

GRADO EN INGENIERÍA INFORMÁTICA DE  
GESTIÓN Y SISTEMAS DE INFORMACIÓN

# TRABAJO FIN DE GRADO

*DISEÑO E IMPLEMENTACIÓN DE UNA  
APLICACIÓN MÓVIL PARA LA  
GESTIÓN DE PEDIDOS*

*MEMORIA*

**Alumno:** Fernández Jiménez, Alexander

**Director:** Villamañe Gironés, Mikel

**Curso:** 2017 - 2018

**Fecha:** 18 de junio de 2018

## Resumen

Este documento corresponde a la memoria del Proyecto de Fin de Carrera de Alex Fernández, desarrollado para la titulación de Grado de Ingeniería Informática de Gestión y Sistemas de Información en la Escuela de Ingeniería de Bilbao de la UPV/EHU.

Este Proyecto de Fin de Carrera tiene el objetivo de facilitar las gestiones de los pedidos para cualquier empresa de paquetería, mensajería, etc. Para ello, se ha realizado el diseño e implementación de una aplicación para dispositivos Android.

Para el desarrollo se utilizó el programa Android Studio. En cuanto al almacenamiento de datos, se ha utilizado una base de datos MySQL remota.

Dokumentu hau Alex Fernández proiektuaren memoria da, UPV/EHU Bilboko Ingenieritza Eskola Kudeaketaren eta Informazio Sistemen Informatikaren Ingenieritza Gradua garatu dena.

Proiektu honen helburua paketeen eta mezularitza enpresentzako aginduak kudeatzea erraztea da. Horretarako, Android gailuetarako aplikazio bat diseinatu eta ezartu egin da.

Garapenarako Android Studio programa erabili zen. Datuen biltegitzeari dagokionez, MySQL datu basea erabili da.

This document belongs to the Alex Fernández Final Degree Project memory, developed to the Degree in Computer Engineering Management and Information Systems in the Faculty of Engineering in Bilbao.

This Final Degree Project has the objective to make easier the order management for any courier company. For it, has been done the design and implementation of an application for Android devices.

For the development, was used the Android Studio program. As for the data storage, has been used the MySQL remote database.



## ÍNDICE GENERAL

Resumen.....	1
Índice de figuras .....	5
Índice de tablas .....	7
1- Introducción .....	9
2- Planteamiento inicial .....	11
2.1- Objetivos .....	11
2.2- Alcance .....	12
2.2.1- Listado de tareas .....	12
2.2.2- Estructura de descomposición del trabajo .....	14
2.2.3- Descripción de tareas.....	17
2.3- Planificación temporal .....	29
2.3.1- Tareas del proyecto.....	29
2.4- Herramientas .....	32
2.5- Gestión de riesgos.....	34
2.6- Evaluación económica.....	36
2.6.1- Gastos.....	36
2.6.2- Inversiones .....	37
2.7- Arquitectura .....	38
3- Antecedentes .....	39
4- Captura de requisitos.....	43
4.1- Modelo de casos de uso.....	43
4.3- Modelo de dominio.....	49
5- Análisis y diseño .....	51
6- Desarrollo.....	59
7- Verificación y evaluación .....	69
7.1- Modificar idioma.....	69
7.2- Calificar pedido .....	70
7.3- Identificarse .....	71
7.4- Recuperar contraseña.....	71
7.5- Registrarse como director.....	72
7.6- Cerrar sesión .....	73
7.7- Modificar datos personales .....	74
7.8- Registrar trabajador .....	75
7.9- Ver sus trabajadores .....	75
7.10- Modificar datos de sus trabajadores .....	76

7.11-. Eliminar trabajadores.....	76
7.12-. Ver estadísticas de pedidos .....	77
7.13-. Crear vehículos.....	77
7.14-. Ver vehículos.....	78
7.15-. Modificar vehículos.....	78
7.16-. Eliminar vehículos .....	79
7.17-. Almacenar objetos .....	79
7.18-. Ver el almacén .....	80
7.19-. Editar objetos.....	80
7.20-. Eliminar objetos .....	81
7.21-. Crear pedidos .....	82
7.22-. Ver pedidos .....	83
7.23-. Editar pedidos .....	84
7.24-. Eliminar pedidos .....	85
7.25-. Observar pedidos asignados .....	85
7.26-. Iniciar pedido .....	86
7.27-. Mostrar ruta pedido.....	86
7.28-. Finalizar pedido.....	87
8-. Conclusiones y trabajo futuro .....	89
8.1-. Conclusiones del proyecto .....	89
8.2-. Conclusiones personales.....	91
8.3-. Trabajo futuro .....	92
9-. Bibliografía .....	95
10-. Anexo 1 – Casos de uso extendidos.....	97
11-. Anexo 2 – Diagramas de secuencia.....	139

## Índice de figuras

Ilustración 1: Primera parte diagrama EDT .....	14
Ilustración 2: Segunda parte diagrama EDT .....	15
Ilustración 3: Tercera parte diagrama ED.....	16
Ilustración 4: Logo Android Studio .....	32
Ilustración 5: Logo WinSCP.....	32
Ilustración 6: Logo Notepad++ .....	32
Ilustración 7: Logo Visual Paradigm .....	33
Ilustración 8: Logo Dropbox .....	33
Ilustración 9: Logo Microsoft Office .....	33
Ilustración 10: Arquitectura de la aplicación .....	38
Ilustración 11: Actividades de los usuarios .....	40
Ilustración 12: Compras online .....	41
Ilustración 13: Casos de uso del cliente .....	43
Ilustración 14: Casos de uso del trabajador .....	44
Ilustración 15: Casos de uso del director .....	46
Ilustración 16: Modelo de dominio .....	49
Ilustración 17: Diagrama de clases Primer prototipo.....	52
Ilustración 18: Diagrama de clases Segundo prototipo .....	52
Ilustración 19: Diagrama de clases Tercer prototipo .....	53
Ilustración 20: Diagrama de clases subprototipo Gestionar personal .....	54
Ilustración 21: Diagrama de clases subprototipo Gestionar almacén .....	55
Ilustración 22: Diagrama de clases subprototipo Gestionar vehículos.....	56
Ilustración 23: Diagrama de clases subprototipo Pedidos asignados .....	56
Ilustración 24: Clase relación con la BD .....	57
Ilustración 25: Actividades que representan los diálogos.....	57
Ilustración 26: Llamada a una actividad con parámetros.....	59
Ilustración 27: Recogida de datos en una llamada .....	59
Ilustración 28: Fragments.....	60
Ilustración 29: Diálogos .....	60
Ilustración 30: Llamadas a diálogos y toast.....	61
Ilustración 31: Toast .....	61
Ilustración 32: Aplicación detenida .....	61
Ilustración 33: Código constructora conexión BD .....	62
Ilustración 34: Llamada a clase para realizar conexión BD .....	62
Ilustración 35: Acción al pulsar un botón.....	62
Ilustración 36: Diseño inicial y diseño final .....	63
Ilustración 37: Uso de listados .....	64
Ilustración 38: Uso de tablas .....	64
Ilustración 39: Archivo strings.xml .....	65
Ilustración 40: Código cambiar idioma.....	65
Ilustración 41: Insertar una ubicación.....	66
Ilustración 42: Realizar petición a Google Maps .....	66
Ilustración 43: Rutas Google Maps .....	67
Ilustración 44: Comparación estimaciones .....	90
Ilustración 45: Diagrama de secuencia Modificar idioma .....	139
Ilustración 46: Diagrama de secuencia Calificar pedido .....	140

Ilustración 47: Diagrama de secuencia Identificarse .....	141
Ilustración 48: Diagrama de secuencia Recuperar contraseña .....	142
Ilustración 49: Diagrama de secuencia Registrarse como director.....	143
Ilustración 50: Diagrama de secuencia Cerrar sesión .....	144
Ilustración 51: Diagrama de secuencia Modificar datos personales.....	144
Ilustración 52: Diagrama de secuencia Registrar trabajador .....	146
Ilustración 53: Diagrama de secuencia Ver sus trabajadores .....	147
Ilustración 54: Diagrama de secuencia Modificar datos de sus trabajadores .....	148
Ilustración 55: Diagrama de secuencia Eliminar trabajadores.....	148
Ilustración 56: Diagrama de secuencia Ver estadísticas de pedidos.....	149
Ilustración 57: Diagrama de secuencia Crear vehículos.....	151
Ilustración 58: Diagrama de secuencia Ver vehículos .....	152
Ilustración 59: Diagrama de secuencia Modificar vehículos.....	153
Ilustración 60: Diagrama de secuencia Eliminar vehículos .....	153
Ilustración 61: Diagrama de secuencia Almacenar objetos .....	154
Ilustración 62: Diagrama de secuencia Ver el almacén.....	155
Ilustración 63: Diagrama de secuencia Editar objetos .....	156
Ilustración 64: Diagrama de secuencia Eliminar objetos .....	157
Ilustración 65: Diagrama de secuencia Crear pedidos .....	157
Ilustración 66: Diagrama de secuencia Ver pedidos .....	158
Ilustración 67: Diagrama de secuencia Editar pedidos - Parte 1.....	160
Ilustración 68: Diagrama de secuencia Editar pedidos - Parte 2.....	161
Ilustración 69: Diagrama de secuencia Eliminar pedidos.....	162
Ilustración 70: Diagrama de secuencia Observar pedidos asignados .....	162
Ilustración 71: Diagrama de secuencia Iniciar pedido .....	163
Ilustración 72: Diagrama de secuencia Mostrar ruta pedido.....	164
Ilustración 73: Diagrama de secuencia Finalizar pedido .....	165

## Índice de tablas

Tabla 1: Tiempo de las tareas.....	29
Tabla 2: Primera parte diagrama Gantt .....	30
Tabla 3: Segunda parte diagrama Gantt .....	31
Tabla 4: Probabilidad de los riesgos.....	34
Tabla 5: Impacto de los riesgos .....	34
Tabla 6: Gastos del proyecto.....	36
Tabla 7: Pruebas Modificar idioma .....	69
Tabla 8: Pruebas Calificar pedido .....	70
Tabla 9: Pruebas Identificarse .....	71
Tabla 10: Pruebas Recuperar contraseña .....	71
Tabla 11: Pruebas Registrarse como director .....	72
Tabla 12: Pruebas Cerrar sesión.....	73
Tabla 13: Pruebas Modificar datos personales .....	74
Tabla 14: Pruebas Registrar trabajador.....	75
Tabla 15: Pruebas Ver sus trabajadores.....	75
Tabla 16: Pruebas Modificar datos de sus trabajadores .....	76
Tabla 17: Pruebas Eliminar trabajadores .....	76
Tabla 18: Pruebas Ver estadísticas de pedidos.....	77
Tabla 19: Pruebas Crear vehículos .....	77
Tabla 20: Pruebas Ver vehículos .....	78
Tabla 21: Pruebas Modificar vehículos .....	78
Tabla 22: Pruebas Eliminar vehículos.....	79
Tabla 23: Pruebas Almacenar objetos.....	79
Tabla 24: Pruebas Ver el almacén .....	80
Tabla 25: Pruebas Editar objetos .....	80
Tabla 26: Pruebas Eliminar objetos.....	81
Tabla 27: Pruebas Crear pedidos .....	82
Tabla 28: Pruebas Ver pedidos.....	83
Tabla 29: Pruebas Editar pedidos.....	84
Tabla 30: Pruebas Eliminar pedidos .....	85
Tabla 31: Pruebas Observar pedidos asignados.....	85
Tabla 32: Pruebas Iniciar pedido .....	86
Tabla 33: Pruebas Mostrar ruta pedido .....	86
Tabla 34: Pruebas Finalizar pedido .....	87



## 1-. Introducción

El proyecto trata de crear una aplicación móvil que permita a las empresas de mercancías, paquetería, etc. gestionar sus envíos, su almacén, sus trabajadores y sus vehículos. Es decir, el director de la empresa crea y asigna los pedidos a sus trabajadores.

Estos trabajadores ven los pedidos que tienen asignados con los detalles correspondientes como la hora de salida, el destino, el contenido de la carga, etc. Además, la aplicación mediante Google Maps les ayuda indicando la ruta a seguir. También, se puede verificar la entrega del pedido, el estado del pedido y la eficiencia de los trabajadores. Para obtener estos detalles, los destinatarios valoran el pedido describiendo el estado del pedido, la puntualidad, etc.

Por otro lado, la aplicación permite gestionar el almacén de la empresa. El encargado almacena objetos en la base de datos, desde la cual puede crear los pedidos a entregar.

Además, el director puede llevar un control de todos sus trabajadores y observar los datos personales y las estadísticas que tienen a la hora de entregar los pedidos.

Finalmente, el encargado puede observar toda su flota de vehículos para controlar el estado de estos. Esto es, por cada vehículo se le muestra la última ITV realizada, la fecha en torno a la que se debe realizar la próxima ITV, número de km recorridos, fecha de obtención del vehículo, etc. y a la hora de realizar el pedido puede asignar el vehículo al trabajador en función del destino o de la carga. De esta manera, puede crear o eliminar vehículos de la aplicación en los casos de compra o venta de estos.

Para realizar todas las funciones que puede realizar la aplicación, cada usuario debe identificarse con su correspondiente nivel de usuario (director o trabajador). Respecto a los destinatarios, no hará falta una identificación, bastando únicamente el código del envío, el nombre de usuario del trabajador y la valoración correspondiente.

Esta idea surge tras haber realizado charlas con consultores que dan soporte a herramientas dirigidas a la gestión de empresas y que desarrollan mejoras en este tipo de herramientas. Además, esta aplicación permitiría un mayor control de la empresa para toda la gestión relacionada con los pedidos.

Respecto a trabajos realizados en esta área, existen aplicaciones para la gestión de almacenes, es decir, aplicaciones para la gestión de recursos de almacenes y donde se encuentran sus objetos. De esta manera, dichas aplicaciones ya existentes junto con el proyecto a realizar permitiría a las empresas de pedidos una nueva opción para sus gestiones.



## 2-. Planteamiento inicial

En este apartado se plantean los objetivos que se desea cumplir. Además, se listan y describen todas las tareas que se realizan durante el proyecto. Estas tareas se pueden observar en el apartado del alcance.

A continuación, se expone una planificación temporal para estimar el tiempo que llevará la elaboración de este proyecto. De tal manera, también se evalúa el gasto que conlleva el desarrollo y cómo se rentabilizará.

Por otro lado, se describe con qué herramientas se implementará la aplicación y cómo se ha redactado la memoria. Además, se detallará la arquitectura de la aplicación para ir conociendo cómo será.

Por último, se indican qué riesgos pueden encontrarse durante el desarrollo de este proyecto y cómo se pueden solventar.

### 2.1- Objetivos

El objetivo principal de este proyecto es desarrollar la aplicación anteriormente descrita. En base a este objetivo, se listan los objetivos que se deben cumplir:

- El objetivo principal de este proyecto es diseñar e implementar una aplicación que permita la gestión de los pedidos con una interfaz sencilla. Es decir, el desarrollo de la aplicación se centrará en el correcto funcionamiento de las funcionalidades dejando en segundo plano el diseño de las interfaces. Una vez finalizada la aplicación, se tomará en cuenta el tiempo restante para mejorar el diseño si es posible. Este objetivo se comprobará teniendo en cuenta si al finalizar ha sido posible mejorar el diseño de las interfaces.
- En segundo lugar, mi objetivo es cumplir y desarrollar todos los requisitos según mi estimación temporal, para así tener tiempo para plantear posibles mejoras una vez finalizado el proyecto.
- Aumentar mi conocimiento a la hora de desarrollar aplicaciones para dispositivos Android.
- Por último, aumentar mi experiencia a la hora de desarrollar una correcta organización y un buen desarrollo, para futuros proyectos. Esto se debe al ser un proyecto de larga duración, individual y complejo.

Una vez descritos todos los objetivos, al final del proyecto se comprueban cuantos se han cumplido, y en caso de que alguno no se haya cumplido, se expone el motivo.

## 2.2.- Alcance

En este apartado se muestran todas las tareas que se van a llevar a cabo. Finalmente, se describe cada una de ellas para comprender mejor su labor.

### 2.2.1.- Listado de tareas

Como se puede observar, el proyecto se divide en 6 principales fases.

1. Organización
  - a. Elaborar un horario semanal a seguir para realizar el proyecto
  - b. Elegir donde almacenar la documentación
  - c. Elegir donde almacenar el código
2. Realizar documentación previa
  - a. Realizar los objetivos
  - b. Realizar el alcance
  - c. Realizar la planificación temporal
  - d. Realizar el análisis de las herramientas a utilizar
  - e. Realizar la gestión de riesgos
  - f. Realizar la evaluación económica
  - g. Realizar el estado del arte
3. Realizar análisis y diseño de la aplicación
  - a. Realizar la captura de requisitos
  - b. Realizar un boceto de las posibles pantallas de la aplicación
  - c. Realizar el plan de pruebas de las pantallas
  - d. Realizar el diagrama de clases, casos de uso y modelo de dominio
  - e. Realizar la base de datos
    - i. Listar los datos que se deben almacenar
    - ii. Implementar la base de datos
    - iii. Probarla para asegurar su correcto funcionamiento
4. Iniciar la implementación
  - a. Diseñar las pantallas y realizar las funcionalidades correspondientes
  - b. Implementar las clases necesarias para el correcto funcionamiento de las pantallas
5. Realizar las pruebas de código y de conexión entre la BD y la aplicación
6. Finalizar la documentación
  - a. Actualizar versiones
  - b. Realizar las conclusiones y propuestas para el trabajo futuro
  - c. Unir la documentación
  - d. Preparar la presentación

La primera fase “Organización”, se corresponde a crear un horario y así poder establecer cuantas horas al día u horas semanales se dedicarán al proyecto. Además, se elige donde se almacenará tanto el código como la memoria.

La siguiente fase denominada “Realizar documentación previa” tendrá diversas tareas. Esta fase se centra en realizar toda la planificación previa. De esta manera, se afronta el proyecto conociendo todas las tareas a realizar, el tiempo necesario, los posibles riesgos y como solucionarlos o minimizarlos, etc.

La tercera fase, “Realizar análisis y diseño de la aplicación”, trata de elaborar todos los detalles que tiene nuestra aplicación. Esta fase es la más duradera debido a que tiene como objetivo realizar todo el diseño que tiene la aplicación, que permite realizar, etc. Una vez esté todo correctamente elaborado, la siguiente fase se realiza de forma más sencilla.

En esta fase, “Iniciar la implementación”, se elabora todo el código necesario para que funcione correctamente la aplicación. Tanto la anterior fase como esta están relacionadas. Si se elabora correctamente y claramente la anterior, no habrá muchos problemas en esta fase.

Se debe indicar que la fase “Iniciar la implementación” y “Realizar las pruebas de código y de conexión entre la BD y la aplicación” se realizan por prototipos. Esto es, se divide la implementación en prototipos, exactamente en cuatro.

Los prototipos a realizar son los siguientes. En primer lugar, el diseño e implementación de las funcionalidades que permitan al usuario la identificación y el registro en la aplicación. Además, se tiene en cuenta la funcionalidad de recuperar la contraseña si al usuario se le ha olvidado.

El segundo prototipo a realizar es el que contenga las funcionalidades de calificar el pedido una vez entregado y la de cerrar sesión. El siguiente prototipo contiene las funcionalidades de cambiar el idioma de la aplicación, modificar los datos personales del usuario identificado y de registrar trabajadores, si el usuario identificado es un director.

Por último, el cuarto prototipo contiene todas las funcionalidades relacionadas con la gestión de los trabajadores, de los vehículos, de los pedidos creados y de los pedidos asignados a los trabajadores. Cada gestión contiene las funcionalidades de creación, edición y eliminación de cualquier dato relacionado con la gestión correspondiente. Al ser un prototipo que contiene muchas funcionalidades, se ha dividido en cuatro subprototipos siendo cada uno de ellos un tipo de gestión.

Por cada prototipo se empieza diseñando las interfaces y a continuación, la implementación de las clases necesarias. Una vez se han realizado estos dos pasos, se prueba el prototipo. En caso de encontrar errores se vuelve al prototipo realizando la parte del diseño o implementación según el error. Pero en caso contrario, se da por finalizado y se continúa con el siguiente. Se ha decidido realizar de esta manera la aplicación para tener funcionalidades correctamente probadas y tener parte de la aplicación.

Por último, en la fase “Finalizar la documentación”, se recaba toda la documentación para realizar las conclusiones debido a que la aplicación estaría finalizada. Además, se prepara la presentación para la posterior defensa del proyecto.

## 2.2.2-. Estructura de descomposición del trabajo

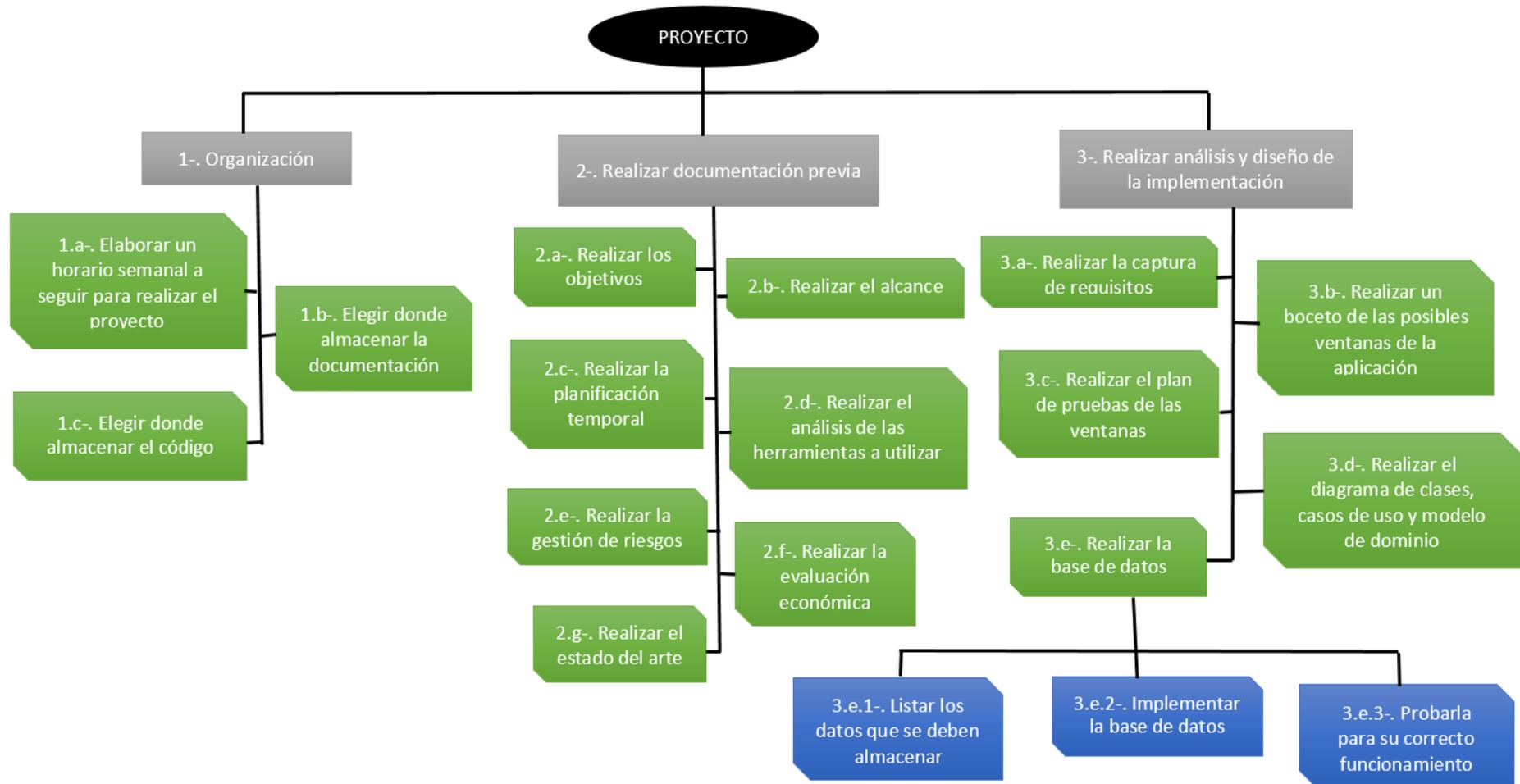


Ilustración 1: Primera parte diagrama EDT

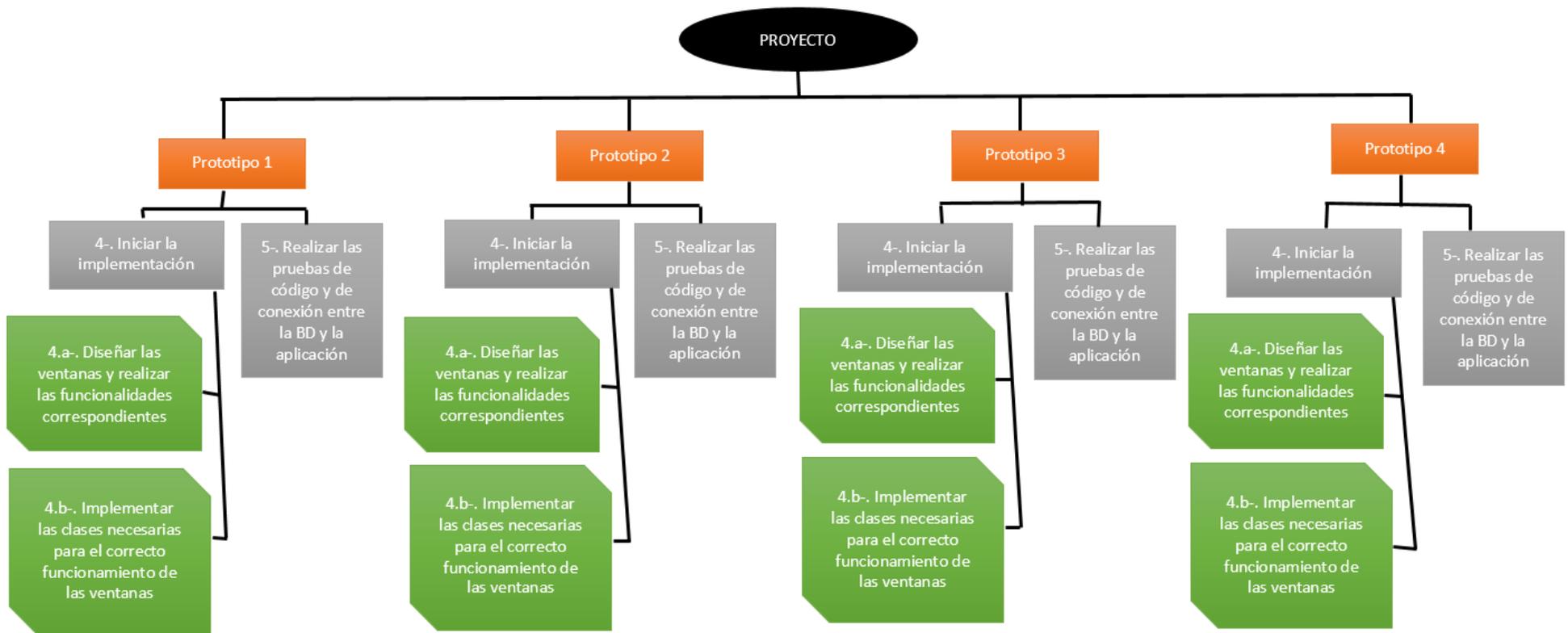
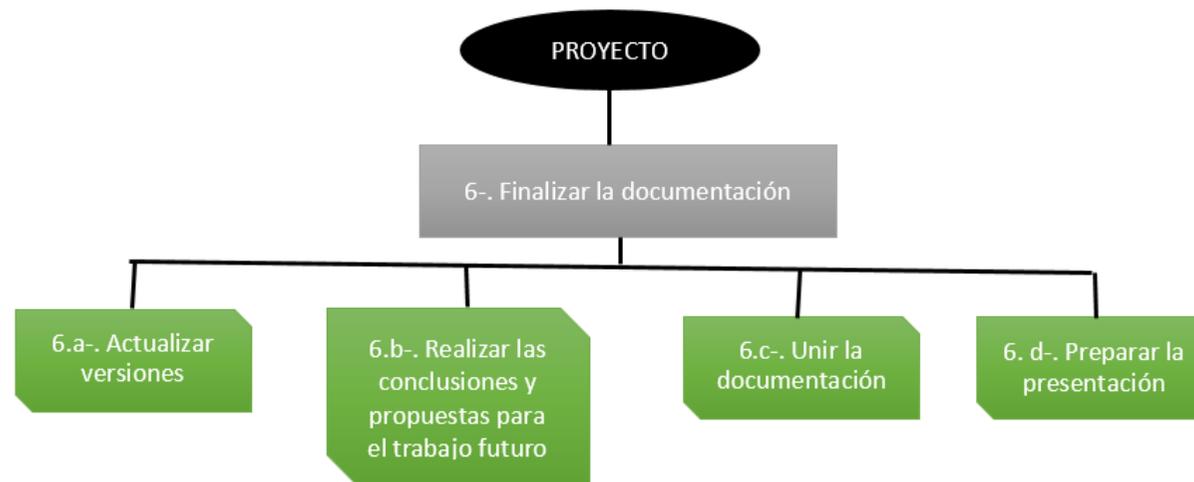


Ilustración 2: Segunda parte diagrama EDT



*Ilustración 3: Tercera parte diagrama ED*

### 2.2.3-. Descripción de tareas

En este apartado se describen todas las tareas que se llevarán a cabo.

1.a-. Elaborar un horario semanal a seguir para realizar el proyecto
<b>Paquete de trabajo:</b> Organización
<b>Duración:</b> 15 minutos
<b>Descripción:</b> Esta tarea consiste en realizar un calendario semanal en el que se indican cuantas horas al día u horas semanales se invertirán en la realización del proyecto.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Horario a seguir
<b>Recursos necesarios:</b> Ninguno
<b>Precedencias:</b> Ninguna

1.b-. Elegir donde almacenar la documentación
<b>Paquete de trabajo:</b> Organización
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en buscar en internet las distintas posibilidades para almacenar la documentación y elegir la más conveniente de acuerdo al tipo de datos que se van a almacenar.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento que justifica la elección de la herramienta para almacenar la documentación.
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Elaborar un horario semanal a seguir para realizar el proyecto

1.c-. Elegir donde almacenar el código
<b>Paquete de trabajo:</b> Organización
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en realizar una búsqueda en torno a gestores para almacenar el código. Una vez se tenga una lista de posibles gestores, se elige uno en el cual se almacenará el código.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento que justifica la elección de la herramienta para almacenar el código.
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Elegir donde almacenar la documentación

2.a-. Realizar los objetivos
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 3.5 horas
<b>Descripción:</b> Esta tarea consiste en listar los objetivos personales. Además, se detallarán los objetivos generales del proyecto.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con todos los objetivos del proyecto.
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Elegir donde almacenar el código

2.b-. Realizar el alcance
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 4 horas
<b>Descripción:</b> Esta tarea trata de listar y describir todas las tareas que se llevarán a cabo en el proyecto. Además de la descripción, se asignan diferentes especificaciones en cada tarea.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento en el que se han descrito las tareas.
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar los objetivos

2.c-. Realizar la planificación temporal
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en realizar un esquema de las tareas del proyecto para tener un seguimiento de los avances del proyecto. Además, se indican fechas para conocer cómo se encontraba el proyecto en un determinado momento.
<b>Entradas:</b> Alcance
<b>Salidas:</b> Documento en el que se esquematizan las tareas
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar el alcance

2.d-. Realizar el análisis de las herramientas a utilizar
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 5 horas
<b>Descripción:</b> Esta tarea consiste en realizar un análisis y comparación entre las herramientas de almacenaje de documentación y de código para obtener la óptima según nuestras necesidades.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento en el que se indica donde almacenar el proyecto y las herramientas a utilizar
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar la planificación temporal

2.e-. Realizar la gestión de riesgos
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 2.5 horas
<b>Descripción:</b> Esta tarea consiste en realizar un listado de todos los posibles riesgos que pueden suceder en el proyecto. Por cada riesgo se mostrarán que pasos se deben realizar para minimizarlo y en caso de que suceda, como solucionarlo
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con todos los riesgos posibles del proyecto
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar el análisis de las herramientas a utilizar

2.f. Realizar la evaluación económica
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en calcular el presupuesto necesario para llevar a cabo el proyecto. Para ello, se tendrán en cuenta las horas necesarias para realizar cada tarea.
<b>Entradas:</b> Documento de la planificación temporal
<b>Salidas:</b> Documento con el presupuesto del proyecto
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar la gestión de riesgos

2.g. Realizar el estado del arte
<b>Paquete de trabajo:</b> Realizar documentación previa
<b>Duración:</b> 3 horas
<b>Descripción:</b> Esta tarea consiste en buscar aplicaciones similares en el mercado para saber si es rentable la creación de nuestra aplicación. Además, se investiga que funcionalidades han sido implementadas con el fin de tener una idea.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con las aplicaciones similares y sus funcionalidades
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar la evaluación económica

3.a. Realizar la captura de requisitos
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 3 horas
<b>Descripción:</b> Esta tarea consiste en listar y describir las funcionalidades que tendrá la aplicación.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con las funcionalidades a realizar
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar el estado del arte

3.b. Realizar un boceto de las posibles pantallas de la aplicación
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 4 horas
<b>Descripción:</b> Esta tarea consiste en realizar un boceto de las pantallas a implementar para tener una idea de cómo serán.
<b>Entradas:</b> Documento con las funcionalidades a realizar
<b>Salidas:</b> Documento con el posible diseño de las pantallas
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar la captura de requisitos

3.c-. Realizar el plan de pruebas de las pantallas
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 6 horas
<b>Descripción:</b> Esta tarea consiste en indicar que pruebas se realizarán sobre las pantallas
<b>Entradas:</b> Documento con el boceto de las pantallas
<b>Salidas:</b> Documento con las pruebas a realizar sobre las pantallas
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar un boceto de las posibles pantallas de la aplicación

3.d-. Realizar el diagrama de clases, casos de uso y modelo de dominio
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 20 horas
<b>Descripción:</b> Esta tarea consiste en realizar el diagrama de clases indicando todas las clases a implementar y su contenido. Además, se elaborarán los casos de uso y el modelo de dominio para conocer las relaciones entre las clases, los usuarios y base de datos.
<b>Entradas:</b> Documento del plan de pruebas de las pantallas
<b>Salidas:</b> Documento con el diagrama de clases, casos de uso y modelo de dominio
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar el plan de pruebas de las ventanas

3.e.1-. Realizar la base de datos – Listar los datos que se deben almacenar
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en listar los atributos y las tablas que estarán contenidos en la base de datos.
<b>Entradas:</b> Documento del diagrama de clases, casos de uso y modelo de dominio.
<b>Salidas:</b> Documento con los datos a almacenar en la base de datos.
<b>Recursos necesarios:</b> Ordenador con conexión a internet.
<b>Precedencias:</b> Realizar el diagrama de clases, casos de uso y modelo de dominio

3.e.2-. Realizar la base de datos – Implementar la base de datos
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 2 horas
<b>Descripción:</b> Esta tarea consiste en crear la base de datos según los datos a almacenar
<b>Entradas:</b> Documento con los datos a almacenar
<b>Salidas:</b> Ninguna.
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar la base de datos – Listar los datos que se deben almacenar

<b>3.e.3-. Realizar la base de datos – Probarla para asegurar su correcto funcionamiento</b>
<b>Paquete de trabajo:</b> Realizar análisis y diseño de la aplicación
<b>Duración:</b> 3 horas
<b>Descripción:</b> Esta tarea consiste en probar la base de datos introduciendo todos los datos posibles según el tamaño, valores... con el fin de asegurarnos que la base de datos funcione correctamente en todos los casos.
<b>Entradas:</b> Documento con todos los datos a almacenar para conocer qué tipo de datos se deben introducir
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados.
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar la base de datos – Implementar la base de datos

A continuación se explican por prototipos las tareas, para una mejor comprensión a la hora de realizarla.

<b>PROTOTIPO 1</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades Identificarse, Registrarse y Recuperar contraseña.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 1</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades Identificarse, Registrarse y Recuperar contraseña.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 1</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el primer prototipo.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el primer prototipo.	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

<b>PROTOTIPO 2</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades Calificar pedido y Cerrar sesión.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 2</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades Calificar pedido y Cerrar sesión.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 2</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el segundo prototipo.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el segundo prototipo.	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

<b>PROTOTIPO 3</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades Registrar trabajador, Modificar datos personales y Cambiar idioma.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 3</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 10 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades Registrar trabajador, Modificar datos personales y Cambiar idioma.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 3</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el tercer prototipo.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el tercer prototipo.	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

Como se ha comentado anteriormente, el cuarto prototipo se ha dividido en cuatro subprototipos. Por lo tanto, por cada subprototipo se describe la tarea a realizar.

<b>PROTOTIPO 4 – Subprototipo 1</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades correspondientes a la gestión del personal	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 1</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades correspondientes a la gestión del personal	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 1</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 1 hora	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el subprototipo de la gestión del personal.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el subprototipo de la gestión del personal	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

<b>PROTOTIPO 4 – Subprototipo 2</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades correspondientes a la gestión del almacén	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 2</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades correspondientes a la gestión del almacén	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 2</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 1 hora	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el subprototipo de la gestión del almacén.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el subprototipo de la gestión del almacén.	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

<b>PROTOTIPO 4 – Subprototipo 3</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades correspondientes a la gestión de los vehículos	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 3</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades correspondientes a la gestión de los vehículos	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 3</b>	<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>
<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación	
<b>Duración:</b> 1 hora	
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el subprototipo de la gestión de los vehículos.	
<b>Entradas:</b> Ninguna	
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el subprototipo de la gestión de los vehículos.	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas	

<b>PROTOTIPO 4 – Subprototipo 4</b>	<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en crear las pantallas para las funcionalidades correspondientes a la gestión de los pedidos asignados.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

<b>PROTOTIPO 4 – Subprototipo 4</b>	<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>
<b>Paquete de trabajo:</b> Iniciar la implementación	
<b>Duración:</b> 2.5 horas	
<b>Descripción:</b> Esta tarea consiste en implementar las clases que serán usadas por las pantallas para las funcionalidades correspondientes a la gestión de los pedidos asignados.	
<b>Entradas:</b> Documento con el boceto de las pantallas, documento con el diagrama de clases, casos de uso y modelo de dominio, y el listado de los datos a almacenar en la base de datos.	
<b>Salidas:</b> Ninguna	
<b>Recursos necesarios:</b> Ordenador con conexión a internet	
<b>Precedencias:</b> Realizar la base de datos – Probarla para asegurar su correcto funcionamiento	

**PROTOTIPO 4 – 5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación**  
**Subprototipo 4**

<b>Paquete de trabajo:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación
<b>Duración:</b> 1 hora
<b>Descripción:</b> Esta tarea consiste en probar el código y en asegurar la correcta conexión entre la base de datos y la aplicación para el subprototipo de la gestión de los pedidos asignados.
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con las pruebas realizadas y sus resultados para el subprototipo de la gestión de los pedidos asignados.
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Implementar las clases necesarias para el correcto funcionamiento de las pantallas

**6.a-. Actualizar versiones**

<b>Paquete de trabajo:</b> Finalizar la documentación
<b>Duración:</b> 10 horas
<b>Descripción:</b> Esta tarea consiste en recabar toda la documentación realizada hasta el momento y actualizarla según la implementación del proyecto
<b>Entradas:</b> Toda la documentación
<b>Salidas:</b> Toda la documentación actualizada
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar las pruebas de código y de conexión entre la BD y la aplicación

**6.b-. Realizar las conclusiones y propuestas para el trabajo futuro**

<b>Paquete de trabajo:</b> Finalizar la documentación
<b>Duración:</b> 4 horas
<b>Descripción:</b> Esta tarea consiste en realizar las conclusiones y las posibles tareas futuras que tendría la aplicación
<b>Entradas:</b> Ninguna
<b>Salidas:</b> Documento con las conclusiones y las propuestas futuras
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Actualizar versiones

**6.c-. Unir la documentación**

<b>Paquete de trabajo:</b> Finalizar la documentación
<b>Duración:</b> 7 horas
<b>Descripción:</b> Esta tarea consiste en unir la documentación actualizada y la implementación para preparar la presentación
<b>Entradas:</b> Documentación actualizada y el código
<b>Salidas:</b> Proyecto actualizado
<b>Recursos necesarios:</b> Ordenador con conexión a internet
<b>Precedencias:</b> Realizar las conclusiones y propuestas para el trabajo futuro

#### 6.d-. Preparar la presentación

**Paquete de trabajo:** Finalizar la documentación

**Duración:** 12 horas

**Descripción:** Esta tarea consiste en realizar y preparar la presentación

**Entradas:** Proyecto actualizado

**Salidas:** Proyecto actualizado y presentación finalizada

**Recursos necesarios:** Ordenador con conexión a internet

**Precedencias:** Unir la documentación

## 2.3-. Planificación temporal

Una vez descritas todas las tareas que se llevarán a cabo en este proyecto, se muestra una tabla para visualizar el tiempo estimado que se invertirá para la realización de cada tarea.

Por otro lado, sabiendo que el proyecto es individual, todas las tareas son críticas y se realizarán una a una. De tal manera, el diagrama de Gantt quedará escalonado.

### 2.3.1-. Tareas del proyecto

Se muestra la tabla con todas las tareas y el tiempo estimado.

Descripción	Tiempo
<b>1.a-. Elaborar un horario semanal para realizar el proyecto</b>	0.25 horas
<b>2.a-. Elegir donde almacenar la documentación</b>	2 horas
<b>3.a-. Elegir donde almacenar el código</b>	2 horas
<b>Organización</b>	<b>4.25 horas</b>
<b>2.a-. Realizar los objetivos</b>	2 horas
<b>2.b-. Realizar el alcance</b>	3.5 horas
<b>2.c-. Realizar la planificación temporal</b>	4 horas
<b>2.d-. Realizar el análisis de las herramientas a utilizar</b>	5 horas
<b>2.e-. Realizar la gestión de riesgos</b>	2.5 horas
<b>2.f-. Realizar la evaluación económica</b>	2 horas
<b>2.g-. Realizar el estado del arte</b>	3 horas
<b>Realizar documentación previa</b>	<b>22 horas</b>
<b>3.a-. Realizar la captura de requisitos</b>	3 horas
<b>3.b-. Realizar un boceto de las posibles pantallas de la aplicación</b>	4 horas
<b>3.c-. Realizar el plan de pruebas de las pantallas</b>	6 horas
<b>3.d-. Realizar el diagrama de clases, casos de uso y modelo de dominio</b>	20 horas
<b>3.e.1-. Listar los datos que se deben almacenar</b>	2 horas
<b>3.e.2-. Implementar la base de datos</b>	2 horas
<b>3.e.3-. Probarla para asegurar su correcto funcionamiento</b>	3 horas
<b>Realizar análisis y diseño de la aplicación</b>	<b>40 horas</b>
<b>4.a-. Diseñar las pantallas y realizar las funcionalidades correspondientes</b>	50 horas
<b>4.b-. Implementar las clases necesarias para el correcto funcionamiento de las pantallas</b>	50 horas
<b>Iniciar la implementación</b>	<b>100 horas</b>
<b>5-. Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>	12 horas
<b>Realizar las pruebas de código y de conexión entre la BD y la aplicación</b>	<b>12 horas</b>
<b>6.a-. Actualizar versiones</b>	10 horas
<b>6.b-. Realizar las conclusiones y propuestas para el trabajo futuro</b>	4 horas
<b>6.c-. Unir la documentación</b>	7 horas
<b>6.d-. Preparar la presentación</b>	12 horas
<b>Finalizar la documentación</b>	<b>33 horas</b>
<b>Tiempo total</b>	<b>211.25 horas</b>

Tabla 1: Tiempo de las tareas

Como se puede observar, se ha estimado que el proyecto será finalizado en torno a las 212 horas. Aun así, en el apartado conclusiones se realiza la comparativa entre el tiempo estimado y el tiempo real elaborado en cada tarea.

