

MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Multiplataforma

VehicleGest

Autor: Carlos Francisco Caruncho Serrano

Tutor: Mario Gago

Fecha de entrega: 05/12/2022

Convocatoria: 2º Semestre - 2022

Documentos del proyecto: [Enlace a la carpeta del Drive](#)



Tabla de contenido

1. INTRODUCCIÓN	3
1.1. Motivación	3
1.2. Abstract.....	4
2. METODOLOGÍA USADA.....	6
2.1. Ciclo de vida del proyecto. Fases	7
3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO	9
4. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN.....	10
5. ANÁLISIS DEL PROYECTO	11
6. DISEÑO DEL PROYECTO	21
7. DESPLIEGUE Y PRUEBAS	34
8. CONCLUSIONES	34
9. VÍAS FUTURAS.....	35
10. BIBLIOGRAFÍA	37

1. Introducción

Vehiclegest es una aplicación móvil desarrollada para el sistema operativo Android, dedicada a gestionar el control y mantenimiento de una flota de vehículos de transporte de carga y personas. Sus principales características serán: control y gestión de los datos e incidencias de una flota de vehículos, control de avisos importantes y fechas de vencimiento, consulta y gestión de las Inspecciones Técnicas de Vehículo, listado de servicios prestados por la empresa y control de asignaciones de personal a vehículos e ITV¹.

VehicleGest está diseñado para darle a la empresa un plus de productividad al eliminar los procesos basados en papel, pizarras, asegurando que los datos sean correctos y de fácil acceso desde cualquier parte del mundo, y en tiempo real. Ayudará a funcionar de manera más eficiente y reducirá los costes operativos. En resumen, mejorará la productividad, y al mejorar la productividad, aumentarán los ingresos.

1.1. Motivación

La motivación principal, surgió debido a la ocupación que tengo en mi puesto de trabajo. Soy responsable de una flota de vehículos de transporte, tanto de carga como de personas. Normalmente todos estos datos se almacenan en hojas de Excel, Word, o pizarras y no son actualizados en tiempo real, dando lugar a errores de datos o duplicidades.

Además, se ha dado el caso de no estar bien anotadas las fechas de inspección o no recordar que día son, así como pasarse la fecha por estar anotado en una hoja de Excel o pizarra que no nos avisa de ella, dando lugar a fallos en la logística que hacen perder tiempo y recursos.

De estas vicisitudes y teniendo en cuenta la gran cantidad de vehículos de que disponemos, surge la necesidad de tener un control de datos centralizado y en tiempo real, desde cualquier dispositivo móvil.

También es necesario tener un control centralizado de los inventarios de los vehículos. Dichos inventarios están también anotados en hojas de Excel, dificultando su control y auditorias.

Este proyecto puede ser una solución para estos problemas tanto en mi ámbito como en ámbito civil, para flotas de autobuses, camiones, vehículos de transporte de personas, etc.

¹ La ITV es una revisión periódica y obligatoria a los vehículos de motor, para mantener un una condiciones de seguridad mínimas y contribuir a mantener el medio ambiente. (¿QUÉ ES ITV? NORMATIVA Y TIPOS)

Los directivos y empleados tendrán todos los datos de la empresa en tiempo real y con datos centralizados en sus terminales móviles Android.

1.2. Abstract

VehicleGest is a complete Android solution to improve the control and maintenance of a logistics company's vehicle fleet, with a centralized database of vehicles, services, materials and personel, taking all this data wherever the company's personnel are.

In the past, where computers and telephones did not exist, all information was stored on paper. In recent decades, technology has evolved very quickly, as well as the way of storing data in businesses, companies and corporations. First of all, the information was stored in digital documents, but it is not productive for many reasons: low availability, inefficient teamwork, low consistency, low security, low reliability, among others. VehicleGest will bring some features to improve it. Firstly, it presents a login screen, because it must be secure. After that, vehicle list appears with some warnings, bringing user technical incident and ITV alerts to avoid errors and improve scheduling control. On the one hand, it notifies when the ITV dates are close or expired. On the other hand, it warns about vehicle technical deficiencies or whatever that affects to each vehicle. VehicleGest have a main menu on this screen bottom that let users navigate to the other sections.

The second section is a list of services provided by our company. It has a search bar with filters to search a service. Each service is associated by a vehicle or vehicle group, and employees.

The third section is an inventory of all the company's tools and where they are assigned. It has a search engine.

The fourth section is a a personel list section that has a search engine too. It be able to search single personel datasheet filtering by some criterias, and assign a person in charge of each vehicle.

The fifth section is a general list of alerts associated with each vehicle, about ITV and deficiencies.

In conclusion, VehicleGest will improve the productivity of the company deleting the paper-based processes ,saving time and ensuring data is correct and easily accessible. It helps work more efficiently reducing operating costs.

1.3. Objetivos propuestos (generales y específicos)

Los **objetivos generales** son:

- Digitalizar y centralizar en una base de datos todos los datos de los vehículos, ITV, material, personal y servicios de la empresa.
- Disponer de toda la información en cualquier lugar en los **terminales móviles Android** de los empleados y directivos de la empresa.
- Poder actualizar y controlar los datos de vehículos, servicios, material y personal asociados de los vehículos.

Los **objetivos específicos** son:

- Crear una base de datos en tiempo real online, evitándonos invertir en una infraestructura propia para la base de datos.
- Diseñar una interfaz de usuario con una usabilidad lo más elevada posible. Debe ser clara y sencilla, con un esquema de colores agradable a la vista.
- Los usuarios deben autenticarse mediante un sistema de usuario y contraseña, para mejorar la seguridad, privacidad y la integridad de los datos de la empresa.
- Las partes de la aplicación deben ser accesibles mediante un menú general, con iconos y nombres.
- Debe dar avisos generados al pasar las ITV por parte de los responsables directos de cada vehículo, de su estado de mantenimiento: niveles, estado de ruedas, limpieza, etc.
- Debe dar avisos de las fechas de vencimiento de las ITV, así como las fechas programadas para estas, y el resultado de cada inspección.
- Debe mostrar un listado de servicios realizados por cada uno de los empleados y los vehículos, pudiendo buscar un registro específico mediante un buscador con filtros.
- Debe mostrar un listado de vehículos que ha usado cada empleado, en que fechas, así saber también que servicios y que conductores ha tenido cada vehículo, pudiendo buscar un registro específico mediante un buscador con filtros.
- Debe mostrar un listado de inventario de herramientas general en la aplicación y otro individual en cada vehículo, pudiendo detectar que herramientas faltan o están deterioradas, pudiendo buscar un registro específico mediante un buscador con filtros.
- Debe disponer de un listado de personal de la empresa, pudiendo acceder a la ficha individual y los listados de servicios y vehículos asociados. Buscar un registro específico mediante un buscador con filtros.

2. Metodología usada

Cómo **metodología de flujo de trabajo** se ha utilizado la metodología ágil **Kanban**², combinada con la de **cascada con retroalimentación**.



Ilustración 1: Ciclos de vida del software. Modelo en cascada con retroalimentación. (IONOS, s.f.)

Combinando Kanban y cascada con retroalimentación surge un proceso **iterativo e incremental**. Con Kanban subdividimos el proyecto en pequeñas tareas individuales. Las tareas se representan en una pizarra con fichas o tarjetas, y por fases. (APD, 2021)

Las etapas de codificación y pruebas del ciclo de software se representan en el tablero kanban cómo “codificando” y “probando”, pudiendo comprobar la calidad del proyecto final a medida que se va desarrollando. (viewnext, s.f.) En el [Anexo I](#) se muestra una fotografía del tablero de corcho usado para el proyecto.

Las fases en las que se encontrarán las tareas y funcionalidades son:

- ❖ **Pendiente:** En este proyecto no se introducen ideas nuevas, las funcionalidades ya están predefinidas y no se introducirán nuevas. Estarán ya en la fase de pendiente desde el inicio del proyecto. En la fase de pendiente estarán en tarjetas individuales cada una de las tareas del proyecto y funcionalidades de las que consta la aplicación.

² Sistema ágil de gestión de proyectos en un tablero físico o digital, escribiendo las tareas en tarjetas dividiendo el tablero en fases, pasando cada una de las tareas por todas las fases.

- ❖ **En curso:** En la parte superior de esta fase se ponen las tareas que no sean de codificación. La parte inferior está subdividida en dos columnas, “**Codificando**” y “**Probando**”. Cuando una funcionalidad vaya a empezar a ser codificada entra automáticamente a la subfase de “Codificación”. Cuando se considera que está terminada la codificación, se mueve a la subfase de “Probando”.
- ❖ **Finalizado:** En la fase de “Probando” se le hacen las pruebas pertinentes y una vez pasadas se dará por terminada y se pasa a la fase de “Finalizado”.

2.1. Ciclo de vida del proyecto. Fases

1.1.1. Iniciación

Se ha **escogido la temática** para la realización del proyecto, es decir una aplicación mejorar la gestión de los vehículos de una empresa logística.

La necesidad a cubrir será mejorar la gestión y la eficiencia, y a su vez la productividad de la empresa. Los empleados con acceso a la aplicación deberán tener la información de todos los vehículos de la flota, de sus deficiencias técnicas, ITVs, material y servicios.

Se hace un **estudio de mercado** para ver la viabilidad de la aplicación, y se ha decidido que es perfectamente viable, ya que no existen aplicaciones parecidas y puede tener su segmento en el mercado.

1.1.2. Planificación

Se **definen las tecnologías se van a usar** en el [apartado 3](#).

Viabilidad: el proyecto es perfectamente viable en tiempo, ya que es un proyecto pequeño, que un programador puede realizar en un mes y medio.

Costes: estimamos los costes en función del tiempo utilizado para la realización del proyecto, desde que se aprueba en este caso la propuesta del mismo. Un programador junior³ cobra de media 1500 euros al mes, por lo que podríamos fijar un presupuesto inventado de 2000 euros.

³ Un programador junior es aquel que tiene menos de dos años de experiencia.

Se ha aplicado un plan de gestión de tiempo con el calendario y se ha transcrito a un **diagrama de Gantt**⁴, para dividir las tareas en el tiempo disponible, y pasarlas también al tablero Kanban. La información de esta fase se desarrolla en el [apartado 4](#) de esta memoria.

1.1.3. Ejecución

Se han ido completando las tareas, a medida que se van terminando, se va comprobando la calidad de los mismos antes de pasarlas a Finalizado.

Para las funcionalidades de la aplicación usamos las fases del método de cascada con retroalimentación ya mencionado.

- **Análisis:** En esta fase hemos definido los objetivos del software a partir de la idea principal, y lo hemos desglosado en tareas más pequeñas, definiendo los requisitos funcionales y no funcionales. Desarrollado en el [apartado 5](#).
- **Diseño:** Se ha decide cómo implementar el software y su estructura. Hemos pasado los requisitos funcionales y no funcionales a un diseño de software. Se han diseñado las estructuras de datos, la estructura de los componentes, la interfaz gráfica de la aplicación, y los componentes en detalle. Desarrollado en el [apartado 6](#).
- **Implementación:** Procedemos a desarrollar el código fuente, tanto de la interfaz con **XML**⁵, como de la lógica con Kotlin. Hemos seguido las convenciones y normas para escribir un código claro y legible. Además, hemos documentado el código a medida que lo vamos escribiendo, para su posterior mantenimiento de una forma más eficiente.
- **Pruebas:** Mediante las pruebas, hemos detectado los errores en la codificación, para corregirlos. Se dice que una prueba es un éxito si encuentra algún error.
- **Mantenimiento:** Esta fase no existe al ser un proyecto académico. Podemos considerar mantenimiento el perfeccionamiento de la aplicación del proyecto, la corrección de errores de programación. No se implementarán nuevas funcionalidades.

2.2. Supervisión

En la supervisión hemos se revisado a si estamos cumpliendo los plazos y los requisitos para la consecución de los objetivos. No se han ido cumpliendo como se planeó en el primer

⁴ Cronograma para gestionar el tiempo de los proyectos.

⁵ XML es un acrónimo de Extensible Markup Language, es un lenguaje de marcas que define reglas para estructurar un documento que contiene datos en texto plano.

diagrama de Gantt por lo que se han ido reduciendo objetivos y funcionalidades. Desarrollado en el [apartado 7](#).

2.3. Cierre

Se ha hecho un juicio crítico para ver que ha ido bien y que se hubiera podido mejorar, y documentarlo todo, se ha usado esta información para desarrollar el [apartado 8](#). Se han subido los archivos a la nube para su acceso y entrega. A partir de aquí entraríamos en mantenimiento si fuese un proyecto para un cliente, corrigiendo los bugs, errores y programando nuevas funcionalidades si fuera necesario, pero no es el caso, porque es un proyecto académico.

3. Tecnologías y herramientas utilizadas en el proyecto

3.1. Software de diseño vectorial - Microsoft Visio

Microsoft Visio es una herramienta de creación de diagramas y gráficos vectoriales⁶. Se ha usado para crear los diagramas de entidad-relación, diagrama relacional, diagrama de casos de uso, y diagrama de clases. He elegido esta herramienta por estar ya familiarizado con ella para hacer planos vectoriales, además es bastante sencilla de utilizar. (Lucidchart)

3.2. Entorno de desarrollo integrado – IDE - Android Studio

Para este proyecto, y tratándose una aplicación nativa para el sistema operativo Android, la mejor elección es **Android Studio**. Se puede programar tanto en Kotlin como en Java. (Wikipedia, 2022). Es el mejor IDE ya que está desarrollado expresamente para el desarrollo de aplicaciones Android, tiene emulador y herramientas Firebase Cloud integradas. Se ha utilizado para pasar los bocetos a mano a layouts

3.3. Lenguaje de programación – Kotlin-XML

El lenguaje de programación seleccionado para la lógica en este proyecto es Kotlin, un lenguaje de programación de código abierto desarrollado por JetBrains⁷, ya que está totalmente integrado con el IDE Android Studio, y posee una serie de ventajas respecto a JAVA. Por otro lado, Android Studio usa XML para las interfaces de usuario, posee un editor visual de interfaces de usuario bastante intuitivo.

⁶ Son gráficos digitales que no se componen por mapas bits, si no por líneas, círculos y curvas.

⁷ Empresa checa de desarrollo de herramientas para desarrolladores de software. Ha desarrollado Android Studio basándose en su IDE IntelliJ Idea.

3.4. Herramientas Firebase para desarrolladores

Se ha escogido usar las herramientas de Google Firebase ya que ofrece una serie de herramientas para desarrolladores. Se han usado las siguientes herramientas de Firebase:

Firestore Auth: Es un sistema de autenticación de usuarios para el backend de software. (Google).

Firestore Cloud: Base de datos NoSql, escalable y flexible, con soporte sin conexión. (Google)

4. Estimación de recursos y planificación

4.1. Diagrama de Gantt esperado

En este apartado vamos a plasmar una estimación del tiempo previsto y el real que se ha empleado en realizar el proyecto en un **diagrama de Gantt**,

Suele estar compuesto de una lista de tareas a la izquierda y un cronograma de barras a la derecha, como vemos en nuestro cronograma simplificado de la [ilustración 3](#). El cronograma ampliado se puede ver en el [Anexo II](#)

Hemos usado una plantilla de Excel sacada de internet (Vertex42).

