MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Multiplataforma

VehicleGest

Autor: Carlos Francisco Caruncho Serrano

Tutor: Mario Gago

Fecha de entrega: dd/mm/aaaa

Convocatoria: 2º Semestre - 2022

Documentos del proyecto: Enlace a la carpeta del Drive

Índice de contenidos

1. Introducción

Vehiclegest será una aplicación móvil 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 ITV.

VehicleGest le dará a la empresa un plus de productividad al eliminar los procesos basados en papel, pizarras y documentos de Office, y permitirá la creación de informes y avisos móviles, ahorrando tiempo, y 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 mi profesión y al trabajo que realizo en ella. 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 cómo en ámbito civil, para flotas de autobuses, camiones, vehículos de transporte de personas, etc. 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, a dashboard appears with some warnings, bringing user technical incident alerts and ITV (Technical Inspection of Vehicles) schedule, 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 vehicle list, that has a vehicle search engine and filters. Also users can navigate to each vehicle datasheet, and know what ITVs and personel have assigned to.

The third section is a ITV list, that has a vehicle search engine and filters. Also users can navigate to each vehicle datasheet, and know what vehicles have assigned to

The third 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 personel.

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

Finally, it has 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.

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 and reduces 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, ITV, servicios, material y personal asociados de los vehículos.

Los **objetivos específicos** son:

- Crear una base 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 formularios y las listas deben ser adaptables al tamaño de las pantallas de los terminales móviles y tablets.
- Los usuarios deben autenticarse mediante un sistema de usuario y contraseña conectado a la base de datos, 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 inspecciones oficiales 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 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.
- Mostrar un listado de ITVs, con la información de que vehículos la han pasado, fechas, e incidencias encontradas.

- 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,** combinada con la de **cascada con retroalimentación**. Kanban es una metodología ágil que se utiliza para definir, administrar y mejorar el trabajo de una manera visual. Además, aumenta la calidad, eficiencia y la productividad en el desarrollo.

El método de **cascada con retroalimentación** se basa en el modelo tradicional en cascada, en la que se sigue una secuencia de pasos y nunca se avanza hasta que se haya completado la fase anterior, pero a diferencia de la tradicional esta permite **corregir, modificar o depurar algún aspecto.** Podemos ver las fases del modelo en cascada tradicional en la **ilustración 1**, y las iremos definiendo una por una.

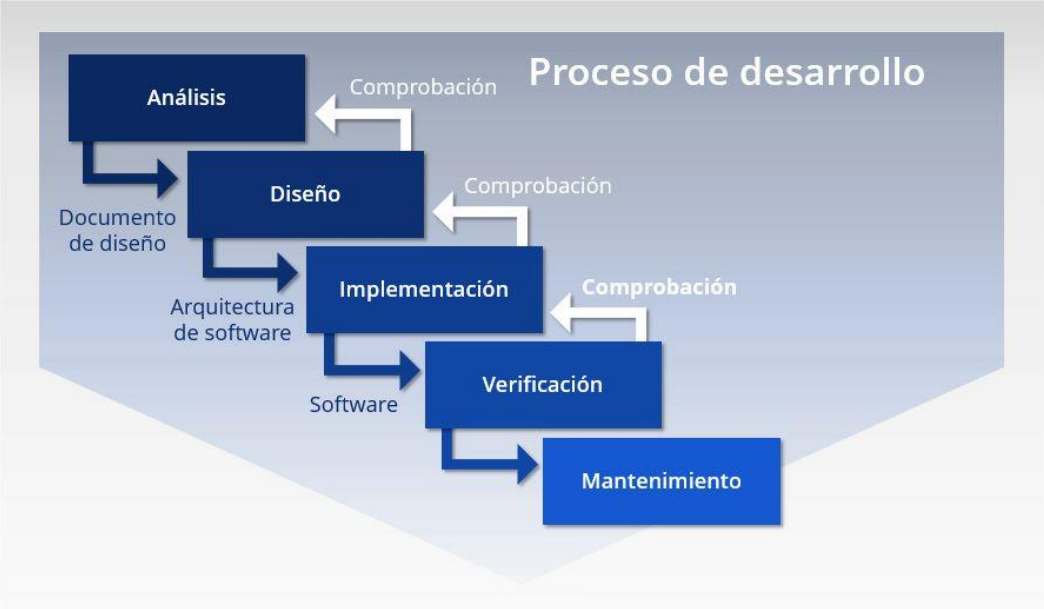


Ilustración 1: Ciclos de vida del software. Modelo en cascada con retroalimentación. (IONOS, s.f.)