Para una mayor visualización gráfica del reparto de las horas de trabajo a lo largo del proyecto, se muestra el diagrama de Gantt.

	Tarea	Tiempo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Organización	1.a-	0,25 horas	■					
	1.b-	2 horas						
	1.c-	2 horas						
Realizar documentación previa	2.a-	2 horas	■					
	2.b-	3,5 horas						
	2.c-	4 horas		■				
	2.d-	5 horas						
	2.e-	2,5 horas			■			
	2.f-	2 horas						
	2.g-	3 horas						
Realizar análisis y diseño de la aplicación	3.a-	3 horas			■			
	3.b-	4 horas						
	3.c-	6 horas				■		
	3.d-	20 horas						
	3.e.1-	2 horas					■	
	3.e.2-	2 horas						
	3.e.3-	3 horas						

Tabla 2: Primera parte diagrama Gantt

			Semana 6	Semana 7	Semana 8	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16	Semana 17	Semana 18
Iniciar la implementación	4.a-	50 horas	Primer prototipo		Segundo prototipo		Tercer prototipo		Cuarto prototipo						
	4.b-	50 horas		Primer prototipo		Segundo prototipo		Tercer prototipo		Cuarto prototipo	Cuarto prototipo				
Realizar las pruebas de código y de conexión entre la BD y la aplicación	5-	12 horas			Primer prototipo		Segundo prototipo		Tercer prototipo				Cuarto prototipo		
Finalizar la documentación	6.a-	10 horas													
	6.b-	4 horas													
	6.c-	7 horas													
	6.d-	12 horas													

Tabla 3: Segunda parte diagrama Gantt

Una vez conocida la estimación del proyecto se puede conocer la fecha en torno a la que es posible acabar este proyecto. En base al horario semanal realizado para este proyecto, unas 12 horas a la semana, este proyecto estaría finalizado en 18 semanas. Hay que tener en cuenta que esta planificación se ha hecho con las estimaciones de los tiempos.

#### 2.4-. Herramientas

En este apartado se van a mostrar las diferentes herramientas que se usan durante el proyecto.

- **Android Studio**

La herramienta Android Studio es la que se usa para elaborar toda la implementación del proyecto. Esta herramienta es el entorno en el que se desarrollan todas las aplicaciones Android. Desde su anuncio en la conferencia de Google, el 16 de mayo de 2013, Android Studio superó y reemplazó a Eclipse como entorno de desarrollo. [1]



Ilustración 4: Logo Android Studio

- **WinSCP**

Esta herramienta permite almacenar los archivos *php* con los cuales se pueden realizar las conexiones y llamadas a la Base de Datos. Esta aplicación es un SFTP que emplea conexiones SSH con las cuales permiten la transferencia de archivos entre dos sistemas, local y remoto. [2]



Ilustración 5: Logo WinSCP

- **Notepad ++**

Notepad ++ es un editor de texto y de código que permite crear los archivos *php*. El uso de este editor da una mayor visualización de las variables, de los métodos, etc. definidos en los archivos *php*.



Ilustración 6: Logo Notepad++

- **Visual Paradigm**

Esta herramienta se usa para la creación de los diagramas de secuencia, de clases y de modelo de dominio. El uso de esta herramienta permite claridad a la hora de mostrar dichos diagramas.



*Ilustración 7: Logo Visual Paradigm*

- **Dropbox**

Se usa esta herramienta para almacenar la documentación en un servidor. El almacenamiento de esta herramienta es de 2GB, suficiente para guardar la memoria del proyecto. La elección de esta herramienta se debe a que permite tener todos los documentos en cualquier momento y lugar, así como la de tener una copia de seguridad en caso de pérdida.



*Ilustración 8: Logo Dropbox*

- **Microsoft Office 2013**

La última herramienta que se usa en este proyecto está contenida en el paquete de Microsoft Office 2013. Esta herramienta se denomina *Microsoft Word*, en la cual se redactan todos los documentos que contendrá la memoria. Además, se usa *Microsoft PowerPoint* para la realización y preparación de la presentación.



*Ilustración 9: Logo Microsoft Office*

## 2.5-. Gestión de riesgos

En este punto se detallan los posibles riesgos que pueden suceder durante el proyecto. Por cada riesgo, se describen qué medidas se deben tomar para minimizar la aparición del suceso y en caso de que suceda, se mencionan qué pasos se deben seguir para solucionar dicho riesgo.

Además, se indica qué probabilidad de suceso tiene cada riesgo. Las probabilidades se detallan en la siguiente tabla.

Tipo de probabilidad	Porcentaje (%)
Probabilidad alta	66.67% – 100%
Probabilidad media	33.34% - 66.66%
Probabilidad baja	0% - 33.33%

Tabla 4: Probabilidad de los riesgos

En caso de que el riesgo haya sucedido se debe tener en cuenta el impacto que tendrá. Este impacto se califica cómo jornadas perdidas, es decir, días perdidos resolviendo el problema. Estos días se clasifican en cuatro grupos, tal y como se detalla en la siguiente tabla.

Impacto	Días perdidos
Bajo	Menos de un día
Medio	Entre uno y dos días
Alto	Entre dos y tres días
Muy alto	Más de tres días

Tabla 5: Impacto de los riesgos

Una vez definida la probabilidad y el impacto, se empiezan a mostrar los posibles riesgos.

Riesgo 1 – Que se caigan los servidores de red
<b>Descripción:</b> Internet o el servidor que almacene la Base de Datos, se caigan por motivos desconocidos.
<b>Plan de prevención:</b> Realizar las conexiones con anticipación. En caso de que no se pueda acceder al sistema, se pueda acceder al plan de contingencia sin retrasar el proyecto.
<b>Plan de contingencia:</b> En primer lugar, se puede esperar hasta que vuelva a establecerse la conexión. En caso de que la espera se alargue, avanzar con la implementación sin poder probarla al no tener acceso a internet y ponerse en contacto con el tutor si el problema es el servidor, o con la compañía telefónica si el problema es la conexión a internet.
<b>Probabilidad:</b> Media
<b>Impacto:</b> Medio

Riesgo 2 – Pérdida de alguna copia
<b>Descripción:</b> Se ha perdido alguna copia de la documentación o de la implementación.
<b>Plan de prevención:</b> En caso de la documentación, se tendrán dos copias. Una en local, es decir, en el ordenador donde se redacta la memoria y otra en la herramienta Dropbox. Y en el caso de la implementación, también se tendrán dos copias. Una en local donde se implementa y otra en una memoria USB. Una vez finalizadas las horas diarias del proyecto, se realizarán las copias en los lugares indicados.
<b>Plan de contingencia:</b> Dependiendo de qué copia se pierda, se restaurarán los archivos de la segunda copia y habrá dos copias de nuevo.
<b>Probabilidad:</b> Media
<b>Impacto:</b> Bajo

Riesgo 3 – Avería de hardware en el equipo informático
<b>Descripción:</b> Se ha averiado algún dispositivo hardware del equipo informático en el que se realiza el proyecto.
<b>Plan de prevención:</b> Realizar un mantenimiento periódico del ordenador, como por ejemplo una limpieza del ventilador.
<b>Plan de contingencia:</b> Se debe arreglar en la mayor brevedad de tiempo o incluso adquirir otro. En caso de tener más de un ordenador, arreglar el primero, usar el segundo ordenador y tener en cuenta el riesgo anterior, pérdida de alguna copia.
<b>Probabilidad:</b> Media
<b>Impacto:</b> Alto

Riesgo 4 – Avería de software en el equipo informático
<b>Descripción:</b> Se ha introducido un virus en el ordenador o algún programa no responde correctamente.
<b>Plan de prevención:</b> Realizar un mantenimiento periódico del ordenador, como por ejemplo análisis de antivirus.
<b>Plan de contingencia:</b> Se debe dedicar tiempo para realizar labores de recuperación o de limpieza para volver a restaurar el ordenador y volver a continuar con el proyecto. En caso de no poder solucionar el problema, se llevará el ordenador a un técnico y se buscará otro ordenador donde continuar el proyecto.
<b>Probabilidad:</b> Media
<b>Impacto:</b> Alto

Riesgo 5 – Pérdida o deterioro del teléfono móvil
<b>Descripción:</b> Este riesgo se debe tener en cuenta debido a que se realizarán las pruebas en un móvil. Este riesgo se da cuando se ha estropeado o se ha perdido el móvil.
<b>Plan de prevención:</b> Tener especial cuidado con el móvil.
<b>Plan de contingencia:</b> Realizar las pruebas sobre los emuladores de Android Studio. Las pruebas sobre estos dispositivos son lentas y se deben tener los emuladores actualizados para intentar minimizar estas pérdidas de tiempo
<b>Probabilidad:</b> Baja
<b>Impacto:</b> Alto

Riesgo 6 – Enfermedad o lesión
<b>Descripción:</b> El desarrollador se ha puesto enfermo o ha tenido alguna lesión realizando algún deporte. Debemos tener en cuenta este riesgo debido a los cambios constantes de temperatura y a la realización de deportes.
<b>Plan de prevención:</b> Tomar medidas para minimizar este caso
<b>Plan de contingencia:</b> Reposo y medicación
<b>Probabilidad:</b> Media – Alta
<b>Impacto:</b> Alto o muy alto dependiendo de la enfermedad o lesión

Riesgo 7 – Problemas a la hora de realizar la implementación
<b>Descripción:</b> Aunque se tenga un nivel de conocimientos medio - alto en el lenguaje de programación Java, el cual es usado en la herramienta Android Studio, es posible que algunas funcionalidades no se sepa implementarlas y se quede atascado el desarrollador.
<b>Plan de prevención:</b> Realizar pruebas de los avances
<b>Plan de contingencia:</b> Dedicar más tiempo del estimado para resolución del programa y en caso de no encontrar soluciones, consultar con el tutor.
<b>Probabilidad:</b> Media – Alta
<b>Impacto:</b> Medio o alto dependiendo de la funcionalidad

## 2.6-. Evaluación económica

En este apartado se identifican los beneficios y gastos asociados al proyecto antes de la implementación del mismo. La evaluación económica se dividirá en dos partes: los gastos y la inversión.

### 2.6.1-. Gastos

Los gastos se clasifican en función de gastos directos o gastos indirectos. Los gastos directos son aquellos que se pueden establecer de manera clara al ser deducidos con precisión teniendo en cuenta las horas trabajadas. Respecto a los gastos indirectos, son difíciles de valorar sobre cuanto valor aportan al proyecto. Por ejemplo, los gastos de la luz o amortizaciones de equipos informáticos.

Los gastos se pueden visualizar en la siguiente tabla:

Descripción del gasto	Tipo de gasto
Sueldo del desarrollador	Directo
Amortización de los equipos informáticos	Indirecto
Amortización del teléfono móvil	Indirecto
Licencia de Microsoft Office 2013	Indirecto
Licencia de Visual Paradigm	Indirecto
Gastos de luz e internet	Indirecto

Tabla 6: Gastos del proyecto

Una vez definidos los tipos de gastos, se analizan cada uno de ellos.

- **Sueldo del desarrollador**

Teniendo en cuenta que un desarrollador cobra de media 30 euros/hora y tras realizar la planificación, el coste de este sería  $30 * 212 \text{ horas} = 6360 \text{ euros}$ . Se debe tener en cuenta que las horas se han realizado en base a estimaciones y que más adelante indicaremos las horas reales invertidas. En base a esas horas invertidas, realizaremos de nuevo el sueldo del desarrollador.

- **Amortizaciones**

Para programar el proyecto se necesitará un ordenador de sobremesa valorado en unos 700 euros con una vida media de 7 años. En cuanto al teléfono móvil, se necesitará uno valorado en unos 300 euros con una vida media de 3 años. Pero como no se utilizan estos equipos al 100% para el proyecto, se tiene en cuenta un 25% a la hora del gasto.

Amortización anual:

Ordenador -> (700 euros / 7 años) \* 25 % = **20 euros** al año

Móvil -> (300 euros / 3 años) \* 25% = **20 euros** al año

- **Licencias**

Se necesitarán dos tipos de licencias, una para Microsoft Office y otra para Visual Paradigm. La licencia de Microsoft Office está valorada en **90 euros** según <https://products.office.com/es-es/home>. Y la licencia de Visual Paradigm está valorada en **99 euros** según <https://www.visual-paradigm.com/shop/vp.jsp?license=perpetual>

- **Gastos de luz e internet**

Es complicado estimar el gasto de luz e internet al mes debido a los constantes cambios. Teniendo en cuenta las últimas facturas de gastos, se pueden realizar la media y valorar las facturas como 80 euros al mes. Debido a que no se dedica toda la luz y el internet para el proyecto, se puede considerar que usamos el 25%. De tal manera, la factura sería de **20 euros al mes**.

De tal manera, la suma para calcular el gasto anual sería la siguiente:

$6360 + 20 + 20 + 90 + 99 + (20 * 12 \text{ meses}) = 6829 \text{ euros}$

Como se puede observar, los gastos ascienden a **6829 euros** teniendo en cuenta que la planificación de las horas es una estimación.

#### 2.6.2-. Inversiones

Una vez conocido el gasto que implica desarrollar esta aplicación, se proponen métodos con los cuales se puede ofrecer la aplicación a las diferentes empresas.

La primera de las opciones es ofrecer licencias a las empresas. Se ofrecería la aplicación gratuitamente durante 15 días para que ellos realicen las pruebas pertinentes y tengan un conocimiento de la aplicación. Transcurridos los 15 días, se ofrecería una licencia de 125 euros al mes durante un año, y a partir de dicho año la licencia aumentaría a 150 euros al mes. Inicialmente no se conocerán muchos clientes pero es posible que haya 4 o 5 interesados. Teniendo en cuenta estos 4 o 5 clientes, se puede rentabilizar la aplicación en un año o incluso en meses si es el caso de 5 clientes.

Otra de las opciones, es introducir publicidad a pie de aplicación. Esta publicidad se cobraría por parte de las empresas que deseen publicitarse. Aunque haya muchos interesados, no se pretende llenar la aplicación de publicidad, con lo que se elegirá a los publicistas con mejor oferta. Además, se ofrecería al cliente dos versiones, la opción sin publicidad con el precio fijo indicado en el párrafo anterior, o la opción con publicidad. Esta última opción disminuiría en 25 euros la licencia. Es decir, 100 euros al mes durante el primer año y 125 euros al mes a partir del primer año. Teniendo en cuenta el inicio con 4 o 5 clientes, se rentabilizaría en un año y pocos meses o en un año respectivamente de los clientes.

## 2.7-. Arquitectura

La arquitectura utilizada para desarrollar esta aplicación trata de cliente-servidor. Para comprender mejor este tipo de arquitectura, se utiliza la siguiente imagen.

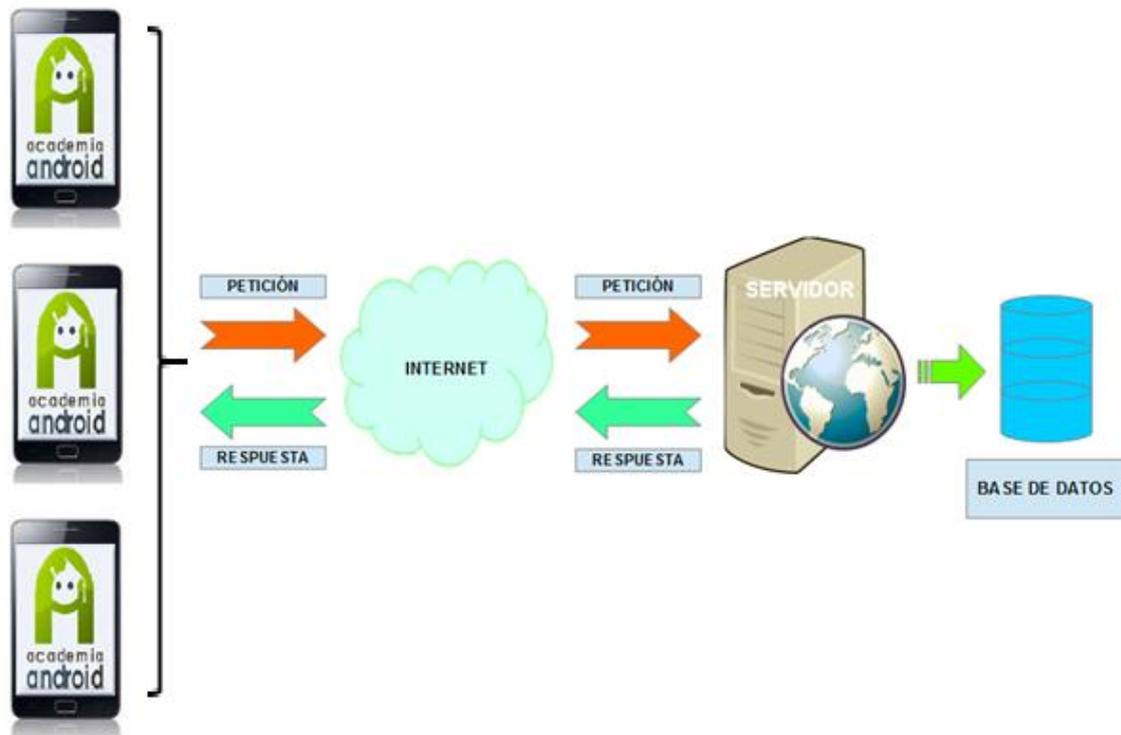


Ilustración 10: Arquitectura de la aplicación

Como se puede observar en la imagen, uno o varios dispositivos móviles realizarán peticiones a un servidor situado en Internet. Estas peticiones tendrán como destino la base de datos almacenada en dicho servidor. Una vez las peticiones hayan sido tratadas, serán devueltas al dispositivo móvil mediante el servidor. Estas respuestas se obtienen en formato texto o JSON. Generalmente estarán en formato texto al necesitar cierta información. Se usará el formato JSON, por ejemplo, cuando se realicen peticiones a Google Maps para obtener coordenadas.

La aplicación puede tener la base de datos en local o en un servidor remoto. Se ha elegido la opción de almacenar la base de datos en un servidor en vez de almacenarla en local debido a que requiere menos capacidad de almacenamiento y a que había disponible un servidor para crear la base de datos. Aun así, el motivo principal ha sido que la aplicación va a estar en diferentes dispositivos y todos ellos deben hacer referencia a la misma base de datos.

### 3-. Antecedentes

La idea de desarrollar esta aplicación surgió tras realizar charlas con consultores que dan soporte a herramientas dirigidas a la gestión de empresas y que desarrollan mejoras en este tipo de herramientas. Estas empresas están enfocadas al transporte de mercancías, paquetería, etc. El objetivo es mejorar la eficiencia de dichas empresas con esta aplicación. Esta eficiencia se vería mejorada a la hora de gestionar los pedidos. Además, permitiría usar la aplicación Google Maps para ayudarnos en nuestra ruta de reparto.

Respecto a las aplicaciones en el mercado, hay tres aplicaciones similares. Estas herramientas son:

- **Natureh Customer Tracking**

La empresa Natureh ofrece una herramienta web que permite el control y monitorización de los pedidos, envíos y entregas al cliente. Las características que presenta esta solución son: [3]

- Multi-idioma
- Multi-usuario
- Guardado del histórico y seguimiento de pedidos
- Integración en cualquier software: ERP, CRM...
- Diseño amigable

- **Logistics**

Logistics es una aplicación Android que permite controlar todas las etapas del servicio de distribución. Consta de las siguientes funcionalidades: [4]

- Control y arreglos de transporte
- Optimización de los procesos de transporte
- Herramienta web
- Creación, modificación y distribución de pedidos
- Enrutamiento por Google Maps

- **TomTom Webflet**

Webflet es una solución de gestión de flotas de transporte que permite conocer en tiempo real donde se encuentran los vehículos conociendo de tal manera el estado del envío. Además, permite comunicación con los conductores. Lleva 17 años y ha prestado servicio a más de 48.000 clientes con más de 785.000 vehículos. Las funcionalidades de esta herramienta son: [5]

- Seguimiento del vehículo y del pedido
- Adaptación al estilo de conducción y planes de ahorro de combustible
- Comunicación con el conductor
- Tiempo de conducción restantes y tacógrafo
- Informes detallados

Una vez conocidas las herramientas que hay en el mercado, hay que resaltar que esta idea no tuvo en cuenta las aplicaciones ya creadas para la gestión de pedidos. El caso de coincidir con alguna existente es casualidad. La inspiración surgió para crear una nueva aplicación con funcionalidades diferentes a las anteriores para ofrecer una nueva aplicación a los usuarios.

Como ya conocemos, las compras por Internet han aumentado considerablemente.

### Porcentaje de usuarios de Internet en los últimos 3 meses<sup>1</sup> por tipo de actividad realizada y sexo. Año 2017

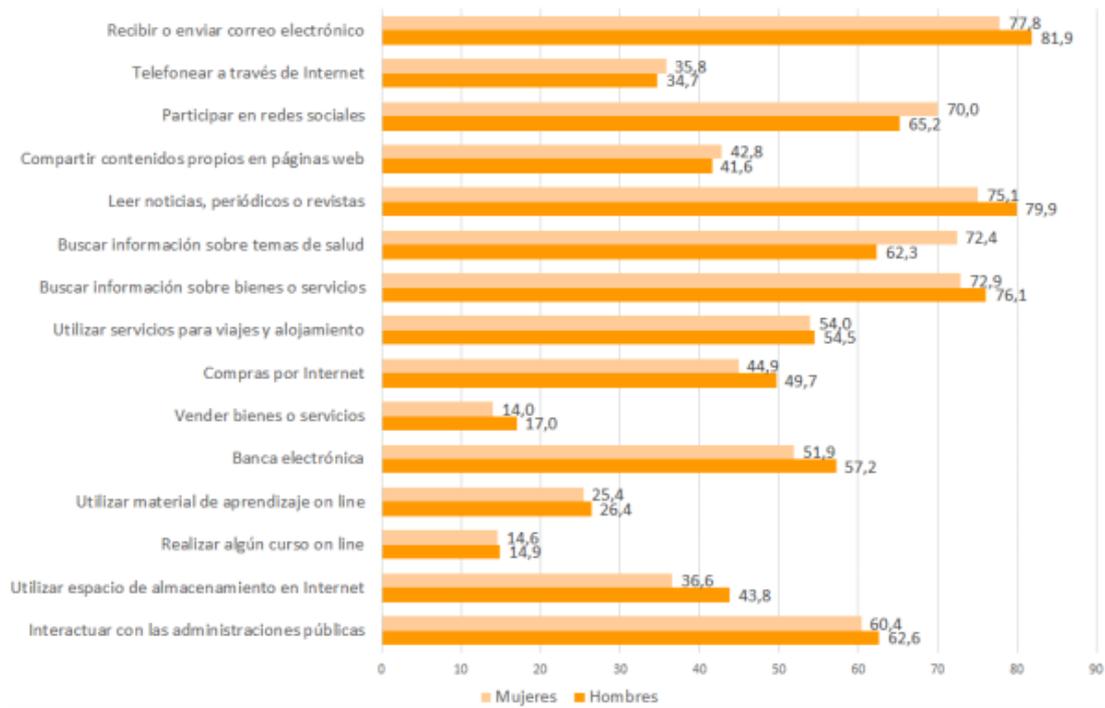


Ilustración 11: Actividades de los usuarios

## Porcentaje de usuarios de TIC por comunidades autónomas. Año 2017

	Han utilizado Internet en los 3 últimos meses	Usuarios frecuentes de Internet	Han comprado por Internet en los 3 últimos meses
<b>Total</b>	<b>84,6</b>	<b>80,0</b>	<b>40,0</b>
Andalucía	83,9	78,2	36,0
Aragón	86,8	82,4	45,3
Asturias, Principado de	82,3	77,0	40,8
Baleares, Illes	88,5	85,7	47,9
Canarias	83,5	80,3	31,2
Cantabria	82,7	77,3	46,3
Castilla y León	81,3	76,9	35,4
Castilla - La Mancha	78,3	73,3	37,3
Cataluña	85,7	81,8	45,5
Comunitat Valenciana	84,0	80,4	36,6
Extremadura	80,2	76,0	34,3
Galicia	79,4	72,1	32,1
Madrid, Comunidad de	90,0	85,5	48,3
Murcia, Región de	84,5	81,1	32,1
Navarra, Comunidad Foral de	86,7	82,1	48,7
País Vasco	85,7	81,8	43,3
Rioja, La	82,0	75,8	39,3
Ceuta	81,4	81,4	30,1
Melilla	88,0	82,5	45,8

*Ilustración 12: Compras online*

Como podemos observar en los anteriores gráficos extraídos del INE (Instituto Nacional de Estadística [www.ine.es](http://www.ine.es)), se asemeja la cantidad de hombres y mujeres que realizan compras online. Además, entorno a la mitad de usuarios frecuentes de internet realizan compras.

Conociendo estos datos y el aumento de las compras online en determinadas fechas, como navidad, rebajas, Black Friday, etc. la eficiencia a la hora de la entrega de los pedidos es clave. Por ello, se pensó en desarrollar esta aplicación. De tal manera, la entrega se pretende realizar en el menor tiempo posible, lo que conllevaría mayor satisfacción al cliente. Sabiendo que el cliente acabará satisfecho, es posible que sus compras aumenten en cantidad y se obtengan mayores beneficios.



## 4-. Captura de requisitos

En este apartado se van a detallar qué funcionalidades han sido implementadas en la aplicación. Además, por cada funcionalidad se expone una breve descripción para conocer mejor su función.

Una vez descritas las funcionalidades, se muestra el modelo de casos de uso y el modelo de dominio para tener una primera idea de la implementación.

### 4.1-. Modelo de casos de uso

En base a la descripción de la aplicación, la aplicación la van a utilizar tres tipos de usuarios: trabajador, director y el cliente. El cliente es el usuario a quien le ha llegado el pedido y su única labor será calificar el estado del pedido.

En primer lugar se muestra el modelo de casos de uso correspondiente al Cliente.

- **Cliente**

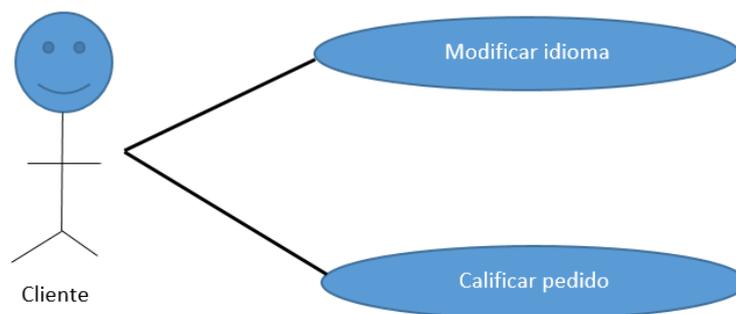


Ilustración 13: Casos de uso del cliente

- *Modificar idioma*

La aplicación esta implementada para tres idiomas: castellano, euskera e inglés. El usuario en cualquier momento podrá puede el idioma al que desee.

- *Calificar pedido*

El cliente final puede dar su opinión en cuanto al pedido recibido. Para ello, al abrir la aplicación aparte de observar las opciones de identificarse y registrarse (las cuales más adelante detallaremos), observa una opción de valorar el pedido. En esta pantalla, introduce su calificación en base a unas puntuaciones y al final puede introducir un texto describiendo su opinión. Además de indicar toda esta información, debe introducir el código del pedido recibido y el nombre de usuario del repartidor. Todos estos de datos se detallarán en el modelo de dominio.

A continuación, se detalla el correspondiente al trabajador

- **Trabajador**

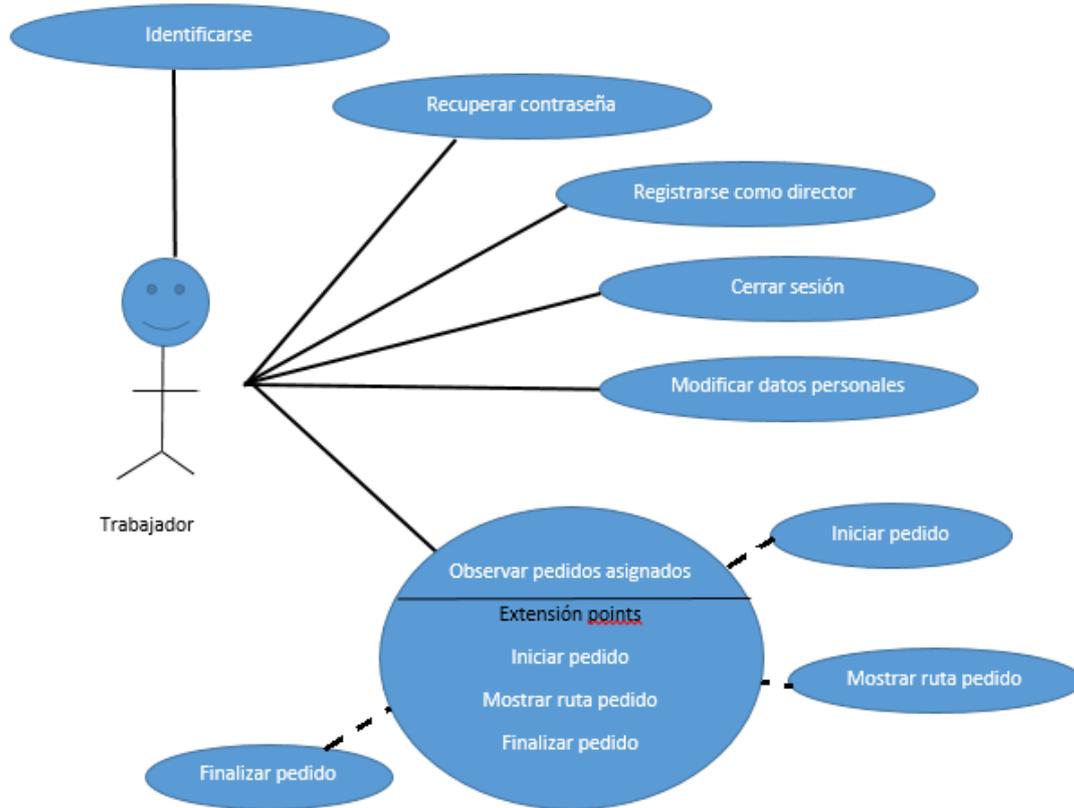


Ilustración 14: Casos de uso del trabajador

- *Identificarse*

El usuario debe identificarse en la aplicación. Para ello, debe especificar su nombre de usuario y su contraseña.

- *Recuperar contraseña*

El usuario en caso de no recordar su contraseña, puede recuperarla introduciendo su nombre de usuario y su email.

- *Registrarse como director*

El director debe registrarse indicando sus datos personales, su nombre de usuario y su contraseña. Además, debe indicar los datos de su empresa.

- *Cerrar sesión*

El usuario puede abandonar su sesión para identificarse con cualquier otro tipo de usuario. Además, en caso de haber cerrado la aplicación sin cerrar sesión, al iniciarla de nuevo no le hará falta identificarse debido a que se almacenarán los datos de la última sesión.

- *Modificar datos personales*

Cada usuario en cualquier momento puede modificar sus datos personales.

- *Observar pedidos asignados*

El trabajador puede observar que pedidos le han sido encargados junto con sus datos. Además, observará todos los detalles del pedido.

- *Iniciar pedido*

El trabajador puede indicar que inicia el pedido y de tal manera, esa información podrá visualizarla el director.

- *Mostrar ruta pedido*

El trabajador si lo desea, puede ver la ruta a seguir en vehículo entre la empresa y el destino.

- *Finalizar pedido*

El trabajador una vez haya entregado el pedido, indicará que el pedido ha sido completado. Además, introducirá los kilómetros realizados por su vehículo, entre otros datos.

Por último, el modelo de casos de uso del Director.

- **Director**

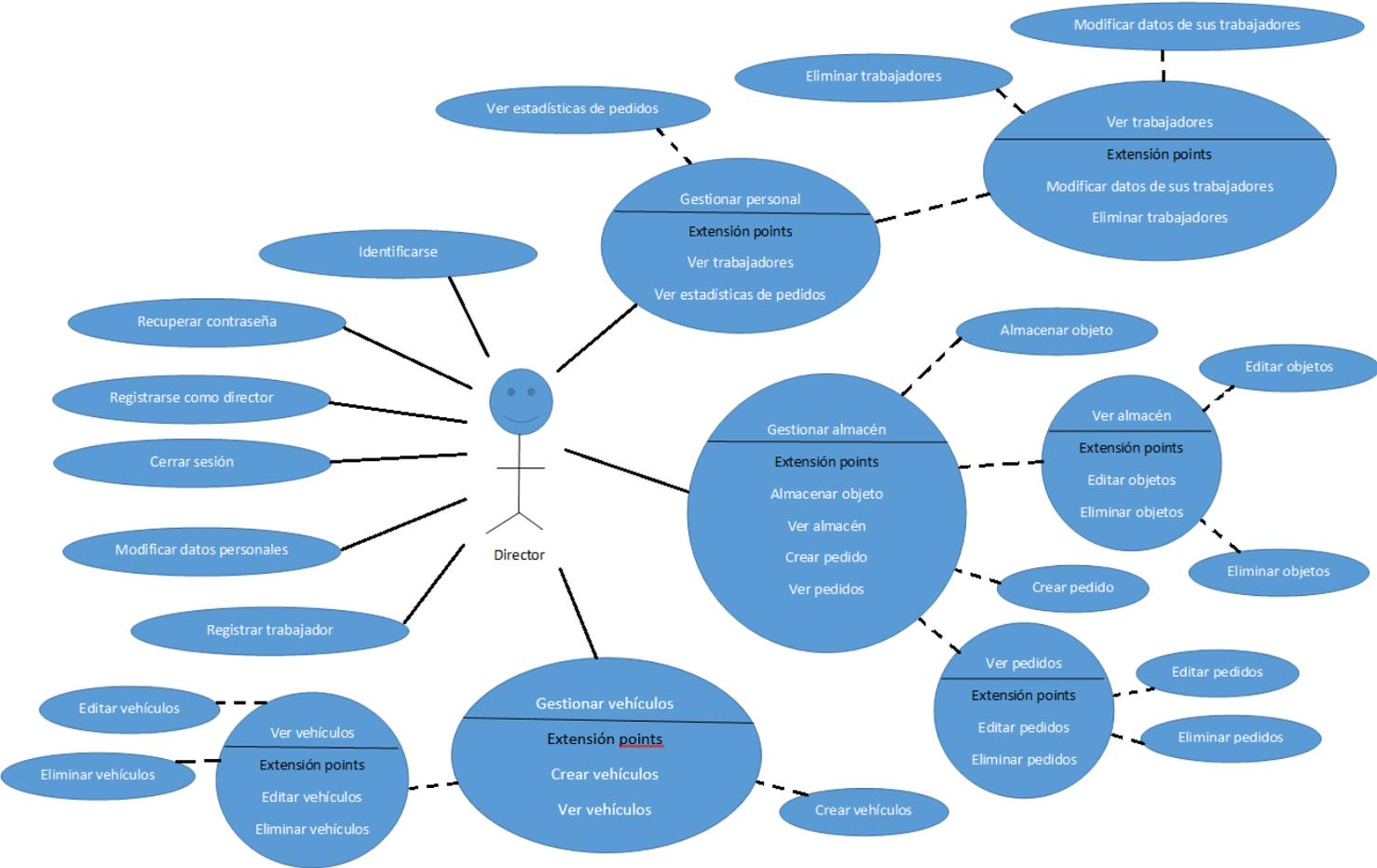


Ilustración 15: Casos de uso del director

- *Identificarse*

El usuario debe identificarse en la aplicación. Para ello, debe especificar su nombre de usuario y su contraseña.

- *Recuperar contraseña*

El usuario en caso de no recordar su contraseña, puede recuperarla introduciendo su nombre de usuario y su email.

- *Registrarse como director*

El director debe registrarse indicando sus datos personales, su nombre de usuario y su contraseña. Además, debe indicar los datos de su empresa.

- *Cerrar sesión*

El usuario puede abandonar su sesión para identificarse con cualquier otro tipo de usuario. Además, en caso de haber cerrado la aplicación sin cerrar sesión, al iniciarla de nuevo no le hará falta identificarse debido a que se almacenarán los datos de la última sesión.

- *Modificar datos personales*

Cada usuario en cualquier momento puede modificar sus datos personales.

- *Registrar trabajador*

El director crea los usuarios para sus trabajadores. Por cada trabajador indica sus datos personales con los cuales podrán identificarse.

- *Ver sus trabajadores*

El director puede visualizar todos los trabajadores suyos, junto con los datos de cada uno de ellos.

- *Modificar datos de sus trabajadores*

El director cuando lo desee puede modificar cualquier dato de sus trabajadores.

- *Eliminar trabajadores*

Por último respecto a los trabajadores, puede eliminar cualquier trabajador que haya creado.

- *Ver estadísticas de pedidos*

El director puede ver las valoraciones que han recibido sus trabajadores a la hora de entregar los pedidos.

- *Crear vehículos*

El director debe crear vehículos con los cuales sus trabajadores entregarán los pedidos. Para ello debe introducir una serie de datos, como la matrícula, modelo...

- *Ver vehículos*

Puede observar todos los vehículos que ha creado junto con sus datos.

- Modificar vehículos

El director puede modificar cualquier dato de los vehículos que haya creado.

- Eliminar vehículos

El director puede eliminar tantos vehículos suyos como desee.

- *Almacenar objetos*

El director puede almacenar objetos en la base de datos con los cuales podrá realizar pedidos más adelante. Por cada objeto indicará una serie de datos.

- Ver el almacén

El director puede visualizar todos los objetos que tiene almacenados, junto con sus detalles.

- Editar objetos

Puede editar cualquier dato de los objetos que haya creado.

- Eliminar objetos

Por último respecto a los objetos, puede eliminar cualquier objeto que haya almacenado.

- *Crear pedidos*

Este usuario puede crear pedidos. El pedido estará compuesto por uno o varios objetos, los cuales deberán ya estar almacenados en la base de datos. Al crear el pedido, debe seleccionar el destino, el vehículo con el que se realiza el pedido y el trabajador encargado de entregar este pedido.

- *Ver pedidos*

En esta funcionalidad, el director puede observar todos los pedidos creados junto con sus detalles del envío.

- Editar pedidos

El director puede editar cualquier pedido creado, siempre y cuando el estado del pedido esté en “En espera”, es decir, no se encuentre “Iniciado” o “Finalizado”.

- Eliminar pedidos

El director cuando lo desee, sin importar el estado del pedido, puede eliminarlo.

### 4.3-. Modelo de dominio

En primer lugar, se observan las relaciones entre las entidades para su posterior descripción.

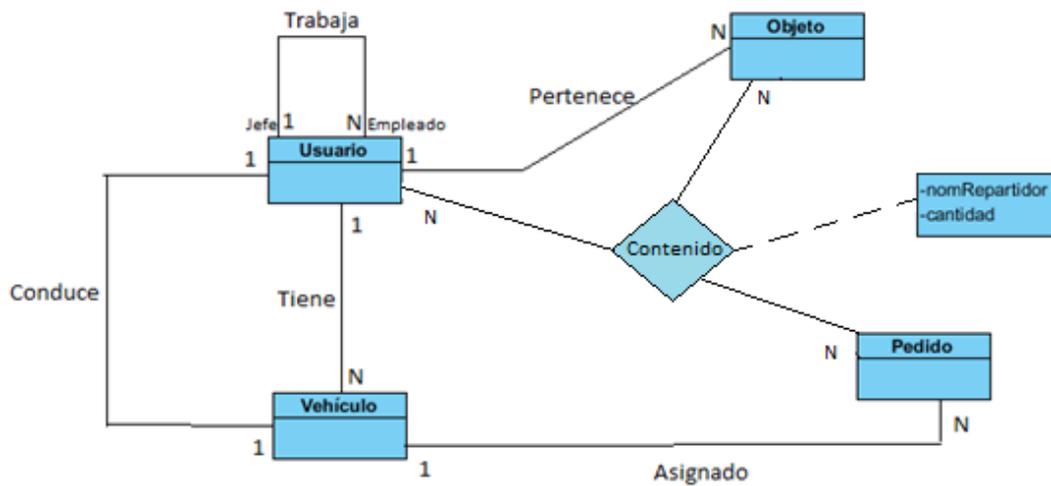


Ilustración 16: Modelo de dominio

En primer lugar se describe que representan las entidades.

La entidad Usuario representa a un usuario de la aplicación, tanto a un trabajador como a un director.

La entidad Vehículo representa un vehículo almacenando los datos de este.

La entidad Pedido representa un pedido creado.

Y por último, la entidad Objeto representa un objeto que ha sido creado.

A continuación, se explican las relaciones.

La relación “Conduce” quiere decir que un usuario actualmente conduce un vehículo y que un vehículo es conducido actualmente por un usuario.

La relación “Tiene” quiere decir que un usuario tiene muchos vehículos y que un vehículo pertenece únicamente a un usuario.

La relación “Trabaja” quiere decir que un jefe tiene muchos empleados y que un empleado solo trabaja para un jefe.

La relación “Pertenece” quiere decir que un objeto pertenece a un usuario pero un usuario puede tener muchos objetos.

La relación “Asignado” quiere decir que un pedido se entregará con un único vehículo. Por otro lado, un vehículo puede entregar muchos pedidos.

Por último, la relación múltiple “Contenido” especifica que usuario ha creado un pedido y con qué objetos.



## 5-. Análisis y diseño

En este apartado se va a describir como se llevó a cabo la implementación de la aplicación. Para ello, se expone la base de datos utilizada, el orden en el que se desarrollan los prototipos y el motivo de este orden. Finalmente, para una mayor aclaración, se acompañará esta descripción con imágenes que representen las relaciones con la base de datos, el diagrama de clases, etc.

En primer lugar, se identifican los atributos de las tablas de la base de datos. Estos campos se obtienen del análisis del anterior modelo de dominio.

**Usuario** ( nomUsuario, nombre, apellidos, contraseña, email, puesto, nomEmpresa, calle, piso, códigoPostal, municipio, numContacto, emailContactoEmpresa, mediaTratoRepartidor, mediaEstadoPaquetes, mediaPuntualidad, cantidadPedidosRealizados, nomUsuarioJefe, matrículaVehículo, cantidadPedidosValorados )

**Objeto** ( codObjeto, descripción, nomPropietario )

**Pedido** ( codPedido, calle, piso, codigoPostal, municipio, estadoPedido, diaSalida, horaSalida, diaLlegada, horaLlegada, pedidoValorado, nomRepartidor, tratoRepartidor, estadoPaquete, puntualidad, opinión, matrículaVehículo , nomJefe)

**Vehículo** ( matrícula, modelo, edad, últimaITV, próximaITV, KmRecorridos, cargaKg, nomUsuarioPropietario )

**Contenido** ( codPedido, codObjeto, nomJefe, nomRepartidor, cantidad )

Como se puede observar, se han marcado las claves primarias con subrayado. Respecto a las claves extranjeras, se han marcado como subrayado con guiones.