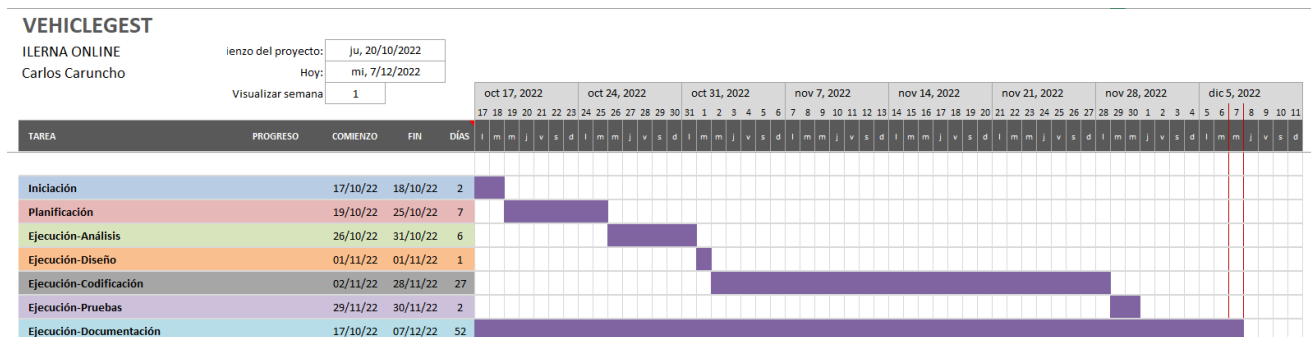


Ilustración 2: Diagrama de Gantt simplificado

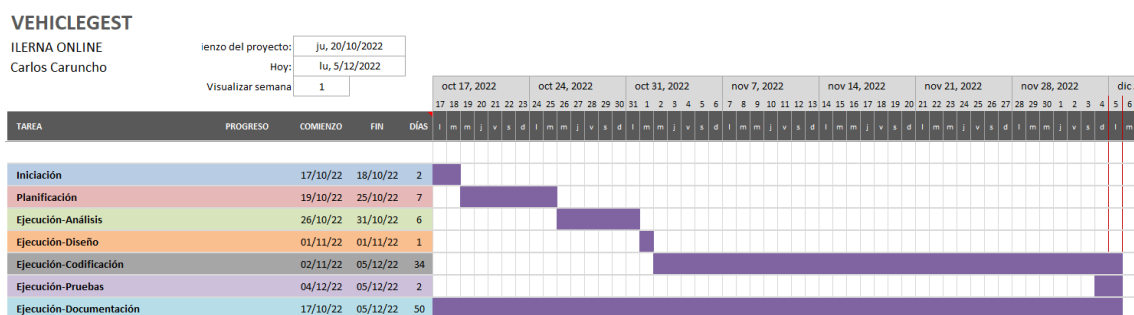
Se ha estimado que la iniciación del proyecto serán pocos días, ya que tiene tareas cortas. La planificación durará unos días más ya que tenemos que elaborar este diagrama, aparte de otras tareas.

La fase de ejecución del proyecto está subdividida en las fases en cascada, donde la que más tiempo nos va a llevar va a ser la de codificar. Para las pruebas hemos estimado que se harían en dos días, ya que no se van a realizar una gran cantidad de las mismas.

Como podemos observar la documentación nos llevará todo el tiempo del proyecto, porque se va a ir realizando a la vez que se hace el proyecto.

4.2. Diagrama de Gantt – Real

Una vez finalizado el proyecto se ha plasmado el diagrama de Gantt real. No se parece en nada al diagrama previsto. El tiempo de codificación ha costado más tiempo del previsto, debido a las dificultades técnicas y aprendizaje sobre la marcha de como implementar los casos de uso. Se ha observado que la cantidad que se estimo de casos de uso fue demasiado grande para el tiempo y ser una sola persona para programarlos todos.



5. Análisis del proyecto

5.1. Funcionamiento de la aplicación

En esta sección se describe la descripción del funcionamiento de la aplicación.

Al iniciar el usuario la aplicación se encuentra una pantalla con un logo, y un formulario para iniciar sesión en la aplicación. No es posible registrarse ya que los usuarios los da de alta un administrador de la empresa.

Al iniciar sesión nos aparecerá la pantalla inicial de la aplicación. En esta ventana la que deben aparecer listados de alertas, e información general de numerales de vehículos, servicios, ITVs y personal activo. Debe tener un menú de navegación inferior para desplazarse por las distintas actividades principales de la aplicación. En la parte superior derecha debe haber un icono de una campana para navegar hasta la actividad de alertas de la aplicación. Al presionar una de estas alertas debe abrirse una ventana emergente con el

detalle de esta alerta, que debe tener un botón con icono de basura para borrarla en la parte superior y una cruceta para cerrarla.

El menú debe constar de 5 iconos, una casa para volver a la pantalla de inicio, un coche para navegar al listado de vehículos, una llave inglesa con un martillo para las ITVs, una hoja con un lápiz para gestión de los servicios, y un muñeco para gestión de personal.

En la parte superior de cada actividad de listado habrá un menú para abrir una pantalla con un formulario de creación de nuevo registro. Presionando un icono de lupa en la parte superior derecha se podrán buscar registros discriminando por datos. Al presionar cada registro se nos abrirá una pantalla con la ficha más detallada con su foto, si debe tenerla, con un menú superior derecha para editar los datos o eliminar el registro.

5.2. Requisitos funcionales y no funcionales

En esta fase vamos a analizar y definir los **requisitos funcionales** y los **no funcionales** a partir de los objetivos propuestos, es decir las características operativas del software, cual es la interfaz que desarrollamos y sus restricciones.

5.2.1. Requisitos funcionales

Un **requisito funcional** es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Se pueden considerar como características que el usuario detecta. (Vlsure Solutions, s.f.)

ID	DESCRIPCIÓN
RF001	Iniciar sesión individual
RF002	Ver listado de alertas
RF003	Ver detalle de alerta
RF004	Eliminar alerta
RF005	Listado de vehículos
RF006	Buscar vehículo
RF007	Ver detalle de vehículo
RF008	Añadir vehículo
RF009	Editar vehículo
RF0010	Eliminar vehículo
RF0011	Ver listado de inventario de cada vehículo

RF0012	Ver listado de servicios de cada vehículo
RF0013	Ver listado de deficiencias concreto
RF0014	Ver listado de ITVs de cada vehículo
RF0015	Ver listado de servicios
RF0016	Buscar servicio
RF0017	Ver detalle de servicio
RF0018	Añadir servicio
RF0019	Editar servicio
RF0020	Eliminar servicio
RF0021	Ver listado de inventario general
RF0022	Buscar item inventario
RF0023	Ver detalle de item de inventario
RF0024	Añadir item de inventario
RF0025	Editar item de inventario
RF0026	Eliminar item de inventario
RF0027	Ver listado de ITVs
RF0028	Buscar ITV
RF0029	Visualizar detalle de ITV
RF0030	Añadir ITV
RF0031	Editar ITV
RF0032	Eliminar ITV
RF0033	Ver listado de empleados
RF0034	Buscar empleado
RF0035	Añadir empleado
RF0036	Editar empleado
RF0037	Eliminar empleado
RF0038	Hacer administrador a empleado
RF0039	Ver listado de vehículos asignados a cada empleado
RF0040	Ver listado de servicios realizados por cada empleado

5.2.2. Requisitos no funcionales

Los **requisitos no funcionales** definen cómo debe funcionar internamente el sistema.

(Vlsure Solutions, s.f.)

ID	DESCRIPCIÓN
RNF001	La interfaz debe tener los controles bien distribuidos
RNF002	La interfaz debe tener una paleta de colores agradable a la vista con colores suaves
RNF003	El menú de navegación debe estar en la parte inferior con iconos distintivos
RNF004	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado
RNF005	La aplicación debe tardar menos de 2 segundos en responder
RNF006	Las alertas serán accesibles en todas las secciones de la aplicación
RNF007	Los accesos a base de datos no bloquearan el hilo principal de la aplicación
RNF008	La aplicación seguirá los patrones de diseño de controles de Google M3 ⁸
RNF009	La aplicación podrá ser ejecutada en la mínima versión de Android que permita el diseño M3.
RNF0010	La aplicación debe señalar al empleado que está realizando una operación asíncrona.
RNF0011	El dispositivo debe tener conexión a internet para acceder a la base de datos Firebase.

5.3. Diagrama de entidad-relación.

Se ha realizado el diagrama con todas las entidades relacionadas con unas normas llamadas formas normales, con esto nos aseguramos que nuestra base de datos es escalable y bien estructurada. Se puede ver el diagrama en el [anexo III](#).

- **Reglas Primera forma normal (1FN)**

- Todos los atributos son atómicos, es decir no tienen múltiples valores.
- Cada entidad tiene una clave primaria única.

- **Reglas Segunda Forma Normal (2FN)**

- Cumple la 1FN
- Los atributos deben depender completamente de la clave primaria de la entidad. Suele pasar que no dependen completamente cuando hay clave primaria compuesta.

- **Reglas Tercera Forma Normal (3FN)**

- Cumple la 2FN

⁸ Google Material Design 3 es un sistema de diseño de código abierto para aplicaciones Android.

- No hay relaciones transitivas, es decir ningún atributo depende de otro elemento que no sea la clave primaria.

(Ortega)

Al realizar el diagrama entidad-relación obtenemos las siguientes entidades y atributos, y las usamos para pasar al diagrama relacional.

- **Empleado** (dni, nombre, apellidos, direccion, telefono, email, fechaNacimiento, usuario, password, esAdmin)
- **Vehiculo** (matricula, idMarca, idModelo, idTipo, idColor, caducidadITV, distanciaTotal, ITVPasada)
- **Servicio** (id, matricula, fecha, observaciones, Cliente)
- **Vehículo_ITV** (matricula, idITV, resultadovehiculo)
- **Deficiencia** (id, solucion, solucionado, fechaSolucion, gravedad, descripcion, idVehiculoITV)
- **ITV** (id, fecha)
- **Alerta** (id, fecha, descripcion resuelta, matricula)
- **Item** (id, tipo fabricante, matricula)
- **Marca** (id, nombre)
- **Modelo** (id, nombre)
- **Color** (id, nombre)

5.4. Diagrama relacional.

Se puede ver el diagrama completo en el [anexo IV](#).

5.5. Diagrama de clases

Se puede ver el diagrama completo en el [anexo V](#).

5.6. Diagrama de casos de uso

Se puede ver el diagrama de clases en el [anexo VI](#)

5.7. Especificaciones de casos de uso

A continuación, explicamos los casos de uso más representativos de la aplicación. El resto de casos de uso están en el anexo VII.

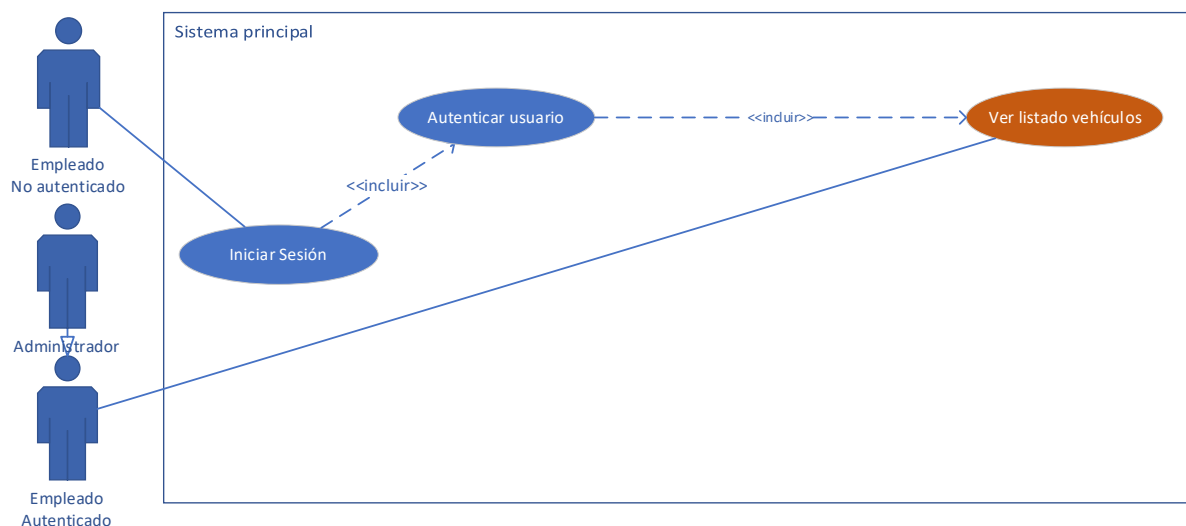


Ilustración 3 : Diagrama casos de uso de sistema principal

Identificador	CU_01	
Nombre	Iniciar sesión	
Descripción	Habrá dos campos, usuario y contraseña al <i>empleado</i> o al <i>empleado administrador</i> , y un botón para aceptar.	
Actores	Empleado, administrador, sistema	
Precondición	Al ejecutar por primera vez solo existe el usuario “admin@admin.com” con contraseña admin123” en la base de datos.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> rellena los campos usuario y contraseña.
	2	El <i>sistema</i> comprueba si las credenciales existen en la base de datos.
Postcondición	El empleado queda autenticado y tiene acceso a toda la aplicación.	
Excepciones	Paso	Acción
	2	Si hay error de credenciales el sistema manda un mensaje de error.
Comentarios		

