

# MEMORIA DEL PROYECTO

*Ecomano: Aplicación web dedicada a la compra y venta de productos de segunda mano con ayuda ecológica.*

Trabajo de Fin de Grado  
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

Julio de 2021

**Autor:**

*Diego Collantes Vicente*

**Tutores:**

*André Filipe Sales Mendes*

*Gabriel Villarrubia González*

*Juan Francisco De Paz Santana*

## **Certificado de los tutores**

D. Gabriel Villarrubia González, D. Juan Francisco de Paz Santana y D. André Filipe Sales Mendes, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR

Que el trabajo titulado “Ecomano: Aplicación web dedicada a la compra y venta de productos de segunda mano con ayuda ecológica.” ha sido realizado por Diego Collantes Vicente, con DNI 21125043A y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 5 de Julio de 2021

D. Gabriel Villarrubia González   D. Juan Francisco de Paz Santana   D. André Filipe Sales Mendes

## TABLA DE CONTENIDO

Resumen.....	9
Summary .....	10
1. Introducción .....	11
2. Objetivos .....	13
2.1 Objetivos del sistema .....	13
2.1.1 Gestión de usuarios.....	13
2.1.2 Gestión de la compraventa .....	13
2.1.3 Gestión de plástico.....	13
2.1.4 Gestión de comunicación.....	13
2.1.5 Gestión de notificaciones.....	13
2.1.6 Usabilidad.....	13
2.1.7 Disponibilidad.....	14
2.2 Objetivos personales.....	14
3. Conceptos teóricos.....	15
3.1 Material Design .....	15
3.2 Framework .....	15
3.2.1 Framework progresivo .....	15
3.3 Librería .....	15
3.4 Lenguaje de programación.....	15
3.5 Lenguaje de etiquetas .....	16
3.6 Lenguaje de estilos.....	16
3.7 Frontend/Backend.....	16
3.8 Gestor de paquetes.....	16
3.9 Dashboard.....	17
4. Técnicas y herramientas utilizadas.....	18
4.1 Entorno de desarrollo .....	18
4.1.1 Visual Studio Code.....	18
4.1.2 Firefox Developer Edition.....	18
4.2 Frameworks.....	18
4.2.1 VueJS .....	18
4.2.2 Vuetify .....	18
4.3 Base de datos y autenticación.....	19
4.3.1 Firebase .....	19
4.4 Herramientas CASE.....	19

## Memoria del proyecto

4.4.1 Microsoft Project.....	19
4.4.2 EZEstimate.....	19
4.4.3 REM .....	19
4.4.4 Visual Paradigm .....	20
4.5 Lenguajes.....	20
4.5.1 JavaScript.....	20
4.5.2 HTML .....	20
4.5.3 CSS.....	20
4.6 Otras herramientas o técnicas .....	20
4.6.1 NodeJS.....	20
4.6.2 Npm.....	20
4.6.3 VueSe .....	21
4.6.4 GitHub .....	21
4.6.5 Haversine.....	21
4.6.6 Creador de logos .....	21
5. Aspectos relevantes del desarrollo .....	22
5.1 Marco de trabajo.....	22
5.2 Estimación esfuerzo .....	23
5.2.1 UUCW, Complejidad de los casos de uso .....	23
5.2.2 UAW, Complejidad de los actores.....	24
5.2.3 TCF, Factores de Complejidad Técnica.....	24
5.2.4 ECF, Factores de Complejidad del Entorno .....	24
5.2.5 Interpretación de los resultados y conclusiones.....	25
5.3 Planificación temporal.....	25
5.3.1 Inicio .....	26
5.3.2 Elaboración.....	26
5.3.3 Construcción.....	26
5.3.4 Transición .....	27
5.3.5 Interpretación de resultados y conclusiones .....	27
5.4 Especificación de requisitos .....	28
5.4.1 Objetivos del sistema .....	28
5.4.2 Catálogo de requisitos del sistema .....	28
5.4.3 Matriz de rastreabilidad .....	29
5.5 Análisis del sistema .....	30
5.5.1 Modelo de dominio .....	30
5.5.2 Paquetes de análisis y vista de arquitectura .....	30

## Memoria del proyecto

5.5.3	Realización de los casos de uso.....	31
5.6	Diseño del sistema .....	32
5.6.1	Patrón arquitectónico .....	32
5.6.2	Subsistemas de diseño .....	34
5.6.3	Vista arquitectónica .....	36
5.6.4	Realización de los casos de uso.....	37
5.6.5	Diseño de la base de datos.....	37
5.6.6	Modelo de despliegue.....	39
6.	Implementación .....	40
6.1	Instalación de NodeJS y el gestor de paquetes npm.....	40
6.2	Primer contacto con VueJS.....	40
6.3	Configuración de firebase .....	43
6.4	Creación de la aplicación.....	44
7	Funcionalidades de la aplicación.....	46
7.1	Interfaz Usuario no identificado.....	46
7.1.1	Interfaz principal .....	46
7.1.2	Interfaz Registro .....	47
7.2	Interfaz de usuario identificado .....	47
7.2.1	Interfaz mis productos .....	49
7.2.2	Interfaz mi perfil.....	50
7.2.3	Interfaz ayuda plástico .....	51
7.2.4	Interfaz home .....	52
7.2.5	Interfaz búsqueda .....	54
7.2.6	Interfaz proporcionar plástico.....	55
7.2.7	Interfaz chat .....	56
7.2.8	Interfaz notificaciones.....	57
7.2.9	Interfaz cesta.....	58
7.3	Interfaz administrador/empleado.....	59
8.	Conclusiones.....	60
9.	Líneas de trabajo futuras.....	62
10.	Referencias.....	63

## ÍNDICE DE FIGURAS

Figura 1: Producción de plástico [4] .....	11
Figura 2: Material design [5] .....	15
Figura 3: Frontend y Backend [6] .....	16
Figura 4: Dashboard .....	17
Figura 5: Logo aplicación .....	21
Figura 6: Ciclo de vida Proceso Unificado [20] .....	23
Figura 7: Métrica UCP.....	25
Figura 8: Inicio, iteración 2 .....	26
Figura 9: Construcción, iteración 4 .....	27
Figura 10: Resultados y conclusiones.....	27
Figura 11: Requisitos funcionales.....	28
Figura 12: Gestión de plástico .....	29
Figura 13: Diagrama de clases.....	30
Figura 14: Diagrama de secuencia He olvidado contraseña .....	31
Figura 15: Vista-VistaModelo-Modelo .....	32
Figura 16: Modelo-VistaModelo-Vista .....	33
Figura 17: Subsistemas de diseño .....	34
Figura 18: Vista.....	34
Figura 19: Vistas .....	35
Figura 20: Vista arquitectónica .....	36
Figura 21: Diagrama de secuencia He olvidado la contraseña.....	37
Figura 22: BBDD asociada a la gestión de usuarios.....	38
Figura 23: Modelo de despliegue.....	39
Figura 24: Complementos del proyecto.....	41
Figura 25: Dependencias del proyecto.....	41
Figura 26: Configuración del proyecto .....	42
Figura 27: Tareas del proyecto.....	42
Figura 28: Configuración firebase .....	43
Figura 29: Configuración del proyecto con firebase .....	43
Figura 30: Distribución del proyecto .....	44
Figura 31: Documentación generada con Vuese .....	45
Figura 32: Interfaz principal .....	46
Figura 33: Interfaz registro .....	47
Figura 34: Navbar .....	47

## Memoria del proyecto

Figura 35: Toolbar .....	48
Figura 36: Interfaz mis productos .....	49
Figura 37: Interfaz mi perfil.....	50
Figura 38: Interfaz ayuda plástico .....	51
Figura 39: Interfaz home .....	52
Figura 40: Interfaz home con notificaciones.....	53
Figura 41: Interfaz búsqueda .....	54
Figura 42: Interfaz proporcionar plástico.....	55
Figura 43: Interfaz chat .....	56
Figura 44: Interfaz notificaciones.....	57
Figura 45: Interfaz cesta .....	58
Figura 46: Interfaz empleado/administrador.....	59

*ÍNDICE DE TABLAS*

Tabla 1: Complejidad de los casos de uso ..... 24  
Tabla 2: Complejidad de los actores ..... 24

## Resumen

Este proyecto se desarrolla con el objetivo de realizar el Trabajo de Fin de Grado y por consiguiente el fin del grado en Ingeniería Informática. El funcionamiento principal trata sobre la compra y venta de productos con el aliciente de proporcionar plástico para así ser una aplicación amigable con el medio ambiente que falta hace.

Siempre se están realizando campañas para incentivar la recogida de plástico, su reciclaje y desuso debido al fuerte impacto que está teniendo este con respecto al medio ambiente. Estas propuestas son muy necesarias y vitales para no destruir lo más importante que tenemos que es nuestro planeta.

El cometido principal de esta aplicación es aportar una nueva idea para seguir influyendo sobre el reciclaje de plástico adaptándose a los nuevos tiempos. En la actualidad la gente actúa cada vez más en un mercado de compraventa online, las personas cuando se está hablando de ahorrar un dinero se sienten más atraídas, y prácticamente todo el mundo dispone de un dispositivo con conexión a internet. Pues bien, esta aplicación cumple con todos esos requisitos.

Si hubiese que hacer una división de la aplicación, esta se dividiría en dos partes, por un lado, tendríamos al usuario que se da de alta en la aplicación y puede poner en venta sus propios productos y también comprar los que más le convenga, pudiendo este efectuar un pago con créditos previamente conseguidos con la ayuda de proporcionar plástico y disponiendo también de un chat para la comunicación entre los diferentes usuarios y por consiguiente negociación. Por otro lado, tendríamos ese apartado de proporcionar plástico en donde el usuario puede donar plástico a la empresa con el objetivo de obtener esos créditos que te ayudan a pagar los productos a un precio más barato. Para validar que esa proporción de plástico que efectúa el usuario sea correcta, se dispone de un apartado donde el empleado de la empresa recibirá todas esas peticiones, una vez este vaya a la dirección donde el usuario haya marcado realizar esa validación, el empleado podrá decir si el plástico es válido o no, lo que conllevaría obtener créditos o no por parte del usuario.

En conclusión, las funcionalidades a llevar a cabo son la autenticación y registro de usuarios, comunicación entre usuarios, gestión de productos y gestión de plástico.

Para el desarrollo de este proyecto se han utilizado tecnologías como el framework vueJS que utiliza el patrón de arquitectura MVVM(Model-View-ViewModel) para la construcción de interfaces de usuario que engloba una estructura predefinida donde se utilizan el lenguaje de programación JavaScript, el lenguaje de etiquetas HTML5, y el lenguaje de estilos CSS. También se van a utilizar servicios de base de datos y autenticación proporcionados por la plataforma Firebase de Google. También cabe destacar que se ha utilizado Vuetify para el diseño de la aplicación, este es un framework de diseño que combina la potencia de VueJS con la estética de Material Design proporcionando componentes listos para usar.

Por último, cabe destacar que para la realización del proyecto se ha utilizado el marco de desarrollo del proceso unificado para así conseguir la memoria y anexos.

**Palabras clave:** aplicación, framework, gestión, reciclaje, compraventa.

## Summary

This project is developed with the objective of carrying out the End of Degree Project and, as a conclusion, the end of the degree in Computer Engineering. The main operation deals with the purchase and sale of products with the incentive of providing plastic to be an environmentally friendly application that is lacking.

Campaigns are always being carried out to encourage the collection of plastic, its recycling and disuse due to the strong impact it is having on the environment. These proposals are very necessary and vital so as not to destroy the most important thing we have, which is our planet.

The main purpose of this application is to provide a new idea to continue influencing plastic recycling by adapting to new times. Nowadays, people act more and more in an online sales market, people when talking about saving money feel more attracted, and practically everyone has a device with an internet connection. Well, this application meets all those requirements.

If there were to make a division of the application, it would be divided into two parts, on the one hand, we would have the user who registers in the application and can put their own products on sale and also buy the ones that suit them best, being able to this make a payment with credits previously obtained with the help of providing plastic and also having a chat for communication between the different users and therefore negotiation. On the other hand, we would have that section of providing plastic where the user can donate plastic to the company to obtain those credits that help you pay for the products at a cheaper price. To validate that the proportion of plastic that the user makes is correct, there is a section where the company employee will receive all these requests, once he or she goes to the address where the user has marked perform this validation, the employee will be able to say if the plastic is valid or not which would entail obtaining credits or not by the user.

In conclusion, the functionalities to be carried out are user authentication and registration, communication between users, product management and plastic management.