La aplicación consta de muchas funcionalidades que se pueden ir desarrollando de forma independiente ya que no afectarían a otras. De tal manera, se juntan por prototipos las funcionalidades que son independientes a otras y que su implementación no se demore durante muchas semanas para así tener un equilibrio de duración entre todos los prototipos. Además, para una mayor facilidad a la hora de desarrollar estos prototipos, se recomienda haber creado antes las tablas en la base de datos debido a que tras finalizar cada prototipo se harán las pruebas para que el prototipo funcione correctamente. De tal manera, si la base de datos ya está creada, las pruebas se harán más rápido.

Por cada prototipo se expone los diagramas de clases de las actividades. Una actividad es un tipo de clase que en la herramienta Android Studio se usa para implementar las funcionalidades. Una aplicación sin actividades no es posible crearla. Por ello, se va a explicar cómo es el detalle de una aplicación Android.

Una aplicación posee varios tipos de archivos, siendo todos ellos necesarios. Como se ha comentado, una actividad es donde se implementan las acciones que se van a realizar. Las actividades pueden estar relacionadas con archivos *layout*. Un archivo *layout* se corresponde con una pantalla. Estos archivos se crean con lenguaje *xml*, indicando y describiendo cada componente visual. Además, cada aplicación es posible implementarla en el idioma o idiomas que se desee. Para ello está el archivo en formato *xml* en el que se almacenan las descripciones de cada texto en los idiomas que se desee.

A continuación, se describen los prototipos. La siguiente imagen se corresponde con las clases necesarias para el **primer prototipo**.

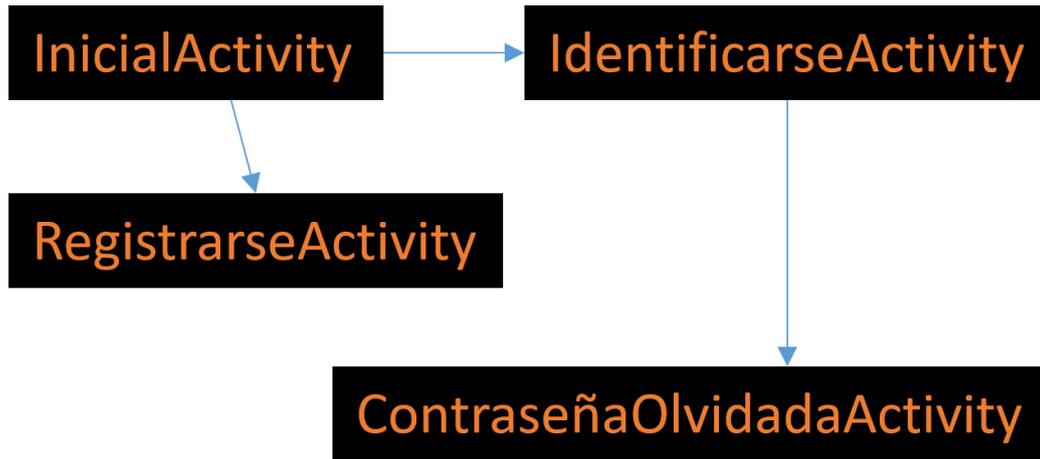


Ilustración 17: Diagrama de clases Primer prototipo

Este prototipo consta de 4 actividades. La actividad “InicialActivity” controla la primera acción que hace el usuario al abrir la aplicación, sin tener iniciada la sesión. Esta actividad controla la acción del usuario, ya que este puede elegir entre identificarse, registrarse o calificar el pedido.

La actividad “IdentificarseActivity” le da la opción de iniciar sesión introduciendo su nombre de usuario y su contraseña. Además, en caso de introducir los datos correctos, crea la instancia de Sesión. Por otro lado, controla si el usuario pulsa sobre la opción de recordar su contraseña.

En ese caso, está la actividad “ContraseñaOlvidada”. Esta actividad recoge algunos datos personales del usuario para comprobar su veracidad y mostrarle su contraseña.

Por otro lado, el usuario puede registrarse en la actividad “InicialActivity”. Para este caso se ha implementado la actividad “RegistrarseActivity”. Esta actividad recoge todos los datos necesarios para la creación de un usuario, y en caso de todos los datos correctos, crea el nuevo usuario.

La siguiente imagen se corresponde con las clases necesarias para el **segundo prototipo**. Además, se implementa el menú lateral contenido en el menú principal.

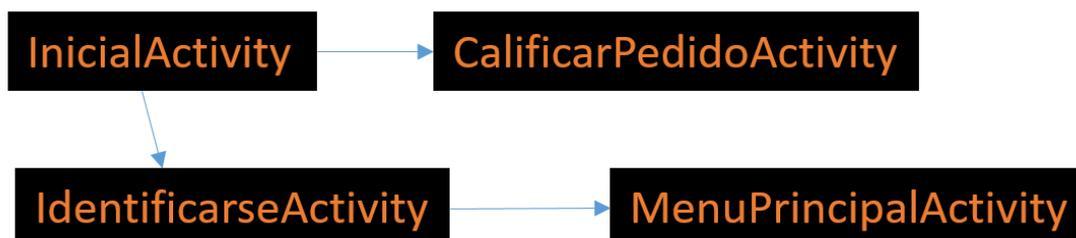
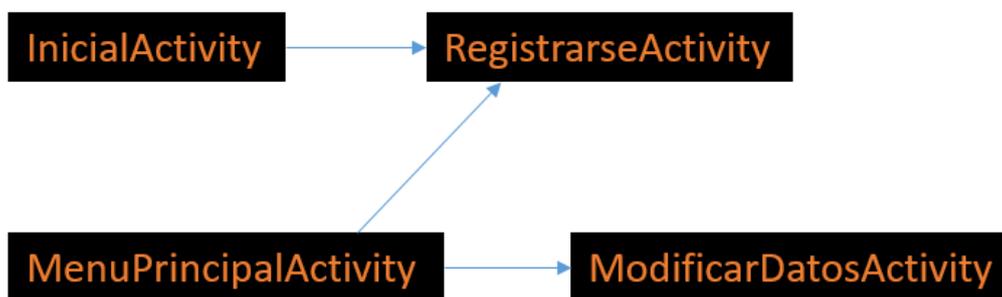


Ilustración 18: Diagrama de clases Segundo prototipo

Para realizar este prototipo, se usan de nuevo las actividades “InicialActivity” e “IdentificarseActivity”. En la actividad “InicialActivity” se controla el caso en el que un usuario desee calificar el estado del pedido recibido. Para ese caso se ha implementado la actividad “CalificarPedidoActivity”, que trata todas las valoraciones de ese pedido y las almacena en la base de datos.

Para la funcionalidad de “CerrarSesión” se creó la actividad “MenuPrincipalActivity”. En esta actividad se le da opción de cerrar sesión al usuario tras haberse identificado en la aplicación. Una vez seleccionada esta opción, la aplicación retorna a la actividad “InicialActivity”. Además, se debe tener en cuenta la opción de que el usuario haya salido de la aplicación con la sesión iniciada. En ese caso, la actividad inicial será “MenuPrincipalActivity” con la sesión iniciada del último usuario.

La siguiente imagen se corresponde con las clases necesarias para el **tercer prototipo**. Se decidieron estas tres funcionalidades dejando para el último paquete todas las funcionalidades que se corresponden con las gestiones. Para realizar la funcionalidad de Registrar trabajador, se reutiliza la actividad de Registrarse bloqueando los campos relacionados con la empresa. Para ello, se implementan las siguientes actividades.



*Ilustración 19: Diagrama de clases Tercer prototipo*

En la actividad “InicialActivity” se añade la opción de que el usuario elija entre los idiomas Castellano, Euskera e Inglés. La elección del idioma implica la modificación del idioma en toda la aplicación. Por defecto, el idioma es el que posea el usuario en el teléfono o Tablet.

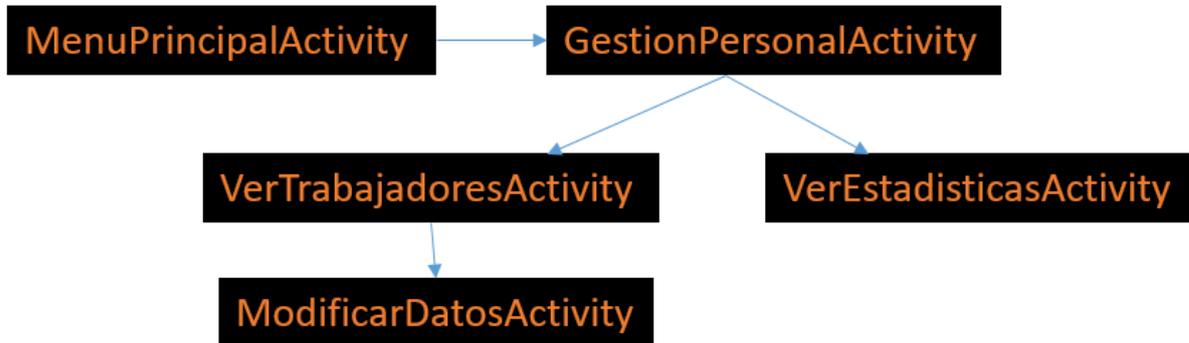
Para desarrollar la funcionalidad registrar trabajador, se enlazan las actividades “MenuPrincipalActivity” y “RegistrarseActivity”. Debido a que a la hora de registrarse como director, el usuario debe introducir todos los campos referentes a una persona, desde la actividad del menú se llama a esa actividad para crear un trabajador y se controla la distinción entre las funcionalidades de registrarse como director y registrar un trabajador.

Respecto a la funcionalidad de modificar los datos personales, está la actividad “ModificarDatosActivity”. Esta actividad muestra al usuario todos los datos actuales del usuario para que este pueda modificarlos, para su posterior actualización en la base de datos.

Por último, se implementa el **cuarto prototipo**. Se ha elegido este contenido para el último paquete debido a que contiene todas las actividades relacionadas con la gestión. Las tres primeras funcionalidades se corresponden al director y la última al trabajador. Como se ha mencionado en el apartado Casos de uso, las gestiones permiten insertar, modificar y eliminar elementos correspondientes a la gestión. Por lo tanto, se debe tener en cuenta todos los casos posibles.

- *Gestionar personal*

En este subprototipo, y en todos los siguientes subprototipos, se implementa una actividad que gestiona la acción que ha seleccionado el usuario. Esta actividad se denomina “GestionPersonalActivity”, cambiando su nombre dependiendo del subprototipo que sea. Para ello, se implementan las siguientes actividades.



*Ilustración 20: Diagrama de clases subprototipo Gestionar personal*

En caso de que desee observar un listado de todos los trabajadores que tiene el director, se implementa la actividad “VerTrabajadoresActivity”. Esta actividad muestra los datos importantes de un trabajador, como son el nombre, apellidos y nombre de usuario.

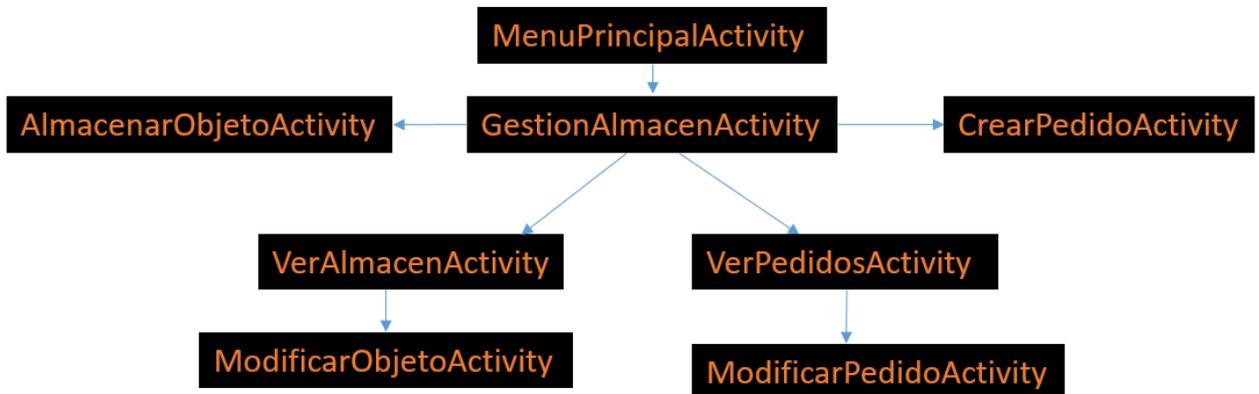
El director puede eliminar o modificar cualquier dato de sus trabajadores. Por ello, si pulsa sobre el nombre de usuario de un trabajador, se le muestran las opciones de eliminar o modificar. La acción de eliminar se controla en esta propia actividad. Pero la opción de modificar los datos se controla en otra actividad.

Esta acción la controla la actividad “ModificarDatosActivity”. Como se ha mencionado anteriormente, esta actividad permite la actualización de cualquier dato de los usuarios, con lo que se le pasa el usuario a modificar.

La última acción que puede realizar en este subprototipo es observar las estadísticas que le han sido dadas a cada trabajador a la hora de entregar los pedidos. Para ello, se implementa la actividad “VerEstadisticasActivity”. Esta actividad le muestra un listado de todos sus trabajadores junto a las valoraciones medias que tiene cada uno de ellos.

- *Gestionar almacén*

Para este subprototipo también se implementa una actividad que gestione todas las opciones, teniendo en cuenta que se presenta la opción de la gestión de objetos con los cuales se crean los pedidos. Para ello, se implementan las siguientes actividades.



*Ilustración 21: Diagrama de clases subprototipo Gestionar almacén*

La actividad “GestionAlmacenActivity” controla la acción del usuario.

Si desea crear un nuevo, debe introducir todos los datos necesarios en la actividad “AlmacenarObjetoActivity”.

Una vez creados los objetos, puede observarlos con un listado en la actividad “VerAlmacenActivity”. Esta actividad le da la opción tanto de editar los datos del objeto como de eliminarlos. La funcionalidad de eliminarlos siempre se trata en la propia actividad. Pero la funcionalidad de editar un objeto se controla en otra actividad.

Esta actividad se denomina “ModificarObjetoActivity”. En este momento, se le muestran todos los datos referentes al objeto para su posible modificación y posterior actualización en la base de datos.

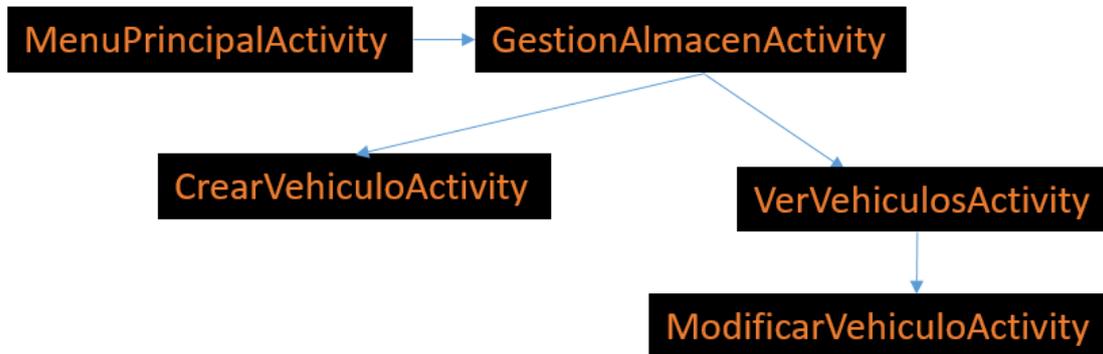
Una vez controlados los objetos, se deben crear los pedidos. Para crearlos se implementa la actividad “CrearPedidoActivity”. Esta actividad recoge de los datos introducidos del usuario el repartidor, los objetos a entregar, el destino, la hora de salida, etc. Una vez insertados todos los datos, se crea el pedido en la base de datos.

Una vez creados, podrá observarlos listados en la actividad “VerPedidosActivity”. En esta visualización, a parte de ver el identificador del pedido y el repartidor, visualizará el estado del pedido. Un pedido tiene tres estados: En espera, Iniciado o Finalizado. En espera quiere decir que el pedido aún no ha iniciado la ruta. Iniciado quiere decir que el repartidor se está dirigiendo al destino. Y por último, finalizado indica que el pedido ya ha sido entregado. Además, puede eliminar todos los que desee o modificar cualquier dato.

En caso de desear modificar algún valor, el usuario se encontraría en la actividad “ModificarPedidoActivity”. Esta actividad le muestra todos los datos del pedido para su modificación.

- *Gestionar vehículos*

Este último subprototipo referente al director, le permite la creación, edición y eliminación de los vehículos que posea. Para ello, se implementan las siguientes actividades.



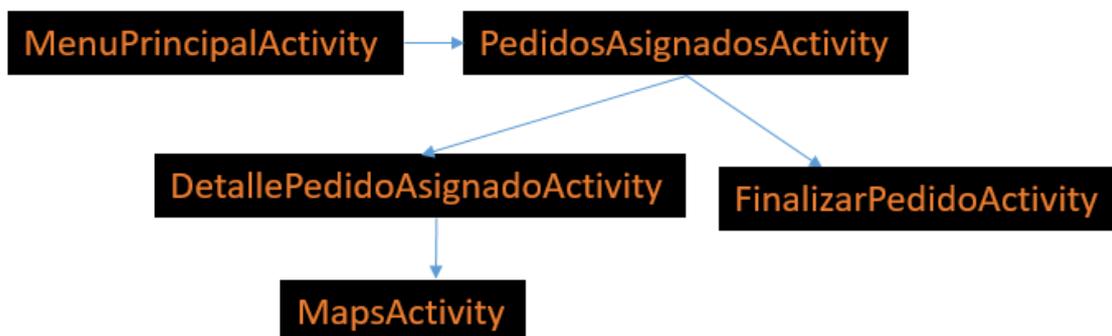
*Ilustración 22: Diagrama de clases subprototipo Gestionar vehículos*

En primer lugar se le mostraría al usuario la actividad Gestión Vehículos desde la cual elegirá que opción va a realizar. En la actividad Ver Vehículos observará todos los vehículos que posee y si desea editar alguno de ellos, le aparecerá la actividad Modificar Vehículos para su edición. Por otro lado, si el usuario en primer lugar seleccionaba la opción crear vehículo se le mostrará la actividad Crear Vehículo.

- *Pedidos asignados*

Por último, se presenta este subprototipo perteneciente únicamente al usuario trabajador. El trabajador mediante este prototipo puede observar los pedidos que le han sido asignados e iniciarlos. Una vez haya iniciado un pedido, la aplicación le dará la opción de utilizar la herramienta Google Maps para mostrarle la ruta a seguir. En cuanto el trabajador haya finalizado el pedido, deberá indicar que el pedido ha sido finalizado y una serie de datos como los kilómetros recorridos, la hora de llegada...

Para ello, se implementan las siguientes actividades.



*Ilustración 23: Diagrama de clases subprototipo Pedidos asignados*

La actividad “PedidosAsignadosActivity” le lista al trabajador todos los pedidos que le han sido asignados, junto con una serie de detalles característicos.

Cuando seleccione uno se le muestra la actividad “DetallesPedidosAsignadoActivity” en el que puede observar todos los detalles del pedido y su estado. En dicha actividad se le

da la opción de mostrarle la ruta usando la herramienta google maps (“MapsActivity”) y cuando haya llegado al destino, debe indicar que el pedido ha sido finalizado mediante la actividad “FinalizarPedidoActivity”.

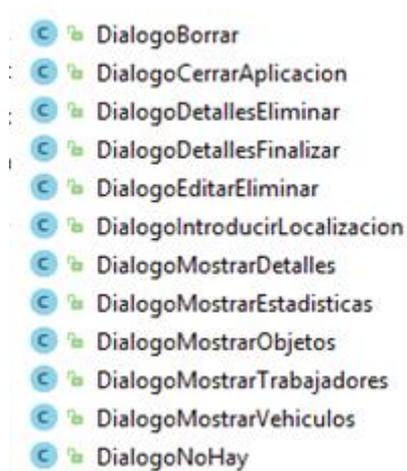
Una vez se han mostrado todos los prototipos con sus correspondientes diagramas de clases, se deben mostrar algunas actividades que no son visuales pero son tan importantes como las anteriores. Una de estas actividades es la que se corresponde con realizar la conexión a la base de datos.

## RemoteDatabase

*Ilustración 24: Clase relación con la BD*

Esta actividad permite la conexión con la base de datos y además recoge todos los parámetros que son necesarios para realizar las consultas, actualizaciones o borrados de datos en la base de datos.

Por último, los diálogos son las pequeñas pantallas en las que se le da la opción al usuario de que tarea realizar o de introducir algún dato. En esta aplicación se han desarrollado los siguientes diálogos y todos ellos siguen el mismo esquema de código.



*Ilustración 25: Actividades que representan los diálogos*

Con toda esta información, se procede al inicio de la implementación de la aplicación, detallando en el siguiente apartado todos los problemas que ha habido y como se han solucionado, las decisiones que se han tomado y sus motivos...



## 6-. Desarrollo

En este apartado se va a indicar como se ha desarrollado este proyecto y los problemas más destacados que han surgido. Además, se indican que elecciones se han optado a implementar y su motivo, para mayor facilidad a la hora de la comprensión.

Por ello, se ha dividido este apartado en la explicación del uso entre las Actividades y los Fragments, el desarrollo de la base de datos remota, el desarrollo de las interfaces, la elección del uso entre las listas y las tablas, el desarrollo de la aplicación multidioma y por último, el uso de la API de Google Maps.

- Actividades vs Fragments

En primer lugar se va a describir que son estos dos objetos y finalmente se contará que elecciones se han realizado con ellos.

Una actividad en Android es como una clase en Java. Consta de sus atributos, sus métodos... pero a la hora de realizar llamadas entre Actividades es donde radica la diferencia. Para ir de una actividad a otra actividad se debe crear un objeto tipo Intent en el que se indica la actividad origen y destino. A este objeto se le puede introducir datos en forma de Hash<K,V>, es decir, indicándole una clave o identificador del dato a almacenar (K), y el dato (V).

```
Intent i = new Intent( packageContext: VerPedidosActivity.this, ModificarPedidoActivity.class);
i.putExtra( name: "pedido", pedido);
startActivity(i);
```

Ilustración 26: Llamada a una actividad con parámetros

Todas las actividades constan de un método denominado *onCreate()* que se corresponde con una constructora en Java. Este método inicializa la actividad y le asigna una interfaz. Las interfaces se detallan en el siguiente punto indicando también las elecciones finales respecto a estas. En caso de haber introducido algún valor en el Intent, se recogen estos valores en el método *onCreate()* de la siguiente manera.

```
43
44 @Override
45 protected void onCreate(Bundle savedInstanceState) {
46     setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE); //poner solo en vertical
47     super.onCreate(savedInstanceState);
48     setContentView(R.layout.activity_modificar_pedido);
49     pedido = getIntent().getExtras().getString( key: "pedido");
```

Ilustración 27: Recogida de datos en una llamada

Los objetos Intent permiten el envío de datos tipo String, Integer, Boolean, CharSequence, ArrayList y Array de bytes principalmente. Debido al limitado tipo de archivos y que no permite transferencia de datos con tamaño muy grande, se deben transferir estos datos en formato de bytes. Para ello, antes de realizar la llamada se convierten los datos en array de bytes y se reconvierten al recoger los datos.

Una vez explicado todo lo correspondiente a las actividades, explicamos los Fragments. Aunque como se puede observar tengan nombres diferentes, el funcionamiento y el tratamiento son muy similares. Los Fragments se implementan sobre una actividad para mantener el contenido de esta. Para una mayor visualización, se muestra la siguiente imagen.

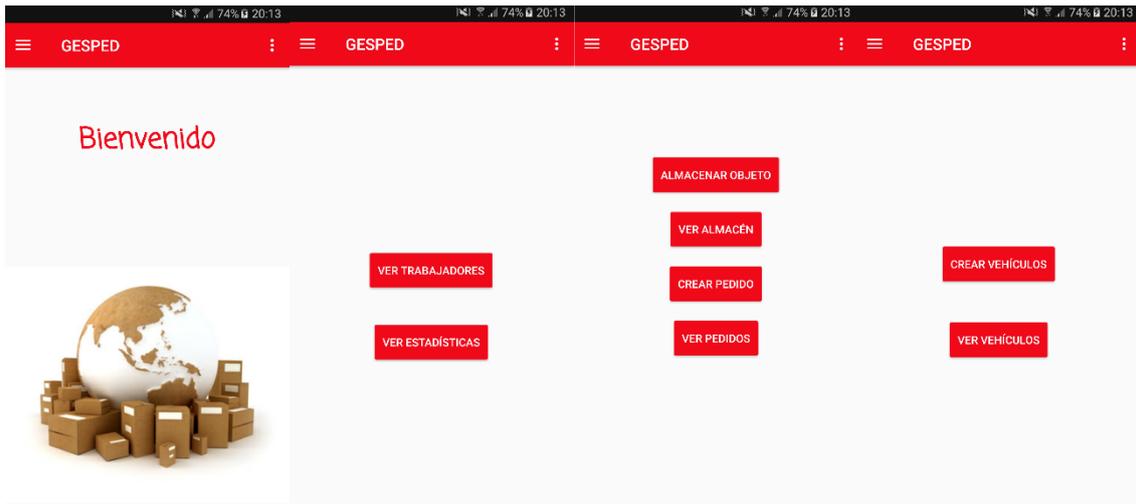


Ilustración 28: Fragments

Como se puede observar, el menú que da la bienvenida contiene un menú desplegable y los tres puntos con la opción de cerrar sesión. Para mantener menú y esta opción de la actividad de bienvenida, se implementan los Fragments. De tal manera, los Fragments de gestión contienen estos aspectos aunque la pantalla sea diferente al contener los botones.

El fin es mantener aspectos de la Actividad “padre” en los Fragments “hijos”. Se ha explicado esta diferencia debido a que se ha encontrado en diversos momentos con la duda de donde implementar los Fragments y cómo. Tras investigar cómo implementar y posibles aspectos a la hora de la mejora, se decidió crear tres Fragments correspondientes a las gestiones para que el usuario pueda navegar con mayor facilidad entre estos, manteniendo el menú y la opción de cerrar sesión.

Por otro lado, se ha utilizado la clase Fragment para implementar diálogos. Los diálogos son un tipo de clase que extienden de DialogFragment. Un diálogo es una alerta en el que se muestra un mensaje, un listado o incluso campos de textos a rellenar como los siguientes.

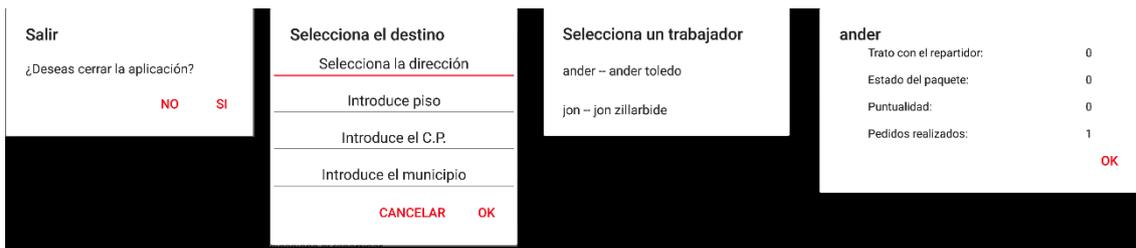


Ilustración 29: Diálogos

El objetivo del uso de estos diálogos es informar al usuario de las acciones que realizará y pedirle datos de forma sencilla sin tener que navegar entre muchas pantallas. Además, se ha tenido que elegir en qué casos se desean mostrar estos diálogos o únicamente un mensaje de información. Estos mensajes de información se denominan Toast, en los que se muestra que acción ha realizado el usuario. Finalmente, se ha optado por mostrar diálogos informando al usuario de que tarea va a realizar, y Toast informando de que tarea acaba de realizar. La forma de llamar a estos objetos es de la siguiente manera:

```
DialogoDetallesEliminar dialogo = new DialogoDetallesEliminar();
dialogo.show(getFragmentManager(), tag: "etiqueta");

Toast.makeText(getApplicationContext(), R.string.pedidoEliminado, Toast.LENGTH_SHORT).show();
```

Ilustración 30: Llamadas a diálogos y toast

Y la visualización de los objetos Toast es la siguiente:



Ilustración 31: Toast

- Base de datos remota

En primer lugar, se ha debido de investigar que detalles caracterizan a los pedidos para su tratamiento y control. Una vez tenida la idea de que información se debe almacenar de los pedidos, se creó la base de datos. Para la creación de esta, se optó por crearla en un lugar remoto, es decir, no crearla en el dispositivo local donde se instalaría la aplicación.

Se optó por implementarla de esta manera para reducir el tamaño de la aplicación, pero sobre todo, porque la aplicación estará en diferentes dispositivos y todos ellos deben estar conectados a la misma base de datos.

La base de datos se creó en un servidor con el cual se han realizado constantemente peticiones para obtener, actualizar, insertar y eliminar datos. Estas peticiones se implementan en archivos *php*.

*PHP* es un lenguaje de programación de código abierto, muy conocido debido a que se puede implementar tanto en desarrollos Web como en códigos HTML. La gran diferencia es que *php* se ejecuta en el servidor en vez de en el cliente. Este lenguaje genera un HTML que es enviado al cliente para que ejecute un script. [7]

Todo archivo PHP se divide en tres partes. La primera de ellas es la recogida de los parámetros de entrada. La segunda de ellas es realizar la conexión con la base de datos. Por último, está la parte de que acción debe hacer la petición PHP. Esto es, una obtención de datos, una actualización de datos o borrado de registros de la base de datos.

Por otro lado, para trabajar con base de datos remota, se necesita una clase en el código. Esta clase se ha denominado "RemoteDatabase" que extiende de AsyncTask<>. Respecto a la clase AsyncTask, es una clase que nos permite realizar progresos en paralelo. Durante nuestra carrera hemos conocido estas clases como *Threads*. Ha sido a la hora de implementar en Android cuando hemos aprendido a utilizar estas clases. En nuestro caso, únicamente la utilizamos para realizar nuestras peticiones en segundo plano mientras se le muestran al usuario diferentes pantallas. En caso de que nuestra tarea que se esté realizando en segundo plano tarda más tiempo de lo debido, salta el siguiente mensaje, con el cual estaremos todos ya familiarizados.

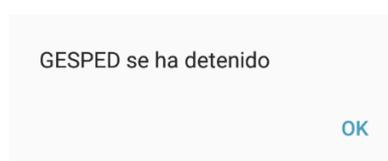


Ilustración 32: Aplicación detenida

La constructora de la clase "RemoteDatabase" y la llamada sería la siguiente:

```
public class RemoteDatabase extends AsyncTask<String, Void, String> {  
  
    // Constructor:  
    public RemoteDatabase(Context context, String reqId, int progressBarId, String [] datos) {
```

Ilustración 33: Código constructora conexión BD

```
String[] datos = new String[3];  
datos[0] = codObjeto;  
datos[1] = descripcion;  
datos[2] = user;  
RemoteDatabase http = new RemoteDatabase( context: this, reqId: "almacenarObjeto", progressBarId: -1, datos);  
http.execute();
```

Ilustración 34: Llamada a clase para realizar conexión BD

Como se puede observar, se le pase el contexto en el que se encuentra, el identificador del archivo *php* en el que se realiza la petición (*reqId*), si se desea que se muestre una barra de progreso mientras se realiza la petición y por último, los datos que se necesitan para realizar correctamente la petición.

- Interfaces

Una interfaz es una pantalla donde se pueden visualizar todo el contenido que se ha añadido (botones, textos, listados...). Cada actividad tiene asociada una interfaz, siendo la que trata todos los aspectos que contiene la interfaz.

Además, se ha elegido el sentido de la pantalla dependiendo del contenido, para que el usuario tenga una mayor visualización.

Para controlar los botones o los textos donde puede pinchar el usuario, se debe llamar al método *setOnClickListener()* y sobrescribir el método *onClick()* para que realice lo que se desea. Para ello, se debe identificar todos los objetos de la pantalla con un identificador, tal y como se muestra en la siguiente imagen.

```
Button botonAlmacenarObjeto = (Button) view.findViewById(R.id.botonAlmacenarObjeto);  
botonAlmacenarObjeto.setOnClickListener((v) -> {  
    Intent i = new Intent(getActivity(), AlmacenarObjetoActivity.class);  
    getActivity().startActivity(i);  
});
```

Ilustración 35: Acción al pulsar un botón

El problema que surgió a la hora de desarrollar las interfaces ha sido el diseño. Inicialmente, para implementar y probar la aplicación, se diseñaron las interfaces con el diseño estándar. Una vez se finalizaron las pruebas y se dio por validada la aplicación, se tuvo en cuenta el tiempo restante para mejorar el diseño. Pero se debe destacar que la mejor opción era dejar el diseño para el final, con el fin de mantener el objetivo principal, el correcto funcionamiento de la aplicación.

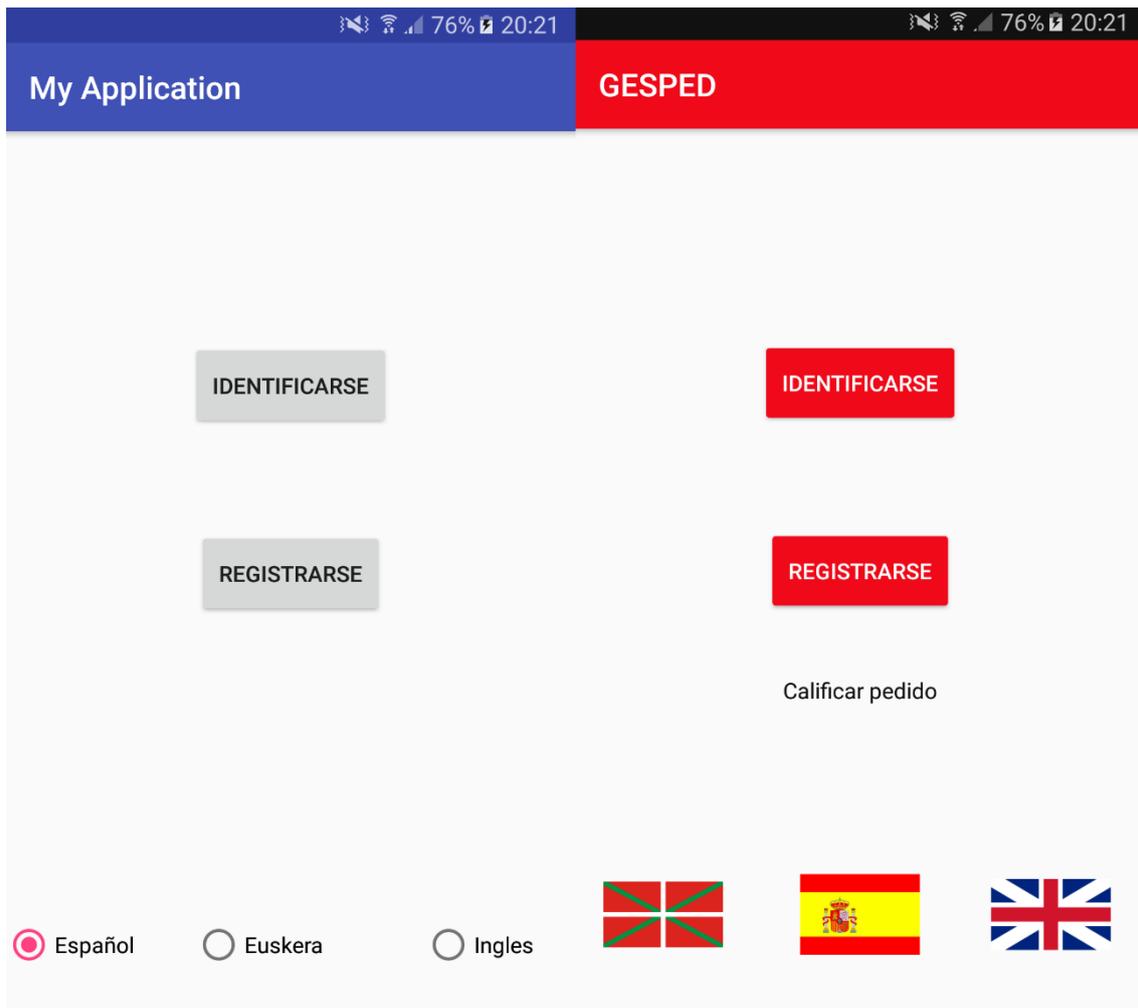


Ilustración 36: Diseño inicial y diseño final

- Listas vs Tablas

Este punto trata del aspecto visual pero también de cómo tratar los datos con mayor facilidad. Durante la implementación de la aplicación, entró la duda de cómo mostrar los datos para el usuario y de cómo recogerlos en caso de modificación, inserción o eliminación. Tras investigar maneras de mostrar datos, se llegó a la conclusión de optar entre mostrarlos en listas o mostrarlos mediante tablas. Al final se optó por las dos opciones, usando cada una de ellas en momentos precisos.

Respecto a las listas, se muestran al usuario únicamente cuando deba insertar un elemento de entre varios para realizar una acción. Por ejemplo, cuando deba crear un pedido y deba elegir que trabajador realizará el pedido entre todos sus trabajadores, o con que vehículo lo entregará de entre toda su flota de vehículos. El otro caso donde se muestra una lista es cuando el director debe añadir o modificar los objetos que se entregarán en un pedido, pinchando sobre algún elemento de la tabla.



Ilustración 37: Uso de listados

Por otro lado, se usan las tablas para mostrar información al usuario, tanto al director como al trabajador. Además, se ha optado por usar tablas a la hora de mostrar información ya que permitía añadir un título a las columnas, y así la información era más clara. En el caso del trabajador, se muestran todos los pedidos que tiene asignados junto a algunos datos descriptivos.

En el caso del director, se muestran las tablas cuando desee visualizar los trabajadores que tiene, las estadísticas de los pedidos de sus trabajadores, los pedidos que ha creado, los objetos almacenados y todos los vehículos que posee. Ambos usuarios pueden observar los detalles de cualquier tabla pulsando sobre el identificador de la fila correspondiente.

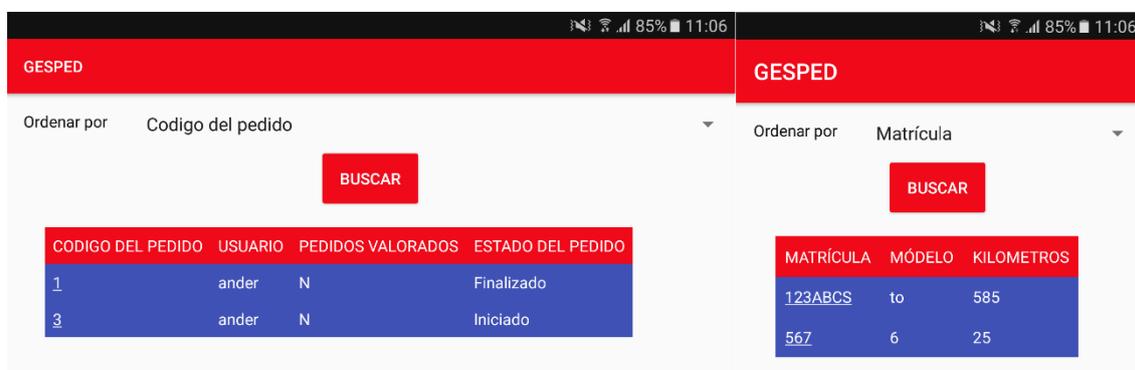


Ilustración 38: Uso de tablas

Para implementar las listas se usan los objetos ListView que permiten listar cualquier tipo de objeto, en este caso, se desea que se listen textos tipo String. Para ello, se debe crear un tipo ArrayList<> con el contenido que se mostrará y pasárselo como parámetro a una nueva clase adaptadora. Esta clase recibirá la actividad en la que se mostrará el listado y el texto que contendrá el listado. Además, se debe indicar que la clase extenderá de BaseAdapter. Se realiza esta acción debido a que se debe sobrescribir el método `getView()` para poder listar los textos.

Una vez esté la clase implementada, se realiza la llamada. También se debe indicar, que la lista tiene la opción de que al pinchar sobre un elemento, el usuario pueda borrarlo si lo desea.

Por último, para crear una tabla con los datos se usa el objeto `TableLayout`. En primer lugar, se introducen los encabezados de las columnas en la primera fila. Una vez se ha rellenado la primera fila, se rellenan el resto de las filas de igual manera pero pasándole los datos correspondientes.

Por ello, se deben crear tantos objetos tipo `TextView` como textos se vayan a poner por fila. Un objeto `TextView` es un tipo de objeto visual en el que se puede introducir cualquier texto. Por ejemplo, si la tabla tiene 4 columnas, se deben crear 4 `TextView`.

Por otro lado, como se dijo anteriormente, si el usuario pinchaba sobre la primera columna de alguna fila con datos, podía realizar alguna acción. Por ello, en el primer `TextView` se sobrescribe el método `onClick()`.

- Idioma

Como ya se ha indicado en la descripción del proyecto, la aplicación permitirá al usuario elegir el idioma entre castellano, euskera e inglés. Se decidió elaborar de tal manera la aplicación para enriquecerla y para que abarque un mayor tipo de clientes.

Para realizar esta característica, se debe hacer uso del archivo `strings.xml`. En este archivo se crean todos los identificadores de los textos y el texto correspondiente a cada idioma.



Key	Resource Folder	Untranslata...	Default Value	Basque (eu)	English (en)	Spanish (es)
datosErroneos	app\src\main\res	<input type="checkbox"/>	Datos erroneos	Datuak okerrak	Incorrect data	Datos erróneos
guardar	app\src\main\res	<input type="checkbox"/>	Guardar	Gorde	Save	Guardar
valoresActualizados	app\src\main\res	<input type="checkbox"/>	Se han actualizado los valores	Baloreak aktualizatu dira	Values are updated	Se han actualizado los valores
seleccionaldioma	app\src\main\res	<input type="checkbox"/>	Selecciona un idioma	Erabaki hizkuntza	Select a language	Selecciona un idioma
español	app\src\main\res	<input type="checkbox"/>	Español	Gaztelaina	Spanish	Español
euskera	app\src\main\res	<input type="checkbox"/>	Euskera	Euskara	Basque	Euskera
ingles	app\src\main\res	<input type="checkbox"/>	Inglés	Ingelesa	English	Inglés

Ilustración 39: Archivo `strings.xml`

Para hacer referencia al texto, se debe hacerlo de la siguiente manera: `R.string.identificadorTexto`. La aplicación por sí misma sabe qué tipo de texto coger al obtener el idioma actual de la aplicación.