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

El más básico puede presentar columnas como “Que hacer, Haciendo y Hecho” como vemos en la **ilustración 2**, pero podemos poner las columnas y filas que sean necesarias. Las tareas avanzan a través de las diferentes fases hasta que estén finalizadas.

Ilustración 2: Modelo Kanban básico. (Tecnosoluciones, s.f.)

Sus ventajas específicas para este proyecto son:

- Facilidad para arrastrar y soltar las tareas de una columna a otra, visualizando cuales están pendientes y activas, permitiendo visualizar el flujo de trabajo.
- Permite la planificación y seguimiento rápido de las tareas.
- Detallado en el seguimiento, permitiendo añadir detalles con notas, subtareas, fechas de inicio y vencimiento, etc.
- Sin límite de información, pudiendo agregar tantas columnas y tareas como se necesiten

(Martins, 2022)

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.)

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

- **Pendiente:** Esta es la fase inicial, ya que he se descarta poner una fase de “Backlog”. Los “Backlog” se suelen usar en Kanban cuando se van a introducir ideas y

funcionalidades nuevas a mitad de proyecto. 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.

- **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

❖ Iniciación

Se **escoge la temática** para la realización del proyecto, es decir 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.

Se **decide que tecnologías se van a usar** en el proyecto con la debida justificación.

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 junior cobra de media 1500 euros al mes, por lo que podríamos fijar un presupuesto inventado de 2000 euros.

❖ Planificación

Se empieza a redactar el plan de proyecto, es decir la memoria, la cual no vamos a terminar del todo hasta el final, ya que hay partes que no podremos hacer hasta tener terminada la memoria.

Se selecciona la metodología, las tecnologías y herramientas con las que vamos a trabajar.

Se hace un plan de gestión de tiempo con el calendario y se transcribe a un **diagrama de Gantt**[1], 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.

❖ Ejecución

Se van 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 definimos los objetivos del software a partir de la idea principal, y lo desglosamos en tareas más pequeñas, definiendo los requisitos funcionales y no funcionales.
- **Diseño:** Se decide cómo implementar el software y su estructura. Pasaremos los requisitos funcionales y no funcionales a un diseño de software. Se diseñan las estructuras de datos, la estructura de los componentes, la interfaz gráfica de la aplicación, y los componentes en detalle.
- **Implementación:** Primero elegimos las herramientas adecuadas para el desarrollo, entre ellas el IDE. Con el IDE y la interfaz ya diseñada, procederemos a desarrollar el código fuente. Seguimos las convenciones y normas para escribir un código claro y legible. Además, documentamos el código a medida que lo vamos escribiendo, para su posterior mantenimiento de una forma más eficiente.
- **Pruebas:** Mediante las pruebas, detectamos los errores en la codificación, para corregirlos. Se dice que una prueba es un éxito si encuentra algún error.
- **Mantenimiento:** Está fase no existe al ser un proyecto académico**.** En esta fase iríamos corrigiendo los errores que el cliente encuentre, adaptaremos el software a las nuevas necesidades de la empresa, y se le añadirían nuevas funcionalidades si fuese

[1] Cronograma para gestionar el tiempo de los proyectos.

necesario. Esta fase es la última y no se realiza en la ejecución, sí que se deja para después del cierre del proyecto.

❖ Supervisión

En la supervisión se revisa si estamos cumpliendo los plazos y los requisitos para la consecución de los objetivos. Si no es así, se corrige el rumbo y hacer ajustes en el plan original.

❖ Cierre

En esta fase se analiza si se han conseguido los objetivos del proyecto. Se debe hacer un juicio crítico para ver que ha ido bien y que se hubiera podido mejorar, y documentarlo todo. Se suben 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

❖ Software de diseño vectorial - Microsoft Visio

Microsoft Visio es una herramienta de creación de diagramas y gráficos vectoriales. 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.

❖ Entorno de desarrollo integrado – IDE - Android Studio

Un IDE no es más que una aplicación informática que facilita a los desarrolladores de software, el desarrollo del mismo, con lo que disminuye el tiempo de desarrollo, la complejidad, con lo que aumenta la productividad. Agrupa un conjunto de herramientas, para codificar, editar, testear y empaquetar. Posee un entorno gráfico para facilitar todas estas tareas. (Ilerna Online S.L.)