For the development of this project, technologies such as the vueJS framework that uses the MVVM architecture pattern (Model-View-ViewModel) have been used for the construction of user interfaces that includes a predefined structure where the JavaScript programming language is used, the HTML5 tagging language, and CSS style language. Database and authentication services provided by Google's Firebase platform will also be used. It is also worth noting that it has been used for the design of the Vuetify application, this is a design framework that combines the power of VueJS with the aesthetics of Material Design providing ready-to-use components.

Finally, it should be noted that the development framework of the unified process has been used to carry out the project in order to obtain the report and annexes.

**Keywords:** application, framework, management, recycling, sale.

## 1. Introducción

El motivo principal por el cual el plástico es un problema a nivel global es porque estos tardan en degradarse entre 100 y 1000 años. El plástico supone más de un 12% de la cantidad de residuos sólidos urbanos. Si se recicla se disminuirá el consumo de energía lo que se traduce como menos emisiones de dióxido de carbono a la atmósfera contribuyendo así a reducir el ya famoso efecto invernadero. [1]

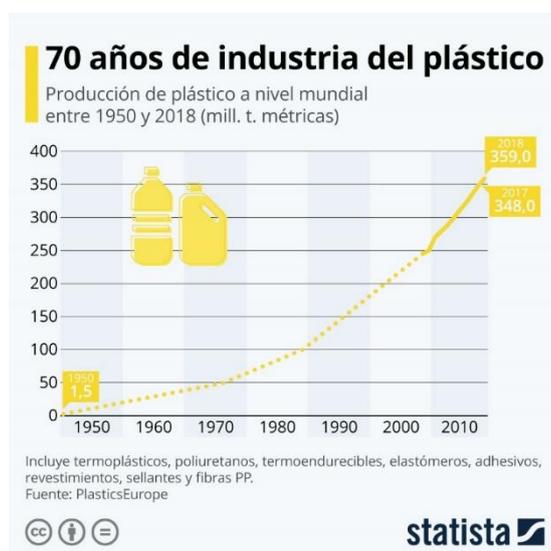
España es uno de los países de Europa que vierte más plástico al océano. Para hacer una idea del impacto que tiene se va a hablar de números, nuestro país vierte al mar unas 120 toneladas de residuos por día, es decir, cada 8 días se está echando al mar una Torre Eiffel de plástico. La mayor parte de estos plásticos tienen como finalidad ser envases de productos de un solo uso. [2]

El plástico representa un 95% de los residuos que flotan en el Mediterráneo, este mar contiene entre un 20% y un 54% de las partículas de micro plásticos del planeta los cuales los humanos ingerimos. [3]

Algunas de las consecuencias que tiene esta contaminación son:

- Problemas cutáneos
- Consumo de plásticos en la alimentación
- Degradación del ecosistema marino
- Formación de islas de basura.

Conclusión, nos estamos cargando el planeta y eso conlleva que la humanidad se está autodestruyendo. A pesar de todas estas consecuencias y efectos negativos que implica este gran problema, cada año se bate un récord de producción de plástico. Podemos observar la producción en la gráfica de la **Figura 1**, la cual representa el crecimiento de la producción, en el “eje y” se puede ver el año, el “eje x” está representado por los millones de toneladas de plástico, en 2018 casi se alcanza la cifra de 360 millones de toneladas [4]:



**Figura 1: Producción de plástico [4]**

## Memoria del proyecto

Por todos estos aspectos de los que hemos hablado anteriormente y muchos más, surge la necesidad de cuidar el planeta.

Existen muchas organizaciones, empresas y movimientos que tratan de acabar con esta contaminación, por eso hay que reforzar estas con iniciativas novedosas y creativas con ayuda de las herramientas que vayan surgiendo.

En los últimos años las tecnologías han avanzado de una forma abismal, por tanto, habría que utilizar estas para hacer cosas buenas, y que mejor que intentar acabar con los residuos de plástico que se generan.

Este proyecto surge de la idea por la preocupación que genera este residuo, creando una aplicación web que va a potenciar, o por lo menos esa es la intención, la participación de las personas a reciclar plástico ya que somos los responsables de esta atrocidad.

El proyecto consiste en aprovechar el auge que tiene la compraventa en internet, este auge viene dado sobre todo el último año, influenciado por el coronavirus.

La aplicación ofrece:

- Poder comprar productos a menor coste
- Vender productos
- Comunicación con otros usuarios dados de alta
- Como funcionalidad destacable y asociada a la principal tarea por la que surge la idea de este proyecto, proporcionar plástico para después ser reciclado.

En este documento se va a llevar a cabo la descripción del proyecto con ayuda de los anexos que constan de una descripción más técnica. La división de estos anexos se realiza en seis partes:

- Anexo I: Plan del proyecto.
- Anexo II: Especificación de requisitos software.
- Anexo III: Análisis del sistema.
- Anexo IV: Diseño del sistema.
- Anexo V: Manual del programador.
- Anexo VI: Manual del usuario.

## 2. Objetivos

En este apartado se exponen dos diferentes tipos de objetivos, los objetivos que debe cumplir el sistema que se va a desarrollar y los objetivos personales que se quieren alcanzar con la realización de este proyecto.

### 2.1 Objetivos del sistema

Para alcanzar estos objetivos la principal funcionalidad que debe alcanzar el desarrollador es la capacidad de diseñar y desarrollar una aplicación web orientada a la compraventa de productos y a promover el reciclaje de plástico. Para ver más en detalle estos objetivos, se va a realizar una división en gestión de usuarios, gestión de compraventa, gestión de plástico, gestión de comunicación, gestión de notificaciones, usabilidad y disponibilidad.

#### 2.1.1 Gestión de usuarios

Se pretende que los usuarios puedan acceder una vez se hayan registrado en la aplicación, puedan editar su perfil, en general puedan usar todas las funcionalidades de las que dispone la zona privada de la aplicación.

#### 2.1.2 Gestión de la compraventa

La aplicación tiene la capacidad de que el usuario previamente registrado, pueda acceder a la zona de sus productos, donde puede poner en venta estos. Por el contrario, estos usuarios también pueden actuar como compradores y acceder a la zona de comprar productos donde después de aplicar una serie de filtros y buscar el producto que ellos deseen puedan añadirlos estos últimos a la cesta para así después poder efectuar un pago que puede ser de diferentes maneras. El comprador puede pagar en efectivo, pagar en créditos más efectivo o pagar en créditos, estos se obtienen gracias a la proporción y posterior validación del plástico.

#### 2.1.3 Gestión de plástico

Es el objetivo distintivo de otras aplicaciones web de segunda mano, en el que se pretende que los usuarios participen en la proporción de plástico. Al utilizar esta funcionalidad se está ayudando a reciclar plástico, y además estos usuarios podrán utilizar los créditos que conseguirán si ese plástico es validado y aceptado por los empleados para así poder comprar productos a más bajo coste.

#### 2.1.4 Gestión de comunicación

Se desea conseguir que los usuarios puedan conversar y negociar el estado de los diferentes productos que deseen comprar, por lo que la aplicación ofrece la oportunidad de esta funcionalidad a través de un chat.

#### 2.1.5 Gestión de notificaciones

Se pretende que el usuario pueda observar diferentes notificaciones dependiendo de si algún usuario quiere conversar con él o tiene peticiones de plástico que han sido concurridas por algún empleado.

#### 2.1.6 Usabilidad

Se desea ofrecer una interfaz que sea fácilmente usable por parte del usuario, se utilizan botones los cuales hagan previsible la funcionalidad, apartados bien marcados ofreciendo un flujo de navegación sencillo, y unos colores y estilos que resulten familiares y cómodos.

## Memoria del proyecto

### 2.1.7 Disponibilidad

Se pretende que la aplicación esté disponible para aquellos dispositivos que tienen conexión a internet y navegador web.

### 2.2 Objetivos personales

La realización de este proyecto ha sido una práctica que he realizado con la intención de poder aprender a utilizar un entorno el cual es muy utilizado y demandado últimamente completamente desde 0.

Personalmente este proyecto me supone un reto, nuevos conocimientos, nuevos conceptos e incluso aprender a planificarme un poco mejor.

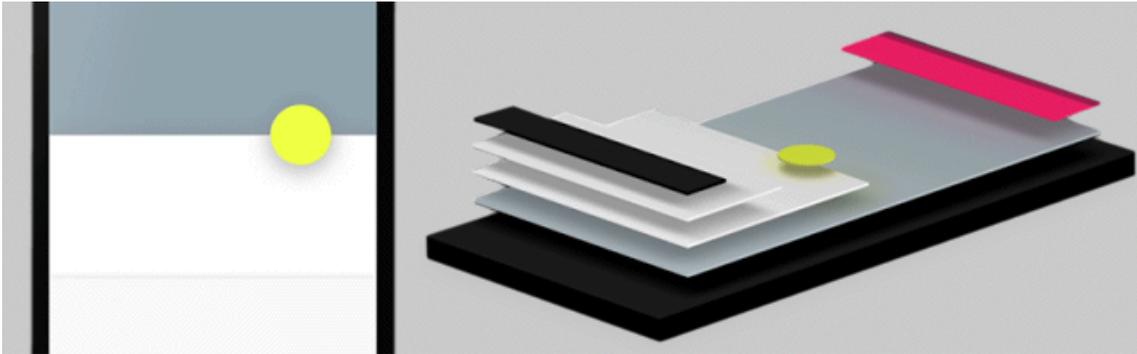
Aprender y construir desde 0 este proyecto no ha sido nada agobiante ya que le he puesto ilusión y ganas, no solo por el hecho de aprender un nuevo entorno, sino porque la idea de me ha parecido interesante, necesaria y motivadora. Desde mi punto de vista me parecen importantes estas ideas para que poco a poco se pueda construir un mundo mejor y más limpio, y si por mi parte puedo aportar mi granito de arena para ayudar al medioambiente lo hago encantado.

Llegando a una conclusión en este apartado, pienso que he ayudado con una nueva idea utilizando un nuevo enfoque para el reciclaje y además he aprendido a utilizar nuevas herramientas que creo que me ayudarán el día de mañana a aprender nuevas cosas y saber utilizarlas.

### 3. Conceptos teóricos

#### 3.1 Material Design

Planteado e implementado en 2014 por Google, para la creación y diseño de aplicaciones móviles y web. Es un lenguaje visual que combina el buen diseño y las nuevas tecnologías. Es una renovación en la interfaz en la que predominan animaciones y transiciones de respuesta, el relleno y los efectos de profundidad, como son la iluminación y las sombras. En la siguiente ilustración, **Figura 2**, se puede ver como se puede dividir una aplicación realizada con este lenguaje visual.



**Figura 2: Material design [5]**

#### 3.2 Framework

Es un marco de trabajo, se podría decir que es un esquema o estructura que se establece y que se utiliza para desarrollar un software determinado. Explicado de forma sencilla, entorno cuyo fin es hacer la programación más sencilla ya sea de una aplicación o herramienta actual. Útil para evitar repetir código, hacen que el desarrollador realice labores que con otros marcos o entornos sería prácticamente imposible. Esto ha provocado que en la actualidad se disponga de muchísimos frameworks diferentes.

##### 3.2.1 Framework progresivo

A diferencia de otros frameworks que están pensados para crear proyectos desde 0, el framework progresivo se puede añadir a cualquier proyecto en cualquier etapa de desarrollo sin complicación, pudiendo aprovechar todas las funcionalidades. Ofrecen una mayor versatilidad, rendimiento y mejor experiencia de desarrollo.

#### 3.3 Librería

Funcionalidades ya compiladas y codificadas en un lenguaje de programación. Estas ofrecen unos métodos o interfaces para ser utilizados en el proyecto. No es un programa que se ejecute, si no que su finalidad es servir a otros programas para que invoquen funcionalidades de esta.

#### 3.4 Lenguaje de programación

Es un conjunto de instrucciones a través del cual los humanos pueden interactuar con los ordenadores, nos permite comunicar con las computadoras a través de algoritmos e instrucciones que son escritos en una sintaxis que la computadora toma y entiende para después interpretar esto en lenguaje máquina. Permiten que los ordenadores procesen una gran cantidad de información de una manera rápida.

### 3.5 Lenguaje de etiquetas

Se utiliza para codificar un documento, con el texto se incorporan unas etiquetas o marcas que contienen información adicional relacionada con la estructuración del texto y de cómo va a ser su presentación.

### 3.6 Lenguaje de estilos

Sirven para la creación de una presentación del documento que se ha codificado con el lenguaje de etiquetas, es decir, se encarga de dar estilo a los diferentes documentos codificados.