Por otro lado, se le da la opción al usuario de modificar el idioma en la actividad Inicial. Se debe controlar que imagen ha seleccionado debido a que cada imagen se corresponde con un idioma. Para cambiar el idioma de la aplicación, se usa el siguiente método.

```
private void ponerIdioma() {
    Locale nuevot = new Locale(idioma);
    Locale.setDefault(nuevot);
    Configuration config = new Configuration();
    config.setLocale(nuevot);
    Resources res = getBaseContext().getResources();
    res.updateConfiguration(config, res.getDisplayMetrics());
    finish();
    startActivity(getIntent());
}
```

Ilustración 40: Código cambiar idioma

- Google Maps

Por último, el uso de la API de Google Maps fue el mayor problema que hubo y la tarea que más tiempo llevó. Se utilizó la actividad Google Maps para mostrar una ubicación o para mostrar una ruta a seguir entre dos puntos. Para realizar estas labores se creó dicha actividad y en el archivo correspondiente a la clave API se debe introducir la clave. Además, se debe introducir el permiso para usar el mapa y del uso de internet en la aplicación, en el archivo *AndroidManifest.xml*

Para obtener esta clave, hay que ir a las API's de Google y crear un proyecto para la aplicación. A continuación, se obtiene la clave y se otorga autorización a la aplicación.

Una vez realizadas estas tareas, en primer lugar se hará la tarea "sencilla": poner un marcador en el mapa. Esta tarea se da cuando el usuario introduce el lugar de su empresa o el destino del pedido. Para introducir este marcador en el mapa, se obtiene la longitud y la latitud del punto, y se añade al mapa con el correspondiente marcador de origen o destino.

```

Geocoder geo = new Geocoder(getApplicationContext());
try {
    List<Address> addressList = geo.getFromLocationName( locationName: nombreCalle+ " "+municipio, maxResults: 1);
    if(addressList.size()==0){
        salir = true;//no ha encontrado la ubicación
    }else{
        latitud = new LatLng(addressList.get(0).getLatitude(),addressList.get(0).getLongitude());
    }
} catch (IOException e) {
    e.printStackTrace();
}
mMap.addMarker(new MarkerOptions().position(latitud).title(marca));
mMap.moveCamera(CameraUpdateFactory.newLatLng(latitud));
CameraPosition cp = new CameraPosition.Builder().target(latitud).zoom(17).build();
mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cp));

```

Ilustración 41: Insertar una ubicación

Ahora se presenta la tarea "difícil", mostrar la ruta entre dos puntos. Insertar dos puntos en el mapa es realizar la labor anterior dos veces tanto para el origen como para el destino.

A la hora de mostrar la ruta, se daba la posibilidad de usar la clase *Polyline* la cual dibuja una recta entre dos puntos. Pero dibujaba una línea recta ignorando los edificios, calles... es decir, unía por ejemplo Barcelona con Roma atravesando el mar mediterráneo. De tal manera, se empezó a investigar y observar métodos de todas las clases relacionadas con mapas, y se encontró una forma de hacerlo.

Una vez obtenidas las coordenadas del origen y del destino se debe realizar una petición al servidor de Google Maps pasándole los correspondientes parámetros.

```

private String obtenerDireccionesURL(LatLng origen,LatLng destino){
    String paramOrigen = "origin="+origen.latitude+","+origen.longitude;
    String paramDestino = "destination="+destino.latitude+","+destino.longitude;
    String sensor = "sensor=false";
    String parametros = paramOrigen+"&"+paramDestino+"&"+sensor;
    String salida = "json";
    String url = "https://maps.googleapis.com/maps/api/directions/"+salida+"?"+parametros;
    return url;
}

```

Ilustración 42: Realizar petición a Google Maps

Como se puede observar en la imagen anterior, se indica el origen y el destino pasándole sus latitudes y longitudes. Respecto al parámetro "sensor", es una variable por defecto que se le

debe pasar a Google Maps, por ello se pone como *false*. Por último, se debe especificar el formato de los datos de la respuesta, en este caso, JSON. De tal manera, con estos parámetros se obtienen los puntos de la ruta óptima por donde el trabajador debería ir para entregar el pedido.

Un dato importante que se ha mencionado en apartados anteriores, es que los trabajadores realizarán sus pedidos en vehículos. De tal manera, surgió la duda de donde se indica dicho detalle ya que en la página de Google Maps da la opción de elegir ruta en vehículo, andando, en transporte público... Por defecto, la ruta se considera que es en vehículo personal y por ello no se detalla nada más.

Una vez obtenida la Url a la que hacer la petición, se ejecutan una serie de tareas en segundo plano. La primera de estas será ejecutar la Url y obtener el resultado. Para ejecutar la Url se usa la clase *HttpURLConnection*, que permite realizar conexiones de tipo HTTP teniendo una dirección Url.

Respecto al resultado, se recibe en formato JSON. Este archivo contendrá todos los datos respecto a la ruta, es decir, origen y destino, distancia... y como no, una serie de puntos que deben ser unidos para formar una línea que indique la ruta.

De tal manera, la siguiente tarea en segundo plano a implementar sería la unión de todos los puntos. Para ello, se empiezan poniendo los puntos origen y destino. Una vez estén situados, se implementa un bucle obteniendo todos los puntos hasta que el punto obtenido coincida con el destino. Por cada punto obtenido se une con el anterior. En el primer caso, se une con el origen. Pero en el resto de los casos, se debe almacenar el último punto colocado para unirlo con este.

Una vez implementados estos pasos, se ejecuta la actividad que mostraría el mapa.

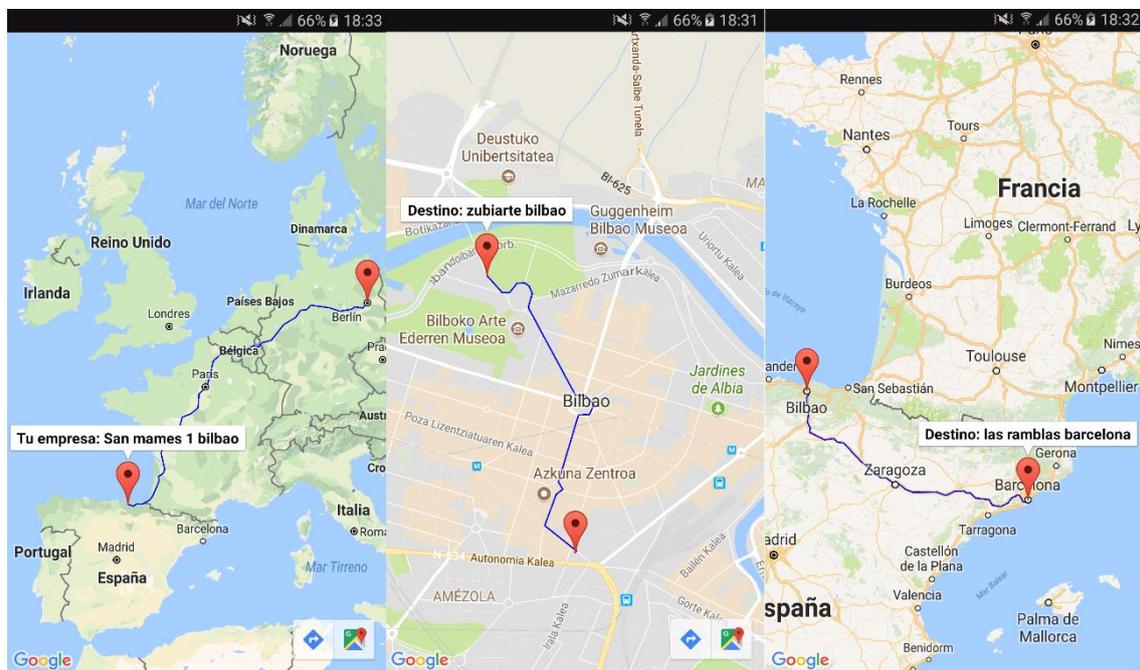


Ilustración 43: Rutas Google Maps

Como se puede observar, la línea indica por donde se debe ir para llegar al destino. Además, para que el trabajador se haga una idea del tiempo que le llevara la entrega o de la distancia que deberá recorrer, se muestra un Toast indicando la distancia. Como se ha comentado

anteriormente, este dato también se obtiene en el archivo JSON, pero en formato *metros*. Es decir, si la distancia es cercana, como por ejemplo “Zubiarte” tal y como se muestra en la imagen, el formato de la distancia en metros es adecuado. Pero ya en el caso de ir a “Las ramblas” o a “Berlín”, queda una distancia muy grande.

Por ello, se decidió cambiar el formato a kilómetros en el caso en el que la distancia sea muy grande.

## 7-. Verificación y evaluación

En este apartado se muestran todas las pruebas que se han realizado a la aplicación. Para ello, se crean diversas tablas por cada funcionalidad indicando el identificador de la prueba, una breve descripción de la prueba realizada, el resultado esperado, el resultado obtenido y el estado de la aplicación.

Respecto al estado de la aplicación, se indica si tras realizar la primera prueba la aplicación funcionaba correctamente (C) o incorrectamente (I). En caso de funcionamiento incorrecto, al final de la tabla se indica como se ha resuelto el problema para que en las siguientes pruebas el funcionamiento haya sido correcto.

Además, se debe indicar que las pruebas correspondientes a introducir un texto de longitud ilimitado no se han considerado debido a que en el código se ha limitado el texto a introducir. Esto es, los campos en los que puede introducir texto el usuario tienen texto limitado a una longitud. Esta longitud es la longitud que tiene el campo correspondiente en la base de datos, menos uno. Por ejemplo, el nombre de usuario tiene longitud máximo 50 en la base de datos. De tal manera, en la aplicación, el usuario puede introducir su nombre de usuario de longitud máxima 49.

Por otro lado, también relacionado con lo anterior, se ha limitado el tipo de texto que puede introducir el usuario. Es decir, si un campo en la base de datos es numérico, al introducir dicho dato en la aplicación se le mostrará un teclado únicamente con valores numéricos. Por ello, tampoco se ha considera como pruebas el tipo de texto introducido.

### 7.1-. Modificar idioma

Se considera que la aplicación inicialmente está en castellano.

Id	Descripción	Resultado esperado	Resultado obtenido	
1	Se pulsa sobre otro idioma que no sea el de por defecto, por ejemplo, euskera.	Se actualiza toda la aplicación a euskera y no se guarda en la pila de pantallas.	Se actualiza toda la aplicación a euskera y no se guarda en la pila de pantallas.	C
2	Se pulsa sobre otro idioma que no sea el de por defecto, por ejemplo, inglés.	Se actualiza toda la aplicación a inglés y no se mantiene en la pila de pantallas.	Se actualiza toda la aplicación a inglés y no se mantiene en la pila de pantallas.	C
3	Se pulsa sobre el idioma que está actualmente.	Se mantiene el idioma y no se guarda en la pila de pantallas.	Se mantiene el idioma y no se guarda en la pila de pantalla.	C
4	Al pulsar atrás se muestra el mensaje: "Pulsa otra vez si deseas salir".	Se muestra un Toast mostrando dicho mensaje.	Se muestra un Toast mostrando dicho mensaje.	C
5	Se pulsa de nuevo atrás una vez que se ha mostrado el mensaje anterior.	Se debe cerrar la aplicación.	Se debe cerrar la aplicación.	C

Tabla 7: Pruebas Modificar idioma

Como se puede observar, nos referimos al término "pila de pantallas". Se explica este término a continuación debido a que en más pruebas se hace uso de él. La "pila de pantallas" es una pila estilo LIFO (*Last In, First Out*). Cada vez que se navega entre pantallas, dichas pantallas se añaden

a la pila y la manera que tiene el usuario de quitar pantallas de la pila es pulsando el botón atrás o cuando se finaliza una tarea, se retorna a la pantalla anterior.

## 7.2-. Calificar pedido

Id	Descripción	Resultado esperado	Resultado obtenido	
6	El usuario deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
7	El usuario introduce un nombre de usuario de repartidor que no existe.	Se le muestra un Toast indicando que dicho nombre de usuario no existe.	Se le muestra un Toast indicando que dicho nombre de usuario no existe.	C
8	El usuario introduce un código de pedido que no existe.	Se le muestra un Toast indicando que el código de pedido no existe.	Se le muestra un Toast indicando que el código de pedido no existe.	C
9	El usuario intenta calificar un pedido que no ha sido finalizado, es decir, está en espera.	Se le muestra un Toast indicando que el pedido aún no ha sido finalizado.	Se le muestra un Toast indicando que el pedido aún no ha sido finalizado.	C
10	El usuario intenta calificar un pedido que no ha sido finalizado, es decir, está iniciado.	Se le muestra un Toast indicando que el pedido aún no ha sido finalizado.	Se le muestra un Toast indicando que el pedido aún no ha sido finalizado.	C
11	El usuario intenta calificar un pedido que ya ha sido valorado.	Se le muestra un Toast indicando que el pedido ya ha sido valorado.	Se le muestra un Toast indicando que el pedido ya ha sido valorado.	C
12	El usuario califica correctamente el pedido.	Se actualizan los valores en la base de datos, se le muestra un Toast indicando pedido valorado y se vuelve a la pantalla anterior.	Se actualizaban incorrectamente algunos valores en la base de datos. El resto de acciones las realizaba correctamente.	I
13	El usuario pulsa el botón atrás.	Se vuelve a la pantalla anterior.	Se cerraba la aplicación.	I

Tabla 8: Pruebas Calificar pedido

- **Prueba 12:** Los valores correspondientes a las estadísticas de pedidos del repartidor no se actualizaban correctamente. Esto se debe a que la fórmula matemática para realizar las medias no tenía en cuenta decimales y no guardaba correctamente dichos valores. Se arregló este error redondeando el valor y eliminando los decimales.
- **Prueba 13:** Tras la primera prueba no se tuvo en cuenta la navegación entre pantallas y el uso del botón atrás. Se solucionó controlando la pila de pantallas y sin eliminar la pantalla anterior de dicha pila. Por ello, de ahora en adelante se indica este tipo de pruebas como correctas, al haberlas solucionado en este punto.

### 7.3-. Identificarse

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>14</b>	El usuario deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
<b>15</b>	El usuario introduce un nombre de usuario que no existe.	Se le muestra un Toast indicando nombre de usuario inexistente.	Se le muestra un Toast indicando nombre de usuario inexistente.	C
<b>16</b>	El usuario introduce una contraseña incorrecta.	Se le muestra un Toast indicando contraseña incorrecta.	Se le muestra un Toast indicando contraseña incorrecta.	C
<b>17</b>	El usuario introduce datos correspondientes a un usuario creado.	Se almacenan sus datos en el archivo sesión. Si el usuario es jefe accede al menú correspondiente al jefe. En caso contrario, accede al menú correspondiente al trabajador.	Se almacenan sus datos en el archivo sesión. Si el usuario es jefe accede al menú correspondiente al jefe. En caso contrario, accede al menú correspondiente al trabajador.	C
<b>18</b>	El usuario pulsa el botón atrás.	Se vuelve a la pantalla anterior	Se vuelve a la pantalla anterior	C

Tabla 9: Pruebas Identificarse

### 7.4-. Recuperar contraseña

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>19</b>	El usuario deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
<b>20</b>	El usuario introduce un nombre de usuario que no existe.	Se le muestra un Toast indicando datos erróneos.	Se le muestra un Toast indicando datos erróneos.	C
<b>21</b>	El usuario introduce un email que no se corresponde al nombre de usuario.	Se le muestra un Toast indicando datos erróneos.	Se le muestra un Toast indicando datos erróneos.	C
<b>22</b>	El usuario introduce datos correspondientes a un usuario creado.	Se le envía una notificación mostrando la contraseña del usuario y volvemos a la pantalla anterior.	Se le envía una notificación mostrando la contraseña del usuario y volvemos a la pantalla anterior.	C
<b>23</b>	El usuario pulsa el botón atrás.	Se vuelve a la pantalla anterior	Se vuelve a la pantalla anterior	C

Tabla 10: Pruebas Recuperar contraseña

## 7.5-. Registrarse como director

Id	Descripción	Resultado esperado	Resultado obtenido	
24	El usuario deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
25	El usuario introduce un nombre de usuario que ya existe.	Se le muestra un Toast indicando que el nombre de usuario está en uso.	Se le muestra un Toast indicando que el nombre de usuario está en uso.	C
26	El usuario introduce contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	C
27	El usuario introduce una dirección que no existe.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	Se detenía la aplicación.	I
28	El usuario deja la calle o el piso sin especificar, o deja alguno de los cuatro campos con los valores del enunciado del dialogo.	Se le muestra un Toast indicando que faltan campos en la dirección.	Se quedaban en blanco los campos calle o piso. Y en el segundo caso, al pasarle al mapa los enunciados del dialogo se detenía la aplicación.	I
29	El usuario introduce una dirección valida.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	C
30	Al pulsar volver cuando está visualizando el mapa.	Retorna a la pantalla del registro manteniendo todos los valores.	Retorna a la pantalla del registro manteniendo todos los valores.	C
31	El usuario consigue registrarse.	Se crea un usuario en la base de datos, se le muestra un Toast indicando director registrado, se almacenan sus datos en el archivo sesión y accede al menú del jefe.	Se crea un usuario en la base de datos, se le muestra un Toast indicando director registrado, se almacenan sus datos en el archivo sesión y accede al menú del jefe.	C
32	El usuario pulsa el botón atrás.	Se vuelve a la pantalla anterior	Se vuelve a la pantalla anterior	C

Tabla 11: Pruebas Registrarse como director

- **Prueba 27:** No se había tenido en cuenta que el usuario introdujera un lugar que no existe, por ello al abrir el mapa la aplicación se detenía. Se solucionó creando un método que comprobase las coordenadas antes de mostrarlas en el mapa. De tal manera, de aquí en adelante este error se dará como arreglado.
- **Prueba 28:** El usuario podía dejar los campos calle o piso vacíos si lo deseaba. En tal caso, se avisa al usuario con un Toast indicándole que quedaban campos por rellenar en la dirección. En el segundo caso, en los campos se mostraba “Introduce piso”, “Introduce el municipio”... Si el usuario le daba al OK sin introducir ningún dato, la aplicación se

detenía. Se solucionó comprobando cada campo con el enunciado y si coincidían, se le avisaba que quedaban campos por rellenar en la dirección con un Toast.

#### 7.6-. Cerrar sesión

Id	Descripción	Resultado esperado	Resultado obtenido	
33	El usuario pulsa sobre la opción cerrar sesión.	Se eliminan los datos almacenados en el archivo sesión y se le muestra la pantalla inicial.	Se eliminan los datos almacenados en el archivo sesión y se le muestra la pantalla inicial.	C
34	Tras cerrar sesión el usuario, pinchaba sobre el botón atrás.	Se le muestra el Toast "Pulsa otra vez si deseas salir" tal y como se indica en la prueba 4.	Volvió al menú principal.	I
35	El usuario abre la aplicación tras haber cerrado sesión la última vez.	Se le muestra la pantalla Inicial.	Se le muestra la pantalla Inicial.	C
36	El usuario abre la aplicación sin haber cerrado sesión la última vez.	Se le muestra el menú correspondiente al usuario.	Se le muestra el menú correspondiente al usuario.	C
37	Abrir la aplicación con sesión iniciada con un usuario que ha sido eliminado.	Se le muestra una alerta indicando que el usuario se ha eliminado y se cierra sesión.	La aplicación se detiene	I

Tabla 12: Pruebas Cerrar sesión

- **Prueba 34:** No se controló correctamente la eliminación de la pantalla del menú y por ello volvía a la pantalla anterior. Se solucionó añadiendo la línea de comandos "this.finish()", la cual elimina la pantalla actual de la pila.
- **Prueba 37:** Cuando se inicia el menú con un usuario trabajador no se tuvo en cuenta si el jefe había eliminado dicho trabajador anteriormente. Se solucionó en la constructora de la actividad poniendo un método que compruebe el usuario. En caso de que haya sido eliminado, se le muestra una alerta indicando usuario eliminado y vuelve a la pantalla inicial.

## 7.7-. Modificar datos personales

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>38</b>	Si el usuario es jefe, le deben aparecer todos los datos personales juntos con los de su empresa.	Se le muestra todos los campos.	Se le muestra todos los campos.	C
<b>39</b>	Si el usuario es trabajador le debe aparecer únicamente los datos personales	Se le muestra solo los campos de datos personales.	Se le muestra solo los campos de datos personales.	C
<b>40</b>	El usuario deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
<b>41</b>	El usuario introduce contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	C
<b>42</b>	El usuario introduce una dirección para la empresa que no existe.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	C
<b>43</b>	El usuario introduce una dirección valida.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	C
<b>44</b>	Al pulsar volver cuando está visualizando el mapa.	Retorna a la pantalla de modificación manteniendo todos los valores.	Retorna a la pantalla de modificación manteniendo todos los valores.	C
<b>45</b>	El usuario modifica sus datos	Se actualizan los datos en la base de datos, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior.	Se actualizan los datos en la base de datos, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior.	C
<b>46</b>	El usuario pulsa el botón atrás.	Se vuelve a la pantalla anterior.	Se vuelve a la pantalla anterior.	C

Tabla 13: Pruebas Modificar datos personales

## 7.8-. Registrar trabajador

Id	Descripción	Resultado esperado	Resultado obtenido	
47	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
48	El director introduce un nombre de usuario que ya existe.	Se le muestra un Toast indicando que el nombre de usuario está en uso.	Se le muestra un Toast indicando que el nombre de usuario está en uso.	C
49	El director introduce contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	C
50	El director registra el trabajador.	Se crea un usuario tipo trabajador asociado al jefe en la base de datos, se le muestra un Toast indicando trabajador creado y vuelve a la pantalla anterior.	Se crea un usuario tipo trabajador asociado al jefe en la base de datos, se le muestra un Toast indicando trabajador creado y vuelve a la pantalla anterior.	C
51	El director pulsa el botón atrás.	Se vuelve a la pantalla anterior.	Se vuelve a la pantalla anterior.	C

Tabla 14: Pruebas Registrar trabajador

## 7.9-. Ver sus trabajadores

Id	Descripción	Resultado esperado	Resultado obtenido	
52	El director no tiene trabajadores	Se le muestra una alerta indicando que no tiene trabajadores.	Se le muestra una alerta indicando que no tiene trabajadores.	C
53	El director tiene trabajadores.	Se le muestran todos sus trabajadores ordenados según el detalle que haya seleccionado.	Se le muestran todos sus trabajadores ordenados según el detalle que haya seleccionado.	C
54	El director da al botón buscar teniendo trabajadores.	Se le muestra la tabla de todos sus trabajadores.	Se le muestra la tabla de todos sus trabajadores.	C
55	El director da al botón buscar, teniendo trabajadores, cuando ya se le mostraban todos sus trabajadores	Se actualiza la tabla.	Se mostraban los trabajadores debajo de la tabla inicial, es decir, se mostraban tantas tablas de trabajadores como veces le daba al botón buscar.	I
56	El director pincha sobre el nombre de usuario de algún trabajador suyo.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	C
57	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del personal.	Se vuelve a la pantalla gestión del personal.	C

Tabla 15: Pruebas Ver sus trabajadores

- **Prueba 55:** Cada vez que le daba al botón Buscar cuando ya estaba observando la tabla de sus trabajadores, se mostraba otra tabla actualizada de sus trabajadores. Es decir, mostraba tantas tablas como veces le daba a buscar. Se solucionó eliminando el contenido de la tabla cada vez que le daba a buscar para mostrarla de nuevo actualizada. Además, este error se dio en todas las pantallas en las que se observaba una tabla con datos, por ello, una vez que se solucionó en este punto, en las próximas tablas se dará como corregido.

#### 7.10-. Modificar datos de sus trabajadores

Id	Descripción	Resultado esperado	Resultado obtenido	
58	Le deben aparecer únicamente los campos personales del trabajador.	Se le muestran los datos personales del trabajador.	Se le muestran los datos personales del trabajador.	C
59	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
60	El director introduce contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	Se le muestra un Toast indicando contraseñas diferentes.	C
61	El director modifica los datos.	Se actualizan los datos en la base de datos, se le muestra un Toast indicando valores actualizados, se vuelve a la pantalla anterior donde no se observa la tabla de trabajadores.	Se actualizan los datos en la base de datos, se le muestra un Toast indicando valores actualizados, se vuelve a la pantalla anterior donde no se observa la tabla de trabajadores.	C
62	El director pulsa el botón atrás.	Se vuelve a la pantalla anterior donde no se observa la tabla de trabajadores.	Se vuelve a la pantalla anterior donde no se observa la tabla de trabajadores.	C

Tabla 16: Pruebas Modificar datos de sus trabajadores

#### 7.11-. Eliminar trabajadores

Id	Descripción	Resultado esperado	Resultado obtenido	
63	El director elimina al trabajador.	Se elimina el trabajador, sus pedidos y el contenido de sus pedidos de la base de datos. Además, se le muestra un Toast indicando trabajador eliminado y vuelve a la pantalla anterior donde no observa la tabla de trabajadores.	Se elimina el trabajador, sus pedidos y el contenido de sus pedidos de la base de datos. Además, se le muestra un Toast indicando trabajador eliminado y vuelve a la pantalla anterior donde no observa la tabla de trabajadores.	C

Tabla 17: Pruebas Eliminar trabajadores

### 7.12-. Ver estadísticas de pedidos

Id	Descripción	Resultado esperado	Resultado obtenido	
64	El director no tiene trabajadores.	Se le muestra una alerta indicando que no tiene trabajadores.	Se le muestra una alerta indicando que no tiene trabajadores.	C
65	El director tiene trabajadores.	Se le muestran todos sus trabajadores ordenados según el desplegable y el check seleccionado.	Se le muestran todos sus trabajadores ordenados según el desplegable y el check seleccionado.	C
66	El director da al botón buscar teniendo trabajadores.	Se le muestra la tabla de todos sus trabajadores y sus estadísticas.	Se le muestra la tabla de todos sus trabajadores y sus estadísticas.	C
67	El director da al botón buscar, teniendo trabajadores, cuando ya se le mostraban todos sus trabajadores.	Se actualiza la tabla.	Se actualiza la tabla.	C
68	El director pincha sobre el nombre de usuario de algún trabajador suyo.	Se le muestra un dialogo mostrándole las estadísticas de sus pedidos.	Se le muestra un dialogo mostrándole las estadísticas de sus pedidos.	C
69	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del personal.	Se vuelve a la pantalla gestión del personal.	C

Tabla 18: Pruebas Ver estadísticas de pedidos

### 7.13-. Crear vehículos

Id	Descripción	Resultado esperado	Resultado obtenido	
70	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
71	Introduce una matrícula de un vehículo ya creado.	Se le muestra un Toast indicando que la matrícula está en uso.	Se le muestra un Toast indicando que la matrícula está en uso.	C
72	El director crea el vehículo.	Se crea el vehículo en la base de datos, se le muestra un Toast indicando vehículo creado y vuelve a la pantalla gestión de vehículos.	Se crea el vehículo en la base de datos, se le muestra un Toast indicando vehículo creado y vuelve a la pantalla gestión de vehículos.	C
73	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión de vehículos.	Se vuelve a la pantalla gestión de vehículos.	C

Tabla 19: Pruebas Crear vehículos

#### 7.14-. Ver vehículos

Id	Descripción	Resultado esperado	Resultado obtenido	
74	El director no tiene vehículos.	Se le muestra una alerta indicando que no tiene vehículos.	Se le muestra una alerta indicando que no tiene vehículos.	C
75	El director tiene vehículos.	Se le muestran todos sus vehículos ordenados según el desplegable.	Se le muestran todos sus vehículos ordenados según el desplegable.	C
76	El director le da al botón buscar teniendo vehículos.	Se le muestran todos los vehículos que tiene.	Se le muestran todos los vehículos que tiene.	C
77	El director da al botón buscar, teniendo vehículos, cuando ya se le mostraban todos sus vehículos.	Se actualiza la tabla.	Se actualiza la tabla.	C
78	El director pincha sobre la matrícula de un vehículo.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	C
79	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión de vehículos.	Se vuelve a la pantalla gestión de vehículos.	C

Tabla 20: Pruebas Ver vehículos

#### 7.15-. Modificar vehículos

Id	Descripción	Resultado esperado	Resultado obtenido	
80	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
81	El director modifica el vehículo.	Se modifican los datos del vehículo y en caso de estar asociados a un pedido, se modifican todos los datos pertinentes en la base de datos. Además, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior donde no observa la tabla de vehículos.	Se modifican los datos del vehículo y en caso de estar asociados a un pedido, se modifican todos los datos pertinentes en la base de datos. Además, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior donde no observa la tabla de vehículos.	C
82	El director pulsa el botón atrás.	Se vuelve a la pantalla anterior donde no se observa la tabla de vehículos.	Se vuelve a la pantalla anterior donde no se observa la tabla de vehículos.	C

Tabla 21: Pruebas Modificar vehículos

### 7.16-. Eliminar vehículos

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>83</b>	El director elimina el vehículo.	Se elimina el vehículo y los pedidos a los que está asignado se quedan sin vehículo. Además, se le muestra un Toast indicando vehículo eliminado y vuelve a la pantalla anterior donde no observa la tabla de vehículos.	Se elimina el vehículo y los pedidos a los que está asignado se quedan sin vehículo. Además, se le muestra un Toast indicando vehículo eliminado y vuelve a la pantalla anterior donde no observa la tabla de vehículos.	C

Tabla 22: Pruebas Eliminar vehículos

### 7.17-. Almacenar objetos

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>84</b>	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
<b>85</b>	El código del objeto ya está en uso por el director	Se le muestra un Toast indicando código del objeto en uso.	Se le muestra un Toast indicando código del objeto en uso.	C
<b>86</b>	Almacena el objeto.	Se almacena el objeto en la base de datos, se le muestra un Toast indicando objeto almacenado y vuelve a la pantalla gestión del almacén.	Se almacena el objeto en la base de datos, se le muestra un Toast indicando objeto almacenado y vuelve a la pantalla gestión del almacén.	C
<b>87</b>	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del almacén.	Se vuelve a la pantalla gestión del almacén.	C

Tabla 23: Pruebas Almacenar objetos

### 7.18-. Ver el almacén

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
88	El director no tiene objetos.	Se le muestra una alerta indicando que no tiene objetos.	Se le muestra una alerta indicando que no tiene objetos.	C
89	El director tiene objetos.	Se le muestran todos sus objetos ordenados según el detalle que haya seleccionado.	Se le muestran todos sus objetos ordenados según el detalle que haya seleccionado.	C
90	El director da al botón buscar teniendo objetos.	Se le muestra la tabla de todos sus objetos.	Se le muestra la tabla de todos sus objetos.	C
91	El director da al botón buscar, teniendo objetos, cuando ya se le mostraban todos sus objetos.	Se actualiza la tabla.	Se actualiza la tabla.	C
92	El director pincha sobre el código de algún objeto.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	C
93	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del almacén.	Se vuelve a la pantalla gestión del almacén.	C

Tabla 24: Pruebas Ver el almacén

### 7.19-. Editar objetos

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
94	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
95	El director modifica el objeto.	Se modifican los datos en la base de datos, se le muestra un Toast indicando campos vacíos y al volver a la pantalla anterior no le aparece la tabla de los objetos.	Se modifican los datos en la base de datos, se le muestra un Toast indicando campos vacíos y al volver a la pantalla anterior no le aparece la tabla de los objetos.	C
96	El director pulsa el botón atrás.	Se vuelve a la pantalla anterior donde no se observa la tabla de objetos.	Se vuelve a la pantalla anterior donde no se observa la tabla de objetos.	C

Tabla 25: Pruebas Editar objetos

## 7.20-. Eliminar objetos

<b>Id</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	
<b>97</b>	El director elimina el objeto.	Se elimina el objeto de la base de datos y de todos los pedidos que lo contenían. Además, se muestra un Toast indicando objeto eliminado y vuelve a la pantalla anterior donde no se observa la tabla de objetos.	Se elimina el objeto de la base de datos y de todos los pedidos que lo contenían. Además, se muestra un Toast indicando objeto eliminado y vuelve a la pantalla anterior donde no se observa la tabla de objetos.	C

*Tabla 26: Pruebas Eliminar objetos*

## 7.21-. Crear pedidos

Id	Descripción	Resultado esperado	Resultado obtenido	
98	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	Se le muestra un Toast indicando campos vacíos.	C
99	El código del pedido está en uso.	Se le muestra un Toast indicando código del pedido en uso.	Se le muestra un Toast indicando código del pedido en uso.	C
100	Introduce un destino inexistente.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	Se detenía la aplicación.	I
101	El usuario introduce una dirección válida.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	C
102	Al pulsar volver cuando está visualizando el mapa.	Retorna a la pantalla de la creación del pedido manteniendo todos los valores.	Retorna a la pantalla de la creación del pedido manteniendo todos los valores.	C
103	No tiene trabajadores.	Se le muestra un dialogo que no tiene trabajadores.	Se le muestra un dialogo que no tiene trabajadores.	C
104	No tiene vehículos.	Se le muestra un dialogo indicando que no tiene vehículos.	Se le muestra un dialogo indicando que no tiene vehículos.	C
105	No tiene objetos	Se le muestra un dialogo indicando que no tiene objetos.	Se le muestra un dialogo indicando que no tiene objetos.	C
106	Crea el pedido	Se crea el pedido en la base de datos, se le muestra un Toast indicando pedido creado y vuelve a la pantalla gestión del almacén.	Se crea el pedido en la base de datos, se le muestra un Toast indicando pedido creado y vuelve a la pantalla gestión del almacén.	C
107	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del almacén.	Se vuelve a la pantalla gestión del almacén.	C

Tabla 27: Pruebas Crear pedidos

- Prueba 100:** En este punto se pensó poner como piso "S/N" queriendo decir sin número. En ese momento la aplicación se detuvo. Se observó que al recoger los valores, los trataba y los separaba por el carácter "/". Esta división lo hacía en todas las ventanas en las que el usuario podía introducir una localización. Este problema se arregló evitando que el usuario introdujera el carácter "/". En caso de introducirlo, se le mostraba un mensaje indicando que no se admite dicho carácter.

## 7.22-. Ver pedidos

Id	Descripción	Resultado esperado	Resultado obtenido	
108	El director no tiene pedidos.	Se le muestra una alerta indicando que no tiene pedidos.	Se le muestra una alerta indicando que no tiene pedidos.	C
109	El director tiene pedidos.	Se le muestran todos sus pedidos ordenados según el detalle que haya seleccionado.	Se le muestran todos sus pedidos ordenados según el detalle que haya seleccionado.	C
110	El director da al botón buscar teniendo pedidos.	Se le muestra la tabla de todos sus pedidos.	Se le muestra la tabla de todos sus pedidos.	C
111	El director da al botón buscar, teniendo pedidos, cuando ya se le mostraban todos sus pedidos.	Se actualiza la tabla.	Se actualiza la tabla.	C
112	El director pincha sobre el código de un pedido que esté en espera.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	Se le muestra un dialogo mostrándole las opciones Eliminar y Editar.	C
113	El director pincha sobre el código de un pedido que esté iniciado.	Se le muestra un dialogo mostrándole las opciones Eliminar y Detalles.	Se le muestra un dialogo mostrándole las opciones Eliminar y Detalles.	C
114	El director pulsa sobre Detalles en un pedido iniciado.	Se le muestra un dialogo mostrando la hora y día de salida. Respecto a la hora y día de llegada se muestra un "-" ya que aún no ha finalizado	Se le muestra un dialogo mostrando la hora y día de salida. Respecto a la hora y día de llegada se muestra un "-" ya que aún no ha finalizado	C
115	El director pincha sobre el código de un pedido que esté finalizado.	Se le muestra un dialogo mostrándole las opciones Eliminar y Detalles.	Se le muestra un dialogo mostrándole las opciones Eliminar y Detalles.	C
116	El director pulsa sobre Detalles en un pedido finalizado.	Se le muestra un dialogo mostrando la hora y día de salida, y la hora y día de llegada.	Se le muestra un dialogo mostrando la hora y día de salida, y la hora y día de llegada.	C
117	El director pulsa el botón atrás.	Se vuelve a la pantalla gestión del almacén.	Se vuelve a la pantalla gestión del almacén.	C

Tabla 28: Pruebas Ver pedidos

## 7.23-. Editar pedidos

Id	Descripción	Resultado esperado	Resultado obtenido	
118	El director deja algún campo vacío.	Se le muestra un Toast indicando campos vacíos.	El director deja algún campo vacío.	C
119	Introduce un destino inexistente.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	Se le muestra un Toast indicando que no se ha podido obtener la ubicación de la dirección introducida.	C
120	El usuario introduce una dirección válida.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	Se le muestra el mapa con un marcador y un título, centrado sobre el marcador.	C
121	Al pulsar volver cuando está visualizando el mapa.	Retorna a la pantalla de modificación del pedido manteniendo todos los valores.	Retorna a la pantalla de modificación del pedido manteniendo todos los valores.	C
122	No tiene trabajadores.	Se le muestra un dialogo que no tiene trabajadores.	Se le muestra un dialogo que no tiene trabajadores.	C
123	No tiene vehículos.	Se le muestra un dialogo indicando que no tiene vehículos.	Se le muestra un dialogo indicando que no tiene vehículos.	C
124	No tiene objetos	Se le muestra un dialogo indicando que no tiene objetos.	Se le muestra un dialogo indicando que no tiene objetos.	C
125	Modifica el pedido.	Se modifica el pedido en la base de datos, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior, donde no se le muestra la tabla de pedidos.	Se modifica el pedido en la base de datos, se le muestra un Toast indicando valores actualizados y vuelve a la pantalla anterior, donde no se le muestra la tabla de pedidos.	C
126	El director pulsa el botón atrás.	Se vuelve a la pantalla anterior donde no se le muestra la tabla de pedidos.	Se vuelve a la pantalla anterior donde no se le muestra la tabla de pedidos.	C

Tabla 29: Pruebas Editar pedidos

## 7.24-. Eliminar pedidos

Id	Descripción	Resultado esperado	Resultado obtenido	
127	Elimina un pedido.	Se elimina el pedido junto con su contenido de la base de datos. Además, se le muestra un Toast indicando pedido eliminado y vuelve a la pantalla anterior donde no se le muestra la tabla de pedidos.	Se elimina el pedido junto con su contenido de la base de datos. Además, se le muestra un Toast indicando pedido eliminado y vuelve a la pantalla anterior donde no se le muestra la tabla de pedidos.	C

Tabla 30: Pruebas Eliminar pedidos

## 7.25-. Observar pedidos asignados

Id	Descripción	Resultado esperado	Resultado obtenido	
128	No tiene pedidos asignados.	Se le muestra una alerta indicando que no tiene pedidos asignados.	Se le muestra una alerta indicando que no tiene pedidos asignados.	C
129	Tiene pedidos asignados.	Se le muestra una tabla con todos los pedidos asignados.	Se le muestra una tabla con todos los pedidos asignados.	C
130	El trabajador da al botón buscar teniendo pedidos asignados.	Se le muestra la tabla de todos sus pedidos.	Se le muestra la tabla de todos sus pedidos.	C
131	El trabajador da al botón buscar, teniendo pedidos asignados, cuando ya se le mostraban todos sus pedidos.	Se actualiza la tabla.	Se actualiza la tabla.	C
132	El trabajador pincha sobre el código de un pedido que esté en espera.	Se le muestra un dialogo mostrándole las opciones Iniciar y Detalles.	Se le muestra un dialogo mostrándole las opciones Iniciar y Detalles.	C
133	El trabajador pincha sobre el código de un pedido que esté iniciado.	Se le muestra un dialogo mostrándole las opciones Finalizar y Detalles.	Se le muestra un dialogo mostrándole las opciones Finalizar y Detalles.	C
134	El trabajador pulsa el botón atrás.	Se vuelve a la pantalla anterior.	Se vuelve a la pantalla anterior.	C

Tabla 31: Pruebas Observar pedidos asignados

## 7.26-. Iniciar pedido

Id	Descripción	Resultado esperado	Resultado obtenido	
135	El trabajador pincha sobre Iniciar	Se indica en la base de datos el pedido como iniciado, se le muestra un Toast indicando pedido iniciado y se oculta la tabla de pedidos asignados.	Se indica en la base de datos el pedido como iniciado, se le muestra un Toast indicando pedido iniciado y se oculta la tabla de pedidos asignados.	C

Tabla 32: Pruebas Iniciar pedido

## 7.27-. Mostrar ruta pedido

Id	Descripción	Resultado esperado	Resultado obtenido	
136	El trabajador pincha sobre Detalles.	Se le muestran todos los detalles sobre el pedido seleccionado.	Se le muestran todos los detalles sobre el pedido seleccionado.	C
137	Dentro de los detalles, el trabajador pincha sobre el botón volver.	Vuelve a la pantalla anterior donde se observa la tabla.	Vuelve a la pantalla anterior donde se observa la tabla.	C
138	Dentro de los detalles, el trabajador pincha sobre mostrar ruta.	Se le muestra un mapa centrado sobre el origen con dos marcadores (origen y destino junto con el título de cada uno de ellos). Además, se observa una línea correspondiente a la ruta a seguir entre los dos puntos. Por último, se muestra un Toast mostrando la distancia entre los dos puntos en metros o kilómetros, dependiendo de la distancia.	Se le muestra un mapa centrado sobre el origen con dos marcadores (origen y destino junto con el título de cada uno de ellos). Además, se observa una línea correspondiente a la ruta a seguir entre los dos puntos. Por último, se muestra un Toast mostrando la distancia entre los dos puntos en metros o kilómetros, dependiendo de la distancia.	C
139	Dentro del mapa el trabajador pincha sobre el botón volver.	Vuelve a la pantalla detalles.	Vuelve a la pantalla detalles.	C

Tabla 33: Pruebas Mostrar ruta pedido

## 7.28-. Finalizar pedido

Id	Descripción	Resultado esperado	Resultado obtenido	
140	El trabajador pincha sobre Finalizar.	Se le muestra una pantalla para que introduzca el día y hora de salida, junto con los kilómetros recorridos.	Se le muestra una pantalla para que introduzca el día y hora de salida, junto con los kilómetros recorridos.	C
141	Si el trabajador deja algún campo vacío.	Se muestra un indicando campos vacíos.	Se muestra un indicando campos vacíos.	C
142	Finaliza el pedido	Se almacena el pedido como finalizado y se actualizan el día y hora de salida, junto con los kilómetros. Además, se muestra un Toast indicando pedido finalizado y vuelve a la pantalla anterior donde no se observa la tabla de pedidos asignados.	Se almacena el pedido como finalizado y se actualizan el día y hora de salida, junto con los kilómetros. Además, se muestra un Toast indicando pedido finalizado y vuelve a la pantalla anterior donde no se observa la tabla de pedidos asignados.	C
143	El trabajador pulsa el botón volver	Vuelve a la pantalla anterior donde no se observa la tabla de pedidos asignados.	Vuelve a la pantalla anterior donde no se observa la tabla de pedidos asignados.	C

Tabla 34: Pruebas Finalizar pedido



## 8-. Conclusiones y trabajo futuro

Una vez se ha finalizado toca realizar un análisis de cómo ha ido el proyecto, es decir, si se han cumplido los objetivos que se habían propuesto, si las planificaciones han estado cerca del tiempo que se ha necesitado para finalizar este proyecto...

Además, hay que hacer un análisis de carácter personal contando los pros y contras que ha habido al realizar el proyecto.