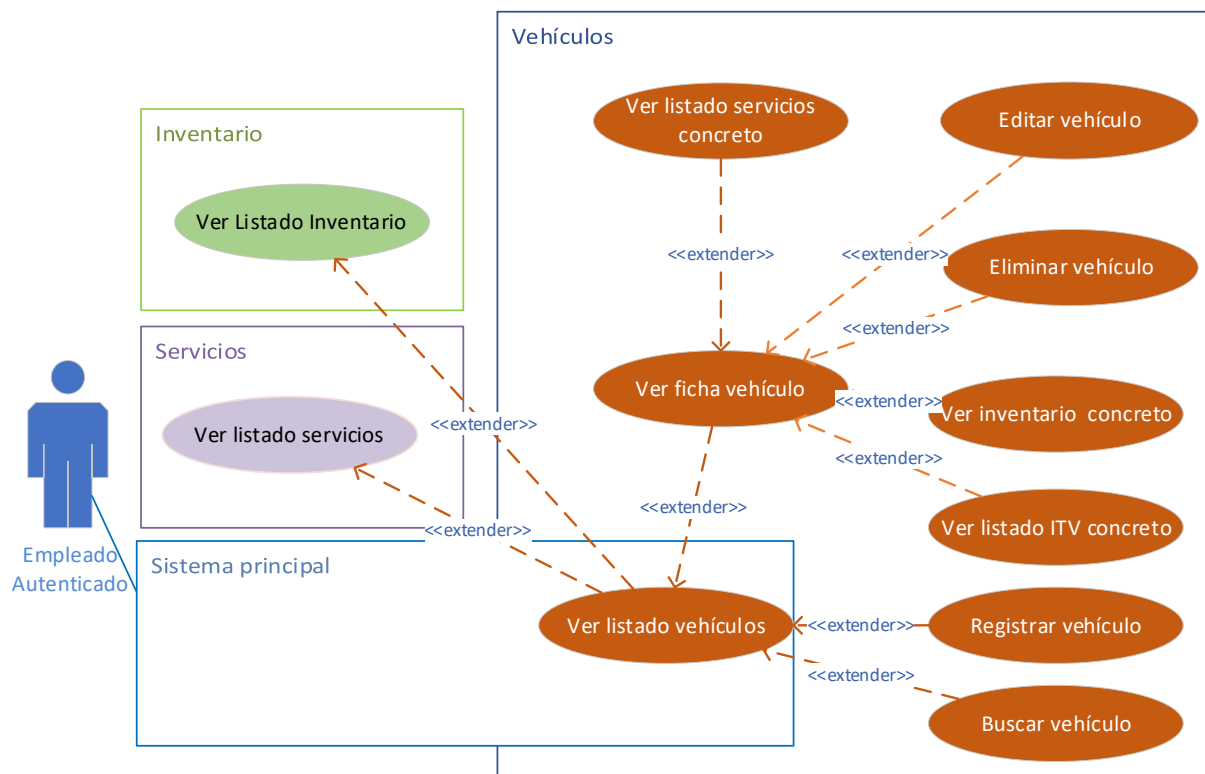


Ilustración 4 - Diagrama de casos de uso de vehículos

Identificador	CU_02	
Nombre	Ver listado de vehículos	
Descripción	<p>El empleado ve un listado de los vehículos datos de alta en la base de datos. El listado es de fichas de los vehículos con una pequeña foto del vehículo y los datos más importantes: matrícula, marca, modelo y color.</p> <p>Se puede presionar un icono de lupa para buscar vehículos por varios criterios.</p>	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona la ficha de un vehículo. Se ejecuta CU_08.
	2	El <i>empleado</i> presiona el icono de lupa arriba derecha. Se ejecuta CU_07
Postcondición		
Excepciones	Paso	Acción
Comentarios		

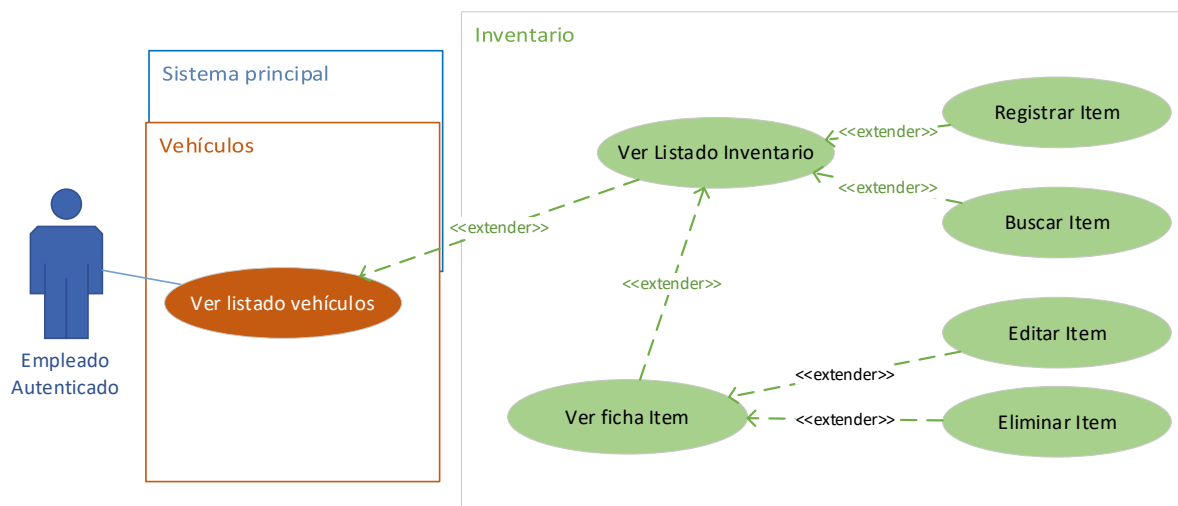


Ilustración 5: Diagrama de casos de uso de inventario

Identificador	CU_03	
Nombre	Ver listado de inventario general	
Descripción	Se ve un listado de herramientas general total que existe en la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre un registro para abrir el detalle. Ejecuta CU_016.
	2	
Postcondición	Solo se ve el listado de Inventario de ese vehículo en concreto	
Excepciones	Paso	Acción
Comentarios		

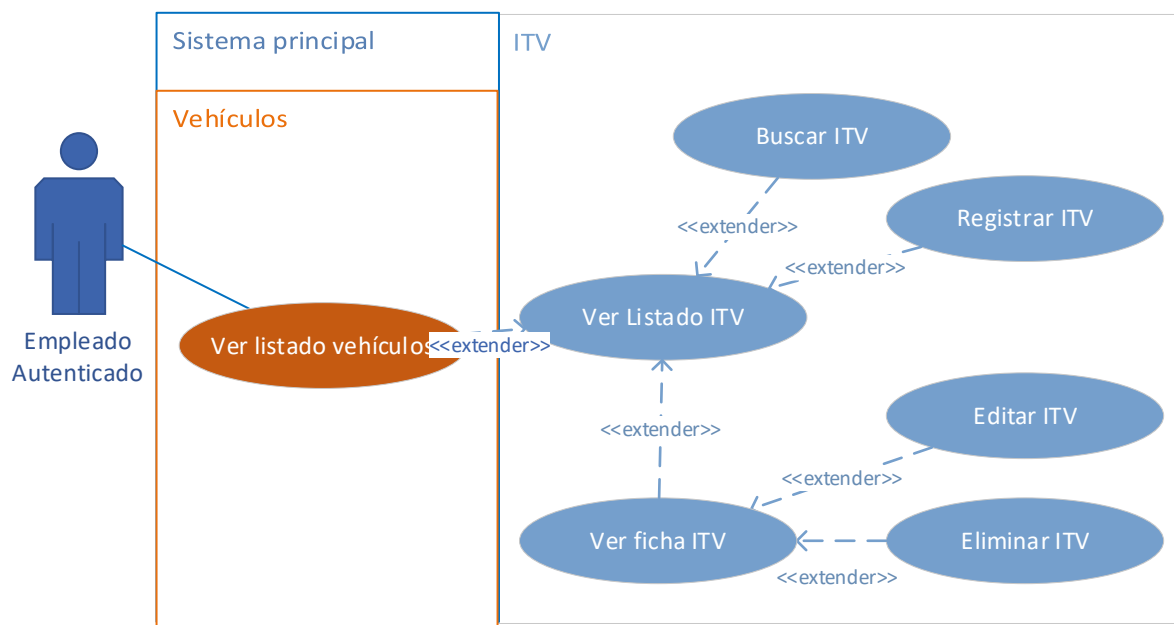


Ilustración 6: Diagrama de casos de uso de ITV

Identificador	CU_04	
Nombre	Ver listado de ITVs	
Descripción	Se ve un listado de ITVs general.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado.	
Secuencia normal	Paso	Secuencia normal
	1	El <i>empleado</i> presiona sobre cada registro para abrir el detalle. Ejecuta CU_22.
	2	
Postcondición		
Excepciones	Paso	Excepciones
Comentarios		

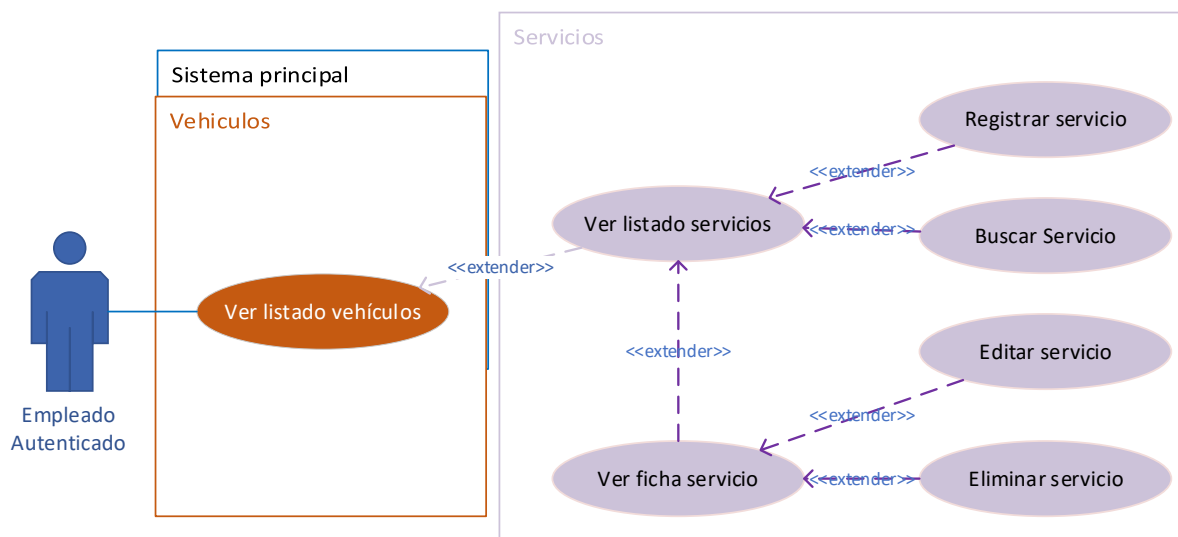


Ilustración 7: Diagrama de casos de uso de servicios

Identificador	CU_05	
Nombre	Ver listado de servicios	
Descripción	Se ve un listado de servicios general.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado.	
Secuencia normal	Paso	Secuencia normal
	1	El <i>empleado</i> presiona sobre un registro para abrir la ficha de detalle. Se ejecuta CU_027.
	2	
Postcondición		
Excepciones	Paso	Excepciones
Comentarios		

6. Diseño del proyecto

6.1. Bocetos de la aplicación

6.1.1. Pantalla de autenticación de usuario

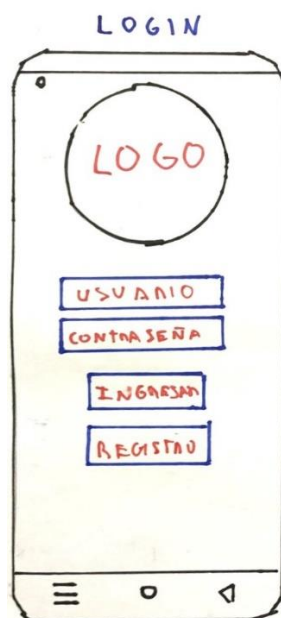


Ilustración 8 - Boceto de pantalla de autenticación

En la ilustración 8 podemos “**activity**”⁹ o **actividad** que ve el usuario al entrar a la aplicación después de ejecutarla, en ella aparece el logotipo de la aplicación en grande, y un formulario con los campos de email y contraseña, con un botón para aceptarlo. Esta vista envía los datos introducidos en el formulario a la base de datos de Firestore, si las credenciales son correctas navegará al listado de vehículos. Los usuarios volverán a esta vista cuando se deslogueen desde cualquier vista. En el caso de que no tengan usuario registrado podrán crear uno presionando el botón “Registro”, lo que les lleva a la siguiente actividad.

6.1.2. Pantalla de registro de usuario

⁹ Una activity o actividad es un punto de entrada a la aplicación, se puede decir que es una pantalla y se puede cargar en cualquier momento.

En la ilustración 9 vemos la pantalla de registro, en esta pantalla el usuario deberá ingresar correo y la contraseña dos veces, para asegurarse que la escribe bien, se realiza una comparativa entre los dos campos, si esta es correcta, el sistema le creará un usuario y navegará hacia la pantalla de **actividad principal**.

6.1.3. Pantalla de listado de vehículos

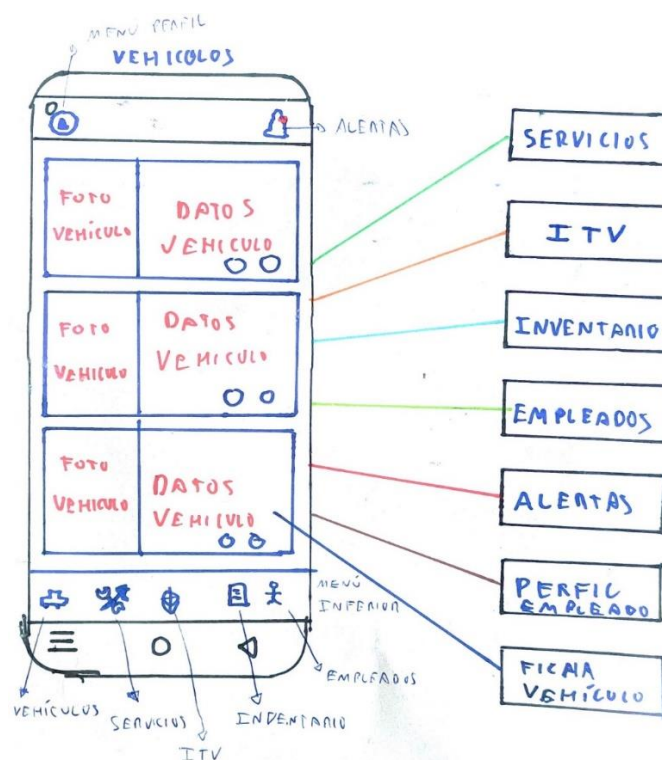


Ilustración 9 - Boceto de pantalla principal

El siguiente boceto es de la actividad principal con el fragmento central cargado de **listado de vehículos**, empezando por la parte de arriba tendremos una barra de herramientas. En la parte derecha de la barra un icono con una campana de notificaciones, podremos navegar al listado de notificaciones y avisos, y otro icono de “deslogueo”, que nos llevará de nuevo a la actividad de login. En la parte central hay un “**recyclerview**¹⁰” que muestra el listado de items, cada ítem tendrá la imagen del vehículo. Esta parte se puede desplazar hacia arriba y hacia abajo con un “**scrollview**¹¹” vertical. Esta parte central será también dinámica para la navegación y cambiará de “**fragment**¹²” o **fragmento** cuando presionemos sobre cada uno

¹⁰ Es una lista dinámica de ítems almacenados en la base de datos con imágenes y posibilidad de presionarlas y seleccionadas. Mejora el rendimiento al reciclar las vistas de los ítems.

¹¹ Un scrollview es un control que permite mostrar una vista más amplia de lo que ocupa la pantalla desplazando hacia arriba o abajo o a los lados.

¹² Un fragmento es una parte reutilizable de la vista, no puede existir por si sola, esta alojada en otra actividad principal.

de los iconos de avisos, o del menú inferior de navegación. El menú inferior de navegación está compuesto de 5 iconos. Para navegar hacia los apartados de vehículos, servicios, ITV, inventario y personal. También habrá icono “+” flotante para agregar registros en cada apartado.

