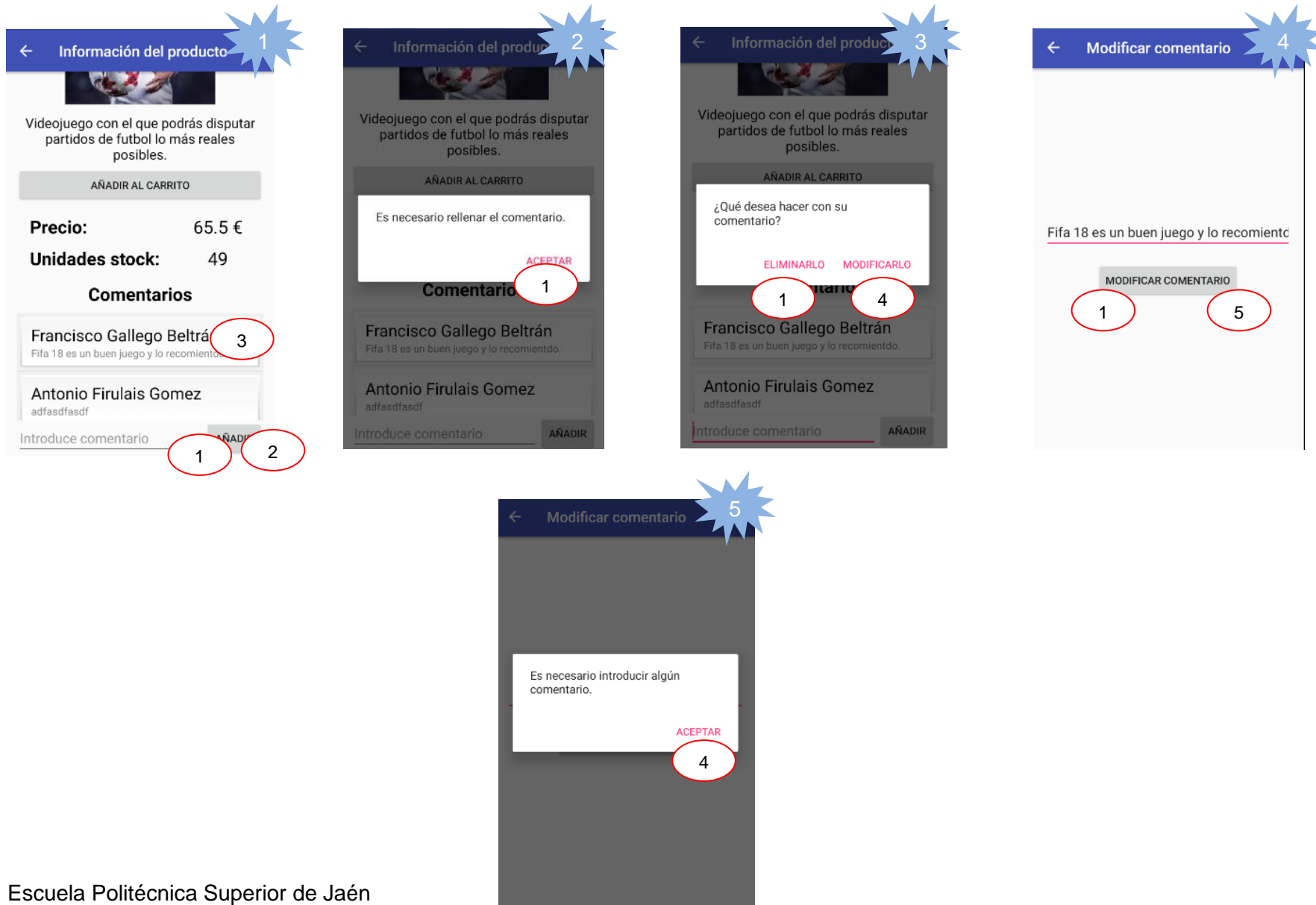
### Storyboard de crear, modificar, eliminar y añadir unidades a un producto



### Storyboard de crear, modificar y eliminar familias



### Storyboard de crear, modificar y eliminar comentarios.



## Bibliografía

**Motivación** [Fecha de última consulta: 04/09/2017]

[1] <http://kelevra.es/aumenta-la-demanda-de-apps-moviles-empresariales/>

**Servicio Web y qué puedo hacer con él** [Fecha de última consulta: 04/09/2017]

[2] <https://www.ibm.com/developerworks/ssa/webservices/newto/service.html>

**Ventajas y Desventajas de los Servicios Web** [Fecha de última consulta: 04/09/2017]

[3] [https://es.wikipedia.org/wiki/Servicio\\_web#Ventajas\\_de\\_los\\_servicios\\_web](https://es.wikipedia.org/wiki/Servicio_web#Ventajas_de_los_servicios_web)

**Qué es MySQL y aplicaciones** [Fecha de última consulta: 04/09/2017]

[4] <https://es.wikipedia.org/wiki/MySQL#Aplicaciones>

**Ventajas y Desventajas de MySQL** [Fecha de última consulta: 04/09/2017]

[5] <https://mysqldaniel.wordpress.com/ventajas-y-desventajas/>

**Por qué usar MySQL** [Fecha de última consulta: 04/09/2017]

[6] <http://indira-informatica.blogspot.com.es/2007/09/qu-es-mysql.html>

**Tomcat** [Fecha de última consulta: 04/09/2017]

[7] <https://es.wikipedia.org/wiki/Tomcat>

**Tipos de tecnologías para realizar servicios Web** [Fecha de última consulta: 04/09/2017]

[8] <http://www.processonline.com.co/blog/servicios-web-y-sus-tecnologias-asociadas/>

**Android** [Fecha de última consulta: 04/09/2017]

[9] <https://es.wikipedia.org/wiki/Android#Caracter.C3.ADsticas>

[10] <http://www.configurarequipo.com/doc1107.html>

**Por qué usar Android** [Fecha de última consulta: 04/09/2017]

[11] <http://esdroids.com/ventajas-de-android-para-el-desarrollo-de-aplicaciones-moviles/>

**Java** [Fecha de última consulta: 04/09/2017]

[12] [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

**JSON** [Fecha de última consulta: 04/09/2017]

[13] <http://www.json.org/json-es.html>

**Android Studio** [Fecha de última consulta: 04/09/2017]

[14] <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>

[15] [https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)

[16] <https://developer.android.com/studio/intro/index.html?hl=es-419>

**Netbeans** [Fecha de última consulta: 04/09/2017]

[17] <https://es.wikipedia.org/wiki/NetBeans>

**PhpMyAdmin** [Fecha de última consulta: 04/09/2017]

[18] <https://es.opensuse.org/PhpMyAdmin>

[19] <https://es.wikipedia.org/wiki/PhpMyAdmin>

**Hibernate** [Fecha de última consulta: 04/09/2017]

[20] <http://blog.educacionit.com/2013/02/07/que-es-java-hibernate/>