Para este proyecto, y tratándose una aplicación nativa para el sistema operativo Android, la mejor elección es **Android Studio**. Está desarrollado por JetBrains basado en su IDE estrella

IntelliJ Idea. 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.

❖ Lenguaje de programación - Kotlin

Con Android Studio podemos programar en los lenguajes **Java o Kotlin** indistintamente, el IDE soporta ambos lenguajes, y es capaz de traducir ficheros de uno a otro, o trabajar con ficheros de ambos lenguajes en un mismo proyecto. El lenguaje de programación seleccionado en este proyecto es Kotlin, ya que posee una serie de ventajas para la programación de aplicaciones Android. (Ilerna Online S.L.)

❖ Formato de archivos JSON

JSON (Javascript object notation) es un formato de archivo sencillo de interpretar y escribir, para almacenar datos estructurados, que sirve para intercambiar datos entre sistemas. La sintaxis viene derivada de lenguajes de programación y se almacena en archivos de texto plano. (NextU, s.f.)

❖ Sistema gestor de base de datos – Firebase Realtime DB

Una base de datos es un conjunto de datos estructurados que pertenecen a un mismo contexto y se utiliza para administrar de forma electrónica grandes cantidades de información. Gracias al avance de la tecnología, hoy en día las bases de datos están en formato digital, evitando tener que guardar como se hacía antaño todos los datos en papel. (Wikipedia, 2022)

El sistema de bases de datos que vamos a utilizar es **Firebase Realtime Database**. Es un conjunto de herramientas que Google pone a disposición de los programadores para facilitar el desarrollo de aplicaciones móviles. (Presta)

Las ventajas que han llevado a utilizar Firebase RB para el proyecto son: (Presta)

1. Es un servicio gratuito en la nube para proyectos pequeños

Firebase RB ofrece un plan gratuito para proyectos pequeños y de pago con precio adaptable a las necesidades de la aplicación.

2. Velocidad de desarrollo

Aumenta la productividad al poder centrase solo en la programación de la interfaz, ahorrando tiempo en la programación de la base de datos y el acceso a esta, es decir nos centraremos

en la programación del “Frontend” y reduciremos coste y complejidad a la programación del “Backend” de la aplicación.

3. **No es necesaria infraestructura de servidores**

Para alojar una base de datos es necesario un servidor o servidores en “cluster”. Podemos tener nuestra propia infraestructura o contratar una. En ambos casos necesitamos dedicación y conocimientos para su administración, esto se traduce en tiempo y menos productividad, a medida que los datos aumentan y el servidor escala.

4. **Monitoreo de errores**

Firebase utiliza una función denominada Crashlytics, para la detección de errores de manera rápida, que puede monitorear errores fatales y no fatales, y los informes se generan en función de cómo los errores afectan la experiencia de los usuarios.

5. **Seguridad**

Firebase garantiza la integridad y disponibilidad de los datos por medio de una copia de seguridad periódica.

Firebase también posee **desventajas,** pero se ha considerado que las ventajas prevalecen sobre estas para este proyecto. Las desventajas más significativas que nos afectan son (Presta):

1. **No es de código abierto**

Esta puede ser la mayor limitación de Firebase y evita que la comunidad mejore el producto, aumentando los niveles de flexibilidad y las opciones de auto alojamiento para los desarrolladores que no pueden pagar los precios de Firebase.

2. **Dependencia del proveedor**

Es un problema importante ya que, si quisiéramos migrar a otro proveedor, deberíamos reescribir todo el “backend”, por falta de acceso al código fuente.

3. **Usa estructuras de datos NoSql**

Por lo que no se pueden realizar consultas complejas. Migrar a una base de datos Sql no será sencillo.

4. Consultas lentas

Firebase es lenta y tiene limitaciones, en varios tipos de transacciones, consultas y otras características que afectan al rendimiento.

Existen otras desventajas, pero no afectan directamente a nuestro proyecto, debido a su entidad, alcance y tamaño.