Presionando la ficha de vehículos se abrirá una pantalla emergente donde se ampliará la foto del vehículo, y se podrán consultar en detalle la información del vehículo, borrarlo y editarlo.

6.1.4. Dialogo de vista de detalle de vehículo

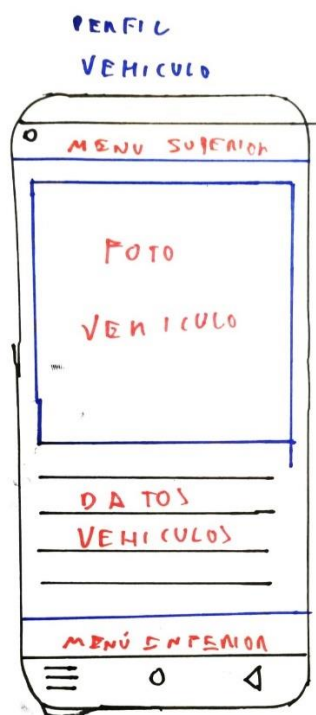


Ilustración 10 - Boceto de detalle de vehículo

En este dialogo podemos ver la foto del vehículo que ocupa media pantalla, debajo de la foto estarán todos los datos del vehículo pudiendo bajar también con un “scrollview” para ver más datos. Habrá tres botones superiores donde se puede volver atrás, borrar el registro y otro para editar el vehículo

6.1.5. Vista de listado de alertas

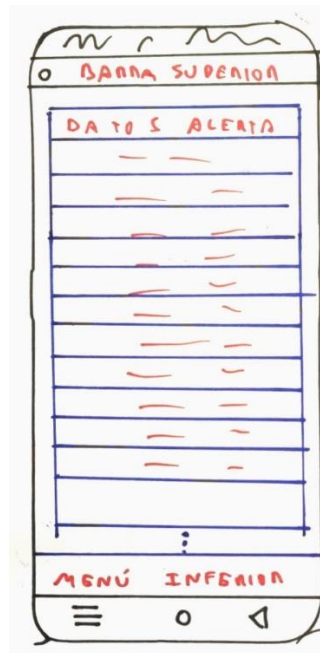


Ilustración 11 - Boceto de listado de alarmas

En este fragmento podremos ver una lista de todas las alertas generadas en una lista, esta lista será una “recyclerview” simple, con la fecha las alertas, el motivo y la fecha de su generación.

Se podrán borrar desde un botón superior y agregar más desde un botón de + inferior.

6.1.6. Vista de listado de servicios

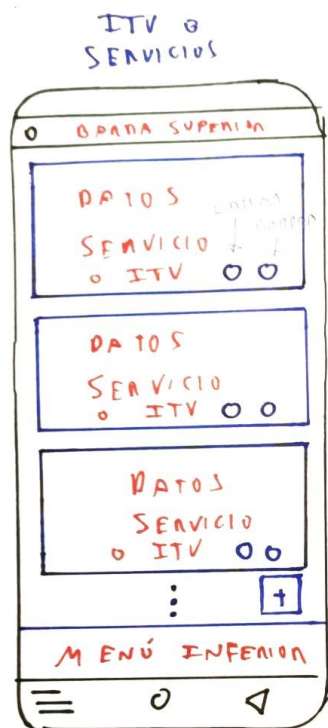


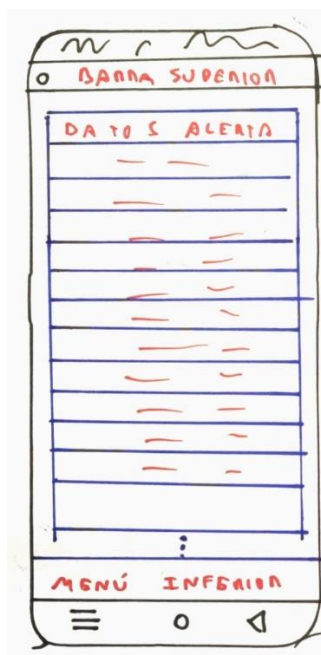
Ilustración 12 - Boceto de listado de servicios

En la ilustración 12 podemos ver el fragmento de listado de servicios, es similar al de ITV, pero variando los datos. Se podrán agregar y abrir en detalle cada uno de los servicios.

6.1.7. Vista de detalle de servicios

En esta vista se podrá ver un fragmento con el detalle del servicio que hemos presionado. Se podrán borrar o editar los datos. Será como la de vehículo pero sin la fotografía.

6.1.8. Vista de listado de inventario



En la ilustración 12 podemos ver el fragmento de listado de items, es similar a las alertas, pero variando los datos. Se podrán agregar y abrir en detalle cada uno de los

Se podrán borrar desde un botón superior y agregar más desde un botón de + inferior.

6.1.9. Vista de detalle de ítem

En esta vista se podrá ver un fragmento con el detalle del ítem que hemos presionado. Se podrán borrar o editar los datos. Es igual que la de vehículo.

6.1.10. Vista de listado de ITV

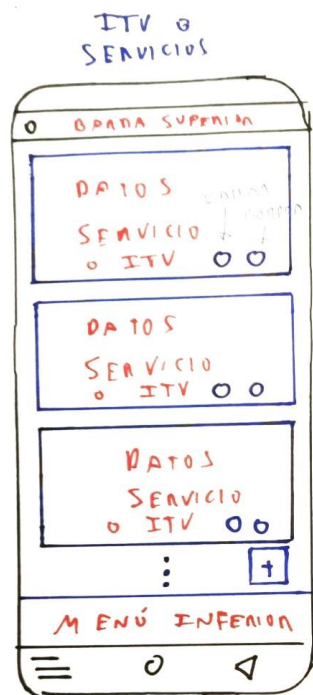


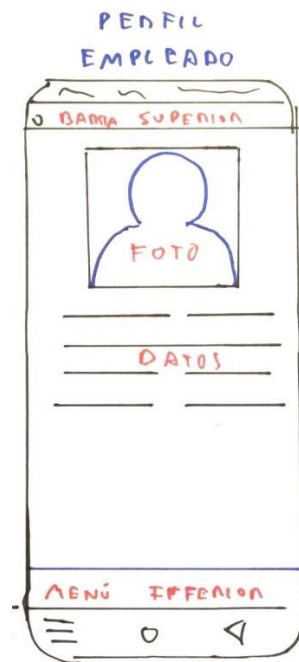
Ilustración 13 - Boceto de listado de ITV

En la ilustración 12 podemos ver el fragmento de listado de servicios, es similar al de servicios, pero variando los datos. Se podrán agregar y abrir en detalle cada uno de las ITV.

6.1.11. Vista de listado de personal

Esta vista es similar a la de listado de vehículos, con los datos esenciales en cada registro y una foto en minatura del empleado, se podrá presionar sobre cada empleado para ver los detalles.

6.1.12. Vista de detalle de empleado



6.2. Implementación en detalle

6.2.1. Configuración de Firestore Database

Nos registramos en Firebase y creamos la base de datos Firestore database

Modelamos los datos pasando del modelo relacional al NoSQL de Firestore. Lo traducimos al inglés como lengua global en la programación y tenerlo todo en ese lenguaje programado.

<https://www.youtube.com/watch?v=UHIhOKctxD8>

- **Colección:** Vehicle
 - **Documento:** 1234ABC (matrícula)
 - **Campos**
 - brand:string
 - color:string
 - expiryDateITV:timestamp
 - model:string
 - type:string
 - totalDistance:number
 - **Subcolecciones**
 - **Employee:**

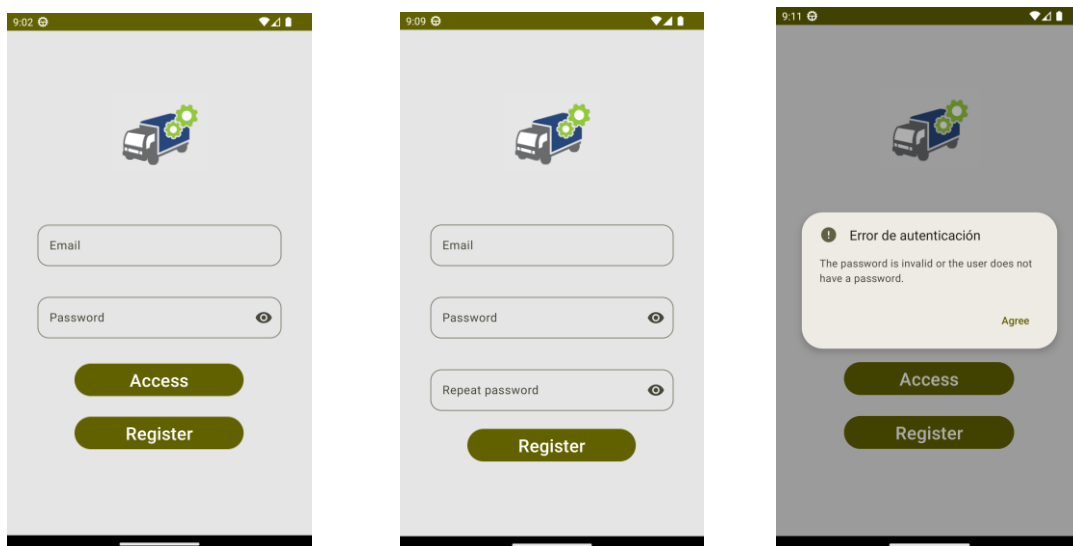
- **Documento:** 12345678A (Employee DNI)
 - **Campos**
 - name:string
 - surname:string
 - **Deficiency:**
 - **Documento:** Id deficiency
 - **Campos**
 - description:string
 - severity:string
 - solution:string
 - fixed:boolean
 - fixedDate:timestamp
 - **ITV:**
 - **Documento:** Id ITV
 - **Campos**
 - date:timestamp
 - **inventory:**
 - **Documento:** Id item
 - **Campos**
 - name:string
- **Colección:** service
 - **Documento:** Id aleatoria
 - **Campos**
 - plateNumber:string (Clave foránea)
 - date:timestamp
 - costumer:string
 - remarks:string
- **Colección:** ITV
 - **Documento:** Id aleatoria
 - **Campos**
 - date:timestamp

- **Subcolecciones**
 - **Vehicle:**
 - **Documento:** 1234ABC (Vehicle plateNumber)
 - **Campos**
 - brand:string
 - model:string
 - color:string
 - type:string
 - **Deficiency:**
 - **Documento:** Id deficiency
 - **Campos**
 - description:string
 - fixed:boolean
 - severity:string
- **Colección:** Vehicle_ITV
 - **Documento:** Id aleatoria
 - **Campos**
 - plateNumber:String(Clave foránea)
 - id_ITV:String(Clave foránea)
 - result_vehicle:Boolean
- **Colección:** Deficiency
 - **Documento:** Id aleatoria
 - **Campos**
 - plateNumber:string(Clave foránea)
 - description:string
 - severity:string
 - solution:string
 - fixed:boolean
 - fixedDate:timestamp
 - date:string

- **Colección:** Alert
 - **Documento:** Id aleatoria
 - **Campos**
 - plateNumber:string(Clave foránea)
 - description:string
 - severity:string
 - solution:string
 - fixed:boolean
 - fixedDate:timestamp
 - date:string

- **Colección:** inventory
 - **Documento:** Id aleatoria
 - **Campos**
 - plateNumber:string(Clave foránea)
 - name:string
 - description:string

6.2.2. Implementación de la autenticación y registro

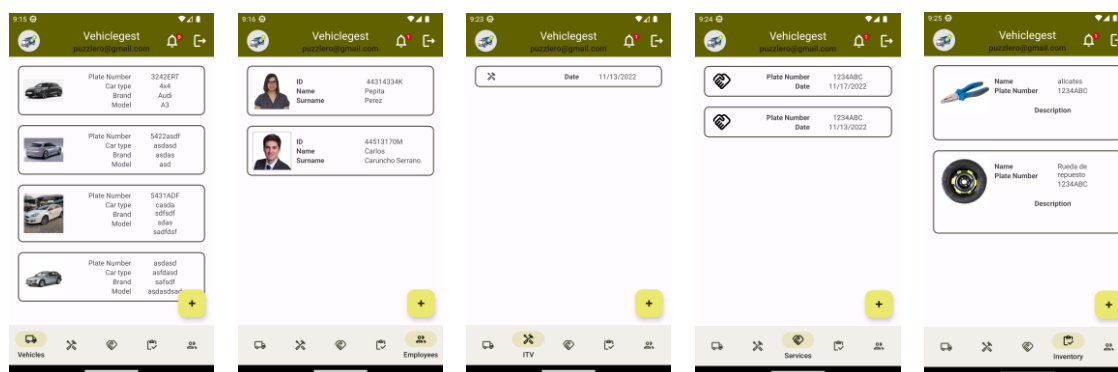


Se implementa la conexión a la base de datos de Firestore auth. Se ha utilizado la documentación de Firebase para implementarlo. <https://firebase.google.com/docs/auth>.

La aplicación puede estar en castellano e inglés, ya que se han traducido cada una de las cadenas de texto con el sistema de android studio, almacenadas en los archivos strings.xml

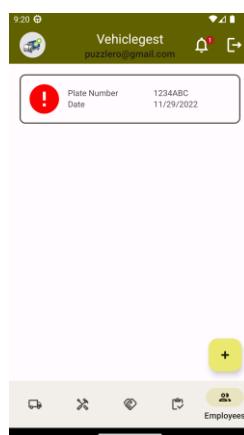
Se ha implementado dialogo de información de errores de usuario y contraseña.

6.2.3. Menú de navegación de fragmentos



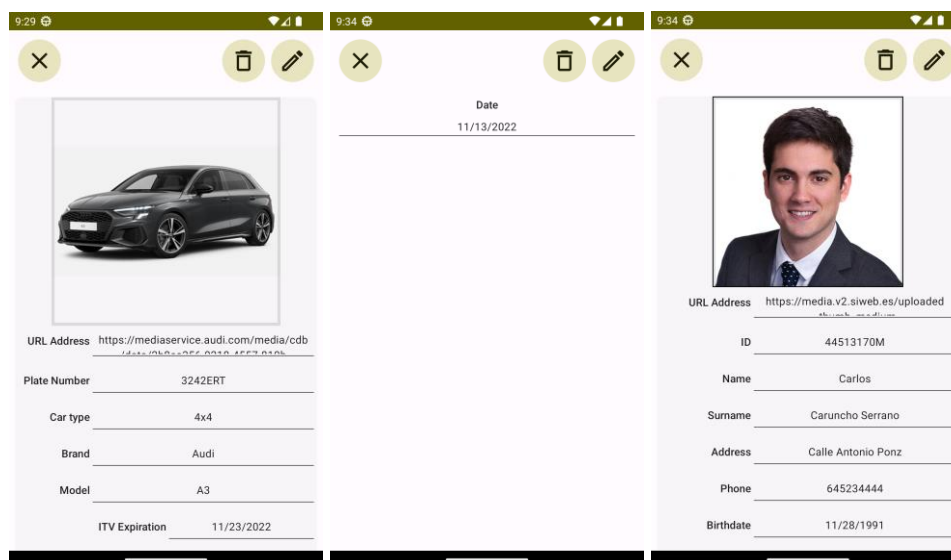
Mediante la implementación del menú inferior podemos navegar entre los fragmentos, que aparecen en un frame central manteniendo los menús superior e inferior, y el botón flotante para agregar registros.