Por último, se propone una serie de funcionalidades que se le pueden añadir al proyecto como trabajo futuro

### 8.1-. Conclusiones del proyecto

En este apartado se va a analizar todo el planteamiento inicial, es decir, si se han cumplido todos los objetivos que se propusieron, si el listado de las tareas del alcance ha sido el llevado a cabo durante el proyecto, si la estimación ha sido la ajustada al tiempo elaborado y por último, si se ha encontrado con algún riesgo mencionado en la gestión de riesgos. Una vez se han analizado todos estos casos, se tiene en cuenta la evaluación económica centrándose en el gasto total que ha conllevado la realización del proyecto.

- Objetivos

Respecto a los objetivos, el principal objetivo era diseñar e implementar la aplicación. Se puede decir que este objetivo ha sido cumplido en su totalidad debido a que todas las funcionalidades han sido desarrolladas y funcionan correctamente. En cuanto al diseño, como ya se comentó, se dejó para el final y se fue mejorando en el tiempo restante que había.

El segundo objetivo era desarrollar la aplicación en base a la estimación temporal inicial. Este objetivo lo se comprueba a la hora de analizar el tiempo que se ha necesitado para realizar el proyecto.

El siguiente objetivo era aumentar mi conocimiento a la hora de desarrollar aplicaciones Android. Se puede decir con total seguridad que ha sido cumplido debido a que he aprendido nuevas maneras de mostrar datos, nuevos diseños para los menús, y finalmente, mejores estilos centrados en el diseño de la aplicación.

Por último, se aumentó mi experiencia a la hora de desarrollar proyectos. Esto se debe a que actualmente conozco cuánto tiempo puede llevar realizar cada tarea. Además, conozco que problemas suelen ser los más usuales y como solucionarlos rápidamente.

- Alcance

Respecto al alcance, se ha de decir que se ha realizado un buen alcance. Esto se debe a que todas las tareas se llevaron a cabo. Algunas tareas no se realizaron en el orden que se indican, pero lo principal es que no ha surgido ninguna tarea que no se haya tenido en cuenta.

- Planificación temporal

Como se han descrito muchas tareas pero cada tarea dentro de un grupo, se analizan los grupos para una mejor comprensión. Para ello se usa el siguiente gráfico.

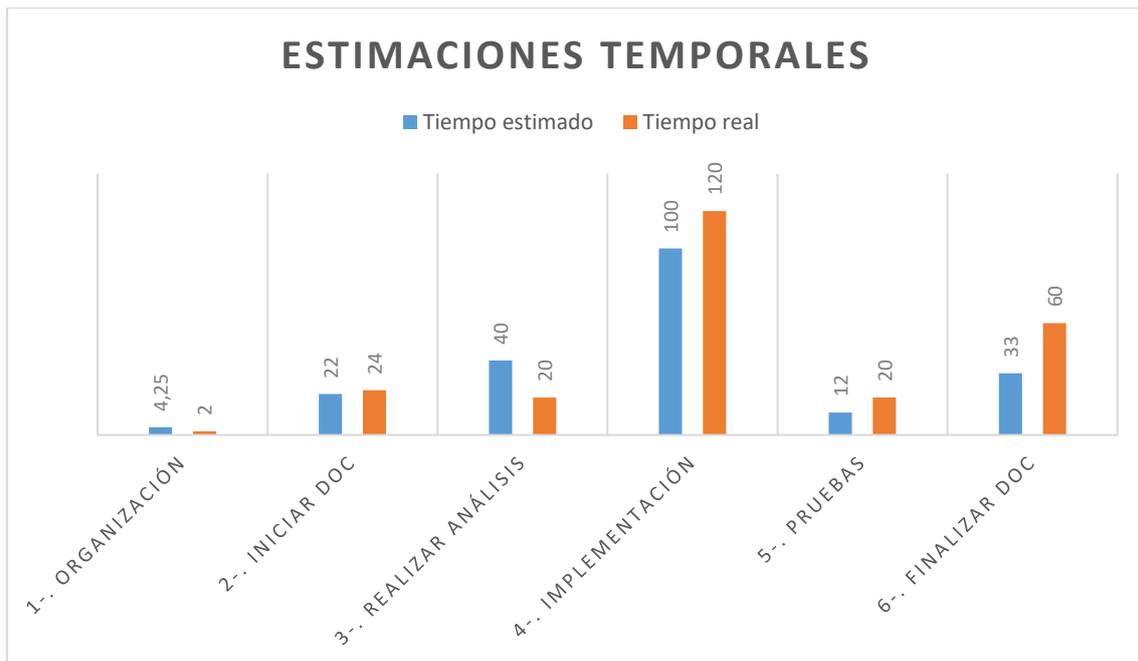


Ilustración 44: Comparación estimaciones

El total del tiempo real es 246 horas, siendo el estimado 211,25 horas. Como se puede observar, en ningún grupo de tareas se ha acertado en cuanto al tiempo que llevaría a cabo, pero en casi todas se ha estado cerca. En cuanto a los dos primeros grupos, se ha estado cerca debido a que ya se sabía lo que había que hacer al haberlo hecho ya en otros proyectos.

En cuanto a los dos siguientes grupos, ha sido en la que más se ha distanciado debido a que había que plasmar las ideas tanto en los diseños como en el código. Ha llevado tiempo debido a que primero se debía saber cómo se podía plasmárselo al usuario y posteriormente mostrarlo con todas las pruebas que conllevaría el código.

Respecto a los dos últimos grupos, ha sido desarrollo de memoria por lo cual se tenía una idea del tiempo que llevaría y por ello se ha estado un poco más cerca de la estimación.

- Riesgos

Sobre los riesgos, se definieron diversos riesgos que no se llevaron a cabo, como la caída de servidores, pérdidas de copias y averías.

Pero se deben destacar los planes de contingencia de los riesgos 6 (Enfermedad o lesión) y 7 (Problemas a la hora de realizar la implementación). Se tienen en cuenta estos dos riesgos debido a que han surgido durante la elaboración, incluso varias veces el riesgo 7.

En cuanto surgieron, el impacto que tuvieron no fue tan grande como el esperado, podría decirse que fue un impacto medio debido a que no se perdió mucho tiempo. El riesgo 6 era inevitable debido a los cambios constantes de tiempo, pero el riesgo 7 se solucionó en la mayor brevedad de tiempo. Debido a la rápida búsqueda de información o diferentes consultas, se pudo solventar y continuar con la realización del proyecto.

Por último, se debe destacar algún riesgo que ha surgido durante el proyecto que no se tuvo en cuenta. El riesgo principal fue el no cumplimiento del horario semanal. Es decir, en la planificación temporal se estableció realizar unas 12 horas semanales. Durante las primeras semanas se llevó a cabo, pero posteriormente ese horario no fue cumplido debido al no poseer

tanto tiempo debido a la realización de prácticas y posteriormente trabajo de jornada completa. Este riesgo tuvo un impacto grande tardando más tiempo en finalizar el proyecto y se solventó dedicando más tiempo durante el fin de semana.

- Evaluación económica

Una vez analizados todos los anteriores aspectos, se realiza de nuevo la evaluación económica en base al tiempo real. Como se ha indicado en dicho apartado, se tiene el sueldo del desarrollador, las amortizaciones, licencias y facturas. El gasto de las licencias y las amortizaciones se mantienen debido a que son anuales. Pero respecto a los otros gastos se deben modificar.

En cuanto al gasto de las facturas, se debe tener en cuenta cuando se inició el proyecto y cuando se finalizó. Se puede decir que el proyecto ha sido desarrollado en unos 7 meses (Noviembre – Mayo). Por lo tanto, en base al gasto mensual indicado de 20 euros, las facturas alcanzan un total de **140 euros**.

Respecto al sueldo del desarrollador, se indicó que un desarrollador cobra de media 30 euros la hora. Por lo tanto, al haber desarrollado el proyecto en 246 horas hay un gasto de **7380 euros**.

Finalmente, el gasto total de este proyecto es:

- Sueldo trabajador	->	7380 euros
- Amortizaciones	->	40 euros
- Licencias	->	189 euros
- Facturas	->	140 euros
- <b>TOTAL</b>	->	<b>7749 euros</b>

Como se puede observar, la diferencia entre la primera estimación de 6829 euros no es tan grande, con lo que se destaca que se hizo una buena evaluación económica. Por lo tanto, el objetivo de desarrollar una estimación acorde al tiempo necesario fue cumplido.

## 8.2-. Conclusiones personales

En cuanto a las conclusiones personales, se puede destacar que una vez se ha finalizado el proyecto he aprendido a realizar una gestión completa. Aunque durante la carrera he realizado diversas gestiones sobre proyectos, esta ha sido la primera vez en la que me he enfrentado a un proyecto individual y complejo.

A pesar de haber modificado mi idea inicial según avanzaba el proyecto y mientras me encontraba con algunos problemas, debo destacar que he realizado una buena planificación inicial.

Por otro lado, he aprendido aspectos en el desarrollo de aplicaciones Android que no conocía, además de mejorar los que ya conocía como por ejemplo, el menú lateral y el uso de la actividad Google Maps.

Otro punto a destacar, es que pese a haber intentado realizar una buena estimación temporal, esta puede no ser la final debido a todas las modificaciones realizadas, los problemas encontrados, y sobre todo, el planteamiento semanal que tenía para avanzar el proyecto. Esto se debe a que inicialmente propuse realizar unas 12 horas a la semana, lo que equivale a una hora y tres cuartos al día. Esta idea ha sido difícil de cumplir por aspectos como la realización de prácticas y finalmente trabajo, mientras se realizaba el proyecto. Además, todos los días no he podido realizar el proyecto con lo que mi idea de finalizarlo en 18 semanas quedó desestimada.

Pero el motivo general se debe a que ha sido un proyecto individual que pese a acordar fechas con el tutor para entregar documentación, había margen para realizar descansos y no avanzar el proyecto todos los días.

Por último, a pesar de tener un gran conocimiento a la hora de gestionar una base de datos, mi base de datos no permitía diversas opciones como la gestión de permisos, bloqueos... De tal manera, son aspectos que no he tenido en cuenta pero se podrían tener en caso de gestionar la base de datos con otra herramienta.

### 8.3- Trabajo futuro

Aunque el proyecto se haya finalizado implementando todas las funcionalidades anteriormente descritas, se puede continuar desarrollando y mejorándolo en caso de que algún cliente esté interesado en este proyecto. A esta aplicación se le pueden añadir varias funcionalidades más que no se le han añadido principalmente por la falta de tiempo y al haber implementado las funcionalidades primordiales de toda aplicación relacionada con la gestión de pedidos.

En cuanto a la aplicación, una posible funcionalidad futura sería la mejora de las interfaces. Como se ha comentado anteriormente, se le ha dado un diseño visual más llamativo que el estándar para mejorar la calidad de la aplicación. Aun así, se pueden mejorar mucho más las interfaces, por ejemplo, un estilo para los botones, otro tipo de forma para mostrar los datos en vez de tablas o listas, otro tipo de estilos para los menús, mayor cantidad de imágenes o fondos de pantalla... Como se puede observar, se pueden mejorar tantos aspectos como cualquier desarrollador o cliente quiera.

Respecto a las funcionalidades de la aplicación, se pueden añadir algunas más. Por ejemplo, un chat entre el director y el trabajador. Este chat permitiría resolver cualquier duda o incidencia que tuviese el trabajador a la hora de entregar el pedido. Además, el director podría conocer en cada momento el estado de la entrega al contactar con el trabajador. Pero sobretodo, esta funcionalidad estaría enfocada a tener una relación más cercana entre director y trabajador. Actualmente, el director asigna el pedido y el trabajador observa la asignación. Implementar esta funcionalidad, podría mejorar la comunicación entre estos y así poder entablar una relación más cercana que solo director – trabajador.

Otra funcionalidad que se podría implementar, es permitir el registro y el control de varios almacenes. Esta aplicación está enfocada a empresas de reparto de mercancías o paquetes. Poniendo el ejemplo de empresas conocidas como Seur o Correos, o aun así empresas no tan conocidas, pueden tener varios almacenes donde guardan sus objetos con los que posteriormente realizan sus pedidos. Actualmente, la aplicación solo permite un origen a la hora de crear el pedido. Este origen se corresponde con el lugar de la empresa. De tal manera, la implementación de esta funcionalidad permitiría poder registrar almacenes indicando sus lugares y así, a la hora de crear el pedido, se podría indicar en que almacén se encuentran los objetos que se deben entregar.

Por otro lado, la aplicación permite tres idiomas: Euskera, Castellano e Inglés. Se decidió usar estos tres idiomas porque son los más hablados en la localidad. De todos modos, si una empresa trabaja en otro país con otro idioma o reparte a países con otro idioma, se puede implantar dicho o dichos idiomas en la aplicación. De esta manera, la aplicación la podría tener cualquier tipo de cliente con el idioma que desee.

Por último, otra posible funcionalidad que se podría implementar trata de mejorar los detalles a la hora de mostrar la ruta. La aplicación le muestra al trabajador un mapa con una línea

indicando la ruta a seguir entre el origen y el destino. El trabajador puede acercar o alejar el zoom o girar la pantalla para una mejor visualización de la ruta. Pero esta funcionalidad le comunicaría al de cuantos metros o kilómetros debería girar o seguir recto, es decir, implementar un GPS. Un GPS te indica cuando debes girar a la izquierda o a la derecha, te indica la distancia que debes recorrer sobre un carril hasta la siguiente salida... La idea sería implementar estos detalles a la hora de mostrar la ruta para que en vez de mirar constantemente al mapa, el trabajador oiga y visualice mejor la ruta a seguir.

Todas estas mejoras llevarían un gran coste de tiempo, pero todas son opcionales. Esto es, ninguna es de obligatoria implantación. El cliente otorgaría un grado de importancia y el desarrollador un grado de coste.



## 9-. Bibliografía

[1]: Información acerca de Android Studio

<https://developer.android.com/studio/index.html> (Accedido el 22/11/2017)

[Ilustración 4]: Logo de Android Studio

[https://commons.wikimedia.org/wiki/File:Android\\_Studio\\_icon.svg](https://commons.wikimedia.org/wiki/File:Android_Studio_icon.svg) (Accedido el 20/11/2017)

[2]: Información acerca de WinSCP

<https://winscp.net/eng/docs/introduction> (Accedido el 22/11/2017)

[Ilustración 5]: Logo de WinSCP

<https://winscp.net/eng/index.php> (Accedido el 20/11/2017)

[Ilustración 6]: Logo de Notepad ++

<https://notepad-plus-plus.org/> (Accedido el 22/11/2017)

[Ilustración 7]: Logo Visual Paradigm

<https://www.visual-paradigm.com/> (Accedido el 22/11/2017)

[Ilustración 8]: Logo Dropbox

<https://www.dropbox.com/> (Accedido el 22/11/2017)

[Ilustración 9]: Logo Microsoft Office

<https://products.office.com/es-es/home> (Accedido el 22/11/2017)

[Ilustración 10]: Arquitectura cliente-servidor con dispositivo móvil (Accedido el 24/11/2017)

<https://academiaandroid.com/aplicaciones-cliente-servidor-y-redes-de-telefonía-móvil/>

[3]: Información acerca de Natureh Customer Tracking

[http://www.natureh.com/pr/Diseno\\_Web/Tienda\\_Online\\_B2C/Seguimiento\\_Cliente](http://www.natureh.com/pr/Diseno_Web/Tienda_Online_B2C/Seguimiento_Cliente) (Accedido el 27/11/2017)

[4]: Información acerca de Logistics

<https://gurtam.com/es/logistics> (Accedido el 27/11/2017)

[5]: Información acerca de TomTom Webfleet

[https://telematics.tomtom.com/es\\_es/webfleet/](https://telematics.tomtom.com/es_es/webfleet/) (Accedido el 27/11/2017)

[Ilustración 11] [Ilustración 12]: Información acerca de las compras online

[http://www.ine.es/prensa/tich\\_2017.pdf](http://www.ine.es/prensa/tich_2017.pdf) (Accedido el (04/03/2018)

[7]: Información acerca del lenguaje de programación PHP

<http://php.net/manual/es/intro-what-is.php> (Accedido el 30/01/2017)



## 10-. Anexo 1 – Casos de uso extendidos

En primer lugar se va a mostrar los casos de usos extendidos correspondientes a las funcionalidades comunes. Después, se muestran las correspondientes al director y por último, los casos de uso de las funcionalidades del trabajador.

### MODIFICAR IDIOMA

**Descripción:** El usuario desea modificar el idioma de la aplicación.

**Actores:** Cualquier usuario.

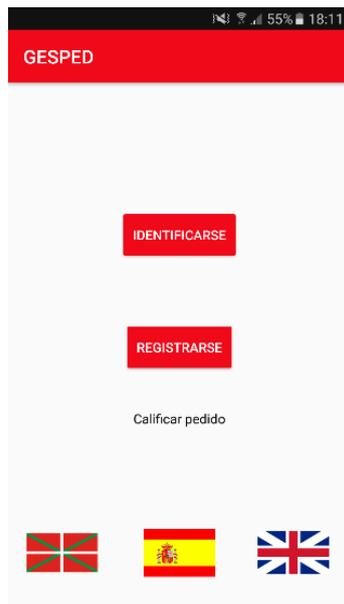
**Precondición:** Ninguna.

**Requisitos no funcionales:** Ninguno.

**Flujo de eventos:**

1-. El usuario pincha sobre cualquier imagen representando el idioma correspondiente (*Ver Pantalla 1*).

**Poscondiciones:** Se ha actualizado el idioma de toda la aplicación.



**Pantalla 1:**  
Pantalla Inicial

## CALIFICAR PEDIDO

**Descripción:** El destinatario tiene la opción de valorar el pedido en base al trato con el repartidor, estado del paquete y puntualidad. Además, podrá ingresar una opinión. Para almacenar la valoración deberá indicar el código del pedido y el nombre de usuario del repartidor.

**Actores:** Cualquier usuario.

**Precondición:** El pedido está finalizado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El usuario pincha sobre el texto "Calificar pedido" (*Ver Pantalla 1*).

2-. El usuario rellena los campos que se le muestran por pantalla (*Ver Pantalla 2*).

*[Si rellena todos los campos]*

*[Si el nombre del repartidor existe]*

*[Si el pedido existe]*

*[Si el estado del pedido es finalizado]*

*[Si el pedido no ha sido valorado]*

3a-. La valoración del pedido se almacena y se le muestra un mensaje indicando valoración guardada (*Ver Toast 1*).

*[Sino]*

3b-. Se le muestra un mensaje indicando pedido ya valorado (*Ver Toast 2*).

*[Sino]*

3c-. Se le muestra un mensaje indicando que el pedido aún no ha finalizado (*Ver Toast 3*).

*[Sino]*

3d-. Se le muestra un mensaje indicando que el pedido no existe (*Ver Toast 4*).

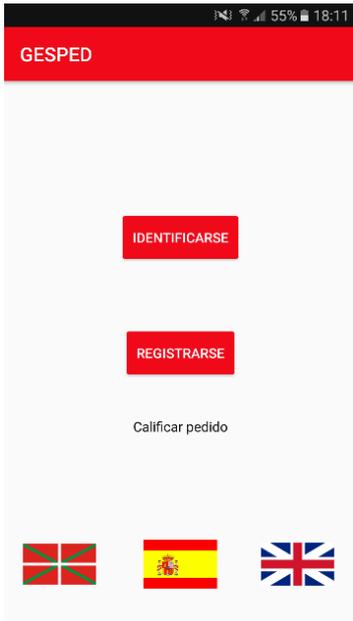
*[Sino]*

3e-. Se le muestra un mensaje indicando que el nombre del repartidor no existe (*Ver Toast 5*).

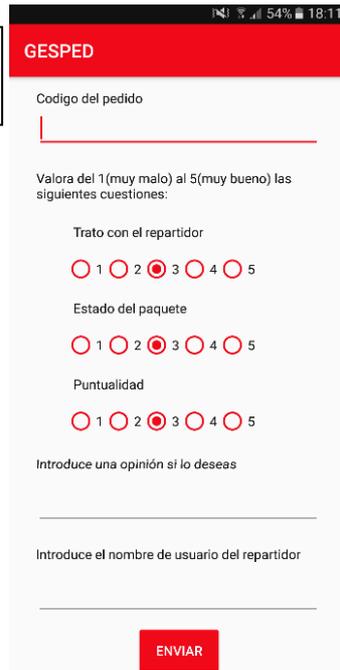
*[Sino]*

3f-. Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

**Poscondiciones:** Se ha guardado la valoración del pedido y se han actualizado las estadísticas de los repartos del usuario.



**Pantalla 1:**  
Pantalla Inicial



**Pantalla 2:**  
Pantalla Calificar  
pedido

Valoración del pedido guardada

**Toast 1**

El pedido ya ha sido valorado

**Toast 2**

El pedido aún no ha finalizado

**Toast 3**

El código del pedido no existe

**Toast 4**

No existe el usuario

**Toast 5**

Hay campos vacíos

**Toast 6**

## IDENTIFICARSE

**Descripción:** Permite a cualquier usuario acceder a la aplicación introduciendo su nombre de usuario y su contraseña.

**Actores:** Cualquier usuario.

**Precondición:** Ninguna.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El usuario pincha sobre el botón “Identificarse” (Ver Pantalla 1).

2-. El usuario introduce su nombre de usuario y su contraseña (Ver Pantalla 3).

[Si rellena todos los campos]

[Si existe el usuario]

[Si la contraseña es correcta]

3a-. El usuario accede a la aplicación (Ver Pantalla 4).

[Sino]

3b-. Se le muestra un mensaje indicando contraseña incorrecta (Ver Toast 7).

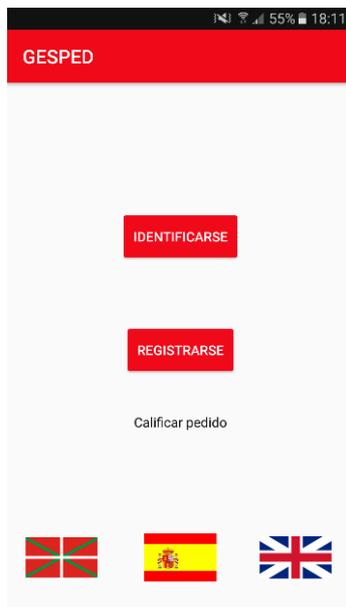
[Sino]

3c-. Se le muestra un mensaje indicando que el usuario no existe (Ver Toast 5).

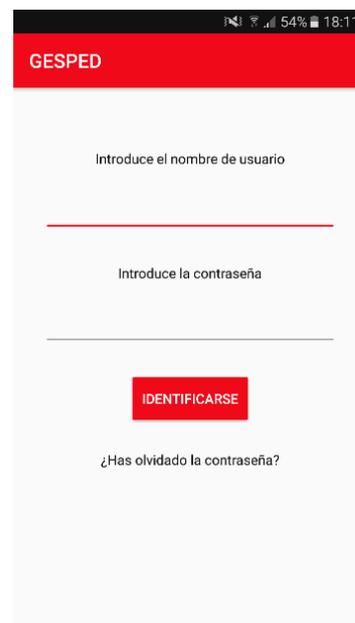
[Sino]

3d-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

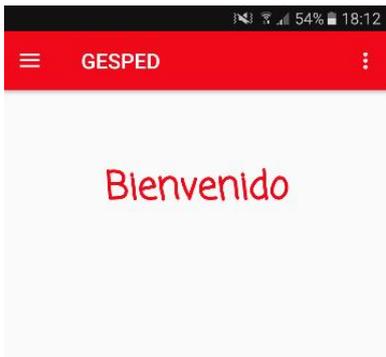
**Poscondiciones:** El usuario ha accedido a la aplicación y se ha almacenado su nombre de usuario, contraseña y nombre de usuario de su jefe (será el mismo si el usuario identificado es jefe) en una sesión.



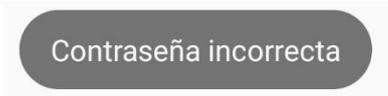
**Pantalla 1:**  
Pantalla Inicial



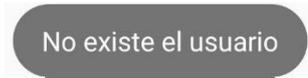
**Pantalla 3:**  
Pantalla Identificarse



**Pantalla 4:**  
Pantalla Menú principal



**Toast 7**



**Toast 5**



**Toast 6**

## RECUPERAR CONTRASEÑA

**Descripción:** El usuario no recuerda su contraseña y desea obtenerla. Para ello, deberá introducir su nombre de usuario y su email.

**Actores:** Cualquier usuario.

**Precondición:** Ninguna.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El usuario pulsa sobre el texto “¿Has olvidado la contraseña?” (Ver Pantalla 3).

2-. El usuario introduce su nombre de usuario y su email (Ver Pantalla 5).

[Si rellena todos los campos]

[Si existe el usuario y el email es correcto]

3a-. Se le envía una notificación mostrándole su contraseña (Ver Notificación).

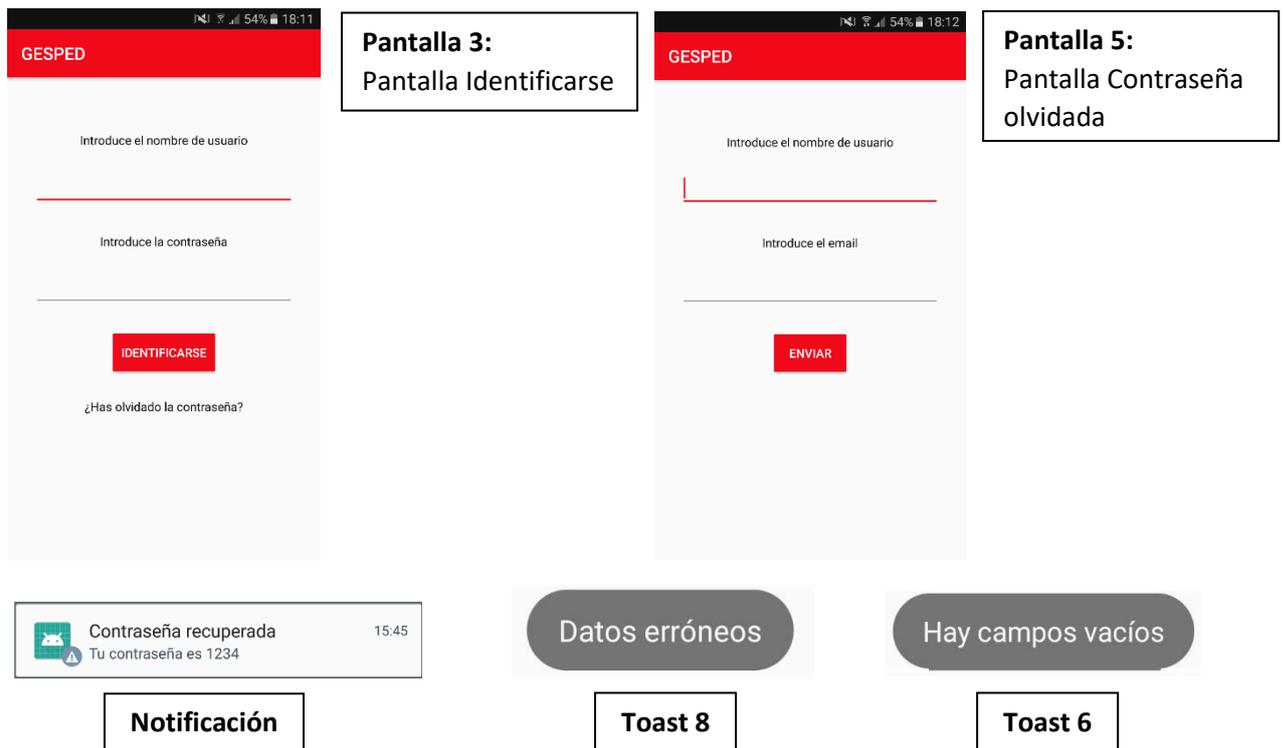
[Sino]

3b-. Se le muestra un mensaje indicando datos erróneos (Ver Toast 8).

[Sino]

3c-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** Se le ha notificado su contraseña para que pueda identificarse en la aplicación.



## REGISTRARSE COMO DIRECTOR

**Descripción:** El usuario desea registrarse como director de una empresa. Para ello deberá introducir una serie de datos personales y datos correspondientes a su empresa.

**Actores:** Cualquier usuario.

**Precondición:** Ninguna.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El usuario pincha sobre el botón “Registrarse” (Ver Pantalla 1).

2-. El usuario introduce todos los datos que se le muestran en pantalla (Ver Pantalla 6).

[Si rellena todos los campos]

[Si las contraseñas coinciden]

[Si el nombre de usuario no existe]

3a-. Se registra y se le muestra un mensaje indicando que se ha registrado como director (Ver Toast 9).

[Sino]

3b-. Se le muestra un mensaje indicando que el nombre de usuario ya existe (Ver Toast 10).

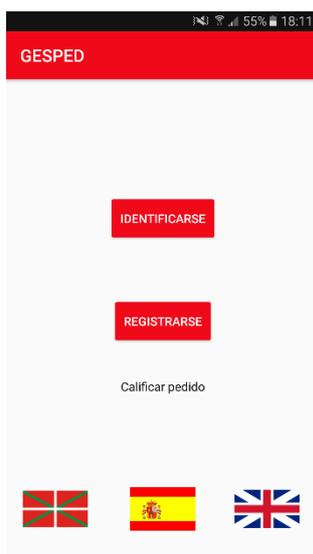
[Sino]

3c-. Se le muestra un mensaje indicando contraseñas diferentes (Ver Toast 11).

[Sino]

3d-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** Se ha registrado en la aplicación y ha podido acceder a ella. Además, se han guardado en la sesión su nombre de usuario y su contraseña.



**Pantalla 1:**  
Pantalla Inicial

**Pantalla 6:**  
Pantalla Registrarse  
como director

Te has registrado como director

**Toast 9**

Nombre de usuario en uso

**Toast 10**

Las contraseñas son diferentes

**Toast 11**

Hay campos vacíos

**Toast 6**

## CERRAR SESIÓN

**Descripción:** El usuario ha cerrado sesión para realizar cualquier tarea de las anteriores.

**Actores:** Director y trabajador

**Precondición:** Debe haber iniciado sesión en la aplicación

**Requisitos no funcionales:** Conexión a internet.

**Flujo de eventos:**

- 1-. El usuario selecciona la opción "Cerrar sesión" (*Ver Opción Cerrar sesión*).
- 2-. Se elimina de la sesión todos los datos almacenados.

**Poscondiciones:** El usuario ha vuelto a la ventana Inicial (*Ver Pantalla 1*).



Opción Cerrar  
sesión

Pantalla 1:  
Pantalla Inicial



## MODIFICAR DATOS PERSONALES

**Descripción:** El usuario desea modificar cualquier dato personal suyo. Se le muestran todos los datos y él modifica los que desee.

**Actores:** Director y trabajador.

**Precondición:** Debe haber iniciado sesión en la aplicación.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El usuario pincha sobre la opción “Modificar datos personales” (*Ver Menú lateral: Director*).
- 2-. Se le muestran todos sus datos almacenados (excepto el nombre de usuario) y modifica los que desee (*Ver Pantalla 7*). Una vez modificados, pulsa sobre el botón “Guardar”.

[Si rellena todos los campos]

[Si las contraseñas coinciden]

- 3a-. Se actualizan los valores en la base de datos y se le muestra un mensaje de valores actualizados (*Ver Toast 11*).

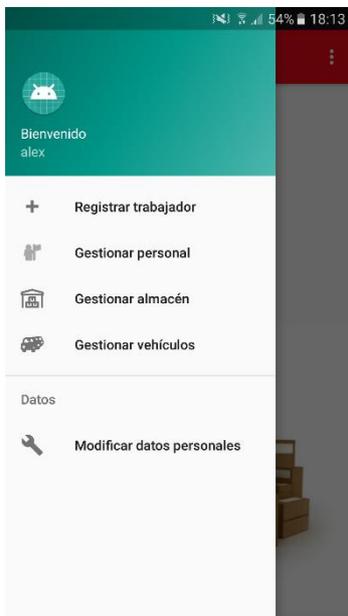
[Sino]

- 3b-. Se le muestra un mensaje indicando contraseñas diferentes (*Ver Toast 12*).

[Sino]

- 3c-. Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

**Poscondiciones:** Se han actualizado todos los valores en la base de datos.



**Menú lateral:  
Director**



**Pantalla 7:  
Pantalla  
Modificar datos**

Se han actualizado los valores

**Toast 11**

Las contraseñas son diferentes

**Toast 12**

Hay campos vacíos

**Toast 6**

A continuación se muestran los casos de uso para el usuario “director”.

## REGISTRAR TRABAJADOR

**Descripción:** El director desea registrar trabajadores, a los cuales les asignará el envío de los pedidos.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1- El director selecciona la opción “Registrar trabajador” (*Ver Menú lateral: Director*).

2- Introduce todos los datos correspondientes al nuevo trabajador (*Ver Pantalla 8*).

[Si rellena todos los campos]

[Si las contraseñas coinciden]

[Si el nombre de usuario no existe]

3a- Se crea el nuevo usuario y se le muestra un mensaje indicando nuevo trabajador registrado (*Ver Toast 13*).

[Sino]

3b- Se le muestra un mensaje indicando nombre de usuario en uso (*Ver Toast 10*).

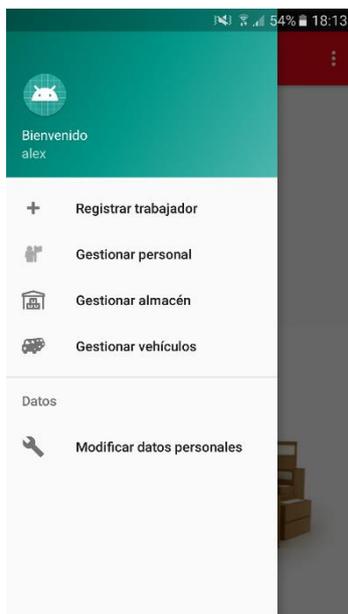
[Sino]

3c- Se le muestra un mensaje indicando contraseñas diferentes (*Ver Toast 12*).

[Sino]

3d- Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

**Poscondiciones:** Se ha almacenado los datos del nuevo usuario en la base de datos.



**Menú lateral:**  
Director

**Pantalla 8:**  
Pantalla Registrar  
trabajador

Nuevo trabajador registrado

**Toast 13**

Nombre de usuario en uso

**Toast 10**

Las contraseñas son diferentes

**Toast 12**

Hay campos vacíos

**Toast 6**

## VER SUS TRABAJADORES

**Descripción:** El director desea ver todos sus trabajadores y sus datos personales. Además, le damos la opción de ordenar sus trabajadores por nombre de usuario, nombre o apellidos.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación y tener algún trabajador creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director selecciona la opción "Gestionar personal" (*Ver Menú lateral: Director*).
- 2-. El director pincha sobre el botón "Ver trabajadores" (*Ver Pantalla 9*).
- 3-. Selecciona en el desplegable el orden por el que se mostrarán sus trabajadores y pincha sobre el botón "Buscar".

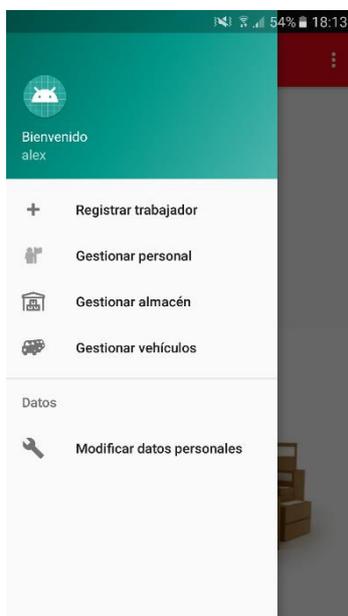
[*Si tiene trabajadores*]

4a-. Se le carga una tabla con el nombre de usuario, nombre y apellidos de todos sus trabajadores. (*Ver Pantalla 10*).

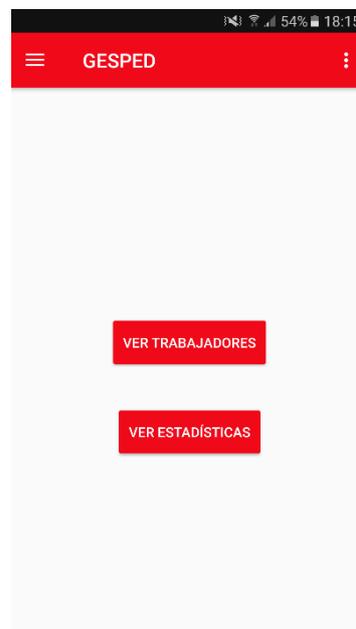
[*Sino*]

4b-. Se le muestra un dialogo indicando que no tiene trabajadores y que registre primero algún trabajador (*Ver No tiene trabajadores*).

**Poscondiciones:** El director ha observado todos los trabajadores que ha creado junto con algunos datos personales.



**Menú lateral:**  
Director



**Pantalla 9:**  
Pantalla Gestionar personal



**Pantalla 10:**  
**Pantalla Ver**  
**trabajadores**

No tienes trabajadores  
Registra primero algún trabajador  
**OK**

**No tiene**  
**trabajadores**

## MODIFICAR DATOS DE SUS TRABAJADORES

**Descripción:** El director desea modificar los datos personales de sus trabajadores.

**Actores:** Director.

**Precondición:** Tener algún trabajador creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El director pincha sobre el nombre de usuario del trabajador que desea modificar (Ver Pantalla 10).

2-. Se le muestra un dialogo indicando si desea eliminar o editar el trabajador seleccionado (Ver Dialogo 1).

3-. El director pincha sobre la opción "Editar" y se le muestran todos los datos del usuario, menos su nombre de usuario (Ver Pantalla 7).

4-. Modifica los datos que desee y pincha sobre el botón "Guardar".

[Si rellena todos los campos]

[Si las contraseñas coinciden]

5a-. Se actualizan los valores en la base de datos y se le muestra un mensaje de valores actualizados (Ver Toast 11).

[Sino]

5b-. Se le muestra un mensaje indicando contraseñas diferentes (Ver Toast 12).

[Sino]

5c-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** Se han actualizado los datos del trabajador en la base de datos.

**Pantalla 10:** Pantalla Ver trabajadores

**Detalles del trabajador**  
¿Que deseas hacer?  
**ELIMINAR** **EDITAR**

**Dialogo 1:** Eliminar o editar un trabajador

**Toast 12:** Las contraseñas son diferentes

**Toast 11:** Se han actualizado los valores

USUARIO	NOMBRE	APELLIDOS
a	a	a
jon	jon	zillarbide

19:22 65%

**GESPED**

Nombre  
jose mari

Apellidos  
azpilicueta

Introduce la contraseña  
....

Introduce de nuevo la contraseña  
....

Introduce el email  
josemari@hotmail.com

**GUARDAR**

**Pantalla 7:**  
Pantalla  
Modificar datos

Hay campos vacíos

**Toast 6**

## ELIMINAR TRABAJADORES

**Descripción:** El director desea eliminar trabajadores que ha creado.

**Actores:** Director.

**Precondición:** Tener algún trabajador creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director pincha sobre el nombre de usuario del trabajador que desea eliminar (*Ver Pantalla 10*).
- 2-. Se le muestra un dialogo indicando si desea eliminar o editar el trabajador seleccionado (*Ver Dialogo 1*).
- 3-. El director pincha sobre la opción "Eliminar".
- 4-. Se le muestra un mensaje indicando que ha eliminado el trabajador (*Ver Toast 14*).

**Poscondiciones:** Se ha eliminado el trabajador de la base de datos. Además, se han eliminado todos los pedidos a los que estaba asignado junto con el contenido de estos.



**Pantalla 10:**  
Pantalla Ver  
trabajadores

### Detalles del trabajador

¿Que deseas hacer?

ELIMINAR EDITAR

**Dialogo 1:**  
Eliminar o editar  
un trabajador

El trabajador ha sido eliminado

**Toast 14**

## VER ESTADÍSTICAS DE PEDIDOS

**Descripción:** El director puede ver las estadísticas de los pedidos de cada trabajador, es decir, la media del trato con el destinatario, la media de los estados de los paquetes entregados, la media de la puntualidad a la hora de la entrega y la cantidad de pedidos realizados. Además, estas estadísticas al ser numéricas las puede obtener ordenadas de mayor a menor o viceversa.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación y tener algún trabajador creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director selecciona pincha sobre el botón “Ver estadísticas” (*Ver Pantalla 9*).
- 2-. Selecciona en el desplegable el detalle por el que se mostrarán sus trabajadores, elige el orden y pincha sobre el botón “Buscar”.

[Si tiene trabajadores]

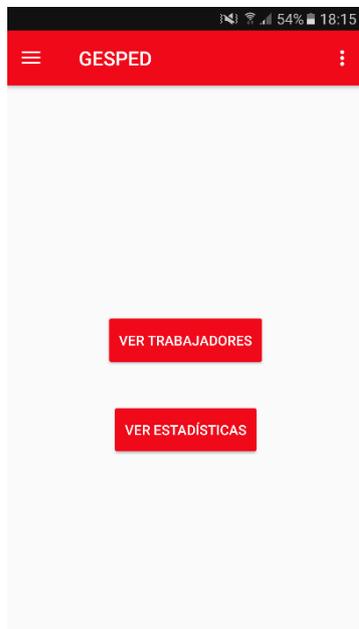
3a-. Se le carga una tabla con el nombre de usuario y las cuatro estadísticas. (*Ver Pantalla 11*).

4a-. El director pincha sobre un nombre de usuario y se le muestra un dialogo mostrándole toda la información de la tabla más detallada (*Ver Estadísticas trabajador*).

[Sino]

3b-. Se le muestra un dialogo indicando que no tiene trabajadores y que registre primero algún trabajador (*Ver No tiene trabajadores*).

**Poscondiciones:** El director ha podido observar todas las estadísticas de sus trabajadores.



**Pantalla 9:**  
Pantalla Gestionar personal

jon		
Trato con el repartidor:		0
Estado del paquete:		0
Puntualidad:		0
Pedidos realizados:		0
		OK

**Estadísticas trabajador**

No tienes trabajadores  
Registra primero algún trabajador

OK

**No tiene  
trabajadores**

GESPED

Ordenar por Trato con el repartidor

De mayor a menor
  De menor a mayor

BUSCAR

USUARIO	TRATO	ESTADO	PUNTUALIDAD	PEDIDOS REALIZADOS
a	0	0	0	0
jon	0	0	0	0

**Pantalla 11:**  
Pantalla Ver estadísticas

## CREAR VEHÍCULOS

**Descripción:** El director crea vehículos con los cuales sus trabajadores entregarán los pedidos.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director selecciona la opción “Gestionar Vehículos” (Ver Menú lateral Director).
- 2-. Pincha sobre el botón “Crear vehículos” (Ver Pantalla 12).
- 3-. Introduce todos los campos que se le muestran por pantalla y pincha sobre el botón “Guardar” (Ver Pantalla 13).

[Si rellena todos los campos]

[Si la matrícula no existe]

4a-. Se crea el vehículo y se le muestra un mensaje indicando vehículo creado (Ver Toast 15).