4. Estimación de recursos y planificación

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**, plasmado en el **anexo 1.**

Un diagrama de Gantt es una herramienta de análisis que sirve para medir y planificar el tiempo de realización de un proyecto, asignar las tareas del proyecto a realizar a unos periodos de tiempo concretos, quien los debe realizar, y la relación entre ellas. (Teamleader, 2021)

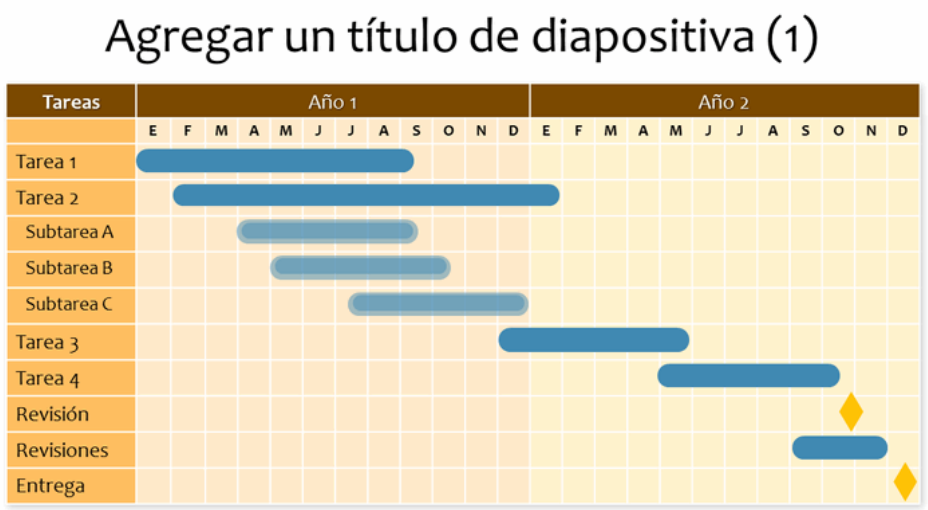


Ilustración 3: Diagrama de Gantt para proyecto de dos años (Microsoft, s.f.)

El diagrama de Gantt suele está compuesto de una lista de tareas a la izquierda y un cronograma de barras a la derecha, como vemos en la plantilla de Excel de la **ilustración 3**.

Al combinar el método ágil Kanban con un diagrama de Gantt para la planificación podremos tener las ventajas de las dos herramientas, es decir, tener una planificación visual en un tablero con los estados de las tareas y por otro lado un cronograma con la temporización para cada una de las tareas.

5. Análisis del proyecto

❖ 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.

❖ 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.

- **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.

ID	DESCRIPCIÓN
RF001	Iniciar sesión individual
RF002	Mostrar información general y avisos en la pantalla principal
RF003	Listado de alertas
RF004	Visualizar detalles de alertas
RF005	Eliminar alertas
RF006	Listado de vehículos
RF007	Buscar vehículos
RF008	Visualizar detalle de vehículos
RF009	Editar vehículos
RF0010	Eliminar vehículos
RF0011	Listado de inventario de cada vehículo
RF0012	Listado de servicios de cada vehículo
RF0013	Buscar items inventario
RF0014	Editar items de inventario
RF0015	Eliminar items de inventario
RF0016	Listado de ITVs
RF0017	Buscar ITVs
RF0018	Visualizar detalle de ITVs
RF0019	Editar ITVs
RF0020	Eliminar ITVs
RF0021	Listado de averías de cada ITV
RF0022	Listado de vehículos de cada ITV
RF0023	Listado de servicios
RF0024	Buscar servicios
RF0025	Editar servicios
RF0026	Eliminar servicios
RF0027	Listado de personal
RF0028	Buscar personal
RF0029	Editar personal

RF0030	Eliminar personal
RF0031	Listado de vehículos asignados a cada persona
RF0032	Listado de inventario general
RF0033	Buscar en inventario general
RF0034	Editar registro de inventario general
RF0035	Eliminar registro de inventario general

- **Requisitos no funcionales**

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

(VIsure 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 dan datos no bloquearan el hilo principal de la aplicación
RNF008	La aplicación seguirá los patrones de diseño de controles de Google M3[2]
RNF009	La aplicación podrá ser ejecutada en la mínima versión de Android que permita el diseño M3.
RNF0010	La aplicación debe señalar al usuario que está realizando una operación asíncrona.
RNF0011	El dispositivo debe tener conexión a internet para acceder a la base de datos Firebase.