6.2.4. Barra superior de información y navegación



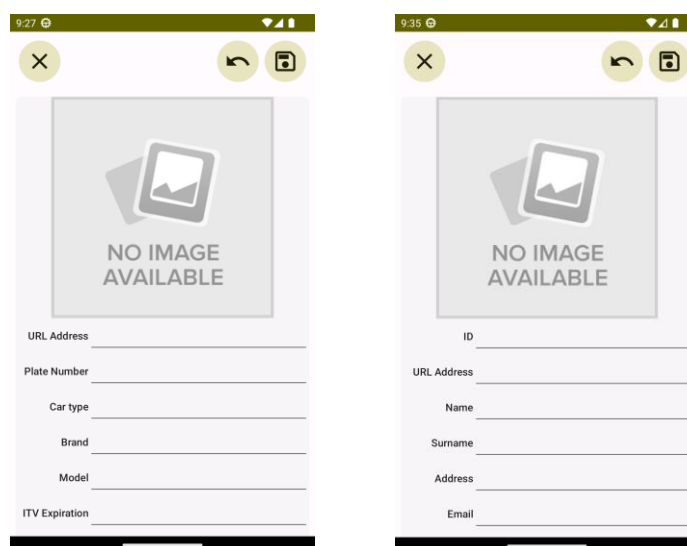
En la barra superior podemos encontrarnos un botón para desloguear y volver a la actividad de autenticación. El icono de avisos con un contador de avisos implementado. También aparece el título de la aplicación y la dirección de correo electrónico del usuario que está logueado.

6.2.5. Pantallas de detalle de registros



En estas pantallas nos encontramos un icono para cerrar la ventana, uno para borrar el registro y otro para editar la información del vehículo. La fotografía se carga mediante la URL almacenada en la base de datos. Los menús se esconden al entrar a los fragmentos.

6.2.6. Pantallas de formularios de registro



En los formularios de registro nos encontramos un botón de cerrar el fragmento, otro de borrar los datos ingresados en el formulario y otro para guardar los datos ingresados en el formulario. Los menús se esconden al entrar a los fragmentos.

7. Despliegue y pruebas

Nº	ESPECIFICACIÓN DE PRUEBAS
1	<p><u>Objetivo probado:</u> Apertura de la aplicación</p> <p><u>Requisitos probados:</u> Debe aparecer la actividad de login para solicitar los datos de ingreso</p> <p><u>Pruebas que realizar:</u> Se abre la aplicación y se ingresan los datos de usuario.</p>
2	<p><u>Objetivo probado:</u> Registro de usuario</p> <p><u>Requisitos probados:</u> Debe aparecer la actividad de login para solicitar los datos de ingreso</p> <p><u>Pruebas que realizar:</u> Se abre la aplicación y se ingresan los datos de usuario.</p>
3	<p><u>Objetivo probado:</u> Ver actividad principal</p> <p><u>Requisitos probados:</u> Debe aparecer la actividad principal con el fragmento central de vehículos</p> <p><u>Pruebas que realizar:</u> Se abre la actividad y se puede presionar sobre los iconos de los menús.</p>
4	<p><u>Objetivo probado:</u> Ver listados de registros</p> <p><u>Requisitos probados:</u> Deben aparecer los registros de vehiculos, servicios, personal, inventario, ITV y alertas.</p> <p><u>Pruebas que realizar:</u> Se conecta a la base de datos y muestra los registros de cada una de las listas,</p>
5	<p><u>Objetivo probado:</u> Crear registros</p> <p><u>Requisitos probados:</u> Crear registros en la base de datos.</p> <p><u>Pruebas que realizar:</u> Se conecta a la base de datos y registra los datos dados ingresados en los formularios de datos de cada una de las secciones</p>
5	<p><u>Objetivo probado:</u> Borrar registros</p> <p><u>Requisitos probados:</u> Borrar registros en la base de datos.</p> <p><u>Pruebas que realizar:</u> Se conecta a la base de datos y borra los datos dados ingresados en los formularios de datos de cada una de las secciones</p>

8. Conclusiones

Como conclusiones finales podemos hablar de varios puntos que se van a desarrollar a continuación.

Las **herramientas** escogidas han sido las adecuadas y no es algo difícil en este caso ya que son casi todas gratuitas, y bastante sencillas de utilizar.

Respecto a la **planificación y la estimación de recursos** es de las partes más importantes de un proyecto. Hay que planificar el tiempo que se va a tardar en hacer un proyecto planteado a partir de los objetivos y sin experiencia en esto no es algo fácil de hacer. Hay que saber cuanto tiempo va a llevar hacer cada uno de los requisitos con el equipo de desarrollo de que se dispone, ya que se dan unos plazos con los clientes. En este proyecto se ha tardado mucho más del tiempo estimado, debido a la falta de experiencia y conocimientos en el ámbito de Android.

Debido a esto último no se han podido alcanzar todos los objetivos planteados, ya que ha llevado más de la cuenta aprender y afianzar muchos de los conocimientos necesarios para llevarlos a cabo.

Después de realizar este proyecto he podido aprender como funciona la elaboración de un proyecto real, su documentación y como llevar a cabo la planificación y desarrollo del mismo.

Se ha adquirido experiencia en el sistema de datos Firestore, el cual está basado en JSON y funciona en texto plano con documentos anidados, y las relaciones y consultas son totalmente diferentes al modelo relacional.

También se ha adquirido conocimientos en el IDE Android Studio, en programación Kotlin y aprendidos muchos conceptos nuevos de programación para el sistema operativo Android.

9. Vías futuras

En las vías futuras se hará lo siguiente para mejorar la aplicación:

- Mejora de la interfaz en general, aplicando el estándar de diseño Material3 en las partes que falten
- Mejor colocación de los menús
- Mejora en la estructura de la base de datos para hacerla más eficiente y adecuada al tipo de aplicación.
- Corrección de errores detectados.
- Posibilidad de imprimir informes de deficiencias de cada vehículo.
- Implementar las funcionalidades de los casos de uso que falten, como por ejemplo buscar registros.
- Incluir en la base de datos a los clientes, para poder marcar en cada servicio a quien se le ha hecho.

- Un tablón central de avisos e información general al entrar el usuario a la aplicación.
- Implementación de sistema de roles para dar permisos según se requiera.

10. Bibliografía

- ¿QUÉ ES ITV? NORMATIVA Y TIPOS. (n.d.). From Servicios ITV:
<https://www.serviciositv.es/que-es-itv>
- APD. (2021, Junio 8). *¿En qué consiste la metodología Kanban y cómo utilizarla?* From APD:
<https://www.apd.es/metodologia-kanban/>
- Curso Kotlin. (n.d.). *Curso Kotlin*. From Capítulo 26 – DatePicker en Kotlin:
<https://cursokotlin.com/capitulo-26-datepicker-en-kotlin/>
- Fox Android. (n.d.). *Bottom Navigation Bar - Android Studio*. From Bottom Navigation Bar - Android Studio: <https://www.youtube.com/watch?v=YIIHxIAoHzU&t=119s>
- Google. (n.d.). *Firebase Auth*. From Firebase Auth: <https://firebase.google.com/docs/auth>
- Google. (n.d.). *Firestore Cloud Plataform*. From Firestore Cloud Plataform:
<https://firebase.google.com/docs/firestore>
- Ilerna Online S.L. (n.d.). Programación Multimedia y Dispositivos Móviles. In I. O. S.L., *Programación Multimedia y Dispositivos Móviles*.
- Ilerna Online SL. (2022). Metodología Kanban. In I. O. SL, *Entornos de desarrollo*. Lleida: Ilerna Online SL.
- Ilerna S.L. (2021). *Entornos de Desarrollo*. Ilerna S.L.
- Invarato, R. (n.d.). *Context de Android*. From Jarroba: <https://jarroba.com/context-de-android/>
- IONOS. (n.d.). *El modelo en cascada: desarrollo secuencial de software*. From IONOS:
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Lucidchart. (n.d.). *Que es Visio*. From <https://www.lucidchart.com>:
<https://www.lucidchart.com/pages/es/que-es-microsoft-visio>
- Meardon, E. (n.d.). *About Gantt Charts*. From Atlassian:
<https://www.atlassian.com/es/agile/project-management/gantt-chart>
- Microsoft. (n.d.). *Diagrama de Gantt de dos años*. From Office.com:
<https://templates.office.com/es-es/diagrama-de-gantt-de-dos-a%C3%B1os-tm56599548>
- Moure, B. (n.d.). *FIREBASE Authentication Android*. From FIREBASE Authentication Android : <https://www.youtube.com/watch?v=dpURgJ4HkMk>

NextU. (n.d.). *¿Qué es Json? ¿Por qué es importante conocerlo?* From NextU: <https://www.nextu.com/blog/que-es-json-por-que-es-importante-conocerlo-rc22/#:~:text=En%20resumen%2C%20JSON%20no%20es,para%20transferir%20informaci%C3%B3n%20entre%20sistemas.>

Ortega, Á. (n.d.). *Cursos de programación*. From Código Online: www.codigoonline.es

Presta, M. (n.d.). *¿Qué es Firebase? Todos los secretos desbloqueados*. From back4app: <https://blog.back4app.com/es/que-es-firebase/>

Teamleader. (2021, Agosto 18). *¿Qué es y para qué sirve un diagrama de Gantt?* From Teamleader: <https://www.teamleader.es/blog/diagrama-de-gantt>

Tecnosoluciones. (n.d.). *Te damos 10 razones para usar la metodología Kanban en tu organización*. From Tecnosoluciones: <https://tecnosoluciones.com/10-razones-para-usar-la-metodologia-kanban-en-tu-organizacion/>

Trello. (n.d.). *Aprende los aspectos básicos del tablero de Trello*. From Trello: <https://trello.com/guide/>

Vertex42. (n.d.). *Simple Gantt Chart*. From Vertex42: https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html?utm_source=v42&utm_medium=file&utm_campaign=templates&utm_term=simple-gantt-chart_ms&utm_content=url

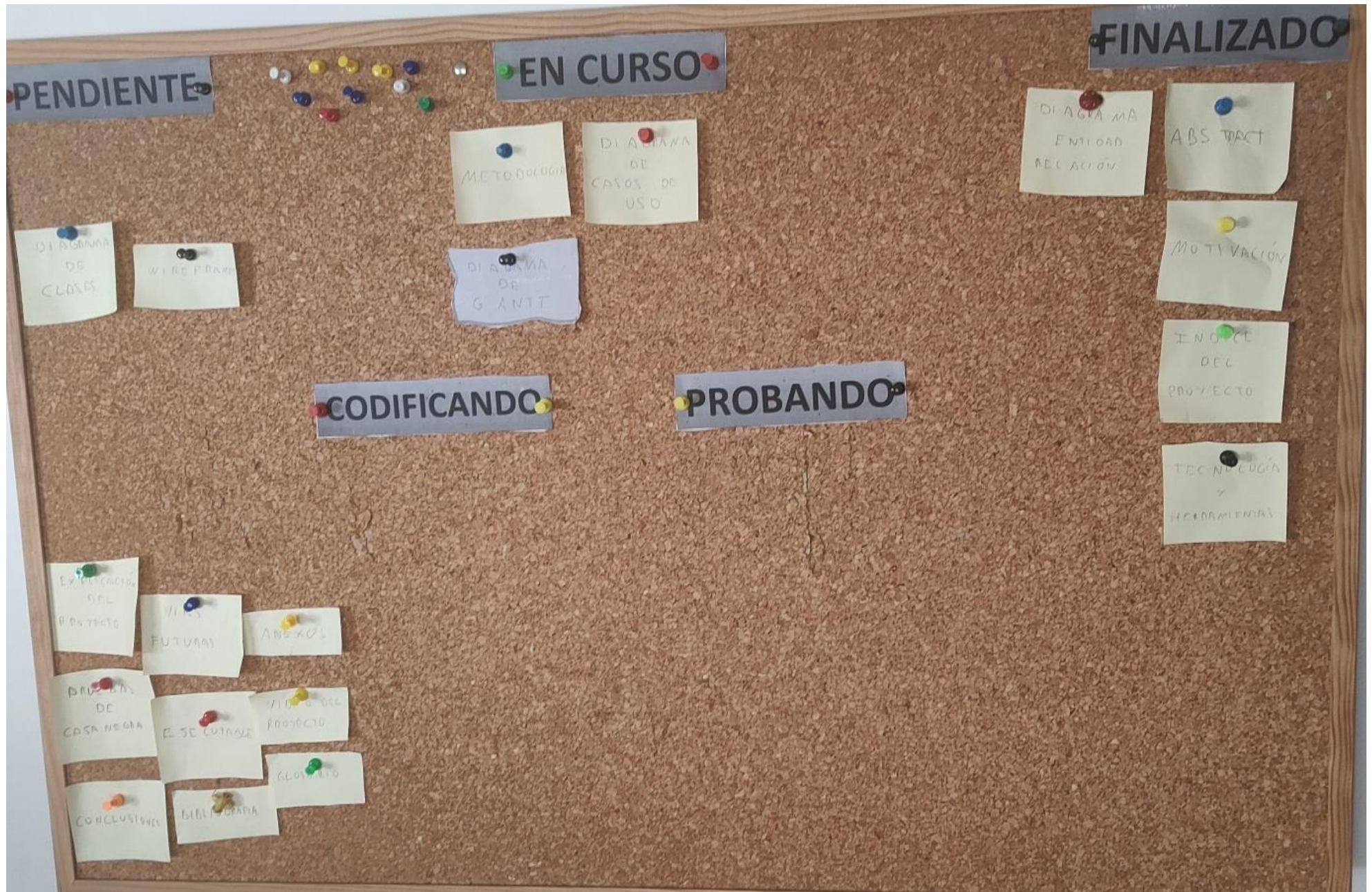
viewnext. (n.d.). *El ciclo de vida de las metodologías ágiles de desarrollo*. From viewnext: <https://www.viewnext.com/el-ciclo-de-vida-de-las-metodologias-agiles-de-desarrollo/>

Vlsure Solutions. (n.d.). *Qué son los requisitos funcionales: ejemplos, definición, guía completa*. From VlsureSolutions: <https://visuresolutions.com/es/blog/functional-requirements/>

Wikipedia. (2022, Octubre 4). *Android Studio*. From Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