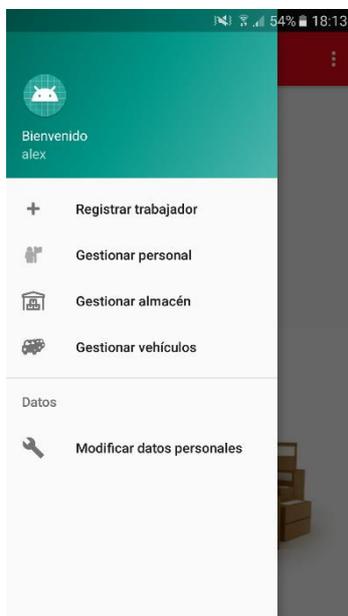
[Sino]

4b-. Se le muestra un mensaje indicando que la matrícula del vehículo ya existe (Ver Toast 16).

[Sino]

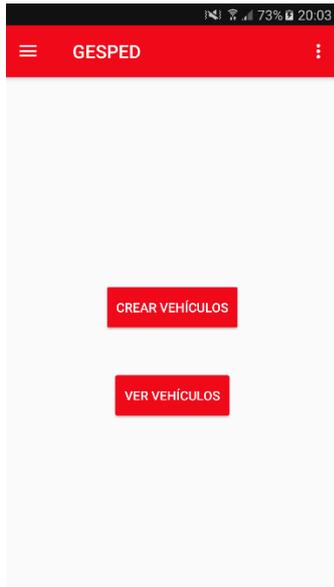
4c-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** El director ha creado vehículos y se han almacenado en la base de datos.



**Menú lateral:  
Director**

**Pantalla 13:  
Pantalla Crear  
vehículos**



**Pantalla 12:**  
Pantalla Gestionar  
vehículos

El vehículo ha sido creado

**Toast 15**

El vehículo ya existe

**Toast 16**

Hay campos vacíos

**Toast 6**

## VER VEHÍCULOS

**Descripción:** El director desea ver todos los vehículos que ha creado para su posterior edición o eliminación, si lo desea.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación y tener algún vehículo creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El director pincha sobre el botón “Ver vehículos” (*Ver Pantalla 12*).

2-. Selecciona en el desplegable el orden por el que se mostraran todos los vehículos y pincha sobre el botón “Buscar”.

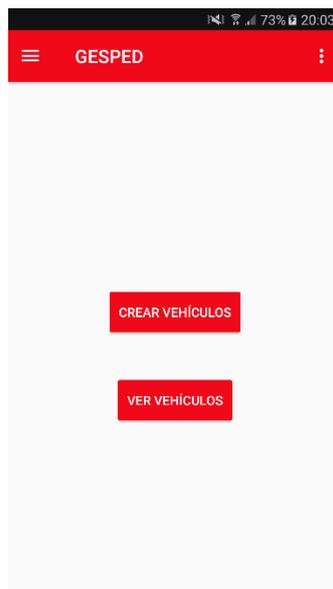
[*Si tiene vehículos*]

3a-. Se le muestra una tabla mostrando todos los vehículos con sus matrículas, modelo y kilómetros recorridos (*Ver Pantalla 14*).

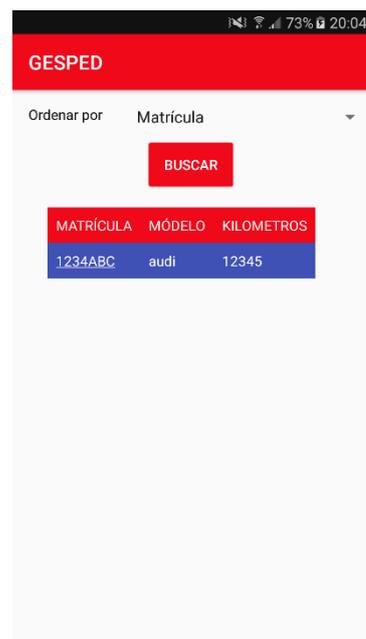
[*Sino*]

3b-. Se le muestra un dialogo indicando que no tiene vehículos y que primero debe crear algún vehículo (*Ver No tiene vehículos*).

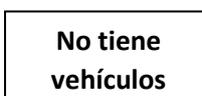
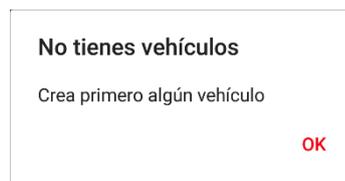
**Poscondiciones:** El director ha visualizado todos los vehículos que ha creado.



**Pantalla 12:**  
Pantalla Gestionar  
vehículos



**Pantalla 14:**  
Pantalla Ver  
vehículos



## MODIFICAR VEHÍCULOS

**Descripción:** El director desea modificar detalles de los vehículos creados.

**Actores:** Director.

**Precondición:** Tener algún vehículo creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director pincha sobre la matrícula del vehículo que desea modificar (*Ver Pantalla 14*).
- 2-. Se le muestra un dialogo indicando si desea eliminar o editar el vehículo seleccionado (*Ver Dialogo 2*).
- 3-. El director pincha sobre la opción “Editar” y se le muestran todos los datos del vehículo (*Ver Pantalla 15*).
- 4-. Modifica los campos y pincha sobre el botón “Guardar”.

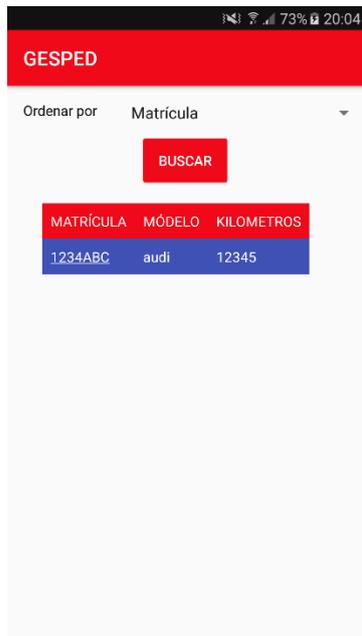
[Si rellena todos los campos]

5a-. Se actualizan los valores en la base de datos y se le muestra un mensaje de valores actualizados (*Ver Toast 11*).

[Sino]

5b-. Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

### Poscondiciones:



**Pantalla 14:**  
Pantalla Ver  
vehículos

**Pantalla 15:**  
Pantalla Editar  
vehículos

GESPED

Introduce la matrícula del vehículo

1234ABC

Introduce el modelo del vehículo

audi

Introduce los años del vehículo

12

Introduce la fecha de la última ITV

15 feb. 2018

Introduce la fecha de la próxima ITV

17 feb. 2018

Introduce los Km recorridos del vehículo

12345

Introduce la carga del vehículo en Kg

123

GUARDAR



**Dialogo 2:**  
Eliminar o editar  
un vehículo

Se han actualizado los valores

**Toast 11**

Hay campos vacíos

**Toast 6**

## ELIMINAR VEHÍCULOS

**Descripción:** El director desea eliminar vehículos que ha creado.

**Actores:** Director.

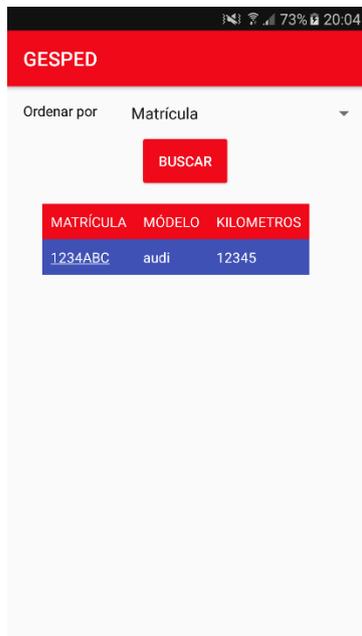
**Precondición:** Tener algún vehículo creado.

**Requisitos no funcionales:** Conexión a internet.

**Flujo de eventos:**

- 1-. El director pincha sobre la matrícula del vehículo que desea eliminar (*Ver Pantalla 14*).
- 2-. Se le muestra un dialogo indicando si desea eliminar o editar el vehículo seleccionado (*Ver Dialogo 2*).
- 3-. El director pincha sobre la opción "Eliminar".
- 4-. Se le muestra un mensaje indicando que ha eliminado el vehículo (*Ver Toast 17*).

**Poscondiciones:** Se ha eliminado el vehículo de todos los vehículos creados. Además, se elimina de los pedidos creados la asignación del vehículo.



**Pantalla 14:**  
Pantalla Ver  
vehículos



**Dialogo 2:**  
Eliminar o editar  
un vehículo

El vehiculo ha sido eliminado

**Toast 17**

## ALMACENAR OBJETOS

**Descripción:** El director debe almacenar objetos con los cuales podrá realizar pedidos.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El director selecciona la opción “Gestionar almacén” (Ver Menú lateral Director).
- 2-. El director pincha sobre el botón “Almacenar Objeto” (Ver Pantalla 16).
- 3-. Introduce el código del objeto y la descripción, y pincha sobre el botón “Guardar” (Ver Pantalla 17).

[Si rellena todos los campos]

[Si el código del objeto no existe]

4a-. Se almacena el objeto y se le muestra un mensaje indicando objeto guardado (Ver Toast 18).

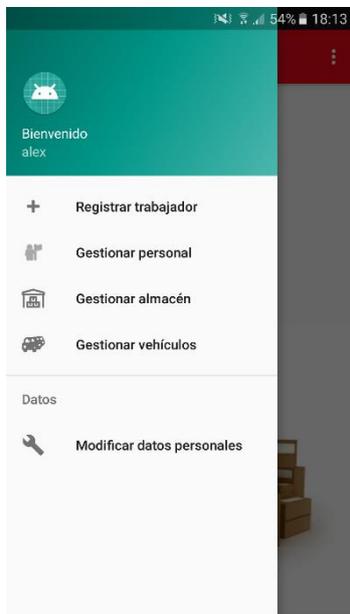
[Sino]

4b-. Se le muestra un mensaje indicando que el código del objeto ya existe (Ver Toast 19).

[Sino]

4c-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

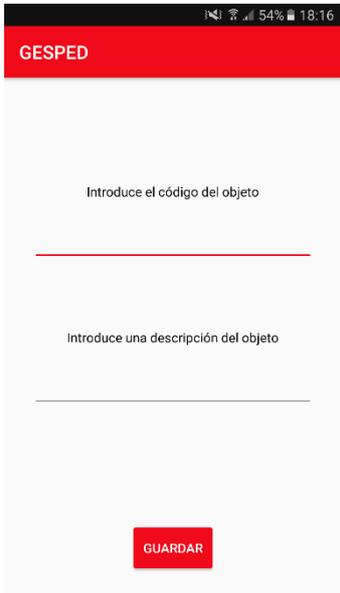
**Poscondiciones:** El director ha podido almacenar objetos en la base de datos.



**Menú lateral:**  
Director



**Pantalla 16:**  
Pantalla Gestionar  
almacén



**Pantalla 17:**  
Pantalla Almacenar  
objetos

El objeto ha sido almacenado

**Toast 18**

El código del objeto ya existe

**Toast 19**

Hay campos vacíos

**Toast 6**

## VER EL ALMACÉN

**Descripción:** El director puede observar todos los objetos que ha almacenado y si lo desea puede editarlos o eliminarlos.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación y tener algún objeto almacenado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El director pulsa el botón “Ver almacén” (*Ver Pantalla 16*).

2-. Selecciona en el desplegable el orden por el que se mostraran los objetos y pincha sobre el botón “Buscar”.

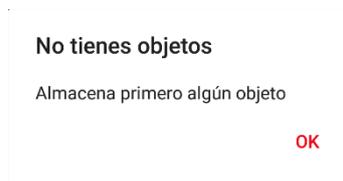
[*Si tiene objetos*]

3a-. Se le muestra una tabla con todos los objetos indicando su código y su descripción (*Ver Pantalla 18*).

[*Sino*]

3b-. Se le muestra un dialogo indicando que no tiene objetos y que debe almacenar primero algún objeto (*Ver No tiene objetos*).

**Poscondiciones:** El director ha visualizado todos los objetos que ha creado.



No tiene  
objetos



**Pantalla 16:**  
Pantalla Gestionar  
almacén



**Pantalla 18:**  
Pantalla Ver  
almacén

## EDITAR OBJETOS

**Descripción:** El director desea modificar la descripción del objeto debido a que el código es único y no se puede modificar.

**Actores:** Director.

**Precondición:** Tener algún objeto almacenado.

**Requisitos no funcionales:** Conexión a internet.

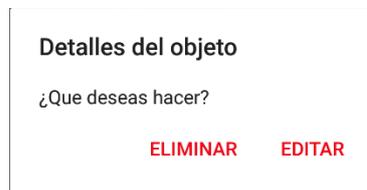
### Flujo de eventos:

- 1-. El director pincha sobre el código del objeto que desea modificar (*Ver Pantalla 18*).
- 2-. Se le muestra un dialogo indicando si desea eliminar o editar el objeto seleccionado (*Ver Dialogo 3*).
- 3-. El director pincha sobre la opción "Editar" y se le muestran todos los datos del objeto (*Ver Pantalla 19*).
- 4-. Modifica la descripción y pincha sobre el botón "Guardar".  
[Si rellena el campo descripción]
  - 5a-. Se actualiza el valor descripción en la base de datos y se le muestra un mensaje de valores actualizados (*Ver Toast 11*).

[Sino]

- 5b-. Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

**Poscondiciones:** El director ha modificado la descripción del objeto seleccionado.



**Dialogo 3:**  
Eliminar o editar  
un objeto

Se han actualizado los valores

**Toast 11**

Hay campos vacíos

**Toast 6**



**Pantalla 18:**  
Pantalla Ver  
almacén



**Pantalla 19:**  
Pantalla Editar  
objeto

## ELIMINAR OBJETOS

**Descripción:** El director desea eliminar objetos que ha almacenado.

**Actores:** Director.

**Precondición:** Tener algún objeto almacenado.

**Requisitos no funcionales:** Conexión a internet.

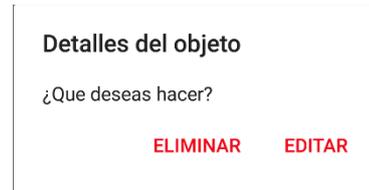
**Flujo de eventos:**

- 1-. El director pincha sobre el código del objeto que desea eliminar (*Ver Pantalla 18*).
- 2-. Se le muestra un dialogo indicando si desea eliminar o editar el objeto seleccionado (*Ver Dialogo 3*).
- 3-. El director pincha sobre la opción "Eliminar".
- 4-. Se le muestra un mensaje indicando que ha eliminado el objeto (*Ver Toast 20*).

**Poscondiciones:** Se ha eliminado el objeto de todos los objetos almacenados y además, se ha eliminado el objeto de todos los pedidos que estén creados.



**Pantalla 18:**  
Pantalla Ver  
almacén



**Dialogo 3:**  
Eliminar o editar  
un objeto

Objeto eliminado

**Toast 20**

## CREAR PEDIDOS

**Descripción:** El director desea crear pedidos. Para ello indicará el destino, el trabajador que entregará el pedido, el vehículo con el que lo realizará y los objetos que entregará.

**Actores:** Director.

**Precondición:** Haber iniciado sesión en la aplicación, tener algún objeto almacenado, tener algún vehículo creado y algún trabajador registrado.

**Requisitos no funcionales:** Conexión a internet

### Flujo de eventos:

1-. El director pincha sobre el botón “Crear pedido” (Ver Pantalla 16).

2-. Rellena todos los campos que se le muestran en la pantalla y añade los objetos al listado (Ver Pantalla 20).

[Si rellena todos los campos y añade los objetos]

[Si el código del pedido no está en uso]

3a-. Se almacena el pedido y se le muestra un mensaje indicando pedido guardado (Ver Toast 21).

[Sino]

3b-. Se le muestra un mensaje indicando código de pedido en uso (Ver Toast 22).

[Sino]

3c-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** Se ha almacenado el pedido junto con el contenido y todos sus detalles.



**Pantalla 16:**  
Pantalla Gestionar almacén

**Pantalla 20:**  
Pantalla Crear pedido



Pedido creado

**Toast 21**

El código de pedido ya existe

**Toast 22**

Hay campos vacíos

**Toast 6**

## VER PEDIDOS

**Descripción:** El director desea ver todos los pedidos creados junto con su estado para su posterior edición o eliminación, si lo desea.

**Actores:** Director.

**Precondición:** Tener algún pedido creado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El director pincha sobre el botón “Ver pedidos” (*Ver Pantalla 16*).

2-. Selecciona en el desplegable el orden por el que se mostraran los pedidos y pulsa sobre el botón “Buscar”.

[*Si tiene pedidos creados*]

3<sup>a</sup>-. Se le muestra una tabla mostrando todos los códigos de los pedidos, el nombre de usuario del repartidor asignado, el estado del pedido y si el pedido está valorado (*Ver Pantalla 21*).

[*Sino*]

3b-. Se le muestra un dialogo indicando que no tiene pedidos y que primero debe crearlos (*Ver No tiene pedidos*).

**Poscondiciones:** El director ha observado todos los pedidos que ha creado.

**Pantalla 16:**  
Pantalla Gestionar almacén

**Pantalla 21:**  
Pantalla Ver pedidos

**No tienes pedidos**  
Crea primero algún pedido  
OK

**No tiene pedidos**

CODIGO DEL PEDIDO	USUARIO	PEDIDOS VALORADOS	ESTADO DEL PEDIDO
1	jon	S	Finalizado
2	jon	N	En espera
3	jon	N	En espera
4	lucas	N	En espera

## EDITAR PEDIDOS

**Descripción:** El director desea editar detalles de los pedidos creados.

**Actores:** Director.

**Precondición:** Tener algún pedido creado

**Requisitos no funcionales:** Conexión a internet

**Flujo de eventos:**

1-. El director pincha sobre el código del pedido que desea editar (*Ver Pantalla 21*).

[Si el estado del pedido es "En espera"]

2a-. Se le muestra un dialogo indicando si desea eliminar o editar el pedido seleccionado (*Ver Dialogo 5*).

3a-. El director pincha sobre "Editar" y observará todos los datos del pedido para su modificación (*Ver Pantalla 22*).

[Si rellena todos los campos y hay algún objeto en la lista]

4aa-. El pedido es actualizado y se le muestra un mensaje indicando valores actualizados (*Ver Toast 11*).

[Sino]

4ab-. Se le muestra un mensaje indicando campos vacíos (*Ver Toast 6*).

[Sino]

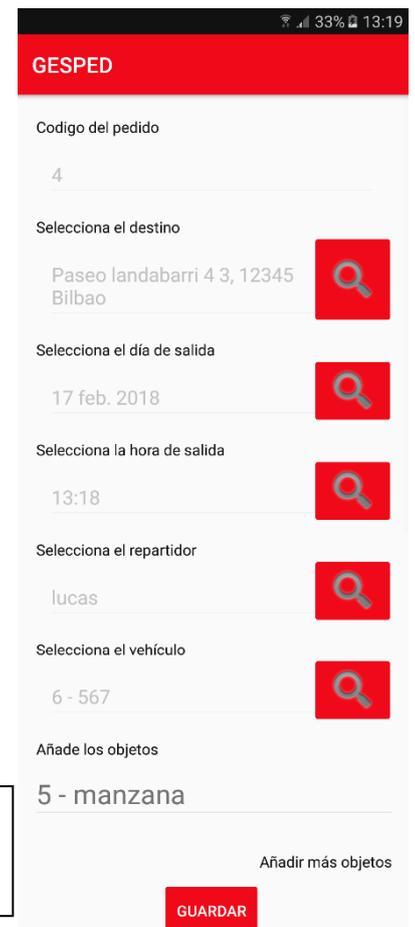
2b-. Se le muestra un dialogo si desea eliminar u observar los detalles del pedido (*Ver Dialogo 4*).

3b-. El director pincha sobre "Detalles" y se le muestra un dialogo mostrándole el día y la hora de salida. Además, en caso de que el estado del pedido sea "Finalizado", observará el día y la hora de llegada (*Ver Detalles de un pedido*).

**Poscondiciones:** El director ha editado el pedido seleccionado, o si el pedido no estaba en espera, ha observado las fechas del pedido.



**Pantalla 21:**  
Pantalla Ver pedidos



**Pantalla 22:**  
Pantalla Editar pedido

**Detalles del pedido**  
¿Que deseas hacer?  
**ELIMINAR** **DETALLES**

**Dialogo 4:**  
Eliminar o detalles  
un pedido

**Detalles del pedido**  
¿Que deseas hacer?  
**ELIMINAR** **EDITAR**

**Dialogo 5:**  
Eliminar o editar  
un pedido

Se han actualizado los valores

**Toast 11**

Hay campos vacíos

**Toast 6**

**Codigo del pedido: 2**  
Dia de salida: 13 feb. 2018  
Hora de salida: 11:52  
Dia de llegada: -  
Hora de llegada: -  
**OK**

**Codigo del pedido: 1**  
Dia de salida: 13 feb. 2018  
Hora de salida: 11:51  
Dia de llegada: 14 feb. 2018  
Hora de llegada: 18:35  
**OK**

**Detalles de un  
pedido**

## ELIMINAR PEDIDOS

**Descripción:** El director desea eliminar pedidos creados.

**Actores:** Director.

**Precondición:** Tener algún pedido creado.

**Requisitos no funcionales:** Conexión a internet.

**Flujo de eventos:**

- 1-. El director pincha sobre el código del pedido que desea eliminar (*Ver Pantalla 21*).
- 2-. Se le muestra un dialogo indicando si desea eliminar el pedido seleccionado (*Ver Dialogo 4 y Ver Dialogo 5*).
- 3-. El director pincha sobre la opción "Eliminar".
- 4-. Se le muestra un mensaje indicando que ha eliminado el pedido (*Ver Toast 23*).

**Poscondiciones:** El director ha eliminado el pedido y se han eliminado dichos valores de la base de datos.

CODIGO DEL PEDIDO	USUARIO	PEDIDOS VALORADOS	ESTADO DEL PEDIDO
1	jon	S	Finalizado
2	jon	N	En espera
3	jon	N	En espera
4	lucas	N	En espera

**Pantalla 21:**  
Pantalla Ver pedidos

El pedido ha sido eliminado

**Toast 23**

Detalles del pedido

¿Que deseas hacer?

ELIMINAR   DETALLES

**Dialogo 4:**  
Eliminar o detalles un pedido

Detalles del pedido

¿Que deseas hacer?

ELIMINAR   EDITAR

**Dialogo 5:**  
Eliminar o editar un pedido

Por último, se muestran los casos de uso para el usuario “trabajador”.

## OBSERVAR PEDIDOS ASIGNADOS

**Descripción:** El trabajador desea ver los pedidos que tiene asignados.

**Actores:** Trabajador.

**Precondición:** Haber iniciado sesión en la aplicación y tener algún pedido asignado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El trabajador selecciona la opción “Pedidos asignados” (*Ver Menú lateral: Trabajador*).
- 2-. Selecciona el orden por el que se mostraran los pedidos y pincha sobre el botón “Buscar”.

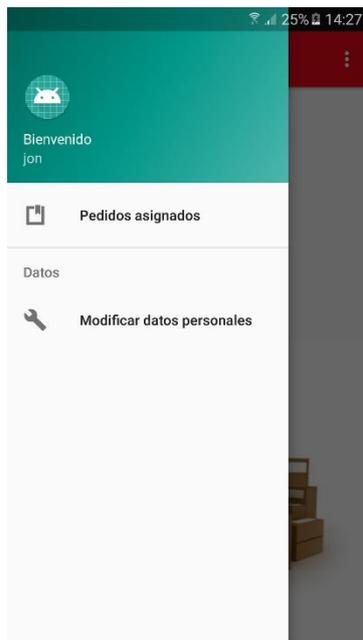
[*Si tiene pedidos asignados*]

3a-. Se le muestra una tabla mostrando los códigos de pedidos, día y hora de salida, el estado del pedido y la matrícula del vehículo con el que realizar el pedido. Además, se deshabilitaran los códigos de pedido cuyo estado sea “Finalizado” (*Ver Pantalla 23*).

[*Sino*]

3b-. Se le muestra un dialogo indicándole que no tiene pedidos asignados y que espere hasta que el jefe le asigne pedidos (*Ver No tiene pedidos asignados*).

**Poscondiciones:** El trabajador ha observado todos los pedidos que tiene asignados



**Menú lateral:  
Trabajador**



**Pantalla 23:  
Pantalla Ver  
pedidos asignados**

**No tienes pedidos**

Espera a que tu jefe te asigne algún pedido

OK

**No tiene pedidos asignados**

## INICIAR PEDIDO

**Descripción:** El trabajador desea iniciar el pedido para realizar la entrega.

**Actores:** Trabajador.

**Precondición:** Tener algún pedido asignado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El usuario pincha sobre un código de pedido cuyo estado sea "En espera" y se le muestra un dialogo con las opciones "Iniciar" y "Detalles". (*Ver Dialogo 6*).
- 2-. El trabajador pincha sobre "Iniciar ", el pedido será iniciado y se le mostrará un mensaje indicando que el pedido ha sido iniciado (*Ver Toast 24*).

**Poscondiciones:** El trabajador ha indicado que el pedido ha sido iniciado y procede a su entrega. Además, se actualizado el estado del pedido en la base de datos.



**Dialogo 6:** Iniciar o detalles un pedido asignado

## MOSTRAR RUTA PEDIDO

**Descripción:** El trabajador desea ver la ruta de entrega entre la empresa y el destino.

**Actores:** Trabajador.

**Precondición:** Tener algún pedido asignado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

- 1-. El trabajador pincha sobre un código de pedido y elige la opción "Detalles" (Ver Dialogo 6).
- 2-. Observa todos los datos del pedido junto con el contenido y pulsa el botón "Mostrar ruta" (Ver Pantalla 24).
- 3-. Se le muestra una pantalla con un mapa centrado sobre el origen y en el cual podrá hacer zoom para una mejor visualización de la ruta. Además, se dibuja sobre una línea la ruta que debe seguir en vehículo para llegar al destino (Ver Pantalla 25). Finalmente, se le muestra un mensaje mostrando la distancia en kilómetros o en metros (Ver Toast 25).

**Poscondiciones:** El trabajador ha visualizado el trayecto a seguir para la entrega.

Detalles del pedido

¿Que deseas hacer?

**INICIAR** **DETALLES**

**Dialogo 6:** Iniciar o detalles un pedido asignado

**Pantalla 25:** Pantalla Ver Mapa

**Pantalla 24:** Pantalla Ver detalles pedido asignado

Distancia: 9558,1 Metros

**Toast 25**

## FINALIZAR PEDIDO

**Descripción:** El trabajador ha entregado el pedido y desea indicar el pedido como finalizado.

**Actores:** Trabajador.

**Precondición:** Tener algún pedido asignado.

**Requisitos no funcionales:** Conexión a internet.

### Flujo de eventos:

1-. El trabajador pincha sobre un código de pedido en estado “Iniciado” y selecciona la opción “Finalizar” (Ver Dialogo 7).

2-. Introduce el día y la hora de entrega y los kilómetros recorridos. A continuación, pulsa sobre el botón “Guardar” (Ver Pantalla 26).

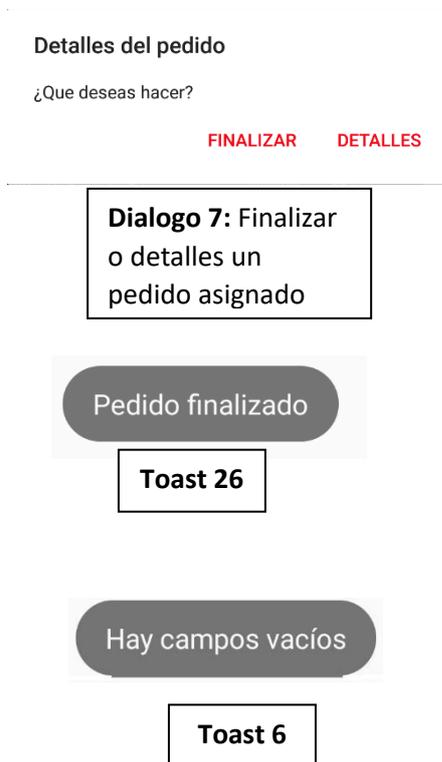
[Si rellena todos los campos]

3a-. El pedido ha finalizado y se guardan los datos introducidos. Además, se muestra un mensaje indicando pedido finalizado (Ver Toast 26).

[Sino]

3b-. Se le muestra un mensaje indicando campos vacíos (Ver Toast 6).

**Poscondiciones:** El pedido se ha actualizado como finalizado y se han almacenado el día y la hora de entrega. Además, se ha aumentado los kilómetros recorridos por el vehículo.



**Pantalla 26: Pantalla Datos pedido**

La pantalla muestra el formulario de datos para finalizar un pedido. El encabezado es "GESPED". El formulario incluye:

- Código del pedido: 2
- Selección del día de salida: campo de texto con un botón de búsqueda.
- Selección de la hora de salida: campo de texto con un botón de búsqueda.
- Introducción de los kilómetros recorridos del vehículo: campo de texto.
- Botón "GUARDAR" rojo en la parte inferior.



## 11-. Anexo 2 – Diagramas de secuencia

- *Modificar idioma*

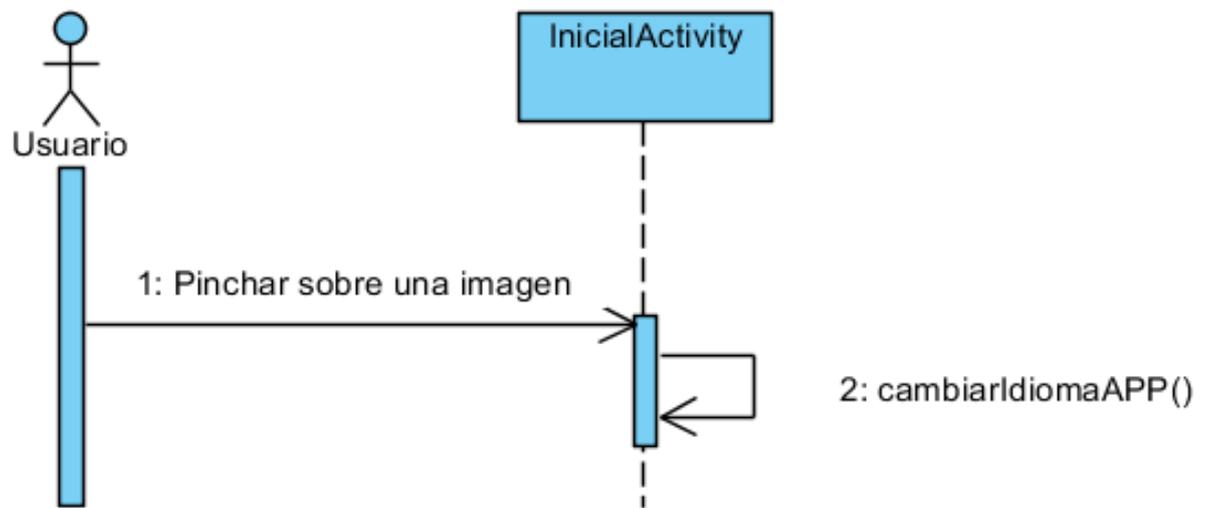


Ilustración 45: Diagrama de secuencia Modificar idioma

- *Calificar pedido*

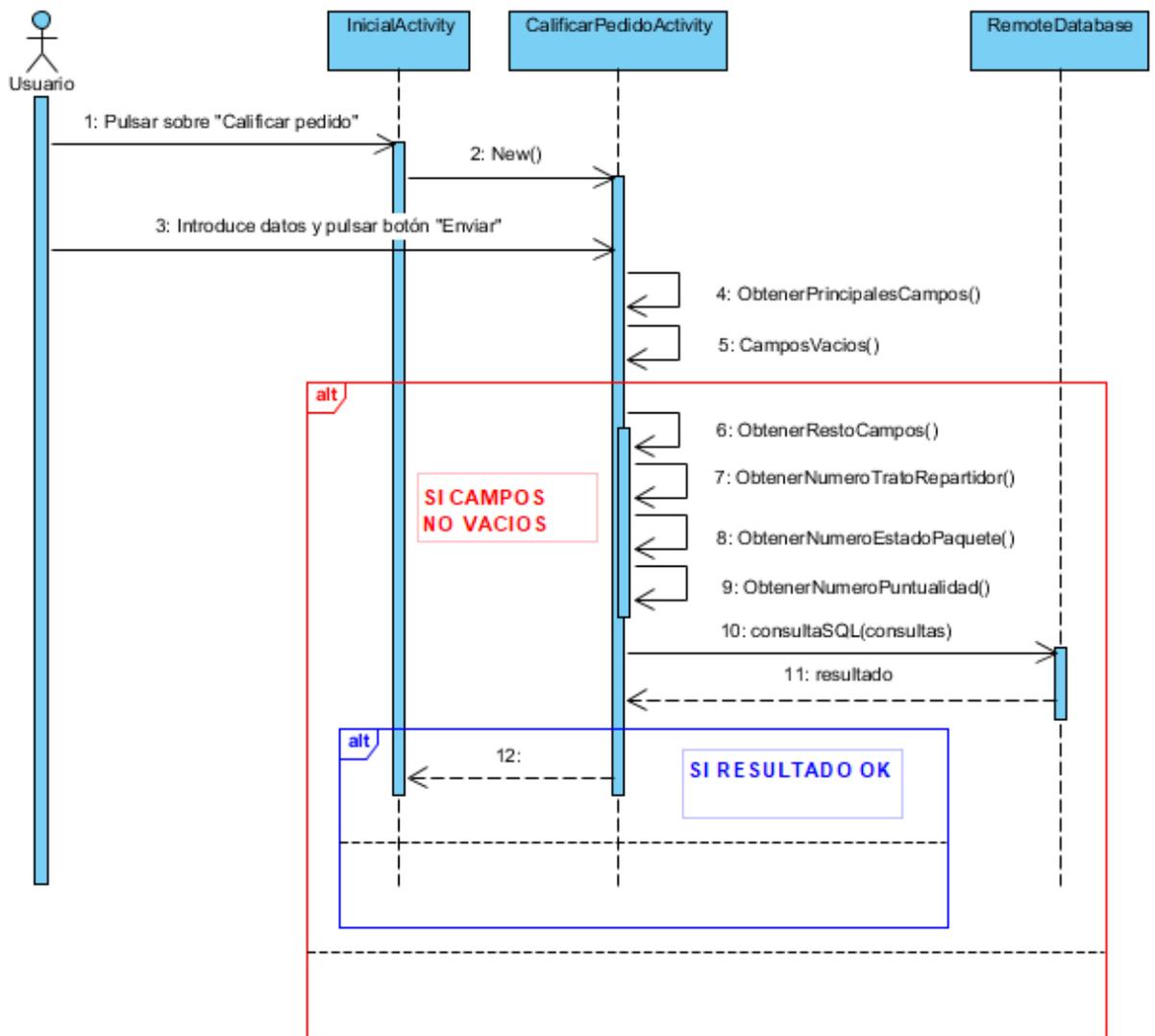


Ilustración 46: Diagrama de secuencia Calificar pedido

**consulta1:** SELECT nomUsuario  
FROM Usuario  
WHERE nomUsuario = \$repartidor;

Se ejecuta la consulta2 en el caso en el que haya algún usuario con ese nombre

**consulta2:** SELECT estadoPedido  
FROM Pedido  
WHERE codPedido = \$codPedido AND nomRepartidor = \$repartidor;

Se ejecuta la consulta3 en el caso de que exista el pedido y el estado sea "Finalizado"

**consulta3:** SELECT pedidoValorado  
FROM Pedido  
WHERE codPedido = \$codPedido AND nomRepartidor = \$repartidor;

Se ejecutan las siguientes consultas en el caso de que el pedidoValorado sea "N"

**consulta4:** UPDATE Pedido

```
Set pedidoValorado = 'S', tratoRepartidor = $tratoRepartidor,
estadoPaquete = $estadoPaquete, puntualidad = $puntualidad,
opini3n = $opinion
WHERE codPedido = $codPedido AND nomRepartidor = $repartidor;
```

**consulta5:** SELECT mediaTratoRepartidor, mediaEstadosPaquetes, mediaPuntualidad, cantidadPedidosValorados

```
FROM Usuario
WHERE nomUsuario = $repartidor;
```

**consulta6:** UPDATE Usuario

```
SET mediaTratoRepartidor = $mediaTratoRepartidorActualizada,
mediaEstadosPaquetes = $mediaEstadosPaquetesActualizada,
mediaPuntualidad = $mediaPuntualidadActualizada,
cantidadPedidosValorados = $cantidadPedidosValorados + 1
WHERE nomUsuario = $repartidor;
```

- *Identificarse*

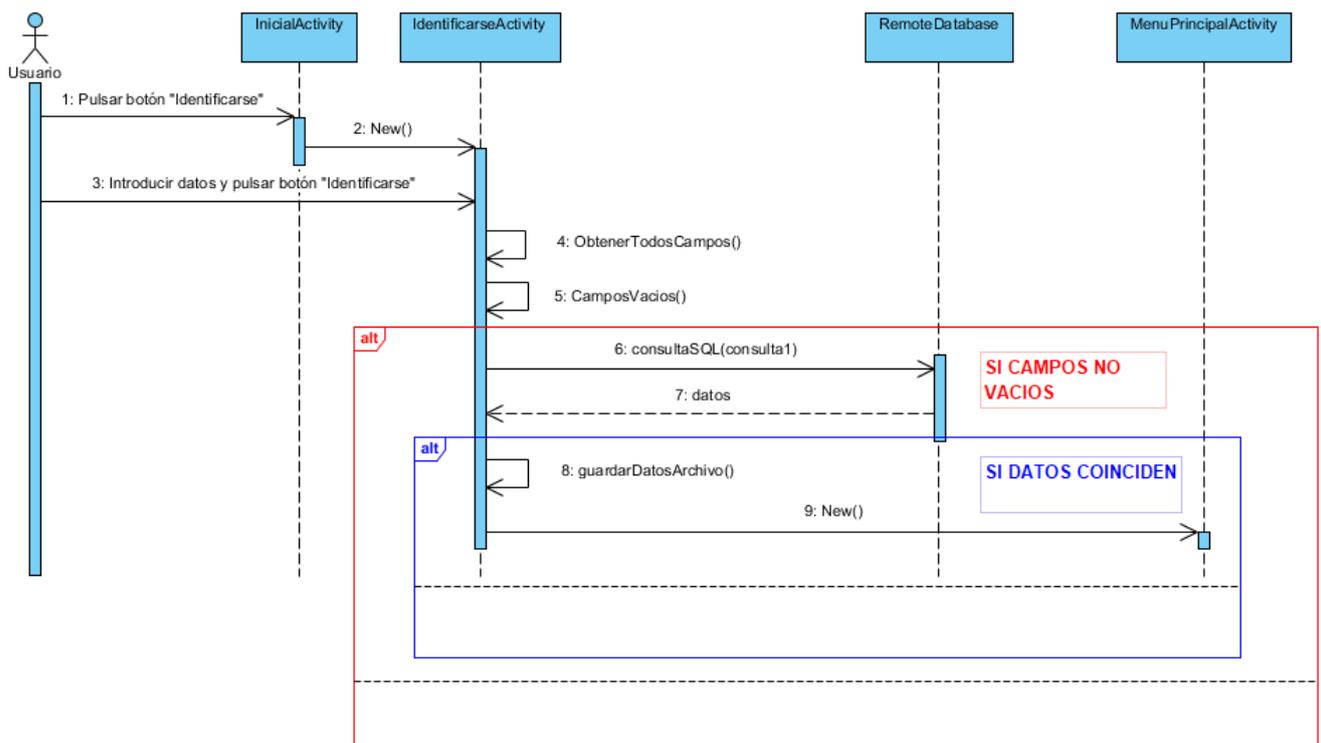


Ilustración 47: Diagrama de secuencia Identificarse

**consulta1:** SELECT contrasena, nomUsuarioJefe

```
FROM Usuario
WHERE nomUsuario = $user;
```

- *Recuperar contraseña*

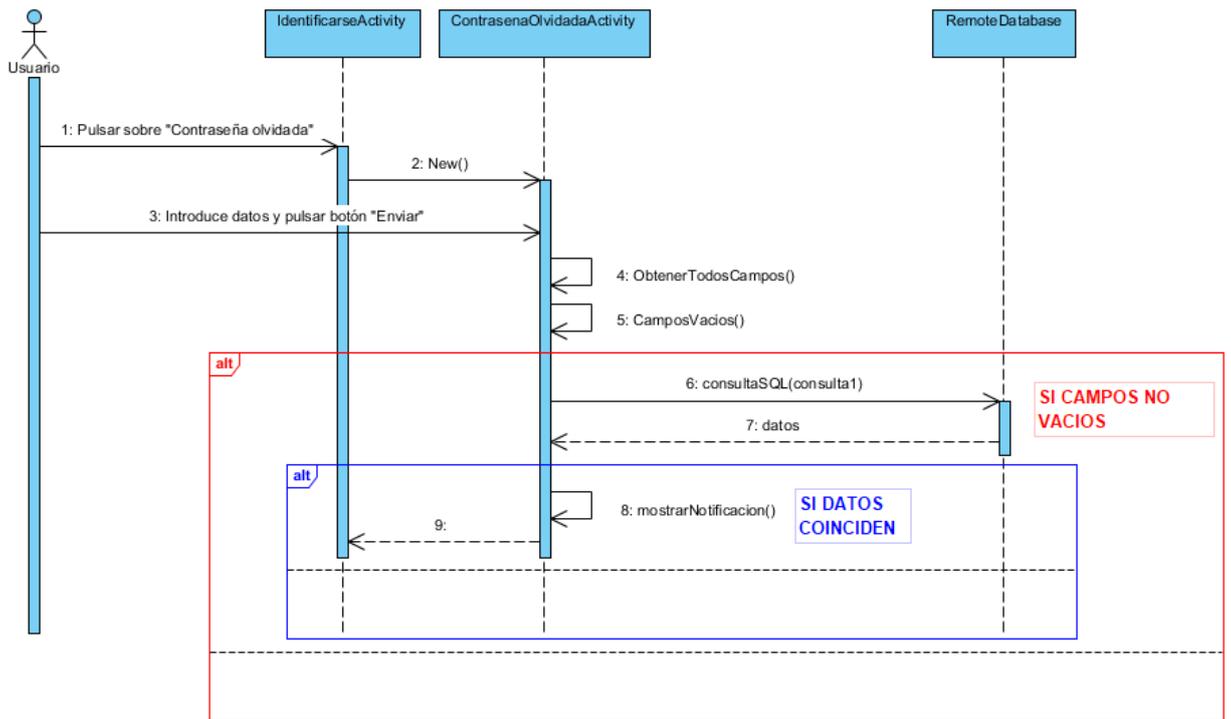


Ilustración 48: Diagrama de secuencia Recuperar contraseña

**consulta1:** SELECT contraseña, email  
 FROM Usuario  
 WHERE nomUsuario = \$nomUsuario;

- Registrarse como director

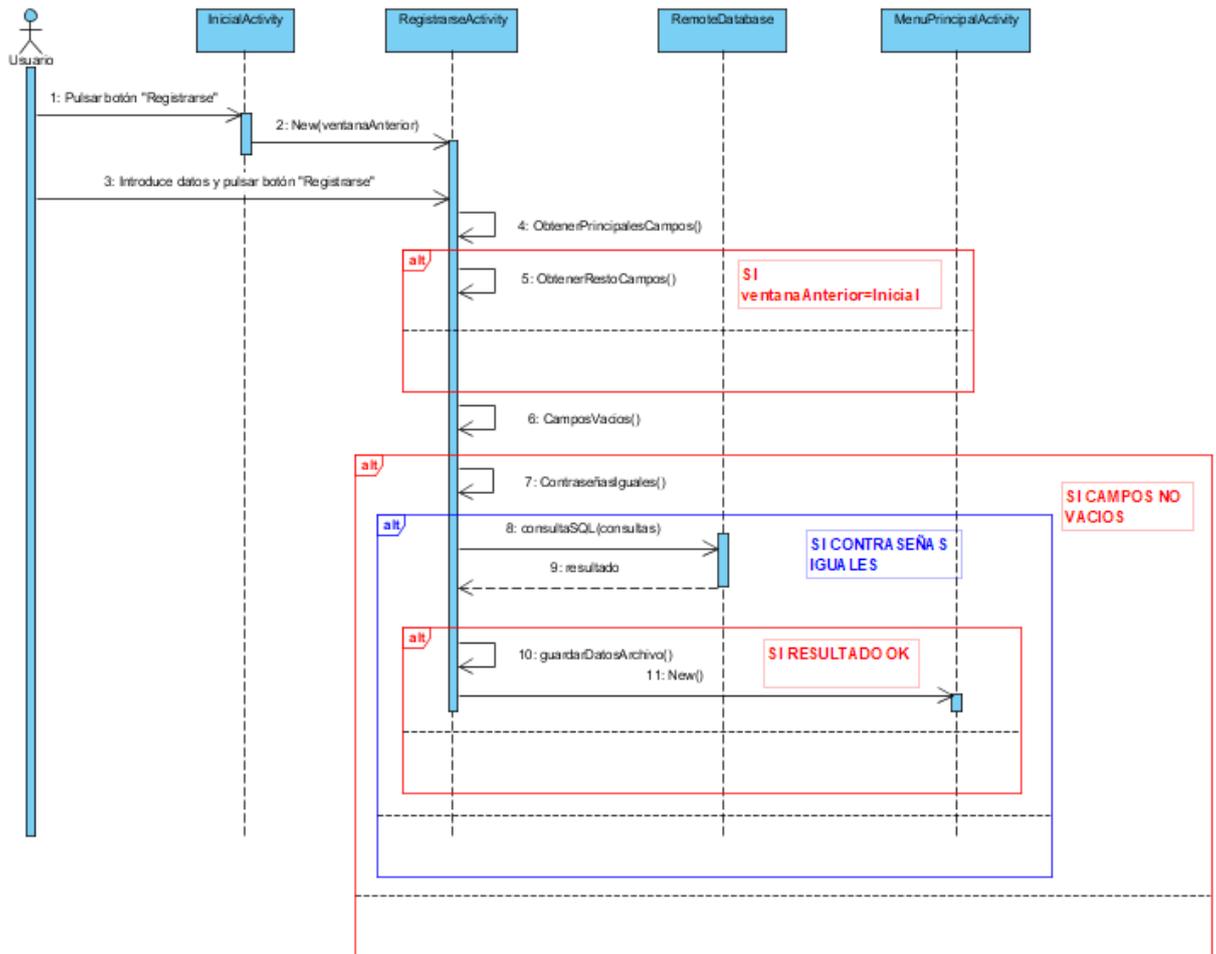


Ilustración 49: Diagrama de secuencia Registrarse como director

En este caso el valor de *ventanaAnterior* será *Inicial* ya que viene de esa ventana.