### 3.7 Frontend/Backend

Frontend es la interfaz de una página web, la cual es la parte visible por los usuarios que visitan un sitio web a través de su navegador. Parte visible y accesible de una web. Frontend también es conocido como el lado del usuario ya que incluye todo lo que el usuario va a experimentar durante la navegación. Aquí se pueden encontrar lenguajes como HTML, CSS o JavaScript.

Por la otra parte se encuentra el Backend, se trata de la parte oculta de un sitio web, los usuarios no tratan con esta directamente. Aquí se almacenan y organizan datos, es decir, trata sobre la lógica de datos de una página o aplicación web para así asegurar que todo funcione correctamente con respecto al usuario.



**Figura 3: Frontend y Backend [6]**

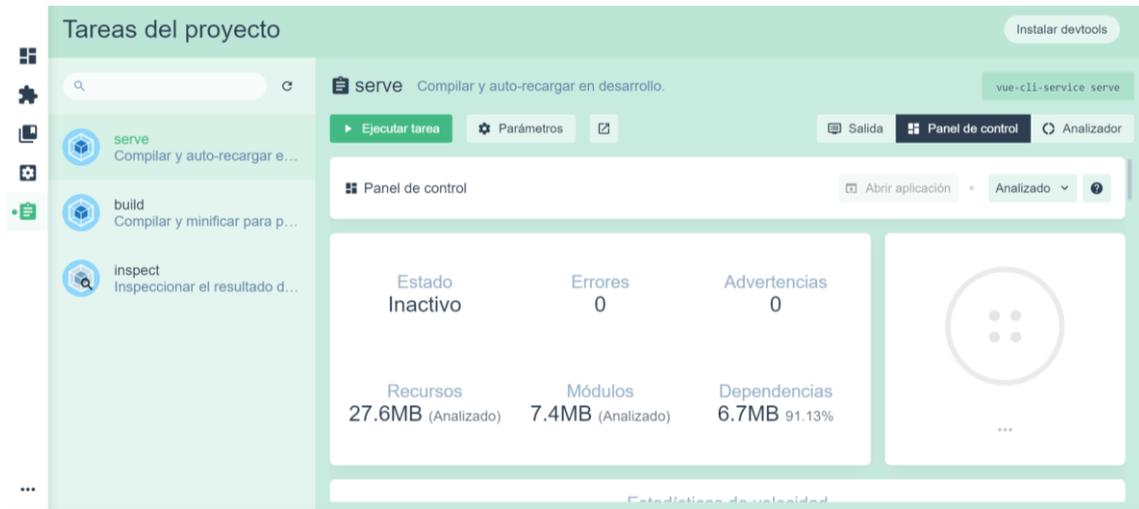
### 3.8 Gestor de paquetes

Tiene un registro software el cual está instalado en el ordenador y permite instalar software nuevo para poder utilizar, también sirve para actualizar este software instalado a versiones más recientes o también se puede eliminar de una forma muy simple, es decir, como su propio nombre indica el gestor de paquetes se encarga de gestionar paquetes, estos paquetes son conjuntos de ficheros.

### 3.9 Dashboard

Es una interfaz gráfica donde un usuario puede gestionar, monitorizar, analizar información o funcionalidades. Este elemento nos lo podemos encontrar en la vida cotidiana como el indicador de un coche, donde podemos ver información agrupada como puede ser el nivel de gasolina, la temperatura, la velocidad, las revoluciones...

Se va a mostrar un ejemplo en la **Figura 4** con un dashboard que ofrece Vue Cli 3:



**Figura 4: Dashboard**

## 4. Técnicas y herramientas utilizadas

### 4.1 Entorno de desarrollo

#### 4.1.1 Visual Studio Code

Este es un editor de código desarrollado por Microsoft disponibles para varios sistemas operativos como Windows, Linux y macOS. Este editor de código es compatible con varios lenguajes de programación. Resalta la sintaxis, ofrece la oportunidad de utilizar snippets (pequeñas partes reusables de código o texto), depuración, refactorización y ayuda a autocompletar para que el desarrollador pueda estar más cómodo y pueda ir a más velocidad. Este también se puede personalizar a gusto del desarrollador pudiendo elegir el tema que este quiera y eligiendo los atajos de teclado que le resulten más familiares. También ofrece la oportunidad de utilizar una pantalla de comandos, para ejecutar comandos, instalar paquetes y demás funcionalidades. **[7]**

##### 4.1.1.1 Vetur

Extensión de Visual Studio Code que sirve para resaltar la sintaxis adecuadamente para archivos con extensión vue. También ofrece el resalto de errores, es decir, si cometemos un error en la sintaxis, se indicará cual es. Ofrece la posibilidad de autocompletar código.

#### 4.1.2 Firefox Developer Edition

Navegador creado para desarrolladores, ofrece la oportunidad de inspeccionar animaciones, editar propiedades de CSS, permite medir la altura y el ancho donde se encuentra el cursor en ese momento entre otras funcionalidades. **[8]**

## 4.2 Frameworks

### 4.2.1 VueJS

Es un framework progresivo cuyo objetivo es crear interfaces de usuario, permite la creación de aplicaciones web de una forma más rápida y sencilla. También es muy importante señalar que este es un framework frontend. La curva de aprendizaje con este es mucho más sencilla que con Angular o React otros frameworks muy famosos destinados a la creación de aplicaciones web. Utiliza una estructura que se divide en tres partes, la primera se encuentra entre las etiquetas “template” se encuentra código HTML5, la segunda se encuentra entre las etiquetas “script” donde se encuentra código JavaScript y por último están las etiquetas “Style” que estas tienen código CSS. **[9]**

#### 4.2.1.1 Vue CLI 3

Es un asistente que ayuda a la creación y administración de los proyectos Vue, proporcionando un Dashboard, donde se pueden instalar dependencias o complementos para tu proyecto, ofrece la oportunidad de configurar, y ofrecer una serie de tareas como compilar y autorrecargar en desarrollo. Todo esto de lo que se ha hablado anteriormente se realiza de una forma gráfica.

#### 4.2.2 Vuetify

Es un framework progresivo, pero de componentes basado en Material Design, este ayuda a desarrollar aplicaciones de una manera más rápida y sencilla con un resultado estiloso utilizando estructuras predefinidas “listas para usar”, como pueden ser formularios, navbars, toolbars, botones, cards, alerts, labels, etc. Ofrece una gran capacidad de personalización de aplicaciones web. **[10]**

### 4.3 Base de datos y autenticación

#### 4.3.1 Firebase

Es una plataforma perteneciente a Google desde el 2014 integrada en la Google Cloud Platform. Se utiliza para el desarrollo de aplicaciones móvil y web. Disponible para iOS Android Web y Unity.

Ofrece una gran cantidad de servicios para el desarrollo de aplicaciones, sincronizándose de una manera sencilla.

De los diferentes servicios que ofrece se destacarán los utilizados en este proyecto, que son Authentication, Firestore database y Storage. **[11]**

##### 4.3.1.1 Authentication

Ofrece la posibilidad de creación de aplicaciones seguras, mejorando el acceso de los usuarios finales. Proporciona servicios de backend, un SDK (kit de desarrollo software, se refiere a un conjunto de herramientas) y a bibliotecas ya elaboradas para que el usuario se pueda autenticar en la aplicación creada. La autenticación que se realiza en este proyecto es mediante correo y contraseña.

##### 4.3.1.2 Firestore database

Base de datos NoSQL (not only SQL) flexible, escalable y en la nube, esta tiene como fin poder almacenar datos y sincronizarse con ellos para el desarrollo de la aplicación. La estructura que esta base de datos sigue está separada en colecciones, compuestas por una serie de documentos de identificador único, por último, estos documentos están compuestos en campos o por otras colecciones.

##### 4.3.1.3 Storage

Ofrece a los desarrolladores de estas aplicaciones la oportunidad de almacenar contenido proporcionado por los usuarios como por ejemplo fotos o videos, en el caso de este proyecto para almacenar imágenes.

### 4.4 Herramientas CASE

#### 4.4.1 Microsoft Project

Software diseñado por Microsoft, para elaborar un calendario de trabajo, identificar tareas o subtareas tomando como referencia el Proceso Unificado, identificar hitos (metas), asignar recursos a las diferentes tareas y estimar sus duraciones. En conclusión, este software sirve para realizar una estimación sobre cuánto va a durar el proyecto dependiendo de la duración de estas tareas, los recursos que están disponibles y los días de los que se dispone para la realización de este. **[12]**

#### 4.4.2 EZEstimate

Herramienta utilizada para la estimación de esfuerzo del proyecto, hace uso de trece factores de complejidad técnica, ocho factores de complejidad del entorno, a parte de estos factores a cada caso de uso se le asigna un peso. Una vez introducidos todos estos datos, la herramienta ofrece una estimación de esfuerzo que va a tener que realizar el desarrollador para realizar ese proyecto.

#### 4.4.3 REM

Esta herramienta ofrece la oportunidad de gestionar requisitos, pueden ser funcionales, no funcionales o de información, también objetivos, casos de uso, matrices de rastreabilidad, etc.

## Memoria del proyecto

Es decir, ayuda a explicar todos estos elementos pertenecientes al desarrollo del proyecto de una manera más sencilla, estructurada y rápida.

### 4.4.4 Visual Paradigm

Esta herramienta admite diagramas de UML de una forma sencilla a partir de unos requisitos dados. En este proyecto esta herramienta se ha utilizado para realizar diagrama de clases, diagramas de secuencia, diagrama de despliegue y diagramas de paquetes. No solo se describen cada uno de estos elementos pertenecientes a cada diagrama, sino que también nos ofrece la oportunidad de poder establecer una relación entre ellos. **[13]**

## 4.5 Lenguajes

Estos tres lenguajes que se van a ver en este apartado son los que van a formar parte de los ficheros con extensión vue, utilizados en este proyecto. En el apartado anterior, en el que se detallaban los conceptos, se habló de las diferentes partes de estos ficheros y que lenguaje se correspondía con cada una de estas partes.

### 4.5.1 JavaScript

Es un lenguaje de programación interpretado orientado a objetos, utilizado para la creación de páginas web. Generalmente es utilizado del lado del cliente con la colaboración de HTML y CSS. Todos los navegadores de la actualidad interpretan el código JavaScript que hay integrado en las aplicaciones web.

### 4.5.2 HTML

Lenguaje de marcas de hipertexto, define la estructura y el significado del contenido web, este se utiliza para la creación de páginas web. Comúnmente este lenguaje se ayuda de otros para la creación de aplicaciones o páginas web como pueden ser CSS o JavaScript.

### 4.5.3 CSS

Lenguaje de estilos el cual es utilizado para dar un diseño gráfico a documentos creados con HTML o XML.

## 4.6 Otras herramientas o técnicas

### 4.6.1 NodeJS

Es un entorno de tiempo de ejecución en tiempo real para el lenguaje de programación JavaScript, construido gracias al motor de JavaScript V8 de Google. Este entorno provoca que estas aplicaciones no solo se ejecutasen en navegadores si no que pudiesen ejecutarse en ordenadores como si fuesen aplicaciones independientes. También convierte el código escrito en JavaScript a código máquina más rápido. **[14]**

### 4.6.2 Npm

Gestor de paquetes que pertenece a NodeJS, este se ejecuta desde la línea de comandos y permite que se puedan instalar dependencias para su posterior uso en el proyecto. También gracias a este gestor se pueden eliminar o actualizar estas dependencias previamente instaladas a su versión más actual. **[15]**

### 4.6.3 Vuese

Se utiliza para la generación sencilla y rápida para documentación de componentes de VueJS, genera automáticamente una página de documentación elegante donde se describirán los componentes que hay en los ficheros de extensión vue, a partir de comentarios y una serie de etiquetas en el código. [16]

### 4.6.4 GitHub

Es una plataforma útil para la colaboración de desarrollo software, aloja proyectos utilizando un sistema de control de versiones. Esta plataforma permite que el usuario aloje su repositorio donde pueden colaborar con otros usuarios, dentro de este repositorio se suele guardar el código. Esta plataforma también ofrece la oportunidad de crear repositorios privados, en el caso de este proyecto se ha utilizado un repositorio publico donde se alojará la documentación que se ha generado automáticamente con la herramienta Vuese, esta se ha visto en el apartado anterior. [17]

### 4.6.5 Haversine

Es una fórmula que se utiliza para el cálculo de distancia entre dos puntos de un globo (en el caso de este proyecto será la Tierra) a partir de la longitud y la latitud de cada uno de esos puntos, en el caso de este proyecto se utiliza para la búsqueda de productos dependiendo de la distancia deseada. Para la obtención de la longitud y latitud en la que se encuentra el dispositivo que está operando sobre la aplicación web en ese momento, se hace uso de la dependencia vue-browser-geolocation. Para más información sobre la fórmula de Haversine [18]

### 4.6.6 Creador de logos

Con esta aplicación móvil disponible para Android destinada a la creación de logos [19], se ha generado un logo para este proyecto, se puede observar en **la Figura 5**:



**Figura 5: Logo aplicación**

## 5. Aspectos relevantes del desarrollo

En este apartado se detallarán los aspectos más relevantes de las distintas fases del ciclo de vida del proyecto.