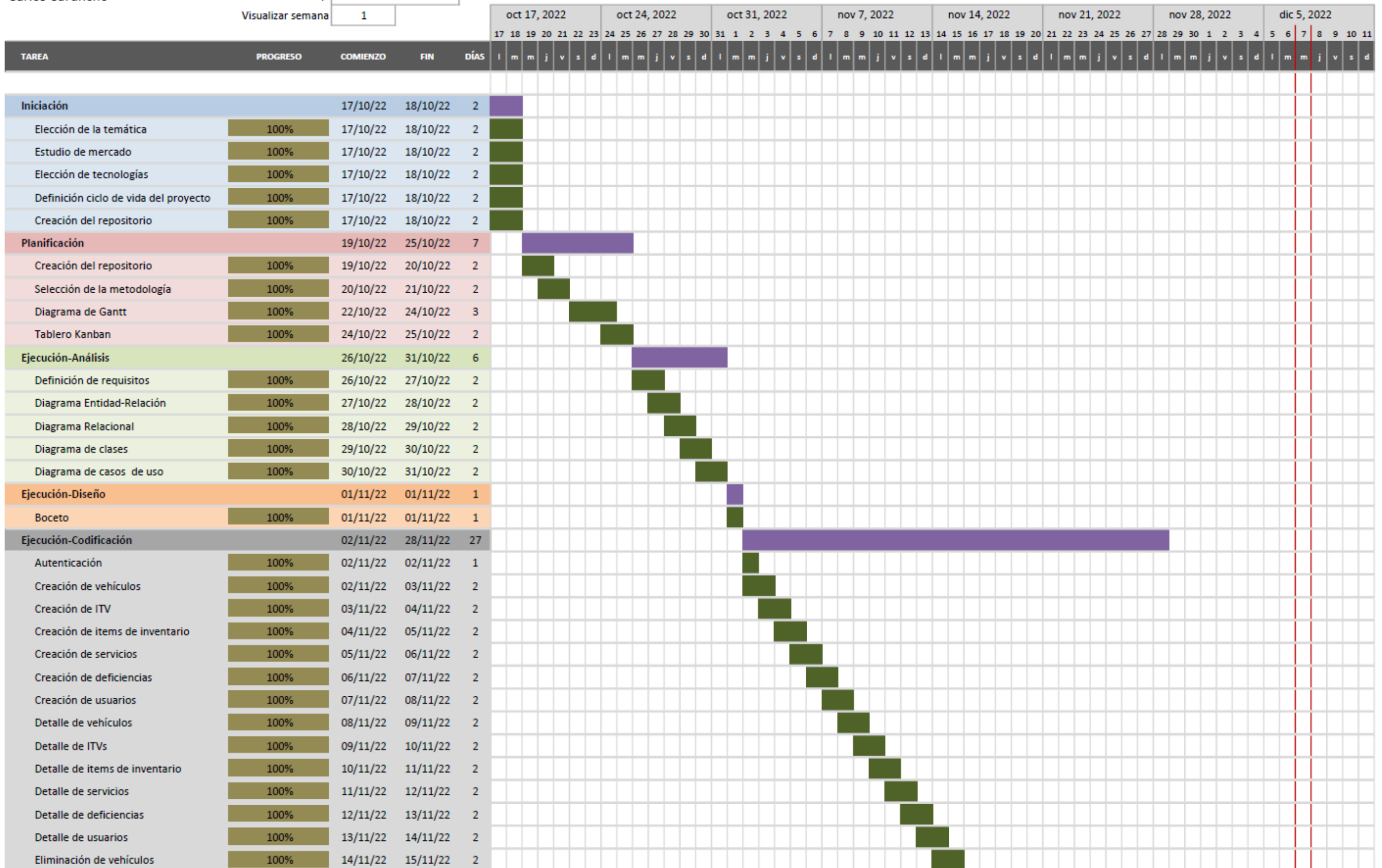
Wikipedia. (2022, Octubre 14). *Bases de datos*. From Wikipedia: https://es.wikipedia.org/wiki/Base_de_datos