**consulta1:** SELECT nomUsuario  
FROM Usuario  
WHERE nomUsuario = \$nomUsuario;

Se ejecuta la consulta2 en el caso en el que no haya algún usuario con ese nombre

**consulta2:** INSERT INTO Usuario  
VALUES (\$nomUsuario, \$nombre, \$apellidos, \$contraseña, \$email, 'director',  
\$nomEmpresa, \$calle, \$piso, \$cp, \$municipio, \$numContacto,  
\$emailContacto, '0', '0', '0', '0', \$nomUsuario, '', '0');

- Cerrar sesión

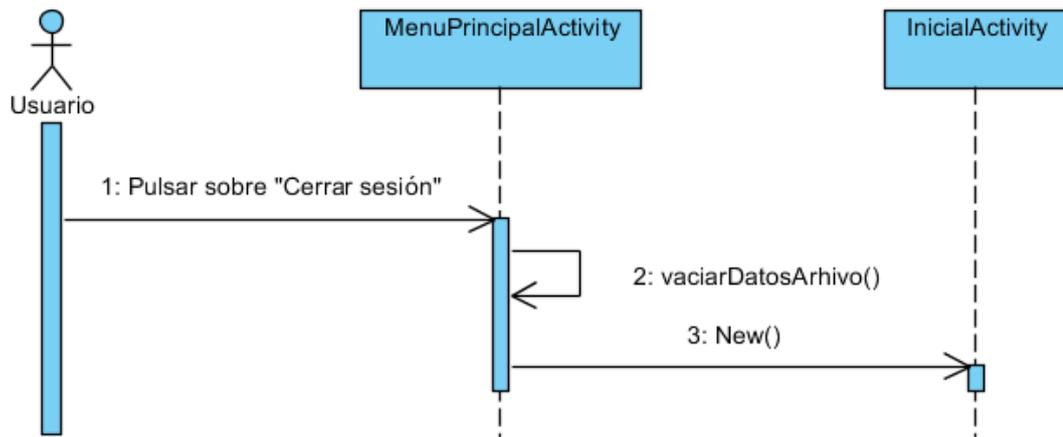


Ilustración 50: Diagrama de secuencia Cerrar sesión

- Modificar datos personales

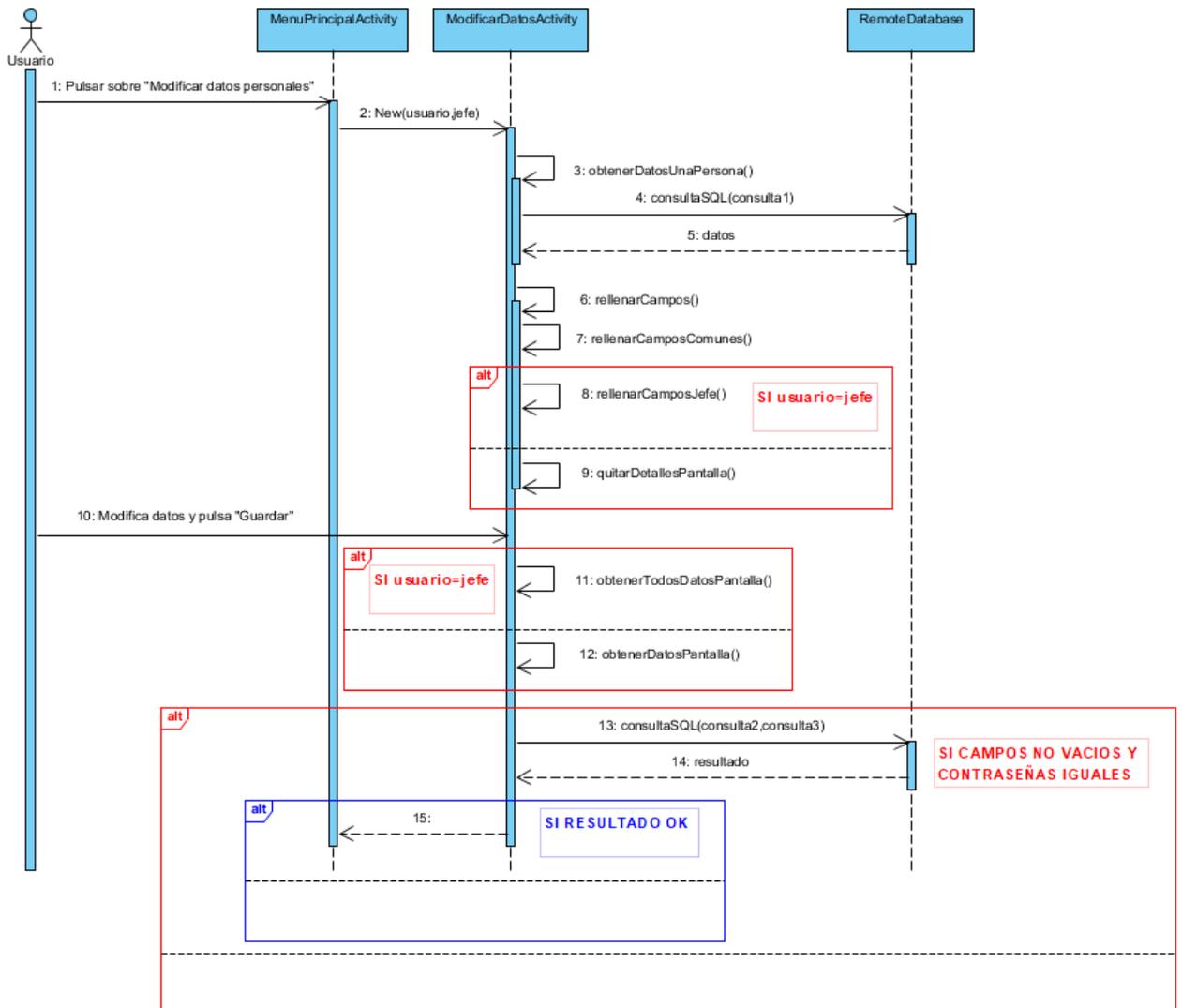


Ilustración 51: Diagrama de secuencia Modificar datos personales

**consulta1:** SELECT nombre, apellidos, contrasena, email, nomEmpresa, calle, piso,  
                  codigoPostal, municipio, numContacto, emailContactoEmpresa  
FROM Usuario  
WHERE nomUsuario = \$user;

Se ejecuta la consulta2 en el caso en el que el usuario sea jefe

**consulta2:** UPDATE Usuario  
              SET nombre = \$nombre, apellidos = \$apellidos, contrasena = \$contraseña,  
                  email = \$email, nomEmpresa = \$nomEmpresa, calle = \$calle,  
                  piso = \$piso, codigoPostal = \$codigoPostal, municipio = \$municipio,  
                  numContacto = \$numContacto,  
                  emailContactoEmpresa = \$emailContactoEmpresa  
WHERE nomUsuario = \$user;

Se ejecuta la consulta3 en el caso en el que el usuario no sea jefe

**consulta3:** UPDATE Usuario  
              SET nombre = \$nombre, apellidos = \$apellidos, contrasena = \$contraseña,  
                  email = \$email  
WHERE nomUsuario = \$user;

- Registrar trabajador

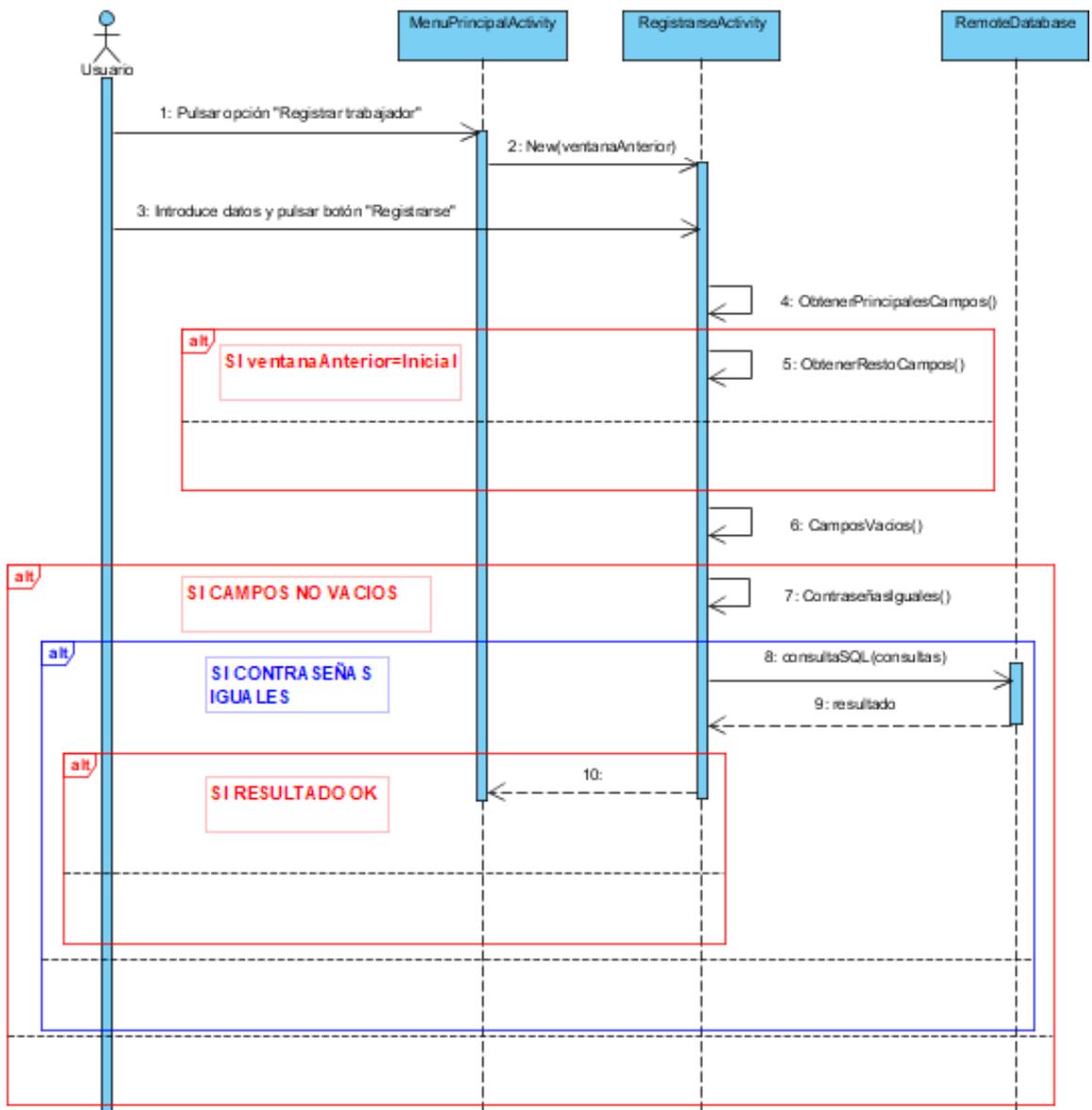


Ilustración 52: Diagrama de secuencia Registrar trabajador

En este caso el valor de *ventanaAnterior* será *Menú* ya que viene de esa ventana.

**consulta1:** SELECT nomUsuario  
FROM Usuario  
WHERE nomUsuario = \$nomUsuarioARegistrar;

Se ejecuta las siguientes consultas en el caso en el que no haya algún usuario con ese nombre.

**consulta2:** SELECT nomEmpresa, calle, piso, cp, municipio, numContacto,  
emailContacto  
FROM Usuario  
WHERE nomUsuario = \$nomUsuarioJefe;

**consulta3:** INSERT INTO Usuario

VALUES (\$nomUsuarioRegistrar, \$nombre, \$apellidos, \$contraseña, \$email, 'trabajador', nomEmpresa, calle, piso, cp, municipio, numContacto, emailContacto, '0', '0', '0', '0', \$nomUsuarioJefe, ' ', '0');

- Ver sus trabajadores

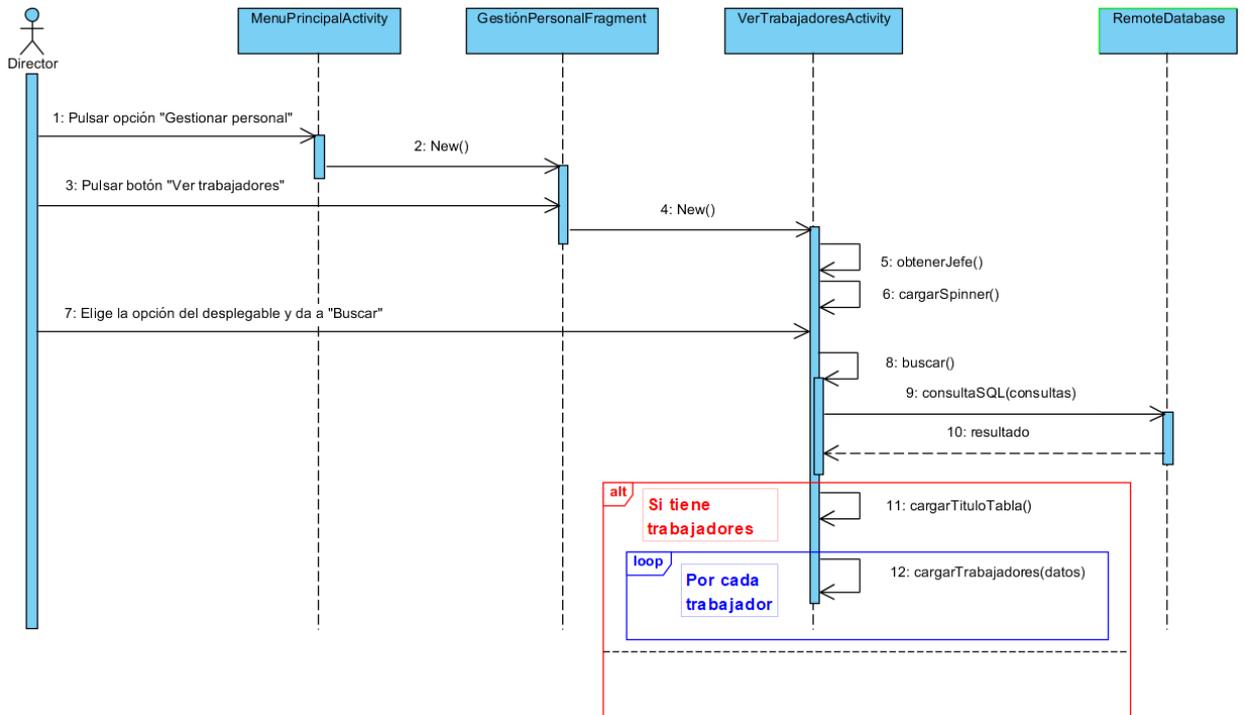


Ilustración 53: Diagrama de secuencia Ver sus trabajadores

Dependiendo de que opción elija el usuario en el desplegable, se realizará una consulta distinta.

Si selecciona "Usuario"

**consulta:** SELECT nomUsuario, nombre, apellidos  
FROM Usuario  
WHERE puesto = 'trabajador' AND nomUsuarioJefe = \$user  
ORDER BY nomUsuario;

Si selecciona "Nombre"

**consulta:** SELECT nomUsuario, nombre, apellidos  
FROM Usuario  
WHERE puesto = 'trabajador' AND nomUsuarioJefe = \$user  
ORDER BY nombre;

Si selecciona "Apellidos"

**consulta:** SELECT nomUsuario, nombre, apellidos  
FROM Usuario  
WHERE puesto = 'trabajador' AND nomUsuarioJefe = \$user  
ORDER BY apellidos;

- *Modificar datos de sus trabajadores*

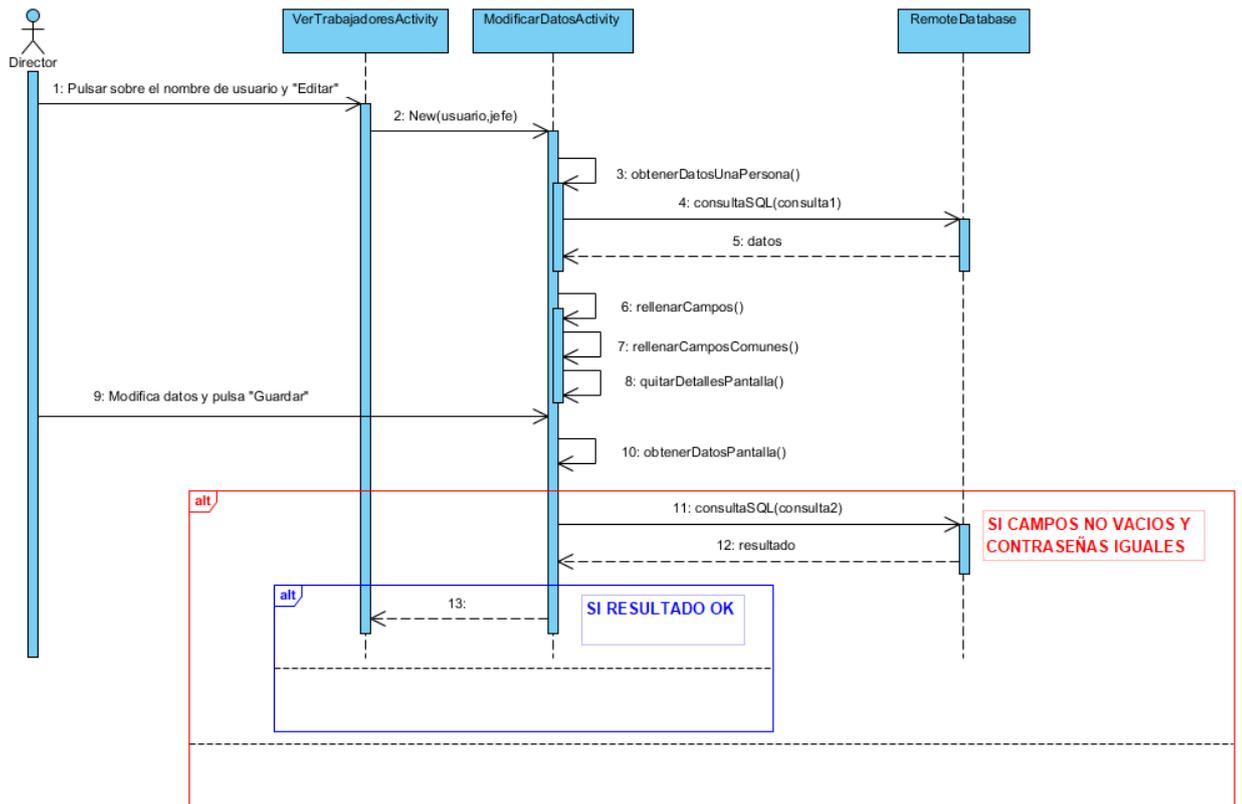


Ilustración 54: Diagrama de secuencia Modificar datos de sus trabajadores

**consulta1:** SELECT nombre, apellidos, contraseña, email, nomEmpresa, calle, piso, codigoPostal, municipio, numContacto, emailContactoEmpresa  
FROM Usuario  
WHERE nomUsuario = \$user;

**consulta2:** UPDATE Usuario  
SET nombre = \$nombre, apellidos = \$apellidos, contraseña = \$contraseña,  
email = \$email  
WHERE nomUsuario = \$user;

- *Eliminar trabajadores*

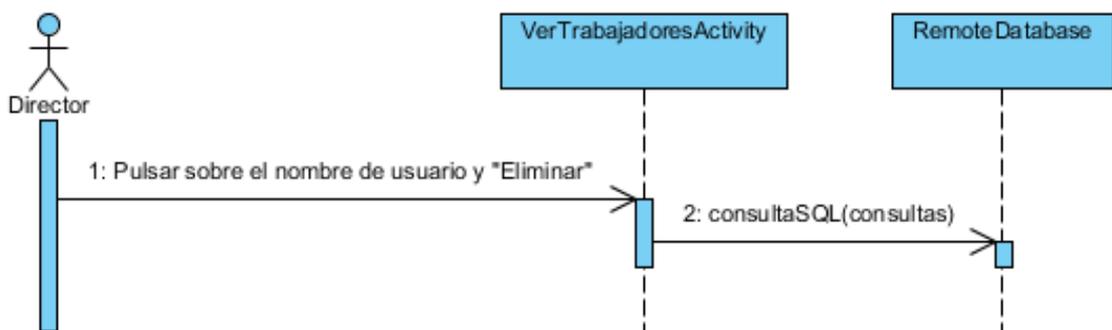


Ilustración 55: Diagrama de secuencia Eliminar trabajadores

**consulta1:** DELETE FROM Usuario  
WHERE nomUsuario = \$user;

**consulta2:** DELETE FROM Contenido  
WHERE nomRepartidor = \$user;

**consulta3:** DELETE FROM Pedido  
WHERE nomRepartidor = \$user;

- Ver estadísticas de pedidos

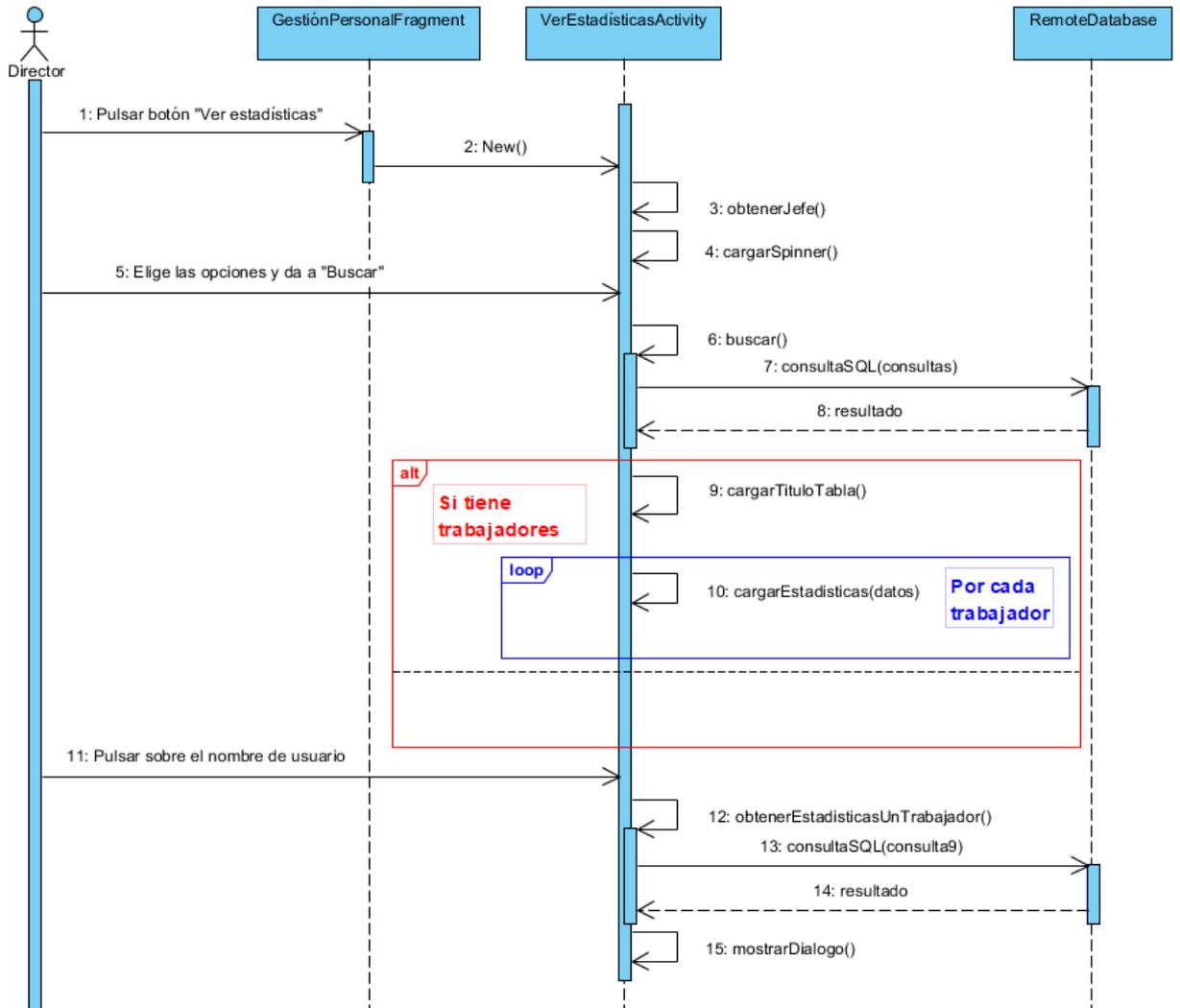


Ilustración 56: Diagrama de secuencia Ver estadísticas de pedidos

Dependiendo que opción elija el usuario en el desplegable y el orden según los checks, se realizará una consulta distinta.

Si selecciona “Trato con el repartidor”

Si selecciona “De mayor a menor”

**consulta1:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaTratoRepartidor DESC;

Si selecciona “De menor a mayor”

**consulta2:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaTratoRepartidor;

Si selecciona “Estado del paquete”

Si selecciona “De mayor a menor”

**consulta3:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaEstadosPaquetes DESC;

Si selecciona “De menor a mayor”

**consulta4:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaEstadosPaquetes;

Si selecciona “Puntualidad”

Si selecciona “De mayor a menor”

**consulta5:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaPuntualidad DESC;

Si selecciona “De menor a mayor”

**consulta6:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = ‘trabajador’ AND nomUsuarioJefe = \$user  
ORDER BY mediaPuntualidad;

Si selecciona “Pedidos realizados”

Si selecciona “De mayor a menor”

**consulta7:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = 'trabajador' AND nomUsuarioJefe = \$user  
ORDER BY cantidadPedidosRealizados DESC;

Si selecciona "De menor a mayor"

**consulta8:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE puesto = 'trabajador' AND nomUsuarioJefe = \$user  
ORDER BY cantidadPedidosRealizados;

La siguiente consulta se da cuando observa las estadísticas de un trabajador.

**consulta9:** SELECT nomUsuario, mediaTratoRepartidor, mediaEstadosPaquetes,  
mediaPuntualidad, cantidadPedidosRealizados  
FROM Usuario  
WHERE nomUsuario = \$trabajador;

- Crear vehículos

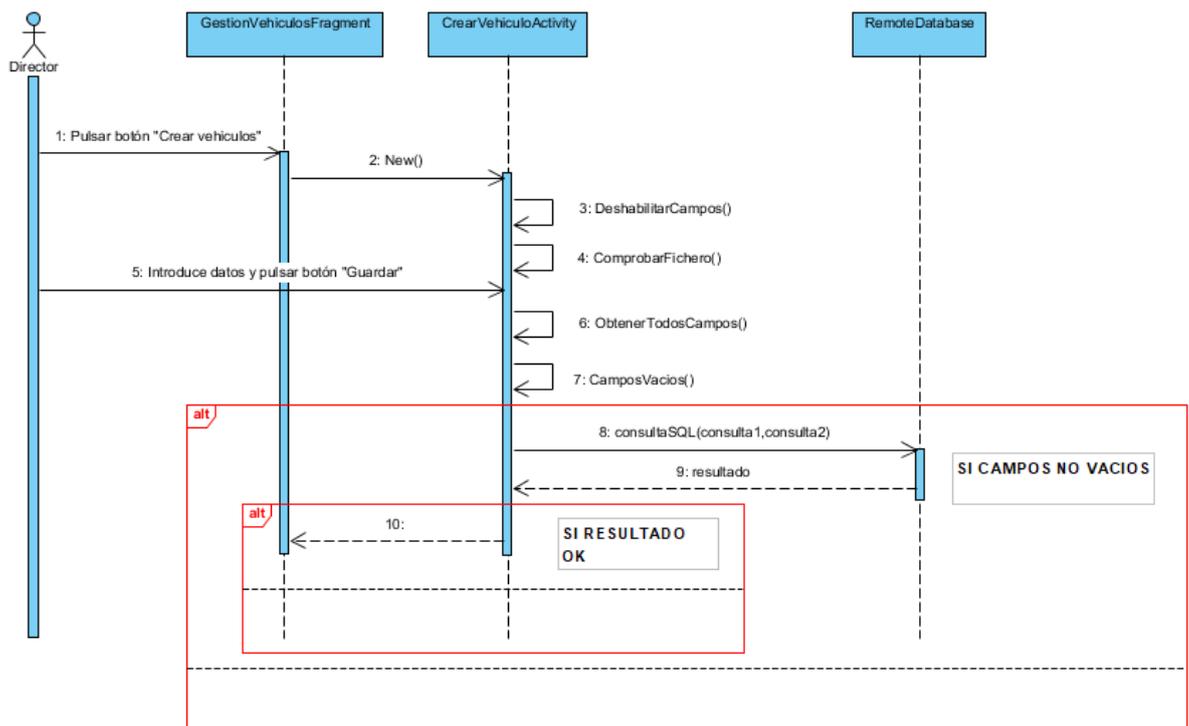


Ilustración 57: Diagrama de secuencia Crear vehículos

**consulta1:** SELECT matricula  
FROM Vehiculo  
WHERE matricula = \$matricula;

Se ejecuta la consulta2 en el caso en el que no haya ningún vehículo con esa matrícula

**consulta2:** INSERT INTO Vehiculo  
VALUES (\$matricula, \$modelo, \$años, \$ultimaITV, \$proximaITV,  
\$kmRecorridos, \$cargaKg, \$jefe);

- *Ver vehículos*

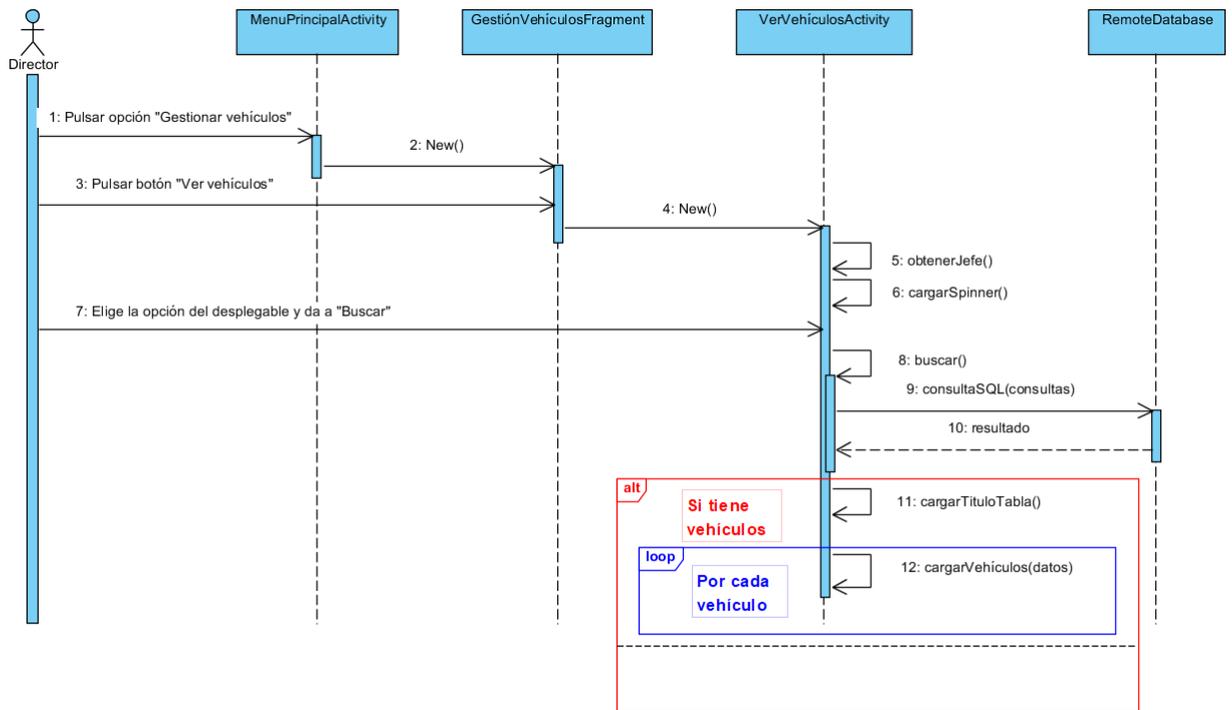


Ilustración 58: Diagrama de secuencia Ver vehículos

Dependiendo que opción elija el usuario en el desplegable, se realizará una consulta distinta.

**Si selecciona "Matrícula"**

**consulta:** SELECT matricula, modelo, kmRecorridos  
 FROM Vehiculo  
 WHERE nomUsuarioPropietario = \$user  
 ORDER BY matricula;

**Si selecciona "Modelo"**

**consulta:** SELECT matricula, modelo, kmRecorridos  
 FROM Vehiculo  
 WHERE nomUsuarioPropietario = \$user  
 ORDER BY modelo;

**Si selecciona "Kilómetros"**

**consulta:** SELECT matricula, modelo, kmRecorridos  
 FROM Vehiculo  
 WHERE nomUsuarioPropietario = \$user  
 ORDER BY kmRecorridos;

- *Modificar vehículos*

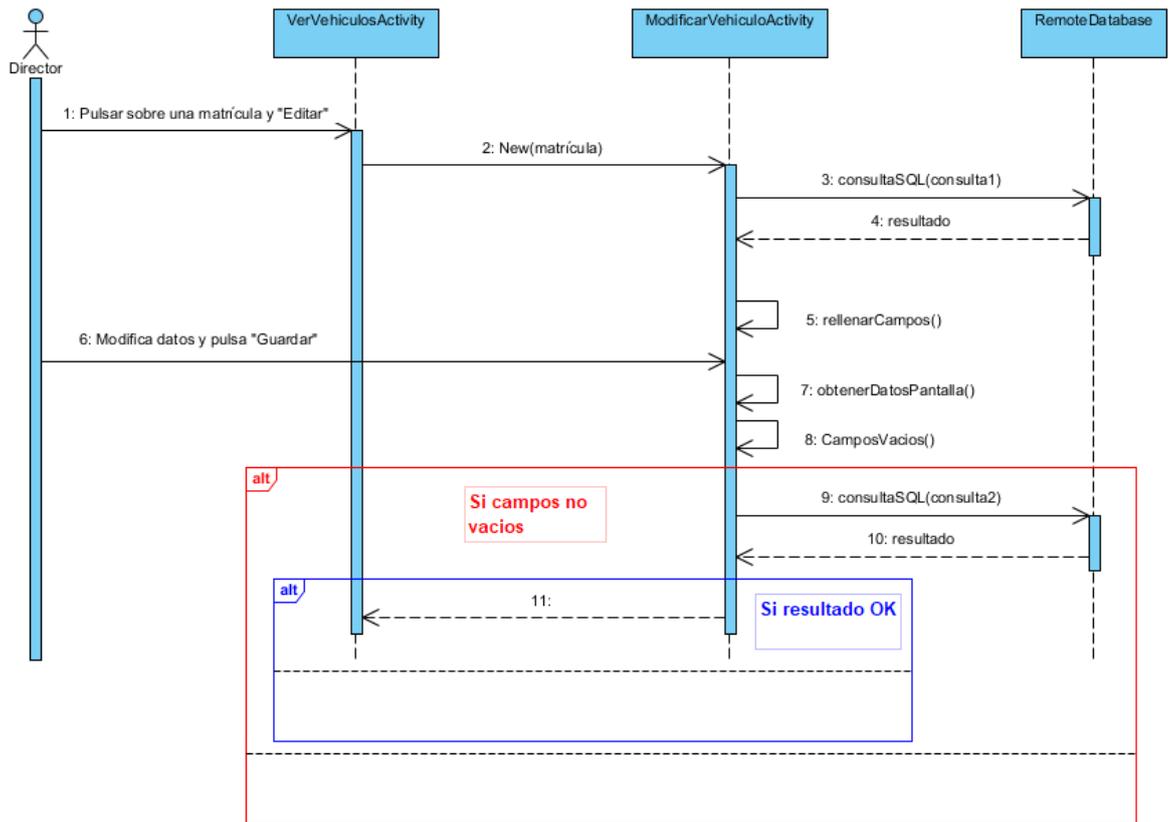


Ilustración 59: Diagrama de secuencia Modificar vehículos

**consulta1:** SELECT modelo, edad, ultimaITV, proximaITV, kmRecorridos, cargaKg  
FROM Vehiculo  
WHERE matricula = \$matricula;

**consulta2:** UPDATE Vehiculo  
SET modelo = \$modelo, edad = \$edad, ultimaITV = \$ultimaITV,  
proximaITV = \$proximaITV, kmRecorridos = \$ mRecorridos,  
cargaKg = \$cargaKg  
WHERE matricula = \$matricula;

- *Eliminar vehículos*

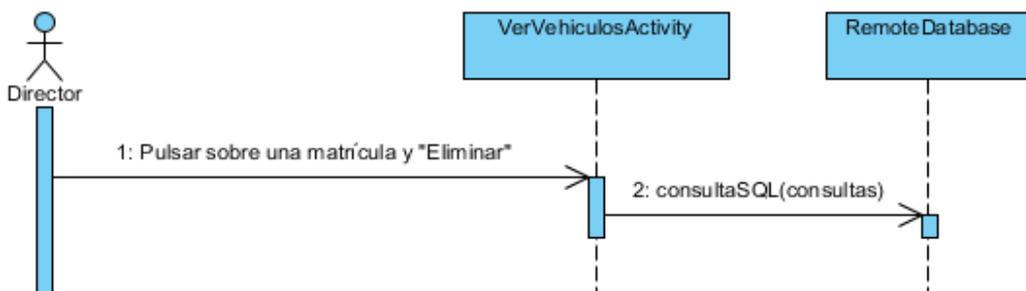


Ilustración 60: Diagrama de secuencia Eliminar vehículos

**consulta1:** DELETE FROM Vehiculo  
WHERE matricula = \$matricula;

**consulta2:** UPDATE Usuario  
 SET matriculaVehiculo = ''  
 WHERE matriculaVehiculo= \$ matricula;

**consulta3:** UPDATE Pedido  
 SET matriculaVehiculo = ''  
 WHERE matriculaVehiculo= \$ matricula;

- Almacenar objetos

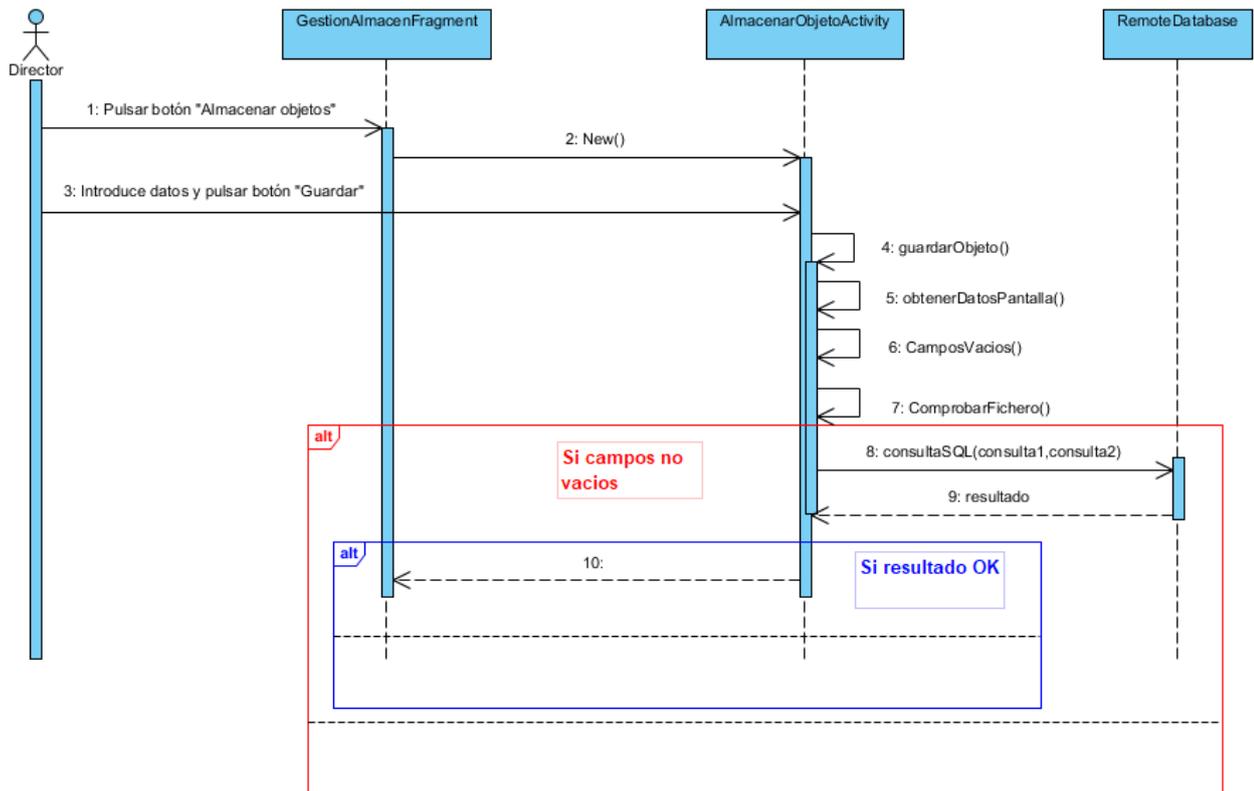


Ilustración 61: Diagrama de secuencia Almacenar objetos

**consulta1:** SELECT codObjeto  
 FROM Objeto  
 WHERE codObjeto = \$codObjeto;

Se ejecuta la consulta2 en el caso en el que no haya ningún objeto con ese código

**consulta2:** INSERT INTO Objeto  
 VALUES (\$codObjeto, \$descripcion, \$user);

- *Ver el almacén*

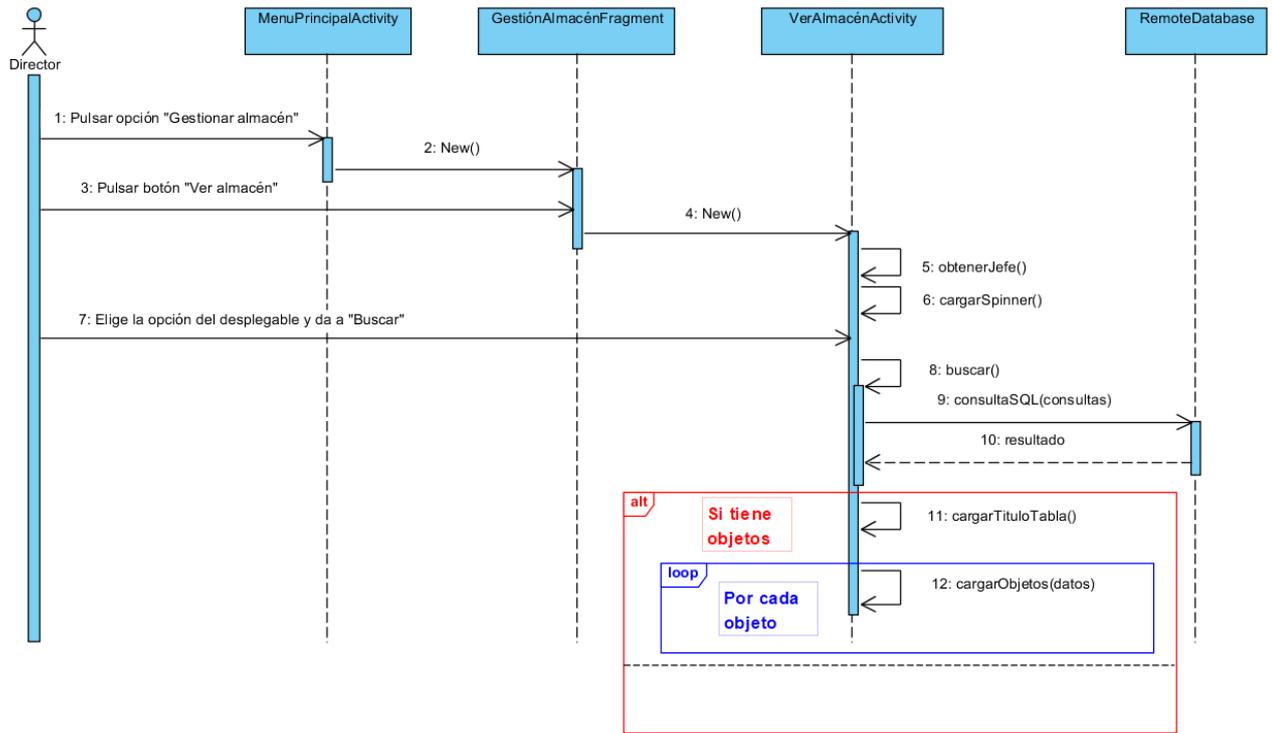


Ilustración 62: Diagrama de secuencia Ver el almacén

Dependiendo de qué opción elija el usuario en el desplegable, se realizará una consulta distinta.