### 5.1 Marco de trabajo

Se utiliza el marco de trabajo del Proceso Unificado para el desarrollo de este proyecto, este marco fue definido por la Rational Software Corporation en el año 2000. Puede utilizarse para una gran cantidad de sistemas software. Utiliza como lenguaje de modelado UML, para modelar los diferentes componentes conceptos del proyecto de una forma gráfica.

El proceso unificado está definido por las siguientes características:

- **Conducido por los casos de uso:** Estos se implementan para que la funcionalidad que se realiza en el sistema es segura y también se encargan de verificar y probar ese sistema. Los casos de uso contienen descripciones de las funciones por lo tanto influyen sobre todas las fases del ciclo de vida.
- **Centrado en la arquitectura:** La arquitectura es descrita a partir de las diferentes vistas que componen el sistema. Existen muchos modelos y vistas que definen la arquitectura del sistema software. Es de gran importancia establecer una arquitectura lo antes posible, realizar prototipos, evaluar esta e ir la refinando a medida que avanza el desarrollo del proyecto.
- **Iterativo e incremental:** Es de gran ayuda dividir el proyecto en varios proyectos más pequeños con el objetivo de poder realizar un análisis más minucioso de cada uno de estos. Cada una de estas partes en las que se ha dividido el proyecto es una iteración que según va avanzando el curso del tiempo resulta en un incremento. En relación con el concepto iteración se refiere a la ampliación y refinamiento del sistema, mientras que con incremental se refiere a que no solo habrá una serie de cambios, sino que también el sistema ira creciendo.

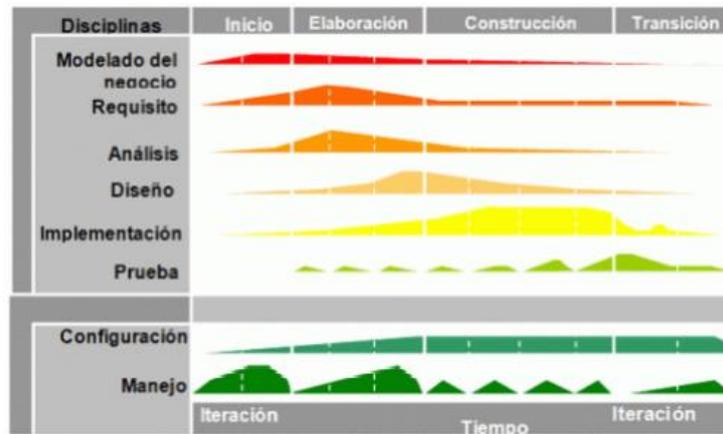
Como se ha definido anteriormente el proceso unificado que se repite a lo largo del ciclo de vida del proyecto, este ciclo de vida esta compuesto por cuatro fases. Cada una de estas fases se puede dividir para así descomponer aún más su trabajo en iteraciones con sus respectivos incrementos. También es importante hablar del concepto hito, este marca cuando termina cada una de las fases, toma decisiones para pasar a la siguiente fase, realiza un control del proyecto y proporciona información para la estimación del tiempo y recursos.

Las cuatro fases que componen este proceso son:

- **Fase de Inicio:** En esta fase, la primera del ciclo de vida, se define el alcance que va a tener el proyecto y se desarrolla el modelo de negocio.
- **Fase de Elaboración:** Aquí se lleva a cabo la planificación del proyecto, de detallan casi todos los casos de uso del sistema, en este apartado también se diseña la arquitectura del sistema.
- **Fase de Construcción:** Se corresponde con la construcción del producto, donde se realizan implementaciones y pruebas de este.
- **Fase de Transición:** Se obtendría la versión beta del producto. Se intentan corregir los problemas que vayan surgiendo, así como incorporar nuevas mejoras, en resumen, se podría decir que esta fase trata sobre el mantenimiento del producto.

Cabe destacar que cada uno de los ciclos de vida del producto correspondería con una versión de este.

A continuación, se va a mostrar la **Figura 6**, la cual se corresponde con las distintas fases que hay en el ciclo de vida del producto, así como también sus iteraciones (eje x) y disciplinas (eje y):



**Figura 6: Ciclo de vida Proceso Unificado [20]**

## 5.2 Estimación esfuerzo

En este apartado se va a realizar una estimación del esfuerzo a partir de la métrica de los puntos de caso de uso (UCP), basada en la funcionalidad representada en forma de casos de uso. A parte de los casos de uso para estimar el esfuerzo también es necesario contar con los actores, escenarios, los factores técnicos y los factores de entorno.

Para el cálculo de los puntos de caso de uso se utiliza la siguiente expresión:

$$UCP = UUCP * TCF * ECF$$

- **UUCP** son los puntos de caso de uso no ajustados, está formado por dos variables:
  - **UUCW**, complejidad y número de casos de uso.
  - **UAW**, complejidad y número de los actores.
- **TCF** se corresponden con 13 factores de complejidad técnica.
- **ECF** se corresponden con 8 factores de complejidad del entorno.

### 5.2.1 UUCW, Complejidad de los casos de uso

A cada uno de los casos de uso se les dará una complejidad que puede ser simple, media o compleja. Esta complejidad depende del número de transacciones que estos casos de uso tengan.

- **Simple:** Número de transacciones de tres o menos.
- **Media:** Número de transacciones entre cuatro y siete.
- **Compleja:** Número de transacciones de siete o más.

En el **anexo I** se pueden ver todas las tablas correspondientes a los diferentes paquetes de gestión que están compuestas por los casos de uso en detalle, en este apartado vamos a tomar como ejemplo la **Tabla 1** que se corresponde con los casos de uso de la gestión de plástico. Esto servirá para ver como a cada caso de uso dependiendo del número de transacciones que tiene se le asocia una complejidad u otra:

Caso de uso	Transacciones	Complejidad
UC-0019	4	Media
UC-0020	4	Media
UC-0021	4	Media
UC-0022	1	Simple
UC-0023	1	Simple
UC-0024	1	Simple
UC-0025	3	Simple
UC-0026	3	Simple

**Tabla 1: Complejidad de los casos de uso**

### 5.2.2 UAW, Complejidad de los actores

En este apartado se hará uso de los actores definidos en el sistema, a cada uno de estos se le asignara una complejidad, esta complejidad tiene los mismos valores que teníamos en la complejidad de los casos de uso, simple, media o compleja.

- **Simple:** Son sistemas o se comunican a través de una API con él.
- **Media:** Son sistemas que mediante un protocolo más complejo se pueden comunicar con él.
- **Compleja:** El actor se comunica con el sistema a través de una interfaz gráfica.

Actor	Complejidad
ACT-0001 (Usuario identificado)	Compleja
ACT-0002 (Usuario no identificado)	Compleja
ACT-0003 (Empleado/administrador)	Compleja

**Tabla 2: Complejidad de los actores**

### 5.2.3 TCF, Factores de Complejidad Técnica

Se compone de 13 factores y a cada uno de estos les asociamos un valor acorde con su impacto, 0-sencillo y 5-complejo, en el **anexo I** se detalla cada valor y el porqué de cada valor.

### 5.2.4 ECF, Factores de Complejidad del Entorno

Se compone de 8 factores y a cada uno de estos le asociamos un valor acorde su impacto, de la misma manera que los factores de complejidad técnica, el valor y por qué se le asocia ese valor a cada factor estará detallado en el **anexo I**.

### 5.2.5 Interpretación de los resultados y conclusiones

Con ayuda de la herramienta EZEstimate, introducimos los valores que hemos ido definiendo para la métrica de los UCP, esto se puede ver en la **Figura 7**:

Id	Module	Type	Name	complexity
1	Gestión de aute...	Usecase	Registro	Average
10	Gestión de venta	Usecase	Listar productos...	Simple
11	Gestión de venta	Usecase	Añadir producto	Average
12	Gestión de venta	Usecase	Eliminar producto	Simple
13	Gestión de venta	Usecase	Ver producto	Simple
14	Gestión de venta	Usecase	Seleccionar pro...	Simple
15	Gestión de cesta	Usecase	Listar productos...	Simple
16	Gestión de cesta	Usecase	Quitar producto...	Simple
17	Gestión de cesta	Usecase	Pagar la cesta	Simple
18	Gestión de cesta	Usecase	Añadir créditos	Simple
19	Gestión de plás...	Usecase	Proporcionar pl...	Average
2	Gestión de aute...	Usecase	He olvidado co...	Average
20	Gestión de plás...	Usecase	Aceptar plástico	Average
21	Gestión de plás...	Usecase	Rechazar plásti...	Average
22	Gestión de plás...	Usecase	Listar peticione...	Simple
23	Gestión de plás...	Usecase	Listar peticione...	Simple
24	Gestión de plás...	Usecase	Listar peticione...	Simple
25	Gestión de plás...	Usecase	Fliminar neticin	Simple

**Figura 7: Métrica UCP**

Como se puede comprobar en el apartado de “Estimation summary”, se ve un desglose de la métrica, también nos ofrece un resultado final en el que sale un valor de 1052 aproximadamente, esto son las horas que necesita el desarrollador para la realización del proyecto, si estas horas se dividen entre 8 horas que dura una jornada, se obtienen 132 días laborales.

### 5.3 Planificación temporal

Este apartado se realizará con la ayuda de la aplicación Microsoft Project, lo primero que habría que ajustar sería el calendario laboral de Castilla y León.

Para la realización de este apartado se tiene en cuenta el marco de Proceso Unificado del que se ha hablado anteriormente en detalle, se tomará en cuenta cada una de las fases e iteraciones que componen el ciclo de vida de este marco.

A continuación, se va a adjuntar un fragmento del Diagrama de barras de Gant en este proyecto, las barras azules se corresponden con la duración de las actividades, las flechas azules las conexiones que va a haber entre actividades, el fin de cada iteración estará representado gráficamente por rombos.

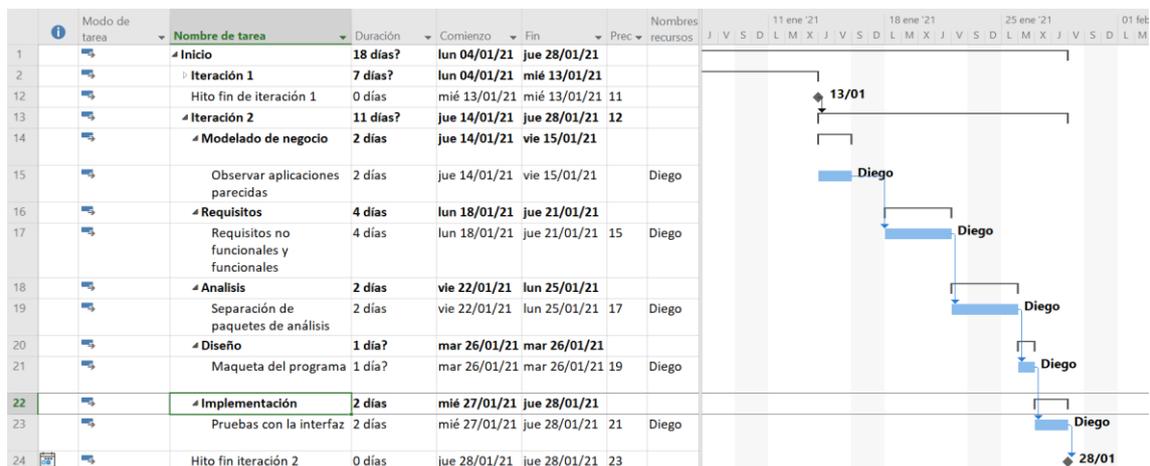
Se va a hacer un seguimiento progresivo de cada una de las fases e iteraciones que componen a cada una de estas fases. En el **anexo I** hay un estudio exhaustivo de cada uno de estos apartados, aquí se van a mostrar dos iteraciones como ejemplo del diagrama de Gant:

## Memoria del proyecto

### 5.3.1 Inicio

Primera fase a realizar, en la que se define el alcance que va a tener la aplicación, este apartado se va a dividir en dos iteraciones, en la primera se hace más énfasis sobre el modelo de negocio y requisitos, en la segunda el análisis cobra un poco más de fuerza con respecto a la anterior iteración. Como ejemplo, se va a mostrar el diagrama de Gant de la iteración 2 en la **Figura 8**:

#### • Iteración 2



**Figura 8: Inicio, iteración 2**

### 5.3.2 Elaboración

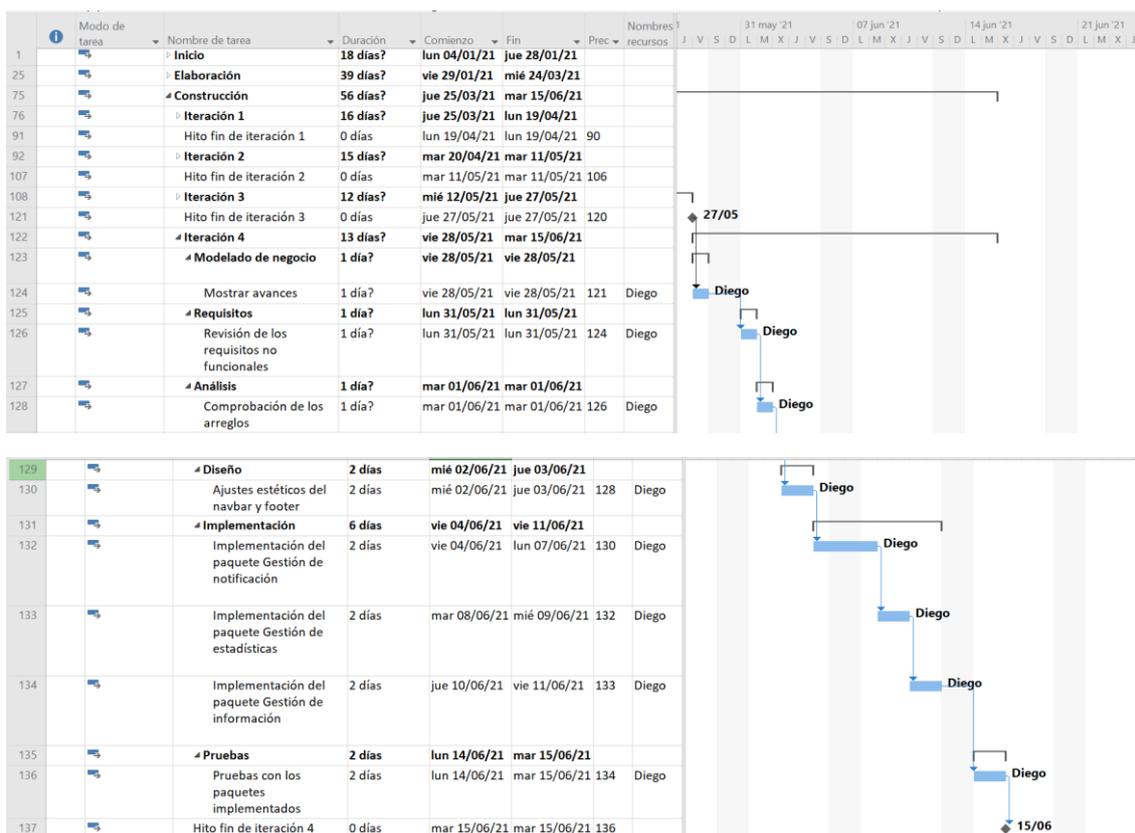
Visión un poco más refinada de la aplicación, implementa el núcleo de la aplicación, nuevos requisitos y las estimaciones previas. Esta fase se divide en tres iteraciones. En la primera iteración el análisis empieza a cobrar casi toda la importancia y el diseño requiere más tiempo. La iteración dos va a ser parecida a la primera en cuanto a la gestión de tiempos, con la diferencia que aquí se van a analizar otros paquetes de gestión. En la iteración tres se apoya y corrigen las dos anteriores iteraciones.

### 5.3.3 Construcción

Se va a construir el proyecto hasta que se consiga la versión beta de la aplicación, afinando detalles anteriores. Esta fase se divide en cuatro iteraciones. En estas iteraciones cabe destacar que la implementación toma casi todo el protagonismo. Como ejemplo en esta fase se muestra en la **Figura 9** el diagrama de Gant de la iteración 4:

# Memoria del proyecto

- **Iteración 4**



**Figura 9: Construcción, iteración 4**

### 5.3.4 Transición

Última fase donde la aplicación debe haber finalizado, aquí se lleva a cabo el mantenimiento y el estudio de futuras mejoras. Esta fase se divide en dos iteraciones. En la primera se realizan las ultimas muestras. En la segunda se harán los últimos remates.

### 5.3.5 Interpretación de resultados y conclusiones

Microsoft Project ofrece una estimación del tiempo que va a tener que trabajar el desarrollador para la realización del proyecto, en este caso al desarrollador que es Diego 1000 horas como se puede observar en la **Figura 10**:

ID	Nombre del recurso	Trabajo
	↳ Sin asignar	0 horas
1	↳ Diego	1.000 horas

**Figura 10: Resultados y conclusiones**

Si se dividen esas 1000 horas entre 8 horas que duraría una jornada, salen 125 días laborables.

## 5.4 Especificación de requisitos

En este apartado se va a recoger la especificación de los distintos requisitos software del sistema a desarrollar, para el desarrollo de este apartado se ha seguido la metodología de Durán y Bernárdez, la estructura que ofrece esta es:

- Organizaciones
- Participantes
- Objetivos del sistema
- Catálogo de requisitos del sistema
  - Requisitos de información
  - Requisitos funcionales
    - Diagrama de paquetes
    - Diagrama de actores
    - Casos de uso del sistema
  - Requisitos no funcionales
- Matriz de rastreabilidad

Para realizar un seguimiento con más detalle sobre este apartado es necesario consultar el **anexo II** donde se pueden ver todos los componentes que forman cada uno de los apartados. En este apartado se va a comenzar a ver directamente desde objetivos del sistema.

### 5.4.1 Objetivos del sistema

Aquí se explican los objetivos que el sistema tiene que cumplir para poder satisfacer los requisitos planteados.

### 5.4.2 Catálogo de requisitos del sistema

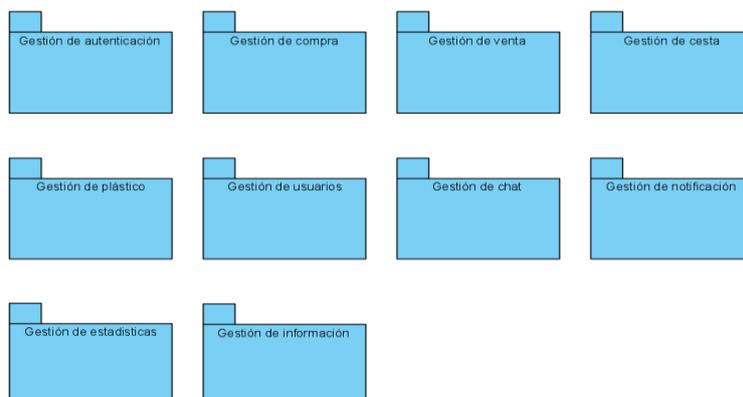
#### 5.4.2.1 Requisitos de información

Indican los datos que el sistema debe almacenar y administrar.

#### 5.4.2.2 Requisitos funcionales

Indican que servicios debe proporcionar el sistema e indicar el comportamiento que debe adoptar en cada situación:

- **Diagrama de paquetes:** Ofrece una división del sistema en paquetes durante la fase de dominio del problema, estos paquetes se pueden observar en la **Figura 11**:



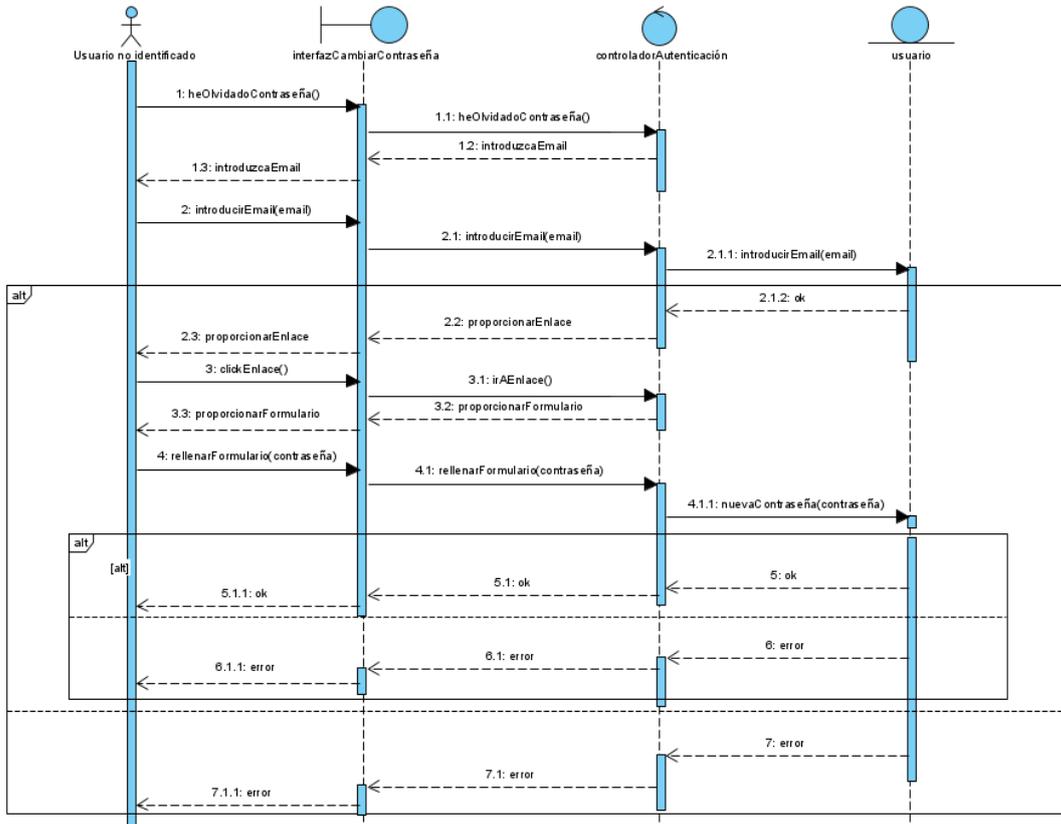
**Figura 11: Requisitos funcionales**





### 5.5.3 Realización de los casos de uso

Esta representación se realiza a través de los diagramas de secuencia donde se refleja un intercambio de mensajes entre los distintos componentes del sistema, como ejemplo tomado del **anexo III** se toma el diagrama de caso de uso de He olvidado contraseña. Este diagrama se observa en la **Figura 14**.



**Figura 14: Diagrama de secuencia He olvidado contraseña**

## 5.6 Diseño del sistema

El diseño es la fase posterior a la especificación de requisitos y la anterior a la fase de implementación. Se refinará lo hallado en el análisis del sistema ofreciendo una solución al problema. Se modelarán y analizarán los requisitos que proporcionarán la información necesaria para la construcción de modelos de diseño.

Se describe la estructura del sistema y se verá en detalle cómo va a estar implementado, se analizarán y se realizarán los requisitos arquitectónicos del sistema. En este apartado se va a obtener una división en componentes y la colocación de estos dentro de una arquitectura a más alto nivel de detalle.

En el **anexo IV** se obtiene la información de este apartado con más detalle.

### 5.6.1 Patrón arquitectónico

Este describe la estructura de diseño, el resultado sería una solución a un problema concreto del diseño dentro de un contexto.

En el caso de este proyecto se ha utilizado el patrón MVVM (Modelo-Vista-VistaModelo), fue creado por "Martin Fowler", apoya plataformas de desarrollo de interfaz de usuario, soportando programación orientada a eventos.

Este patrón separa la vista de la lógica de la aplicación, a diferencia de otros patrones famosos como el MVC (Modelo-Vista-Controlador) el cual ha sido visto en varias asignaturas del grado, el MVVM tiene entre medias del modelo y la vista un nuevo componente llamado ViewModel (VistaModelo) sustituyendo al controlador. En el MVVM la vista es la responsable de la gestión de las acciones indicadas por parte del usuario.