ANEXO I – Tablero Kanban

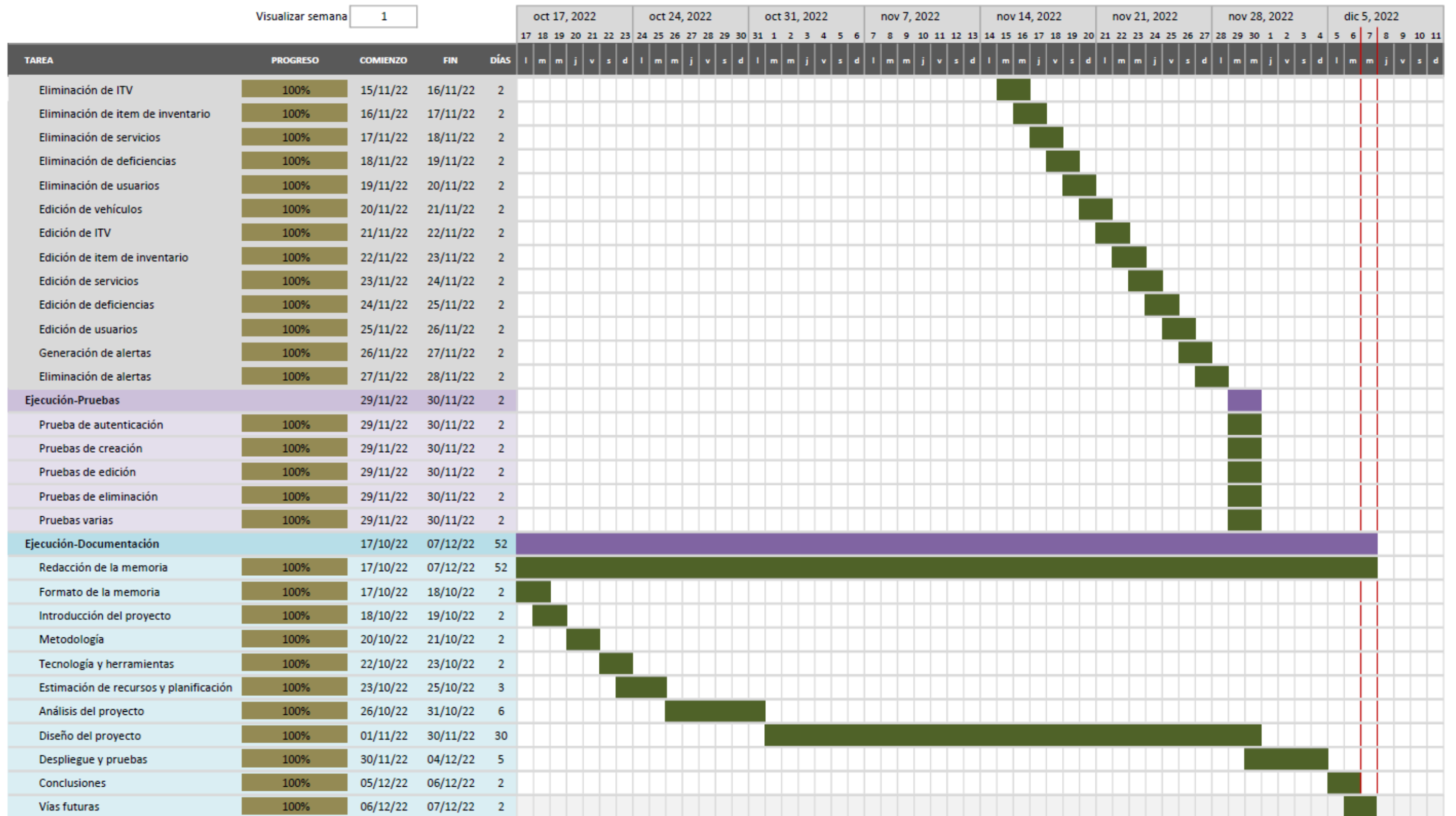


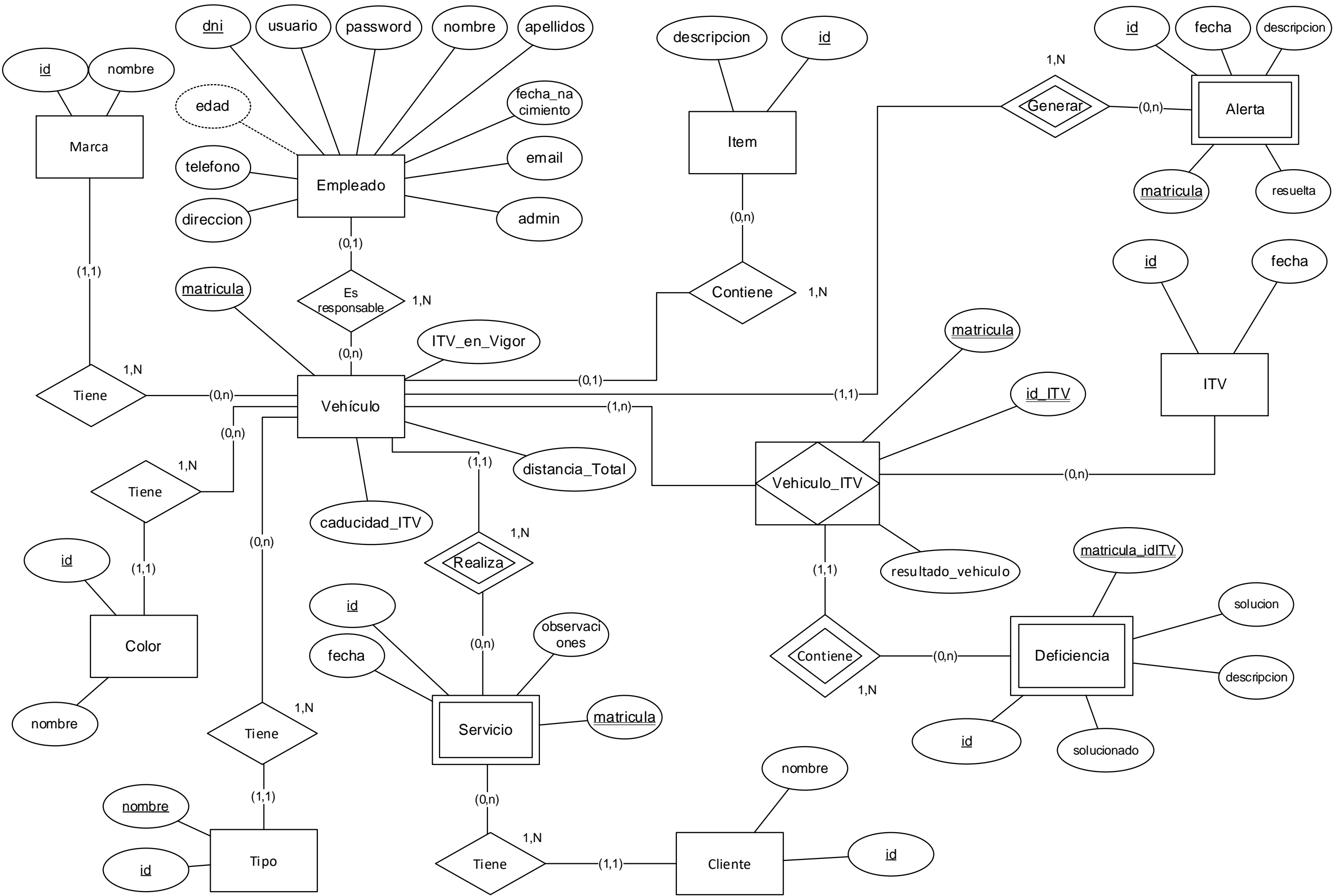
Carlos Caruncho

mienzo del proyecto:	ju, 20/10/2022	
Hoy:	mi, 7/12/2022	
Visualizar semana	1	

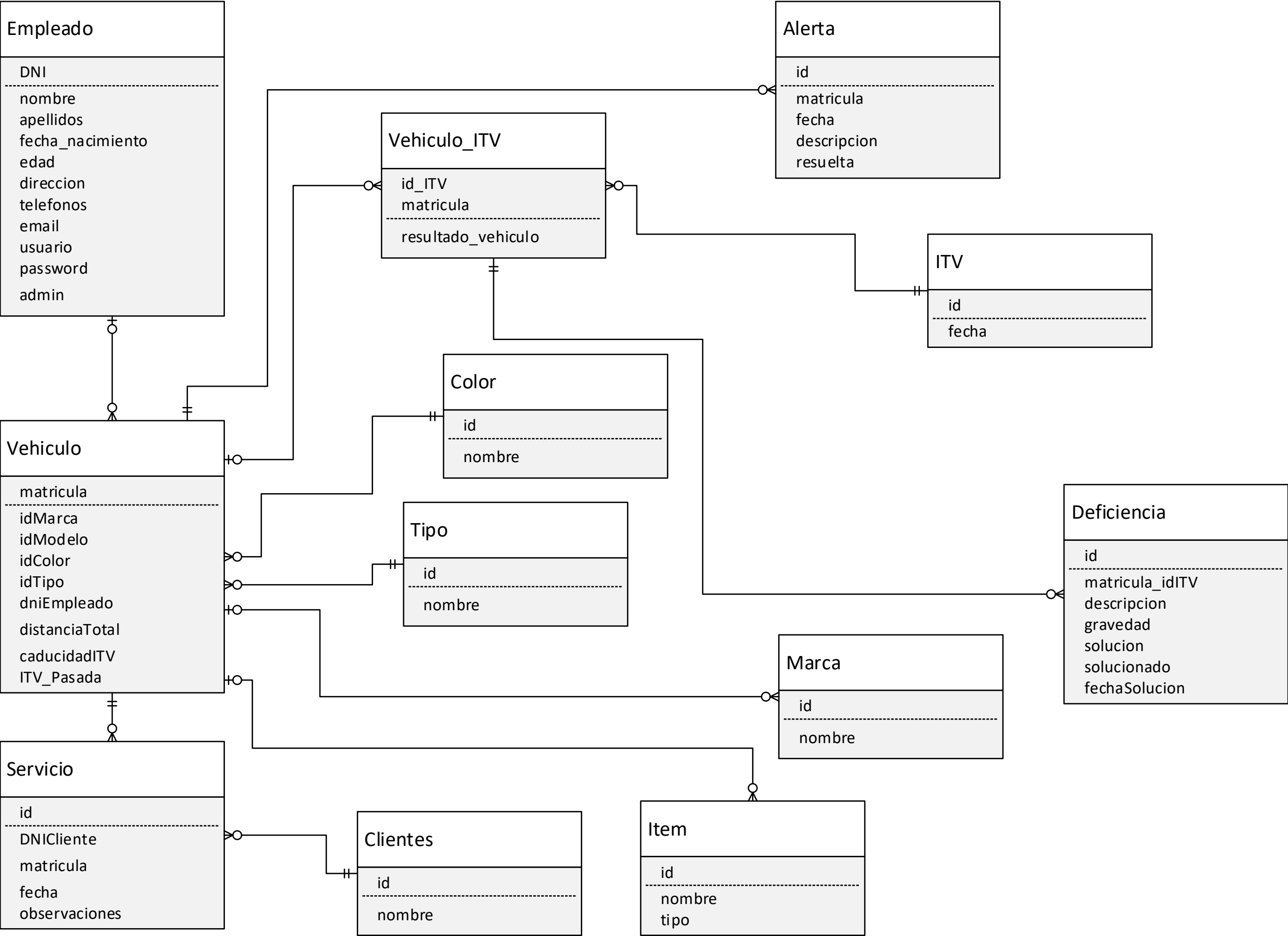


1

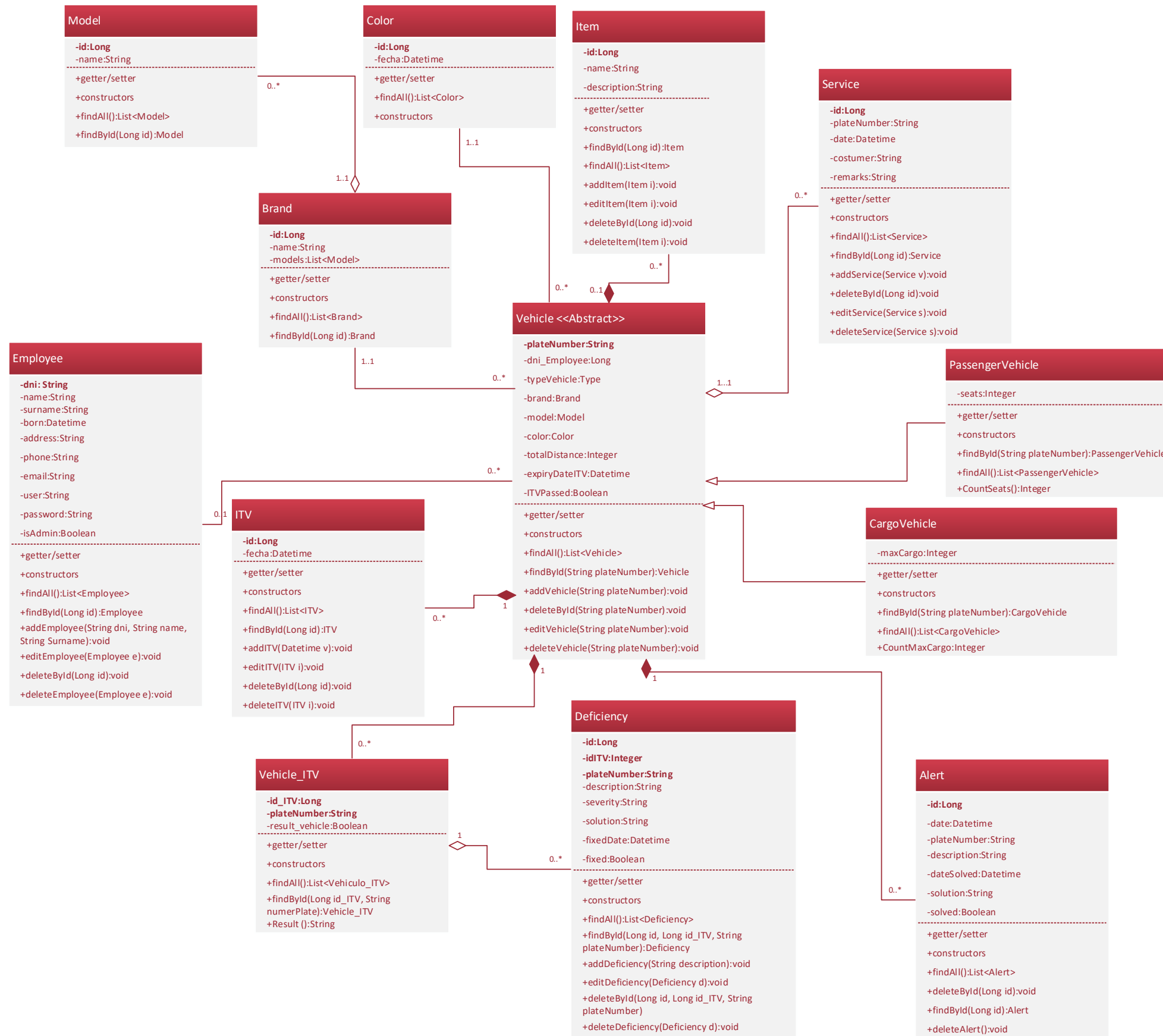




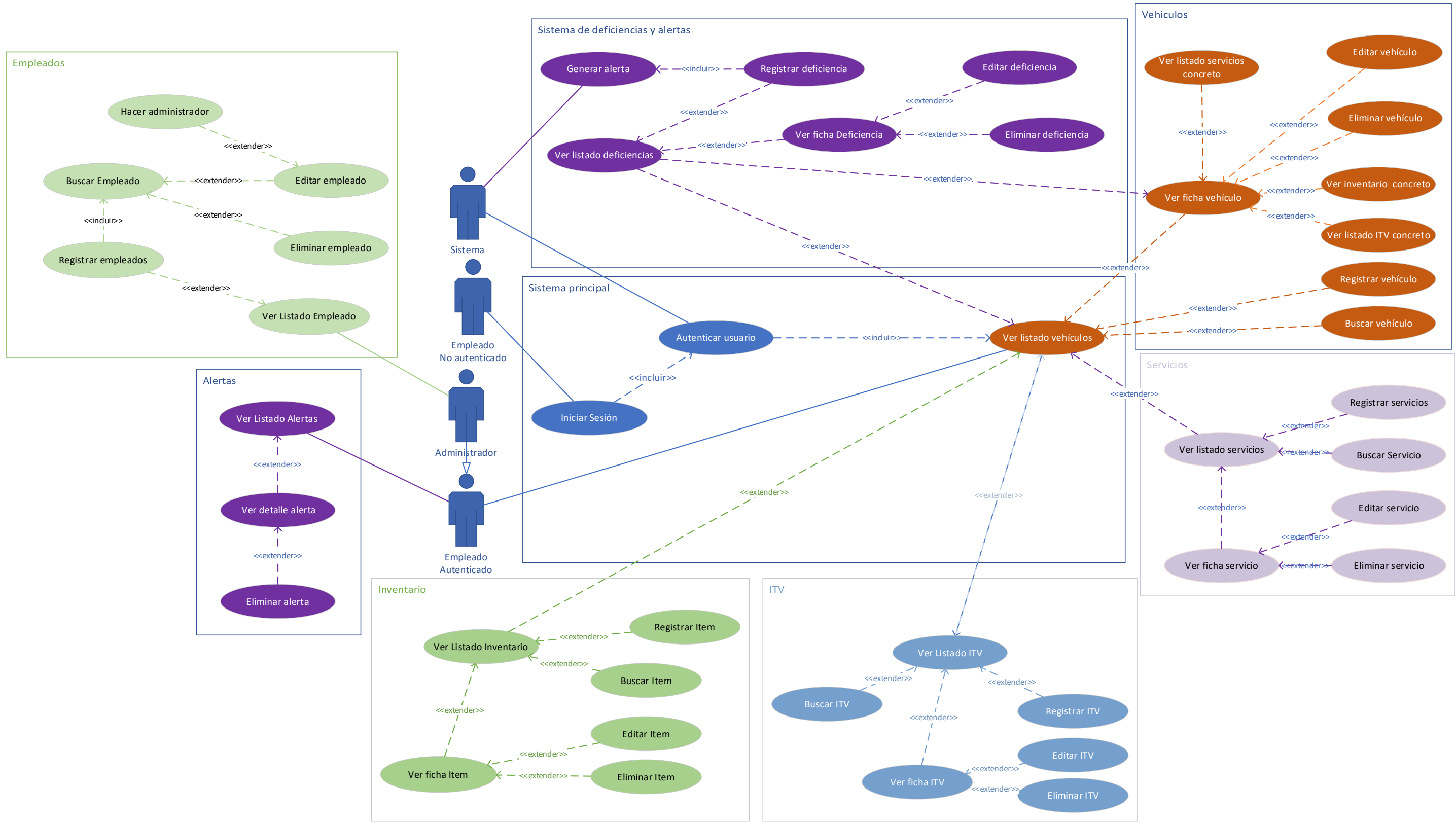
ANEXO IV – Diagrama Relacional



ANEXO V – Diagrama de Clases



ANEXO VI – Diagrama de Casos de Uso



ANEXO VI – Diagrama de Casos de Uso

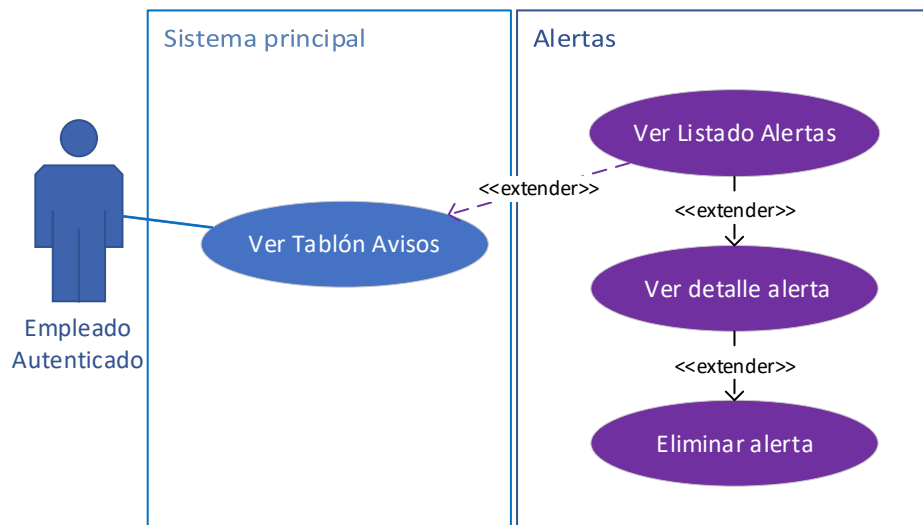


Ilustración 14: Diagrama de casos de uso de alertas

Identificador	CU_06	
Nombre	Ver listado alertas	
Descripción	Se abre al presionar el icono “Alertas” de arriba derecha del tablón de avisos.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre la alerta para abrir el detalle de la alerta. Se ejecuta CU_04
Postcondición		
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_07	
Nombre	Ver detalle alerta	
Descripción	Al presionar sobre una alerta del listado de alertas se abre una ventana emergente con su ficha detallada. En la alerta se detalla si es por fechas próximas o vencidas de ITV, o bien de deficiencias no corregidas.	
Actores	Empleado, administrador.	

ANEXO VI – Diagrama de Casos de Uso

Precondición	Empleado autenticado. Listado de alertas abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona el botón “Borrar alerta”. Se ejecuta CU_05.
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_03.
Postcondición		
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_08	
Nombre	Eliminar alerta	
Descripción	Borra la alerta permanentemente de la base de datos.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Detalle de alertas abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> ha borrado del sistema la alerta de la base de datos.
Postcondición	La alerta desaparece del listado de alertas	
Excepciones	Paso	Acción
	1	Si la deficiencia no está corregida el sistema no permitirá borrar la alerta.
Comentarios		

Identificador	CU_09	
Nombre	Buscar vehículo	
Descripción	Se abre una ventana modal de búsqueda con una línea de texto, y filtros de búsqueda por matrícula, marca o modelo.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Listado de vehículos abierto	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> introduce el texto en el campo de texto. El listado de vehículos se actualiza automáticamente según los filtros y/o el texto. Solo aparecen los vehículos que coinciden con los criterios. Volvemos a CU_06

ANEXO VI – Diagrama de Casos de Uso

	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_06
Postcondición	El listado de vehículos se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.

Identificador	CU_010	
Nombre	Ver detalle de vehículo	
Descripción	Se abre una ficha con toda la información detallada del vehículo, incluida su foto.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que existan registros de vehículos en la base de datos. Listado de vehículos abierto	
Secuencia normal	Paso	Acción
	1	El empleado presiona el botón "Editar". Se ejecuta CU_09.
	2	El empleado presiona el botón "Eliminar". Se ejecuta CU_10
	3	El empleado presiona el botón "Inventario". Se ejecuta CU_11.
	4	El empleado presiona el botón "Servicios". Se ejecuta CU_12.
	5	El empleado presiona el botón "ITVs". Se ejecuta CU_13.
	6	El empleado puede cierra la ventana. Vuelve a CU_06
Postcondición		
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_011	
Nombre	Editar Vehículo	
Descripción	Hace editable la ficha del vehículo para cambiarlos en la base de datos.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que el registro de vehículo exista en la base de datos. Detalle de vehículo abierto.	
Secuencia normal	Paso	Acción
	1	El <i>sistema</i> actualiza el registro en la base de datos.
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_08

ANEXO VI – Diagrama de Casos de Uso

Postcondición	El registro se actualiza en la base de datos y en el listado de vehículos.	
Excepciones	Paso	Acción
	1	Si los datos son erróneos se dará un mensaje de error.
Comentarios		

Identificador	CU_012	
Nombre	Eliminar vehículo	
Descripción	Borra el registro de vehículo y sus datos de la base de datos.	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que el registro de vehículo exista en la base de datos. Detalle de vehículo abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_06
	2	El <i>empleado</i> le da a cancelar en un modal. El registro no varía. Vuelve a CU_08
Postcondición	El registro desaparece de la base de datos.	
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_013	
Nombre	Ver listado de inventario concreto	
Descripción	Se ve un listado de inventario del vehículo filtrado	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que haya registros de inventario de vehículo en la base de datos. Vehículo abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre uno de los items. Ejecuta CU_016
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_08
Postcondición	Solo se ve el listado de Inventario de ese vehículo en concreto	
Excepciones	Paso	Acción
	1	Si el vehículo no tiene asignado nada en el inventario aparecerá en un mensaje de aviso.
Comentarios		

ANEXO VI – Diagrama de Casos de Uso

Identificador	CU_014	
Nombre	Ver listado de servicios concreto	
Descripción	Se ve un listado de servicios de cada vehículo filtrado	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que exista el registro de vehículo en la base de datos. Vehículo abierto.3	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre una de servicios. Se ejecuta CU_022.
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_08.
Postcondición	Solo se ve el listado de servicios de ese vehículo en concreto	
Excepciones	Paso	Acción
	1	Si el vehículo no tiene asignado servicios aparecerá en un mensaje de aviso.
Comentarios		

Identificador	CU_015	
Nombre	Ver listado de ITV concreto	
Descripción	Se ve un listado de ITV de cada vehículo filtrado	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que el registro de vehículo exista en la base de datos. Vehículo abierto.	
Secuencia normal	Paso	Secuencia normal
	1	El <i>empleado</i> presiona sobre una de las ITV. Se ejecuta CU_022
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_08
Postcondición	Solo se ve el listado de ITV de ese vehículo en concreto	
Excepciones	Paso	Excepciones
	1	Si el vehículo no tiene ITV's aparecerá en un mensaje de aviso.
Comentarios		

Identificador	CU_016
----------------------	---------------

ANEXO VI – Diagrama de Casos de Uso

Nombre	Ver listado de deficiencias concreto	
Descripción	Se ve un listado de deficiencias de cada vehículo filtrado	
Actores	Empleado, administrador.	
Precondición	Empleado autenticado. Que el registro de vehículo exista en la base de datos. Vehículo abierto.	
Secuencia normal	Paso	Secuencia normal
	1	El empleado presiona sobre una de las deficiencias. Se ejecuta CU_
	2	El empleado puede cierra la ventana. Vuelve a CU_
Postcondición	Solo se ve el listado de deficiencias de ese vehículo en concreto	
Excepciones	Paso	Excepciones
	1	Si el vehículo no tiene deficiencias aparecerá en un mensaje de aviso.
Comentarios		

Identificador	CU_017	
Nombre	Buscar item	
Descripción	Se abre una ventana modal de búsqueda con una línea de texto, y filtros de búsqueda por fecha	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Listado de inventario general abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> introduce el texto en el campo de texto. El listado de registros se actualiza automáticamente según los filtros y/o el texto. Solo aparecen registros que coinciden con los criterios. Volvemos a CU_015.
	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_15.
Postcondición	El listado de registros se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.
Comentarios		

Identificador	CU_018
----------------------	---------------

ANEXO VI – Diagrama de Casos de Uso

Nombre	Editar item	
Descripción	Hace editable la ficha del item de inventario para cambiarlos en la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de item exista en la base de datos. Detalle de item de inventario abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_016
	2	El <i>empleado</i> le da a cancelar en un modal. El registro no varía. El <i>empleado</i> puede cierra la ventana. Vuelve a CU_16.
Postcondición	El registro se actualiza en la base de datos y en el listado de inventario.	
Excepciones	Paso	Acción
	1	Si los datos son erróneos se dará un mensaje de error.
Comentarios		

Identificador	CU_019	
Nombre	Eliminar item	
Descripción	Borra el registro del inventario y sus datos de la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de item exista en la base de datos. Detalle de item de inventario abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_13
	2	El <i>empleado</i> le da a cancelar en un modal. El registro no varía. El <i>empleado</i> puede cierra la ventana. Vuelve a CU_15
Postcondición	El registro desaparece de la base de datos.	
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_020	
Nombre	Ver detalle de item	
Descripción	Se abre una ficha con toda la información detallada del vehículo, incluida su foto.	