6. Diseño del proyecto

- **Bocetos**
- Diagrama de entidad-relación. Ver **anexo II.**

[2] Google Material Design 3 es un sistema de diseño de código abierto para aplicaciones Android.

- Diagrama relacional. Ver **anexo III.**
- Diagrama de casos de uso. Ver **anexo IV**.

7. Despliegue y pruebas

8. Conclusiones

9. Vías futuras

10. Bibliografía

APD. (2021, Junio 8). *¿En qué consiste la metodología Kanban y cómo utilizarla?* From APD: https://www.apd.es/metodologia-kanban/

Garzas, J. (2011, Noviembre 22). *Kanban*. From Javier Garzas: https://www.javiergarzas.com/2011/11/kanban.html#:~:text=Las%20principales%20reglas%20de%20Kanban,como%20%E2%80%9Clead%20time%E2%80%9D).

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.

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/

Martins, J. (2022, Julio 22). *¿Qué es la metodología Kanban y cómo funciona?* From Asana: https://asana.com/es/resources/what-is-kanban

Meardon, E. (n.d.). *About Gantt Charts*. From Altassian: 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

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.

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/

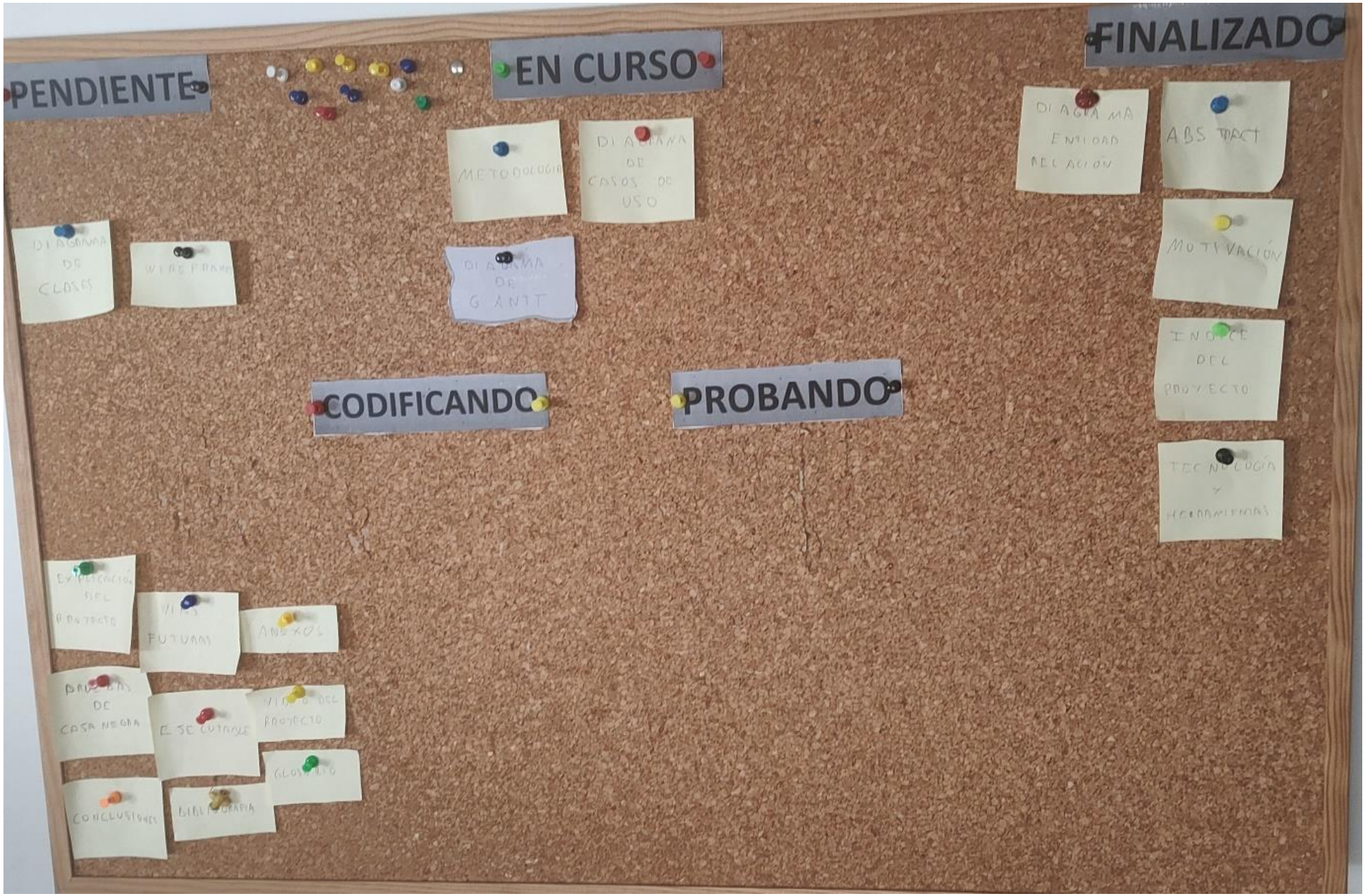
viewnext. (n.d.). *El ciclo de vida de las metodologías ágiles de desarrollo*. From viewnext: https://www.viewnext.com/el-ciclo-de-vida-de-las-metodologias-agiles-de-desarrollo/