A continuación, en la **Figura 15** y **Figura 16** se muestra un esquema del funcionamiento de este patrón arquitectónico Modelo-Vista-VistaModelo:



**Figura 15: Vista-VistaModelo-Modelo**



**Figura 16: Modelo-VistaModelo-Vista**

A continuación, se va a explicar la funcionalidad que tiene cada uno de los componentes del patrón MVVM:

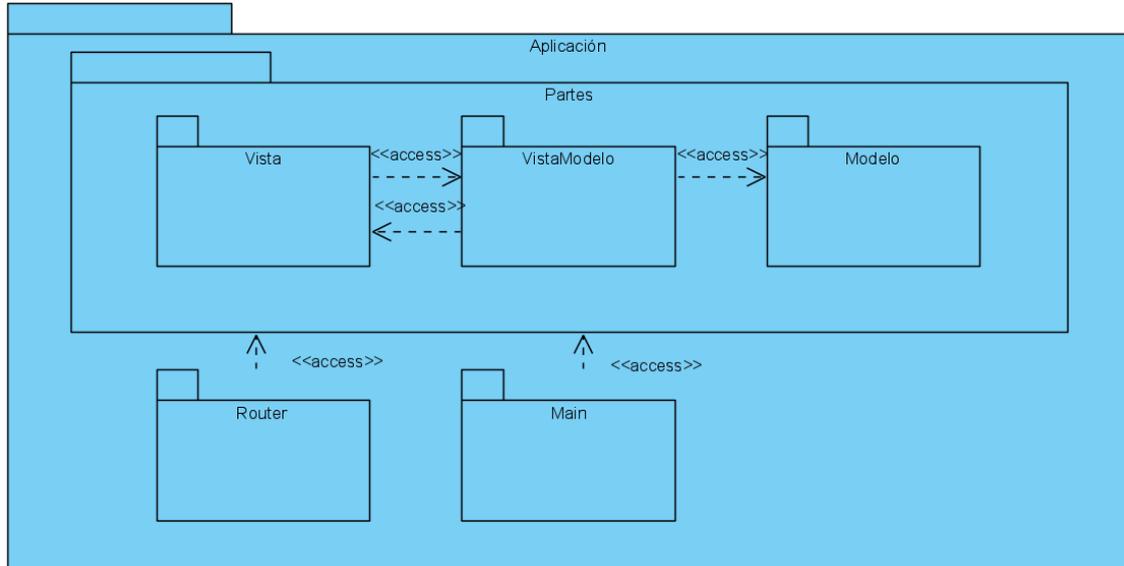
- **Modelo:** Es el encargado de la lógica y del estado/datos de la aplicación, contiene la base necesaria para realizar una manipulación de los datos que se encuentran en la aplicación. El modelo no puede estar en contacto directamente con la visualización de los datos.
- **Vista:** Muestra la información al usuario, recibirá y manejará los eventos, e invocará métodos de la VistaModelo. No tiene influencia sobre el manejo directo de los datos.
- **VistaModelo:** Componente que se encuentra entre los dos anteriores, controla la lógica de la vista y maneja las invocaciones del modelo.

Ventajas de este patrón: Separación entre la visualización de los datos y lógica de estos, las pruebas con el modelo se realizan con independencia a la vista y el rediseño de la interfaz es más sencillo.

Este patrón es el que utiliza el framework VueJS, con el que se ha realizado la aplicación.

### 5.6.2 Subsistemas de diseño

En la **Figura 17** se realizará una división detallada de las partes que componen el sistema con ayuda de diagramas, la división del sistema se realizará en tres partes, una parte estaría compuesta a su vez por otras tres que serían el Modelo, la Vista y la vistaModelo, las otras dos partes serían el Router y Main.

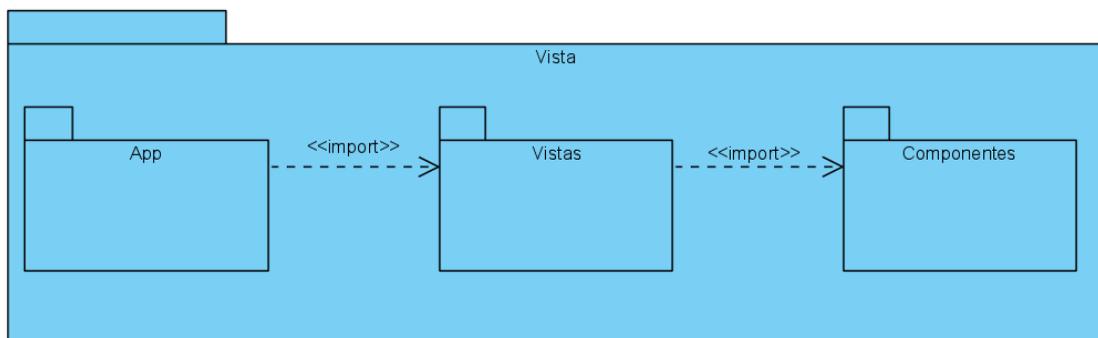


**Figura 17: Subsistemas de diseño**

A continuación, se mostrará cómo se realizaría un desglose de cada una de las partes del diagrama anterior, como ejemplo en este apartado se va a mostrar el desglose de la vista, para ver los demás desgloses se recomienda el acceso al **anexo IV**.

#### 5.6.2.1 Vista

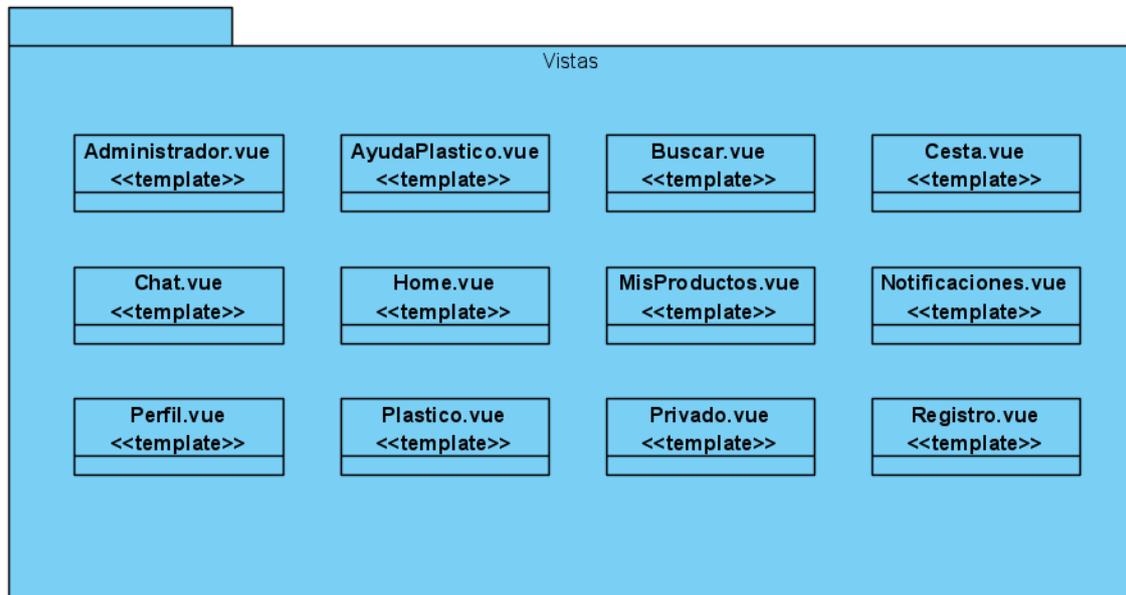
Realizando un diagrama con un nivel más alto de detalle sobre la vista del sistema, en la **Figura 18** se pueden observar tres componentes dentro de ella:



**Figura 18: Vista**

Dentro de la vista se encuentran otros tres, estos son App, Vistas, y Componentes, los dos últimos se diferencian en que los componentes como su propio nombre indica, pueden ser reutilizados en diferentes partes de las Vistas, mientras que la parte de App va a estar presente

en la totalidad de la aplicación y dentro de esta se ira encontrando con cada una de las diferentes vistas. A continuación, en la **Figura 19**, se va a mostrar como ejemplo como está dividido el componente Vistas:

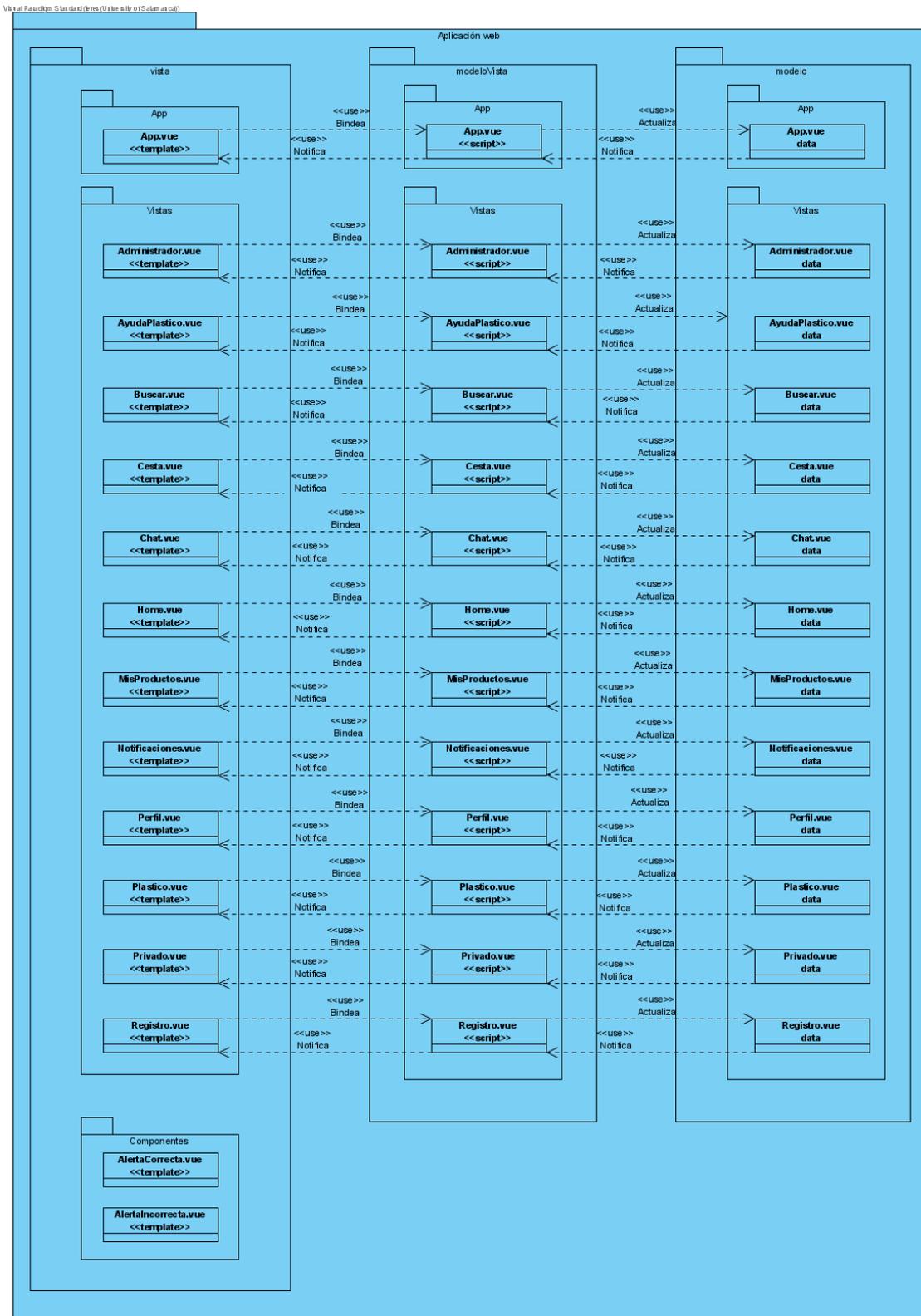


**Figura 19: Vistas**

Se destaca la etiqueta "template" se corresponde con la Vista.