ANEXO VI – Diagrama de Casos de Uso

Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que existan registros de inventario en la base de datos. Listado de inventario abierto	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona el botón “Editar”. Se ejecuta CU_18.
	2	El <i>empleado</i> presiona el botón “Eliminar”. Se ejecuta CU_19.
	3	El <i>empleado</i> presiona el icono de volver. Volvemos a CU_15.
Postcondición	El listado de registros se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.
Comentarios		

Identificador	CU_021	
Nombre	Buscar ITV	
Descripción	Se abre una ventana modal de búsqueda con una línea de texto, y filtros de búsqueda por fecha	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Listado de ITVs abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> introduce el texto en el campo de texto. El listado de ITVs se actualiza automáticamente según los filtros y/o el texto. Solo aparecen las ITV que coinciden con los criterios. Volvemos a CU_020
	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_020
Postcondición	El listado de ITVs se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.
Comentarios		

Identificador	CU_022	
Nombre	Ver detalle de ITV	
Descripción	Se abre una ficha con toda la información detallada de la ITV.	

ANEXO VI – Diagrama de Casos de Uso

Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que exista el registro de ITV en la base de datos. Listado de ITVs abierto	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona el botón “Editar”. Se ejecuta CU_23.
	2	El <i>empleado</i> presiona el botón “Eliminar”. Se ejecuta CU_24
	3	El <i>empleado</i> presiona el icono de volver. Volvemos a CU_20.
Postcondición		
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_023	
Nombre	Editar ITV	
Descripción	Hace editable la ficha del vehículo para cambiarlos en la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de vehículo exista en la base de datos. Detalle de ITV abierto.	
Secuencia normal	Paso	Acción
	1	El sistema actualiza el registro en la base de datos.
	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_022.
Postcondición	El registro se actualiza en la base de datos y en el listado de vehículos.	
Excepciones	Paso	Acción
	1	Si los datos son erróneos se dará un mensaje de error.
Comentarios		

Identificador	CU_024	
Nombre	Eliminar ITV	
Descripción	Borra el registro de vehículo y sus datos de la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de vehículo exista en la base de datos.	
	Paso	Acción

ANEXO VI – Diagrama de Casos de Uso

Secuencia normal	1	El <i>empleado</i> le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_14
	2	El <i>empleado</i> le da a cancelar en un modal. El registro no varía. El <i>empleado</i> puede cerrar la ventana. Vuelve a CU_16
Postcondición	El registro desaparece de la base de datos.	
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_025	
Nombre	Buscar servicio	
Descripción	Se abre una ventana modal de búsqueda con una línea de texto, y filtros de búsqueda por fecha	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Listado de servicios abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> introduce el texto en el campo de texto. El listado de servicios se actualiza automáticamente según los filtros y/o el texto. Solo aparecen los servicios que coinciden con los criterios. Volvemos a CU_025
	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_25
Postcondición	El listado de servicios se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.
Comentarios		

Identificador	CU_026	
Nombre	Ver detalle de servicio	
Descripción	Se abre una ficha con toda la información detallada de la ITV.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que exista el registro de ITV en la base de datos. Listado de ITVs abierto	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona el botón "Editar". Se ejecuta CU_28.
	2	El <i>empleado</i> presiona el botón "Eliminar". Se ejecuta CU_29.

ANEXO VI – Diagrama de Casos de Uso

	3	El <i>empleado</i> presiona el icono de volver. Volvemos a CU_25.
Postcondición		
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_027	
Nombre	Editar servicio	
Descripción	Hace editable la ficha del servicio para cambiarlo en la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de servicio exista en la base de datos. Detalle de servicio abierto.	
Secuencia normal	Paso	Acción
	1	El sistema actualiza el registro en la base de datos.
	3	El <i>empleado</i> presiona el icono de volver. Volvemos a CU_27.
Postcondición	El registro se actualiza en la base de datos y en el listado de servicios.	
Excepciones	Paso	Acción
	1	Si los datos son erróneos se dará un mensaje de error.
Comentarios		

Identificador	CU_028	
Nombre	Eliminar servicio	
Descripción	Borra el registro de servicio y sus datos de la base de datos.	
Actores	Empleado, administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de vehículo exista en la base de datos.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_25
	2	El <i>empleado</i> le da a cancelar en un modal. El registro no varía. El <i>empleado</i> puede cerrar la ventana. Vuelve a CU_27
Postcondición	El registro desaparece de la base de datos.	
Excepciones	Paso	Acción
Comentarios		

ANEXO VI – Diagrama de Casos de Uso

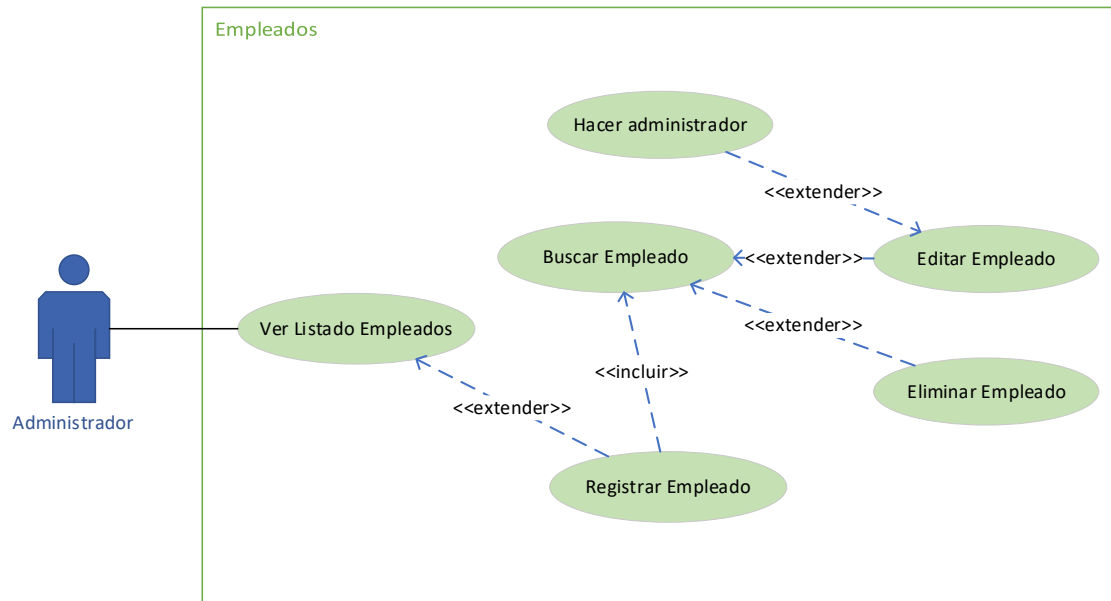


Ilustración 15: Diagrama de casos de uso de gestión de empleados

Identificador	CU_029	
Nombre	Ver listado de <i>empleados</i>	
Descripción	Se ve un listado de <i>empleados</i>	
Actores	<i>Empleado</i> , administrador, sistema.	
Precondición	Administrador o <i>empleado</i> autenticado.	
Secuencia normal	Paso	Secuencia normal
	1	El <i>empleado</i> presiona sobre un registro de <i>empleado</i> para abrir el detalle. Se ejecuta CU_032.
	2	
Postcondición	Solo se ve el listado de servicios de ese vehículo en concreto	
Excepciones	Paso	Excepciones
	1	Si el vehículo no tiene servicios aparecerá en un mensaje de aviso.
Comentarios		

Identificador	CU_030	
Nombre	Buscar <i>empleado</i>	
Descripción	Se abre una ventana modal de búsqueda con una línea de texto, y filtros de búsqueda por fecha	

ANEXO VI – Diagrama de Casos de Uso

Actores	Administrador o <i>empleado</i>	
Precondición	Administrador o <i>Empleado</i> autenticado. Listado de <i>empleados</i> abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> introduce el texto en el campo de texto. El listado de <i>empleados</i> se actualiza automáticamente según los filtros y/o el texto. Solo aparecen los <i>empleados</i> que coinciden con los criterios. Volvemos a CU_025
	2	El <i>empleado</i> cierra el diálogo mediante una flecha de la parte arriba izquierda. Volvemos a CU_25
Postcondición	El listado de servicios se actualiza según criterios de búsqueda	
Excepciones	Paso	Acción
	1	Si no aparecen registros coincidentes aparecerá un mensaje de error de búsqueda.
Comentarios		

Identificador	CU_031	
Nombre	Ver detalle de <i>empleado</i>	
Descripción	Se abre una ficha con toda la información detallada del <i>empleado</i> .	
Actores	Administrador o <i>empleado</i>	
Precondición	Administrador o <i>empleado</i> autenticado. Listado de <i>empleados</i> abierto.	
Secuencia normal	Paso	Acción
	1	El administrador presiona el botón "Editar". Se ejecuta CU_33.
	2	El administrador presiona el botón "Eliminar". Se ejecuta CU_34.
	3	El administrador marca la casilla "Hacer administrador". Se ejecuta CU_35.
	4	El administrador presiona el icono de volver. Volvemos a CU_30.
Postcondición		
Excepciones	Paso	Acción
	1,2,3	Solo el <i>empleado</i> administrador puede editar o eliminar <i>empleados</i> .
Comentarios		

Identificador	CU_032	
Nombre	Editar <i>empleado</i>	
Descripción	Hace editable la ficha del <i>empleado</i> para cambiarlo en la base de datos.	

ANEXO VI – Diagrama de Casos de Uso

Actores	Administrador.	
Precondición	Administrador autenticado. Que el registro de <i>empleado</i> exista en la base de datos. Detalle de <i>empleado</i> abierto.	
Secuencia normal	Paso	Acción
	1	El sistema actualiza el registro en la base de datos.
	3	El administrador presiona el icono de volver. Volvemos a CU_32.
Postcondición	El registro se actualiza en la base de datos y en el listado de <i>empleados</i> .	
Excepciones	Paso	Acción
	1	Si los datos son erróneos se dará un mensaje de error.
Comentarios		

Identificador	CU_033	
Nombre	Eliminar <i>empleado</i>	
Descripción	Borra el registro de servicio y sus datos de la base de datos.	
Actores	Administrador.	
Precondición	Administrador autenticado. Que el registro de <i>empleado</i> exista en la base de datos. Detalle de <i>empleado</i> abierto.	
Secuencia normal	Paso	Acción
	1	El administrador le da a aceptar en un modal. El registro es eliminado de la base de datos. Vuelve a CU_30
	2	El administrador le da a cancelar en un modal. El registro no varía. El administrador puede cerrar la ventana. Vuelve a CU_32
Postcondición	El registro desaparece de la base de datos.	
Excepciones	Paso	Acción
Comentarios		

Identificador	CU_034	
Nombre	Hacer administrador	
Descripción	El administrador nombra administrador a otros <i>empleados</i> marcando una casilla de verificación.	
Actores	Administrador.	
Precondición	Administrador autenticado. Que el registro de <i>empleado</i> exista en la base de datos. El administrador maestro no puede ser degradado. <i>Empleado</i> abierto.	

ANEXO VI – Diagrama de Casos de Uso

Secuencia normal	Paso	Acción
	1	El <i>empleado</i> se convierte en administrador marcando una casilla de la ficha de detalle. Vuelve a CU_32
	2	
Postcondición	El <i>empleado</i> nombrado administrador puede acceder a las opciones avanzadas de edición de <i>empleados</i> .	
Excepciones	Paso	Acción
	1	El administrador maestro tiene la casilla marcada permanentemente y no puede ser desmarcada.
Comentarios		

Identificador	CU_035	
Nombre	Listado de vehículos asignados a cada <i>empleado</i>	
Descripción	En la ficha de <i>empleado</i> se puede ver un listado de vehículos asignados a cada <i>empleado</i> .	
Actores	<i>Empleado</i> o administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de <i>empleado</i> exista en la base de datos. <i>Empleado</i> abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre uno de los vehículos. Ejecuta CU_016
	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_32
Postcondición		
Excepciones	Paso	Acción
	1	
Comentarios		

Identificador	CU_036	
Nombre	Listado de servicios asignados a cada <i>empleado</i>	
Descripción	En la ficha de <i>empleado</i> se puede ver un listado de servicios asignados a cada <i>empleado</i> .	
Actores	<i>Empleado</i> o administrador.	
Precondición	<i>Empleado</i> autenticado. Que el registro de servicio exista en la base de datos. <i>Empleado</i> abierto.	
Secuencia normal	Paso	Acción
	1	El <i>empleado</i> presiona sobre uno de los servicios. Ejecuta CU_027

ANEXO VI – Diagrama de Casos de Uso

	2	El <i>empleado</i> puede cierra la ventana. Vuelve a CU_32
Postcondición		
Excepciones	Paso	Acción
	1	
Comentarios		

VehicleGest

Carlos Fco. Caruncho Serrano