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

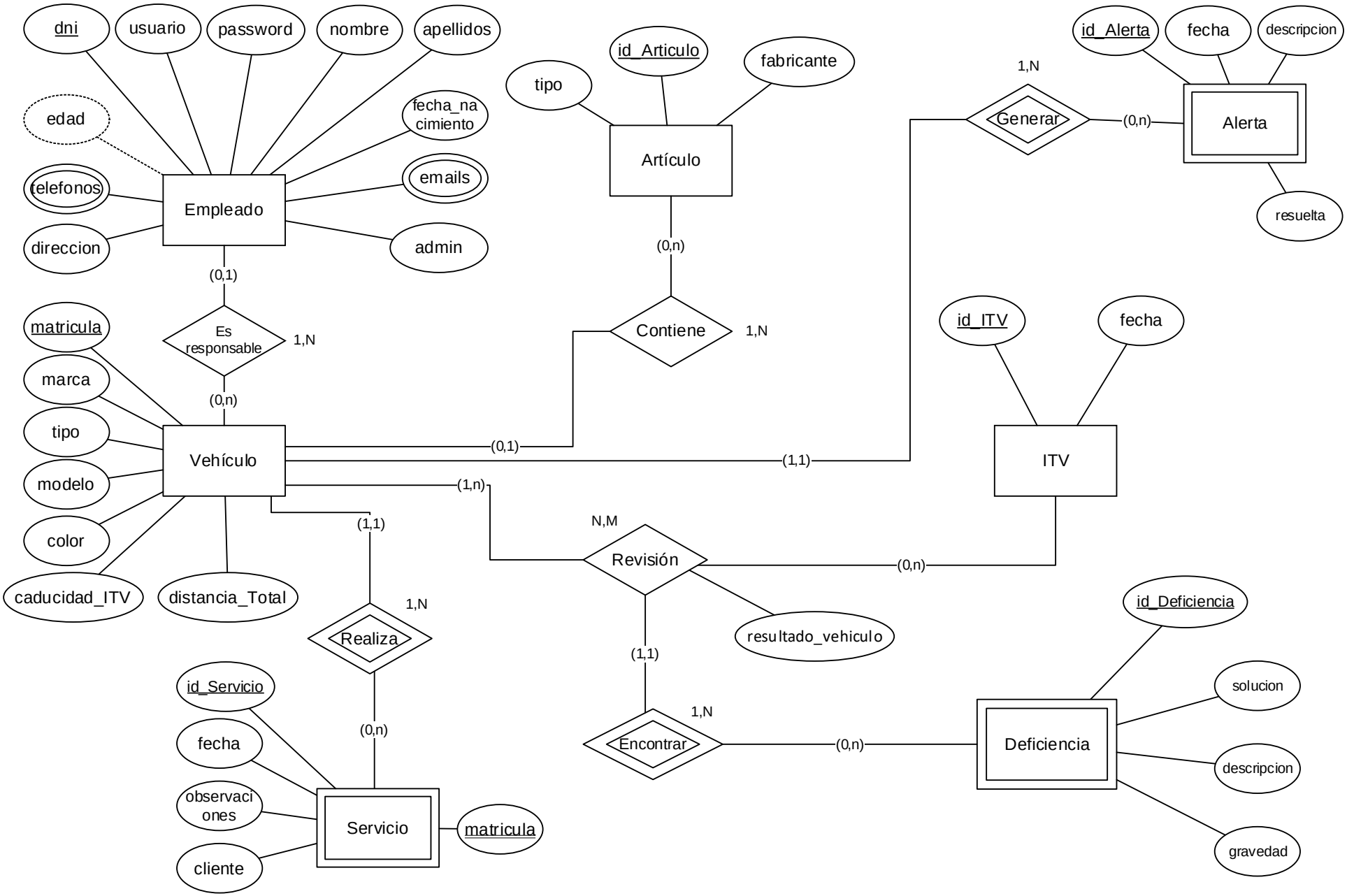
Wikipedia. (2022, Octubre 4). *Android Studio*. From Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