### 5.6.3 Vista arquitectónica

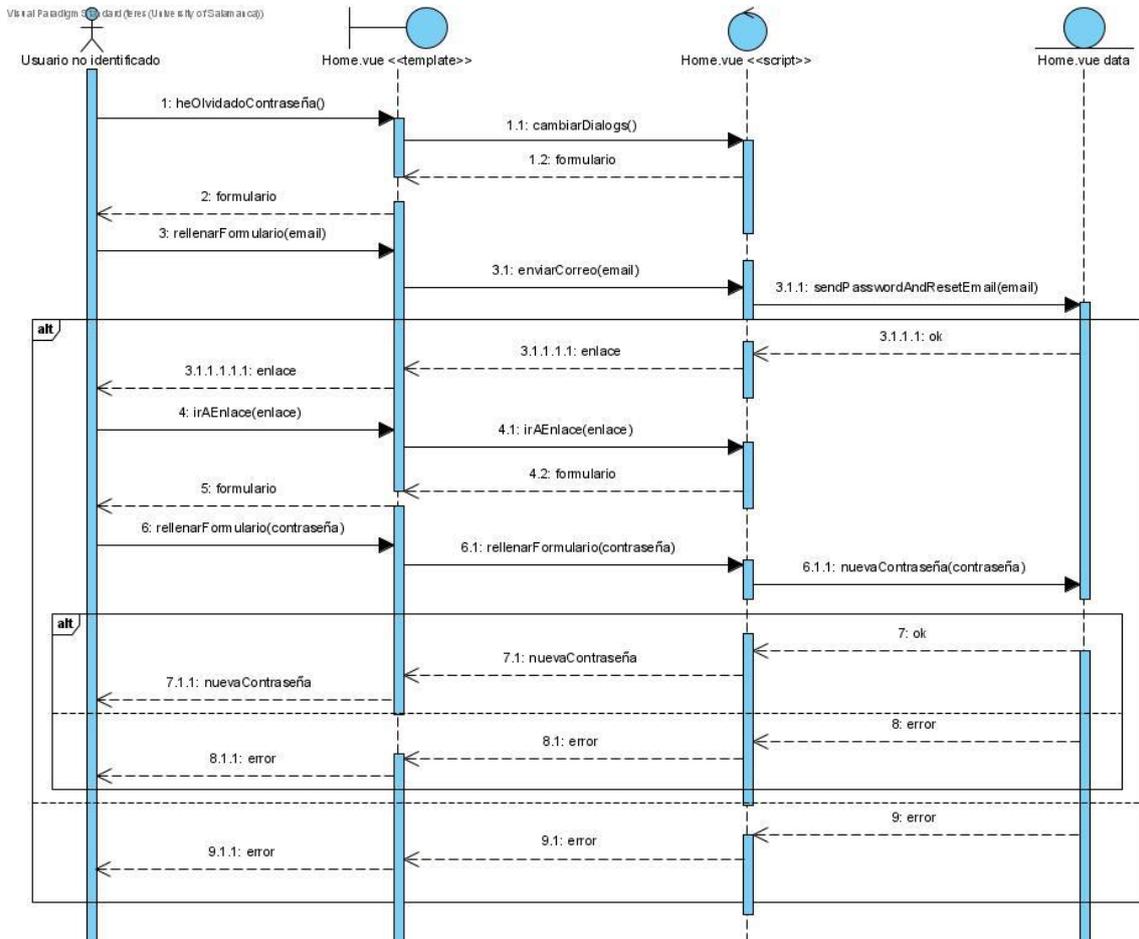
Se va a mostrar una organización global de los paquetes, estos pertenecen al patrón arquitectónico MVVM previamente definido. Esta vista se puede observar en la **Figura 20**.



**Figura 20: Vista arquitectónica**

### 5.6.4 Realización de los casos de uso

Se van a realizar los diagramas de secuencia asociados a los casos de uso, estos diagramas van a estar asociados a las estructuras del sistema, componentes y subsistemas realizados. Se tomará como ejemplo ,en la **Figura 21**, el mismo caso de uso que se eligió en los diagramas de secuencia del análisis del sistema, He olvidado contraseña:



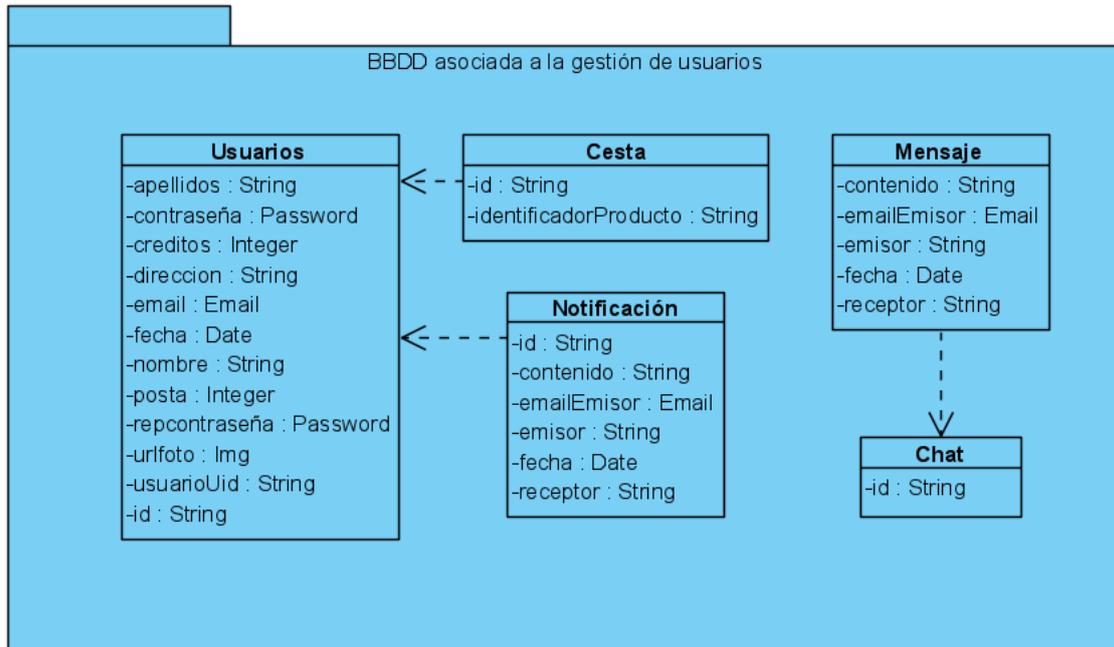
**Figura 21: Diagrama de secuencia He olvidado la contraseña**

### 5.6.5 Diseño de la base de datos

En este apartado se realiza un diseño asociado con la distribución de los datos en la base de datos y la relación que hay entre ellos.

Para la utilización de la base de datos se hace uso de firebase que ofrece servicios de base de datos como por ejemplo firestore, dentro de firestore se van a guardar los datos en forma de colecciones, dentro de estas se encuentran documentos identificados de forma única, por último, dentro de estos documentos se encuentran los diferentes campos que los componen. También cabe destacar el servicio storage, también proporcionado por firebase, en este caso este servicio sirve para guardar archivos, en esta aplicación en concreto para guardar imágenes.

En el **anexo IV** se realiza un desglose de esta base de datos para una mejor apreciación, como ejemplo en este apartado, se va a mostrar la distribución de la base de datos asociada a la gestión de usuarios en la **Figura 22**:

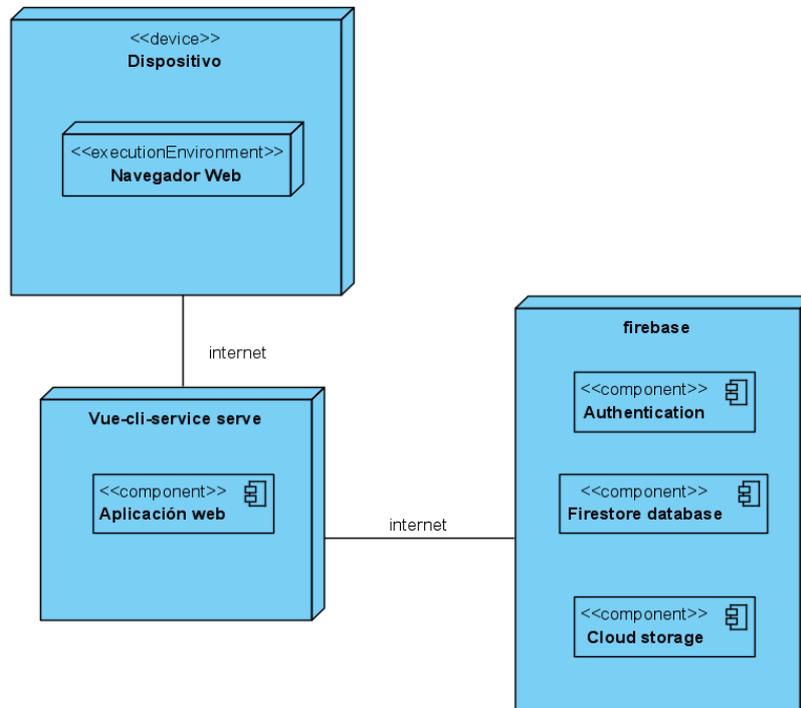


**Figura 22: BBDD asociada a la gestión de usuarios**

El nombre de cada clase se corresponde con el nombre de la colección y los atributos se corresponden con cada uno de los campos. Las flechas significan que una colección pertenece a otra, es decir, la colección Cesta y la colección Notificación son colecciones que están dentro de la colección Usuarios.

### 5.6.6 Modelo de despliegue

Aquí se recogen los nodos y componentes que forman el sistema, y también los canales mediante los que interactúan estos nodos. El modelo de despliegue se puede ver en la siguiente ilustración, **Figura 23**:



**Figura 23: Modelo de despliegue**

El sistema está formado por tres nodos, Dispositivo, Vue-cli-service serve, y firebase.

- **Dispositivo**: va a disponer de un navegador web que va a ser el entorno de ejecución.
- **Vue-cli-service serve**: va a crear una instancia local del servidor para poder ejecutar la aplicación.
- **Firebase**: nos proporcionará servicios de autenticación y base de datos.

## 6. Implementación

Para la realización de la implementación del sistema sobre él que está basado este proyecto, se ha hecho uso de todos los apartados anteriores, desde la utilización de las diferentes herramientas, aprendizaje y aplicación de cada uno de los conceptos descritos, tomando en cuenta el análisis y el diseño del sistema.

Para la implementación del sistema se ha hecho uso de varios foros y tutoriales de internet debido a que es la primera experiencia por parte del desarrollador con aplicaciones de este tipo, aplicaciones web.

Para información sobre la estructura de carpetas y ficheros que forman el proyecto es necesario acudir al **anexo V**, en este también se encontrará información relativa al código generado por parte del desarrollador, no obstante, al final de este apartado se hablará un poco de esto.

### 6.1 Instalación de NodeJS y el gestor de paquetes npm

Lo primero que se necesita instalar es NodeJS desde la página oficial de NodeJS [14], con la instalación de NodeJS se encuentra el gestor de paquetes npm. En este caso se instalará NodeJS para Windows. A continuación, para ver si tenemos ya integrado npm, abrimos una terminal y escribimos el comando:

```
npm -v
```

Este indicará la versión de npm que hay instalada y por tanto indica que se ha instalado bien y se puede utilizar, para cualquier duda con respecto a npm se recomienda utilizar el comando:

```
npm help npm
```

### 6.2 Primer contacto con VueJS

Es necesario ir a la página [21] donde podemos observar cómo se puede instalar VueJS, en esta página se van a mostrar los comandos necesarios que se pueden utilizar con el gestor de paquetes, npm, para la instalación de este framework. Para la instalación de VueJS se introduce en consola el siguiente comando:

```
npm install -g @vue/cli
```

Una vez instalado escribiendo en la terminal el siguiente comando se comprobará que la instalación ha sido correcta:

```
Vue -V
```

Para abrir la interfaz gráfica de Vue, Vue CLI, es necesario ejecutar el comando:

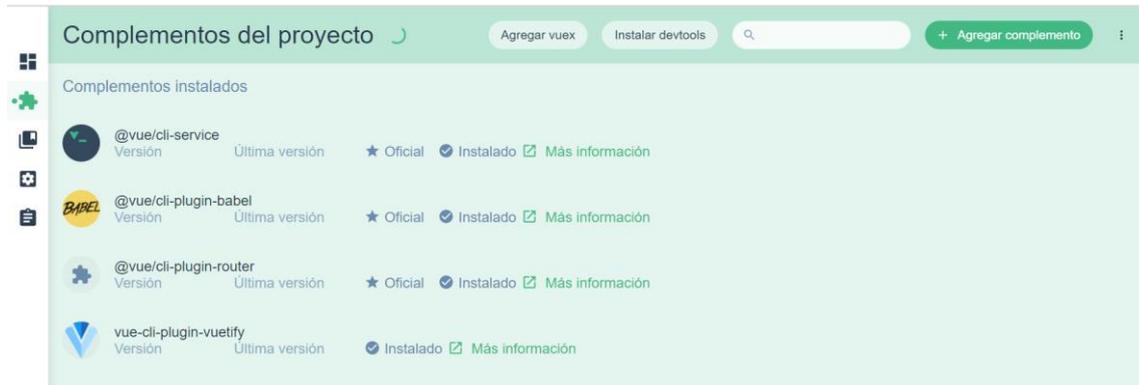
```
Vue ui
```

Este comando abre una ventana donde se va a poder crear el proyecto o configurarlo al gusto del desarrollador de la aplicación.