*Si selecciona "Código"*

**consulta:** SELECT codObjeto, descripcion  
 FROM Objeto  
 WHERE nomPropietario = \$user  
 ORDER BY codObjeto;

*Si selecciona "Descripción"*

**consulta:** SELECT codObjeto, descripcion  
 FROM Objeto  
 WHERE nomPropietario = \$user  
 ORDER BY descripcion;

- *Editar Objetos*

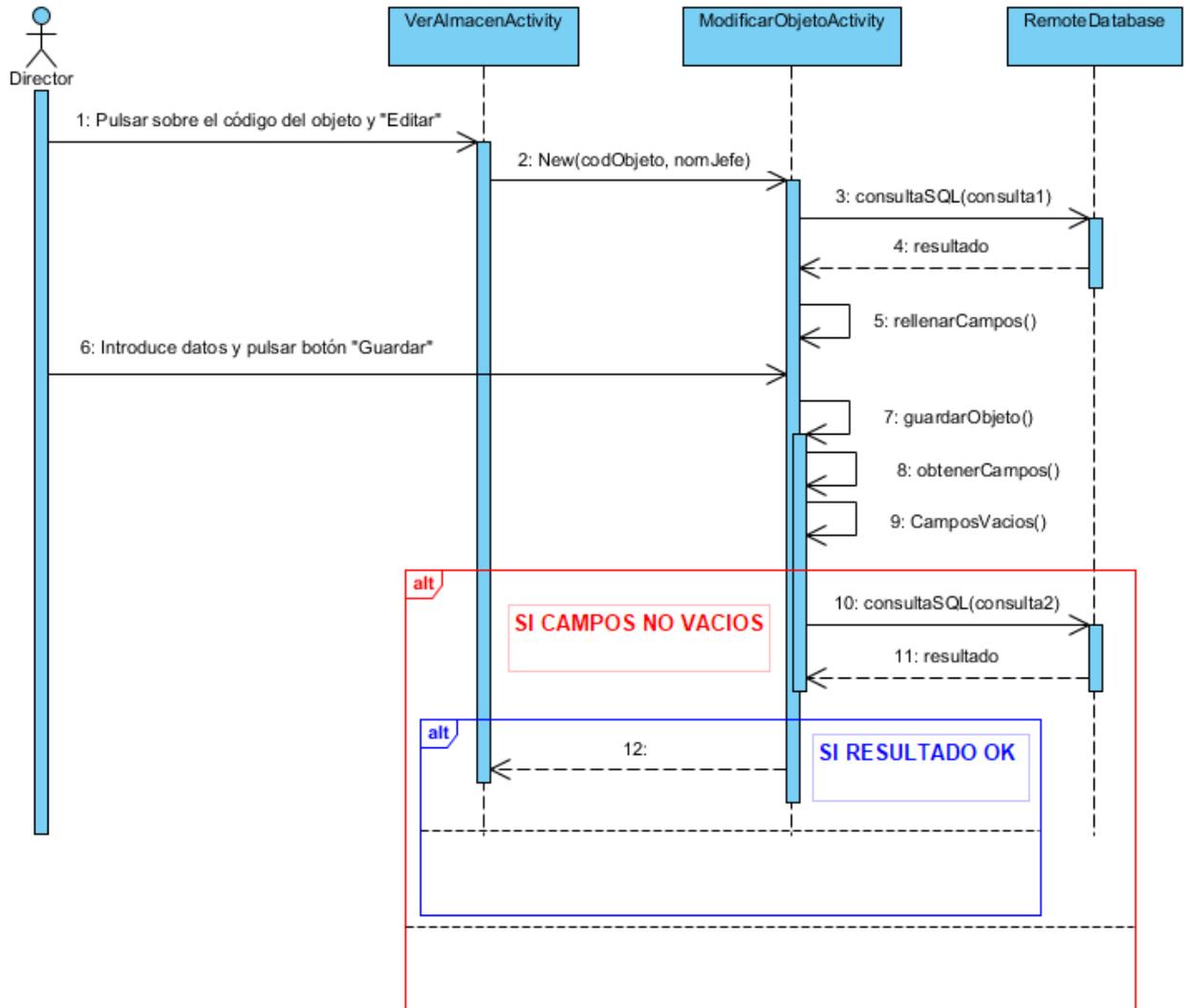


Ilustración 63: Diagrama de secuencia Editar objetos

**consulta1:** SELECT descripcion  
 FROM Objeto  
 WHERE codObjeto = \$codObjeto;

**consulta2:** UPDATE Objeto  
 SET descripcion = \$descripcion  
 WHERE codObjeto = \$codObjeto AND nomPropietario = \$user;

- *Eliminar objetos*

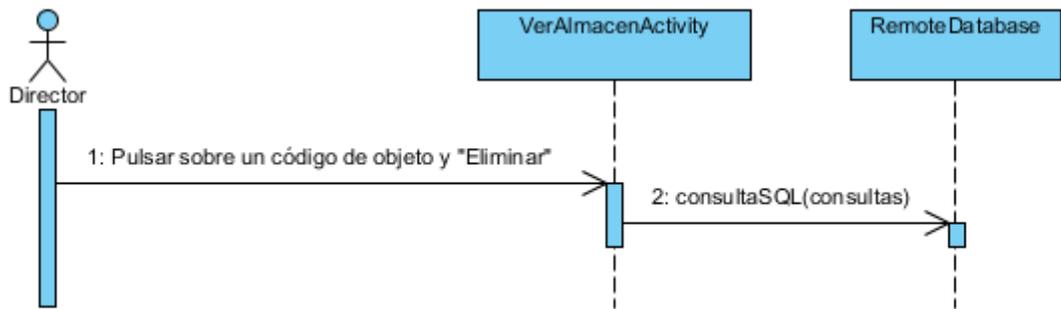


Ilustración 64: Diagrama de secuencia Eliminar objetos

**consulta1:** DELETE FROM Objeto  
 WHERE codObjeto = \$codObjeto AND nomPropietario = \$user;

**consulta2:** DELETE FROM Contenido  
 WHERE codObjeto = \$codObjeto AND nomJefe = \$user;

- *Crear pedidos*

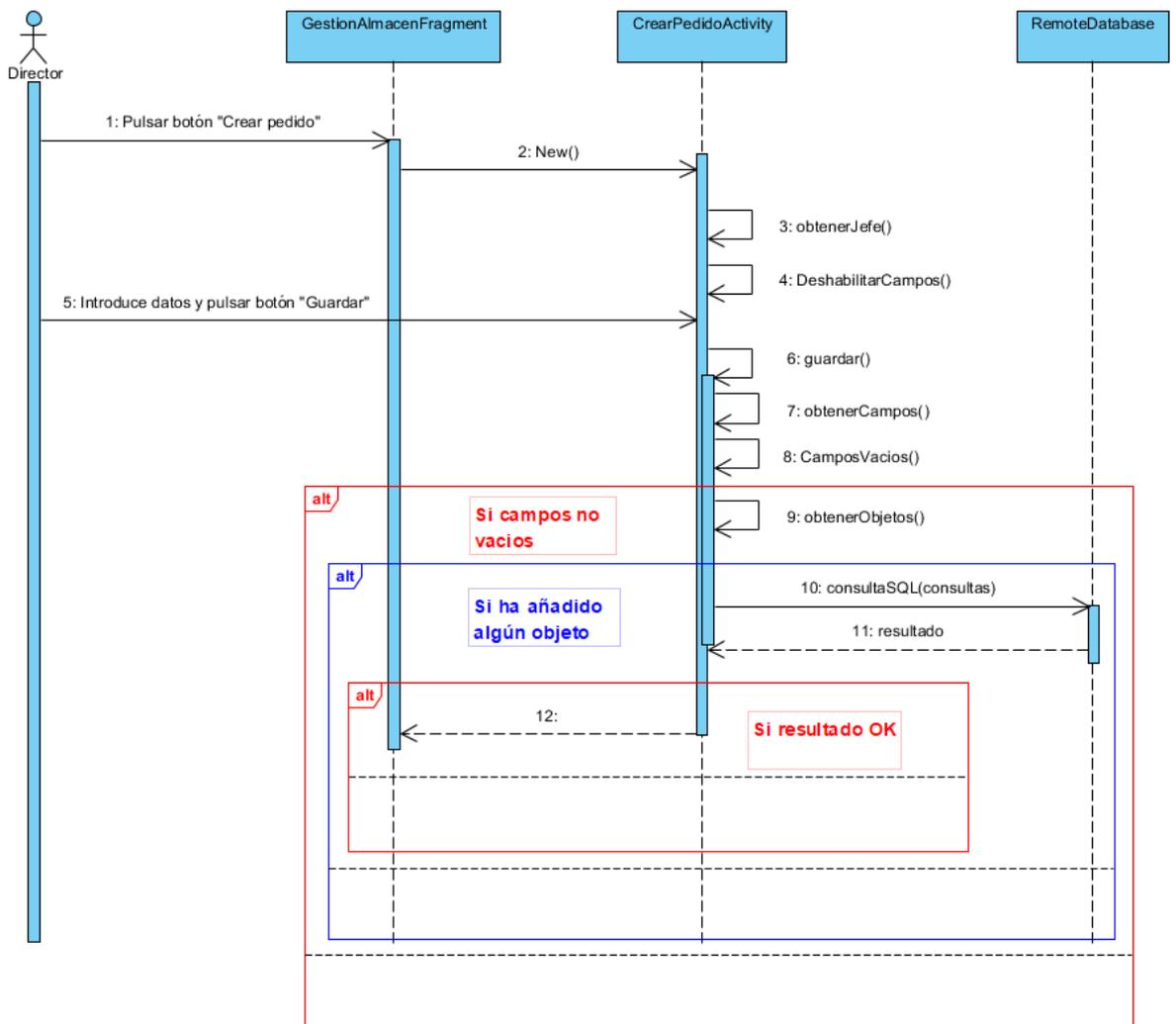


Ilustración 65: Diagrama de secuencia Crear pedidos

**consulta1:** SELECT codPedido  
FROM Pedido  
WHERE codPedido = \$codPedido;

Se ejecuta la consulta2 en el caso en el que no haya ningún pedido con ese código

**consulta2:** INSERT INTO Pedido  
VALUES (\$codPedido, \$calle, \$piso, \$codigoPostal, \$municipio, 'En espera',  
\$diaSalida, \$horaSalida, ' ', ' ', 'N', \$repartidor, '0', '0', '0', '0',  
\$vehiculo, \$jefe);

Se ejecutan las siguientes consultas por cada objeto del pedido

**consulta3:** SELECT cantidad  
FROM Contenido  
WHERE codPedido = \$codPedido AND codObjeto = \$todosObjetos[i] AND  
AND nomJefe = \$nomJefe AND nomRepartidor = \$nomRepartidor;

Se ejecuta la consulta4 en el caso en el que si haya alguna cantidad

**consulta4:** UPDATE Contenido  
SET cantidad = \$cantidad + 1  
WHERE codPedido = \$codPedido AND codObjeto = \$todosObjetos[i] AND  
AND nomJefe = \$nomJefe AND nomRepartidor = \$nomRepartidor;

Se ejecuta la consulta5 en el caso en el que no haya alguna cantidad

**consulta5:** INSERT INTO Contenido  
VALUES (\$codPedido, \$todosObjetos[i], \$jefe, \$repartidor);

- Ver pedidos

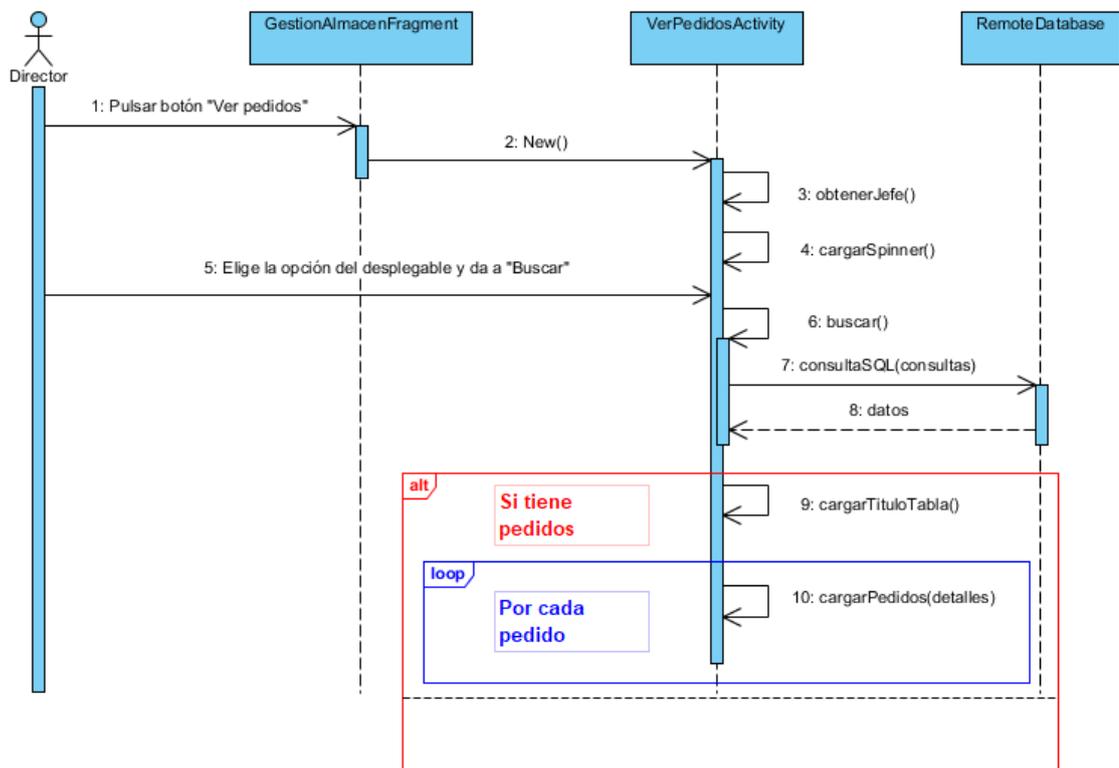


Ilustración 66: Diagrama de secuencia Ver pedidos

Dependiendo que opción elija el usuario en el desplegable, se realizará una consulta distinta.

Si selecciona “Código del pedido”

```
consulta: SELECT codPedido, nomRepartidor, pedidoValorado, estadoPedido
            FROM Pedido
            WHERE nomJefe = $user
            ORDER BY codPedido;
```

Si selecciona “Estado del pedido”

```
consulta: SELECT codPedido, nomRepartidor, pedidoValorado, estadoPedido
            FROM Pedido
            WHERE nomJefe = $user
            ORDER BY estadoPedido;
```

Si selecciona “Pedidos valorados”

```
consulta: SELECT codPedido, nomRepartidor, pedidoValorado, estadoPedido
            FROM Pedido
            WHERE nomJefe = $user
            ORDER BY pedidoValorado;
```

- *Editar pedidos*

Debido al gran tamaño de este diagrama de secuencia, se muestra por partes. Como ya se ha comentado en los casos de uso, dependiendo sobre que código de pedido pulse el director, visualizará diferentes cosas. De tal manera, se divide el diagrama en esas dos partes.

La primera parte se corresponde con la edición de un pedido que se encuentra en “En espera”.

La segunda parte se corresponde con la visualización de los detalles de un pedido que se encuentre “Iniciado” o “Finalizado”.

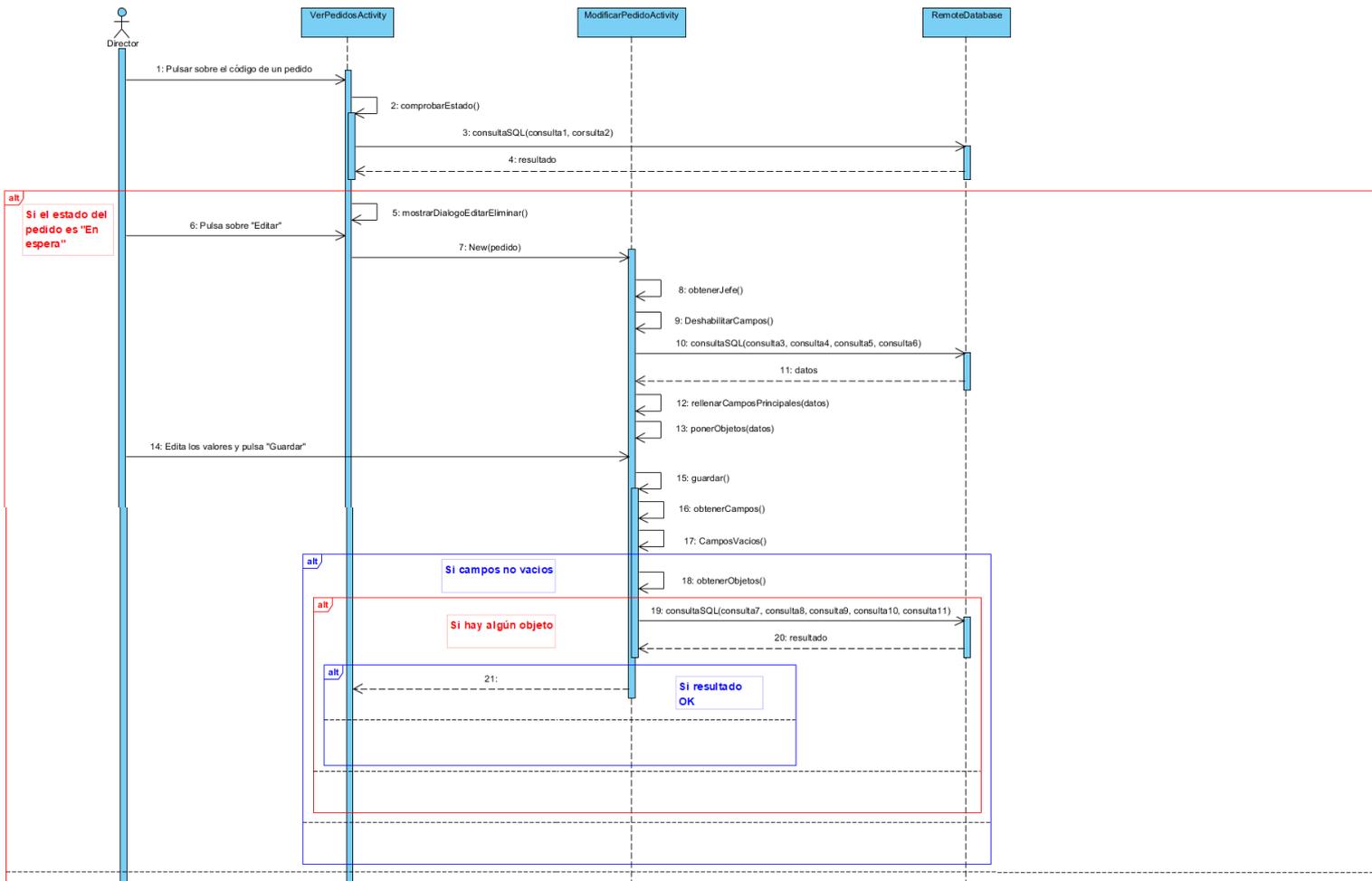


Ilustración 67: Diagrama de secuencia Editar pedidos - Parte 1

Como se puede observar, las dos primeras consultas son independientes de la acción que elija el usuario.

Estas dos consultas se corresponden con la obtención del estado del pedido teniendo en cuenta si es director o no:

Si es jefe

**consulta1:** SELECT estadoPedido  
FROM Pedido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

Si es repartidor

**consulta2:** SELECT estadoPedido  
FROM Pedido  
WHERE codPedido = \$pedido AND nomRepartidor = \$user;

Las siguientes cuatro consultas se corresponden con la obtención de todos los datos de un pedido.

**consulta3:** SELECT \*  
FROM Pedido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

**consulta4:** SELECT modelo  
FROM Vehiculo  
WHERE matricula = matricula;

**consulta5:** SELECT codObjeto, cantidad  
FROM Contenido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

Por cada objeto de la tabla contenido

**consulta6:** SELECT codObjeto, descripcion  
FROM Objeto  
WHERE codObjeto = codObjeto AND nomPropietario = \$user;

Las últimas cinco consultas de la primera se corresponden con la edición de los datos introducidos por el director.

**consulta7:** UPDATE Pedido  
SET calle = \$calle, piso = \$piso, codigoPostak = \$codigoPostal, municipio = \$municipio, diaSalida = \$diaSalida, horaSalida = \$horaSalida,  
nomRepartidor = \$nomRepartidor, matriculaVehiculo = \$matricula,  
WHERE codPedido = \$pedido AND nomJefe = \$user;

**consulta8:** DELETE FROM Contenido  
WHERE codPedido = \$pedido AND nomJefe = \$user AND nomRepartidor = \$repartidor;

En la consulta 8 se borran todos los objetos que contenía el pedido, para volverlos a insertar los que haya elegido el usuario. De tal manera, las siguientes tres consultas se repiten por cada objeto que contenga el pedido.

**consulta9:** SELECT cantidad  
FROM Contenido  
WHERE codPedido = \$pedido AND codObjeto = \$todosObjetos[i] AND  
nomJefe = \$user AND nomRepartidor = \$repartidor;

Si ya hay algún objeto con una cantidad

**consulta10:** UPDATE Contenido  
SET cantidad = \$cantidad +1  
WHERE codPedido = \$pedido AND codObjeto = \$todosObjetos[i] AND  
nomJefe = \$user AND nomRepartidor = \$repartidor;

Si no hay algún objeto con una cantidad

**consulta11:** INSERT INTO Contenido  
VALUES ( \$codPedido, \$todosObjetos[i], \$user, \$repartidor);

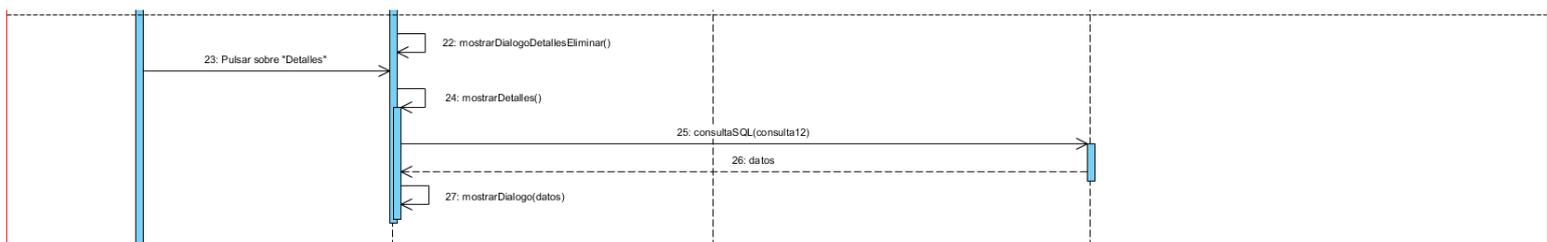


Ilustración 68: Diagrama de secuencia Editar pedidos - Parte 2

Este caso se da cuando el usuario pincha sobre un código de pedido que se encuentre “Iniciado” o “Finalizado. En ese caso, no podrá editar el pedido con lo que visualizará los detalles de este.

La consulta que se requiere para este caso es:

**consulta12:** SELECT codPedido, diaSalida, horaSalida, diaLlegada, horaLlegada  
FROM Pedido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

- *Eliminar pedidos*

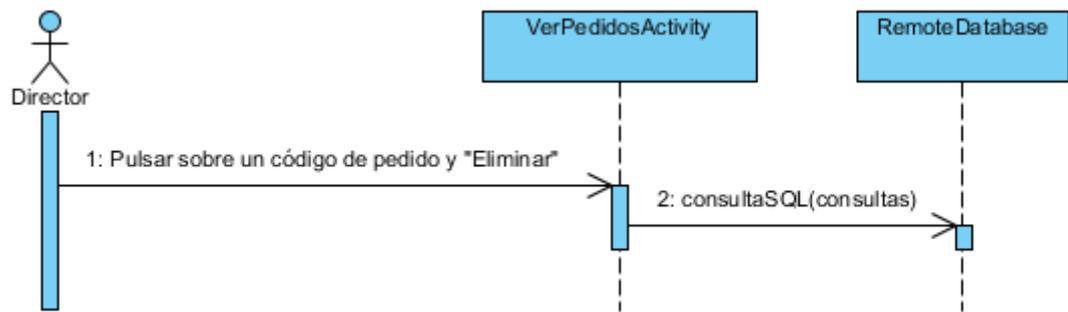


Ilustración 69: Diagrama de secuencia Eliminar pedidos

**consulta1:** DELETE FROM Pedido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

**consulta2:** DELETE FROM Contenido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

- *Observar pedidos asignados*

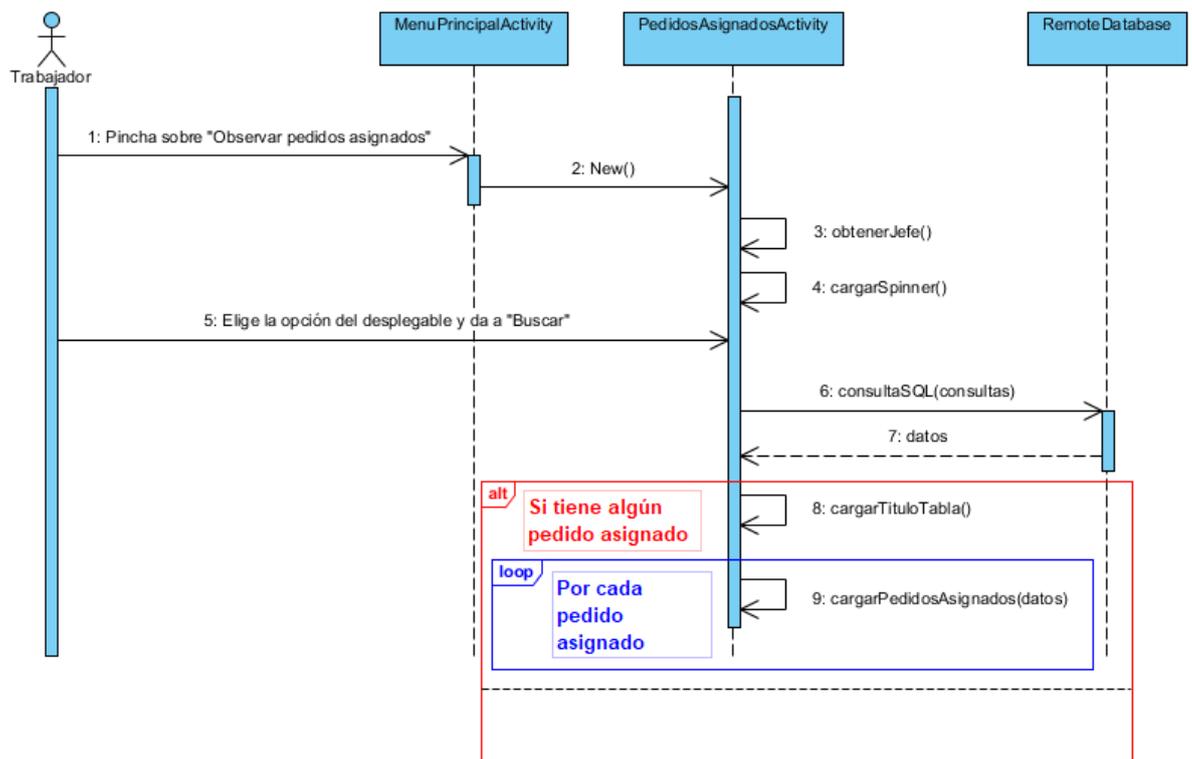


Ilustración 70: Diagrama de secuencia Observar pedidos asignados

Dependiendo que opción elija el usuario en el desplegable, se realizará una consulta distinta.

Si selecciona “Código del pedido”

```
consulta: SELECT codPedido, diaSalida, horaSalida, estadoPedido, matriculaVehiculo
FROM Pedido
WHERE nomRepartidor = $user
ORDER BY codPedido;
```

Si selecciona “Dia de salida”

```
consulta: SELECT codPedido, diaSalida, horaSalida, estadoPedido, matriculaVehiculo
FROM Pedido
WHERE nomRepartidor = $user
ORDER BY diaSalida;
```

Si selecciona “Estado del pedido”

```
consulta: SELECT codPedido, diaSalida, horaSalida, estadoPedido, matriculaVehiculo
FROM Pedido
WHERE nomRepartidor = $user
ORDER BY estadoPedido;
```

Si selecciona “Matrícula”

```
consulta: SELECT codPedido, diaSalida, horaSalida, estadoPedido, matriculaVehiculo
FROM Pedido
WHERE nomRepartidor = $user
ORDER BY matriculaVehiculo;
```

- *Iniciar pedido*

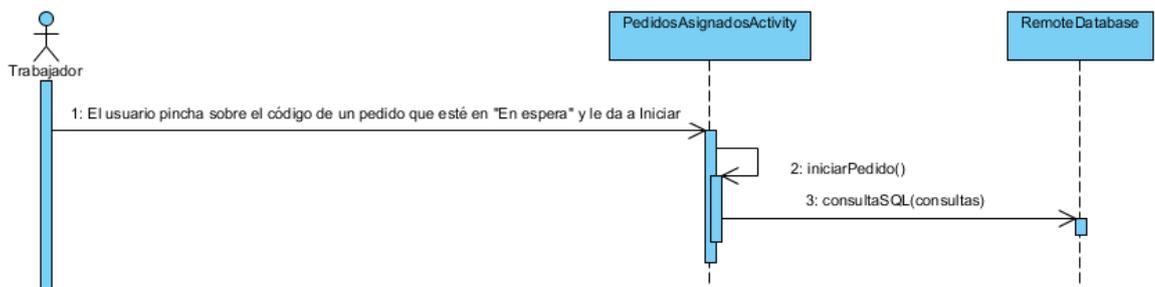


Ilustración 71: Diagrama de secuencia Iniciar pedido

```
consulta1: UPDATE Pedido
SET estadoPedido = 'Iniciado'
WHERE codPedido = $pedido AND nomRepartidor = $user;
```

```
consulta2: SELECT matriculaVehiculo
FROM Pedido
WHERE codPedido = $pedido AND nomRepartidor = $user;
```

```
consulta3: UPDATE Usuario
SET matriculaVehiculo = matriculaVehiculo
WHERE nomUsuario = $user;
```

- Mostrar ruta pedido

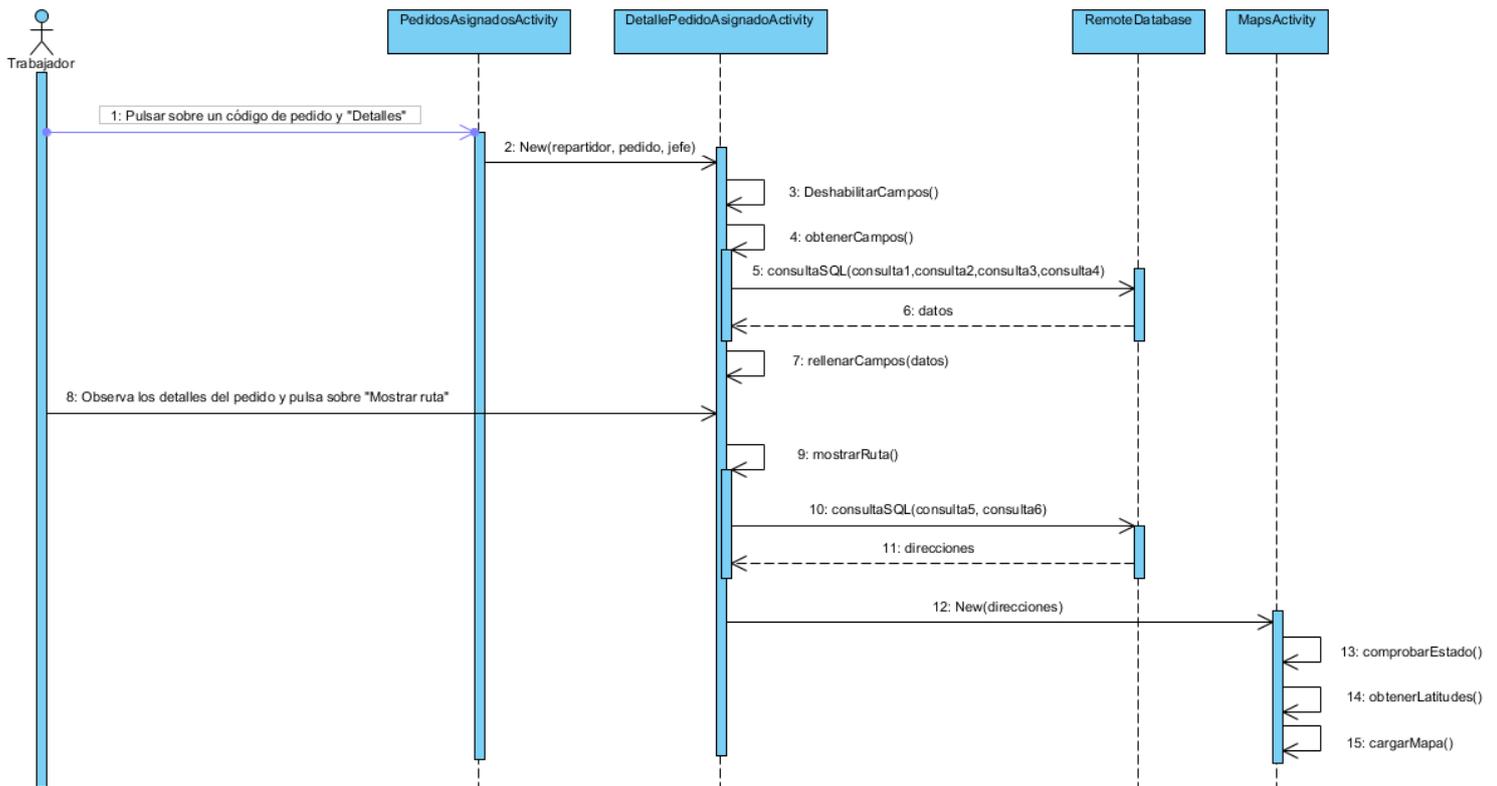


Ilustración 72: Diagrama de secuencia Mostrar ruta pedido

Las cuatro primeras consultas se corresponden a la obtención de todos los datos del pedido seleccionado por el usuario. Estas consultas son las siguientes:

**consulta1:** SELECT \*  
FROM Pedido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

**consulta2:** SELECT modelo  
FROM Vehiculo  
WHERE matricula = matricula;

**consulta3:** SELECT codObjeto, cantidad  
FROM Contenido  
WHERE codPedido = \$pedido AND nomJefe = \$user;

Por cada objeto de la tabla contenido

**consulta4:** SELECT codObjeto, descripcion  
FROM Objeto  
WHERE codObjeto = codObjeto AND nomPropietario = \$user;

Las dos últimas consultas son las que se realizan para obtener las direcciones de origen y de destino respectivamente.

**consulta5:** SELECT calle, municipio  
FROM Usuario  
WHERE nomUsuario = \$repartidor;

**consulta6:** SELECT calle, municipio  
FROM Pedido  
WHERE codPedido = \$pedido AND nomRepartidor = \$repartidor;

- *Finalizar pedido*

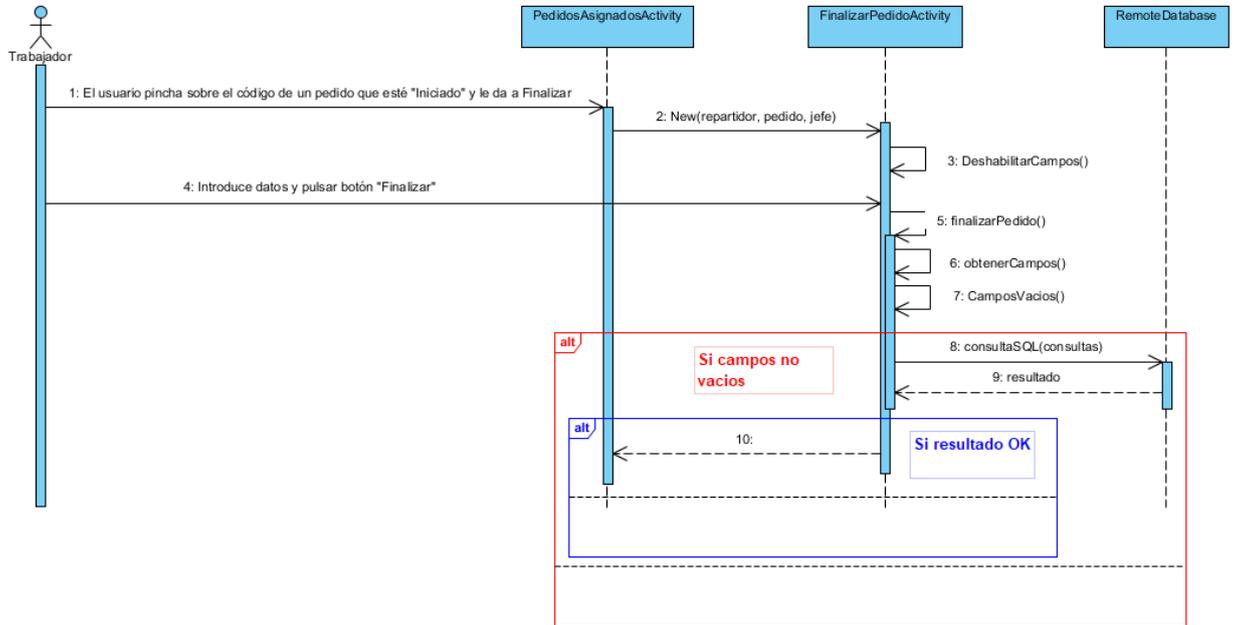


Ilustración 73: Diagrama de secuencia Finalizar pedido

**consulta1:** UPDATE Pedido  
SET diaLlegada = \$diaLlegada, horaLlegada = \$horaLlegada,  
estadoPedido = 'Finalizado'  
WHERE codPedido = \$pedido AND nomRepartidor = \$repartidor;

**consulta2:** SELECT matricula, kmRecorridos  
FROM Vehiculo  
WHERE nomUsuarioPropietario = \$jefe;

**consulta3:** UPDATE Usuario  
SET matriculaVehiculo = ''  
WHERE nomUsuario = \$repartidor;

**consulta4:** UPDATE Vehiculo  
SET kmRecorridos = kmRecorridos + \$kmRecorridos  
WHERE matricula = \$matricula AND nomUsuarioPropietario = \$jefe;

**consulta5:** SELECT cantidadPedidosRealizados  
FROM Usuario  
WHERE nomUsuario = \$repartidor;

**consulta6:** UPDATE Usuario  
SET cantidadPedidosRealizados = \$ cantidadPedidosRealizados + 1  
WHERE nomUsuario = \$repartidor;