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

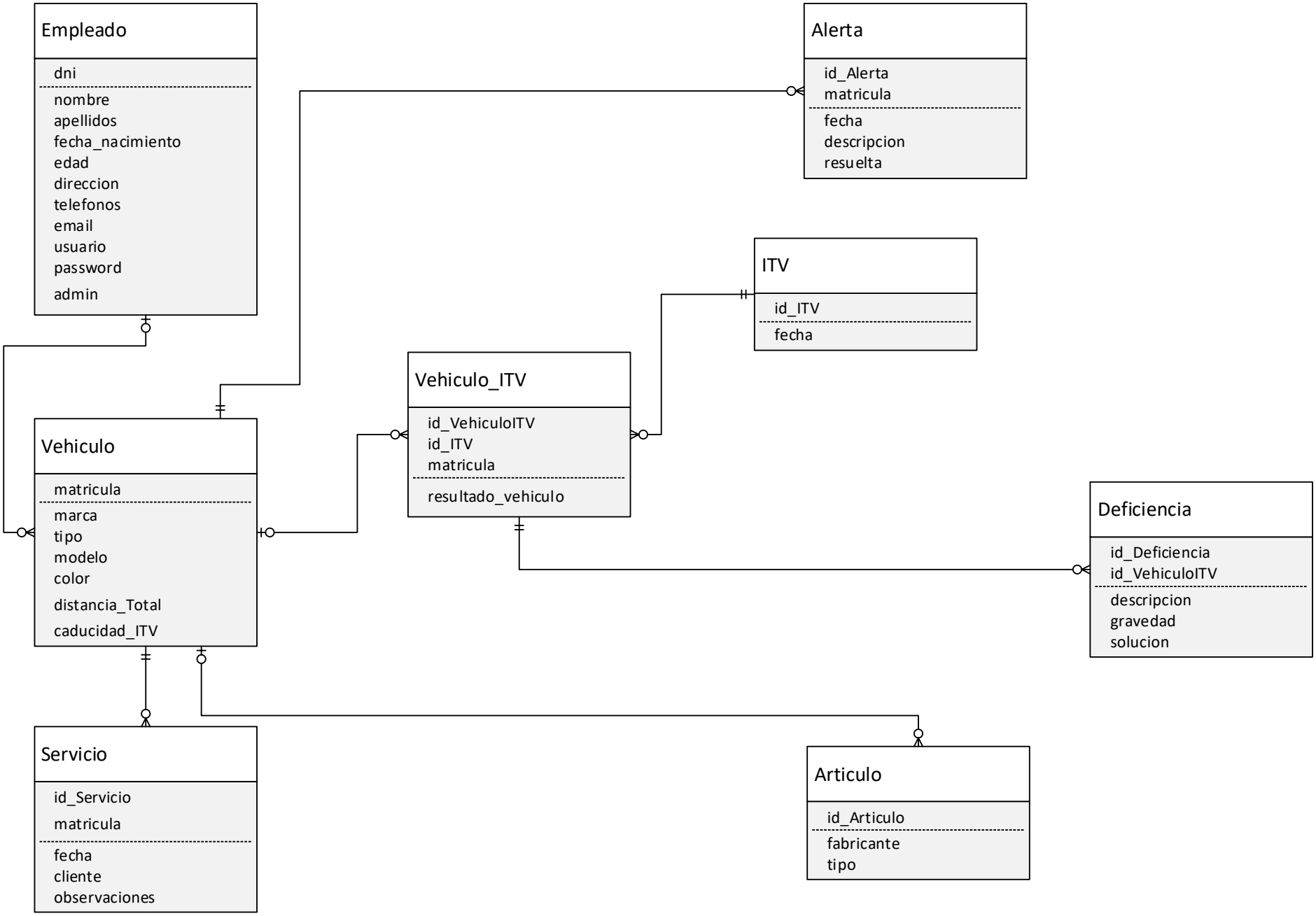
ANEXO I – Tablero Kanban



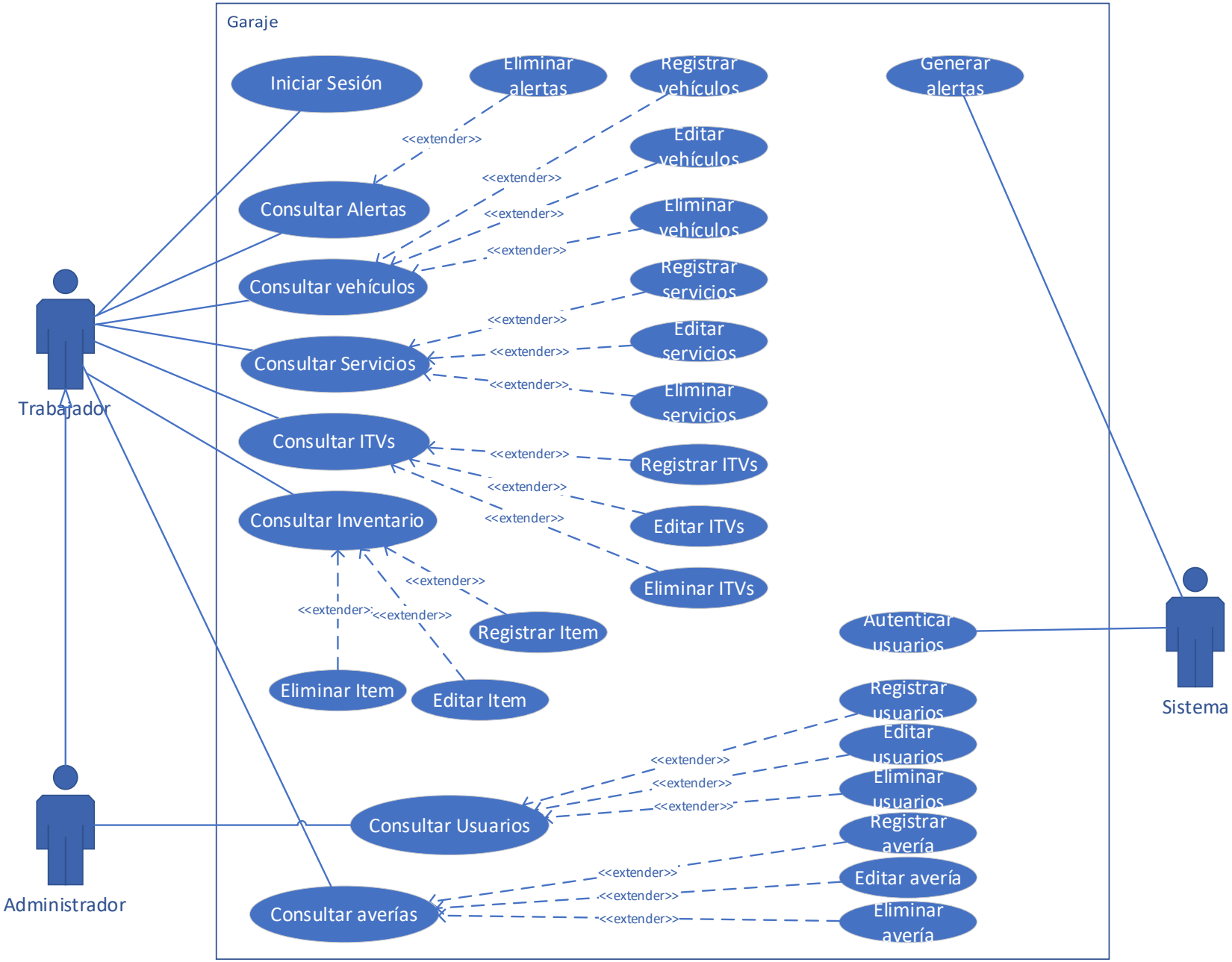
ANEXO II – Diagrama Entidad-Relación



ANEXO III – Diagrama Relacional



ANEXO IV – Diagrama de Casos de Uso



ANEXO V – Diagrama de Clases