Una vez creado el proyecto se creará una carpeta en el sistema que contendrá la estructura básica de una aplicación de vue.

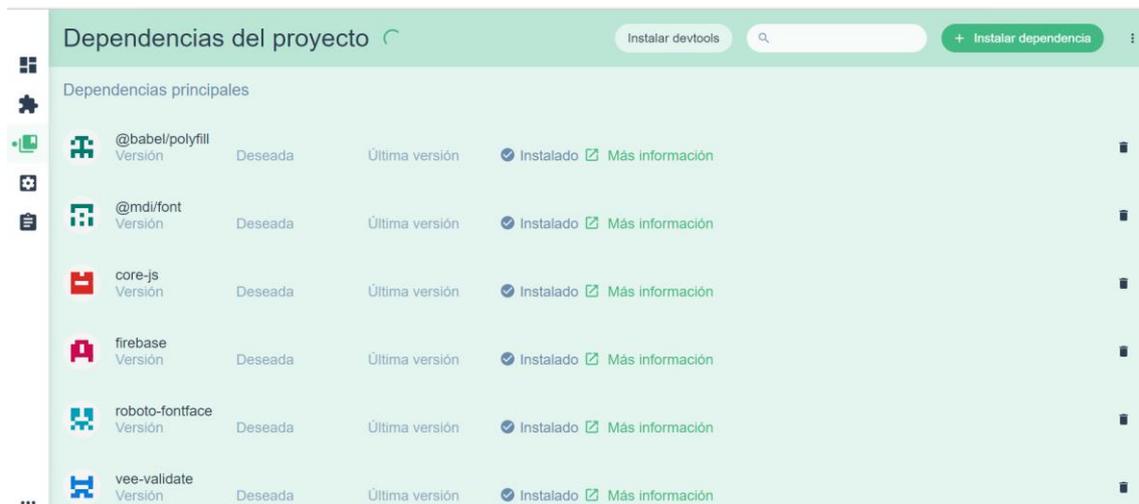
## Memoria del proyecto

En la **Figura 24** se puede observar los complementos instalados en el proyecto o como agregar nuevos complementos:



**Figura 24: Complementos del proyecto**

También se van a mostrar en la **Figura 25** algunas de las dependencias instaladas, en este apartado también se ofrece la posibilidad de desinstalar estas o instalar nuevas:



**Figura 25: Dependencias del proyecto**

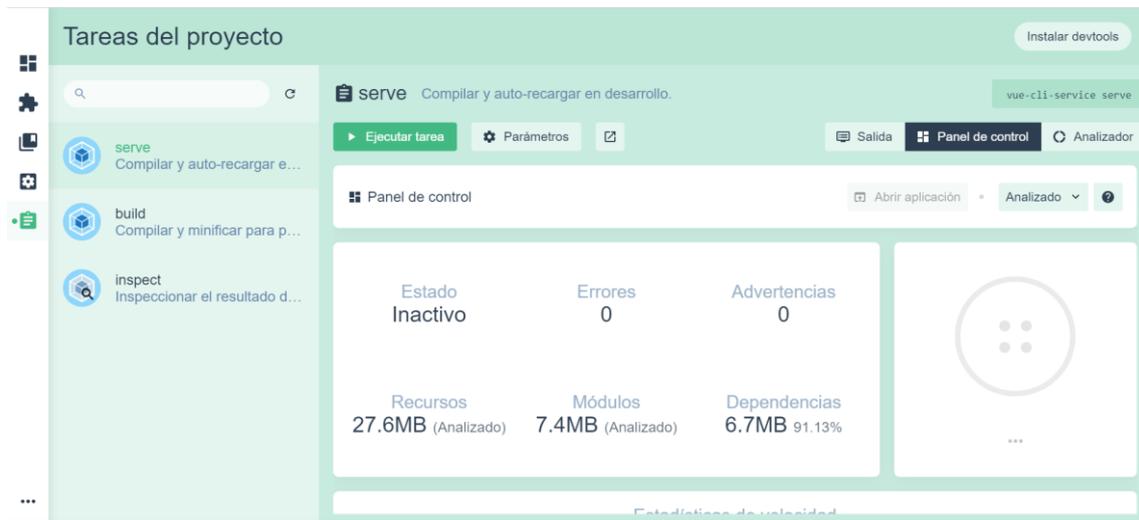
## Memoria del proyecto

Otro apartado que ofrece esta interfaz es la posibilidad de configurar el proyecto como el desarrollador deseé, esto se puede observar en la **Figura 26**:



**Figura 26: Configuración del proyecto**

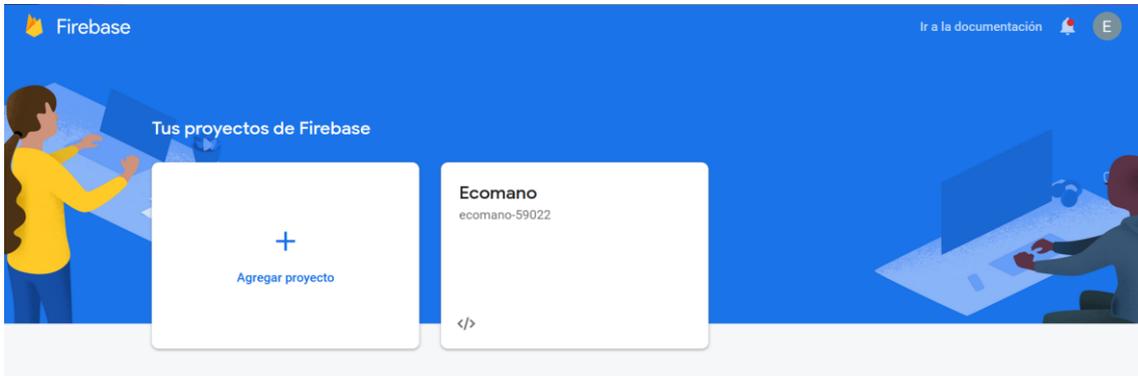
Y por último ofrece una serie de tareas, donde por ejemplo se puede ejecutar una instancia local del servidor para poder ejecutar la aplicación, se puede observar esto en la **Figura 27**.



**Figura 27: Tareas del proyecto**

### 6.3 Configuración de firebase

Lo primero es acceder a firebase console [22] donde es necesario acceder con la cuenta de Google, una vez aquí se empieza a crear un proyecto, se introduce el nombre del proyecto que se deseé, en el caso de este proyecto el nombre es Ecomano, se puede ver en la **Figura 28**:

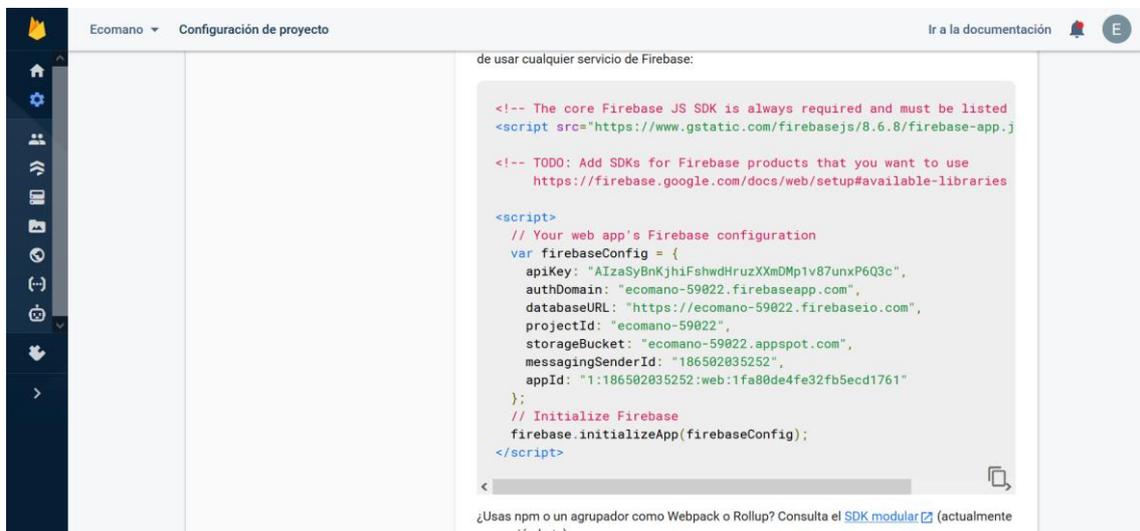


**Figura 28: Configuración firebase**

Una vez creado el proyecto es necesario indicar el tipo de aplicación a utilizar, en este caso aplicación web.

Dentro de firebase es necesario que en el apartado de Authentication se habilite la opción de email que es el método de autenticación que se va a utilizar.

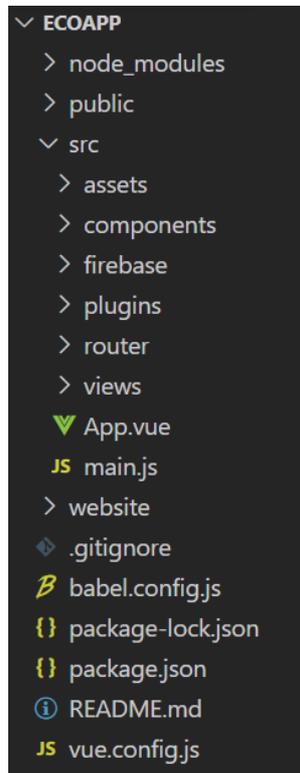
Para poder utilizar firebase en la aplicación web que se está creando, es necesario importar la siguiente configuración, que se muestra en la **Figura 29**, en el proyecto vue:



**Figura 29: Configuración del proyecto con firebase**

## 6.4 Creación de la aplicación

Una vez realizado los anteriores apartados, se puede comenzar instalando las dependencias y complementos necesarios para la aplicación, después de esto se importaría en el proyecto creado cada una de las dependencias. A partir de aquí el desarrollador iría programando a su gusto cada una de las vistas, la lógica de datos que quiera establecer, así como su control. La estructura del proyecto se puede ver en la aplicación Visual Studio Code que ha sido el editor utilizado. Esta distribución se puede apreciar en la ilustración de la **Figura 30**.



**Figura 30: Distribución del proyecto**

Cada uno de los apartados de la figura anterior se ha ido viendo y detallando con detenimiento en el **anexo V**, desde `node_modules` que es un directorio que se crea automáticamente cuando se instalan las dependencias con npm, guardando el código relacionado con estas para después importar este código en el proyecto cuando sea necesario, hasta la carpeta `src` que es donde se guarda el código fuente de la aplicación.

Por último, en este apartado cabe señalar que se ha creado un repositorio en GitHub donde se documentan los elementos principales de las vistas y componentes que hay en el proyecto.

Para la generación de este documento se ha hecho uso de Vuese, esta herramienta genera documentación automáticamente en un HTML estilizado. En el código fuente desarrollado (los ficheros `.vue`) mediante comentarios, la etiqueta `@vuese` y `@args` se describen los elementos más importantes del código.

Este repositorio se encuentra en el siguiente enlace, es necesario disponer de un navegador Firefox o por el contrario una vez accedido al enlace es necesario recargar la página.

<https://diegocollantes.github.io/documentacionCodigo/#/>

Como ejemplo de esta página se muestra la **Figura 31**:

Components

---

▼ BASIC

- Administrador
- AlertaCorrecta
- AlertaCorrecta
- App
- AyudaPlastico
- Buscar
- Cesta**
  - Methods
  - Computed
  - Data
- Chat
- Home
- MisProductos
- Notificaciones
- Perfil
- Plastico
- Privado
- Registro

## Cesta

### Methods

Method	Description	Parameters
eliminarCesta	Se elimina de la cesta el producto	se pasa como argumento el precio del producto a eliminar y el id del producto
leerDatosTotal	Se cogen los productos que pertenecen a la cesta elegida por el usuario	-
leerDatos	Se actualiza el precio total de la cesta	-
pagar	Este metodo actua cuando el usuario desea	-

**Figura 31: Documentación generada con Vuese**

## 7 Funcionalidades de la aplicación

En este apartado se pretende explicar el funcionamiento de la aplicación EcoMano, para ello se hará uso de ilustraciones sacadas de la ejecución de la aplicación. Cabe destacar que en el **anexo VI** se realiza una descripción minuciosa de la funcionalidad, no obstante, en este apartado se va a ir explicando poco a poco la funcionalidad de cada interfaz encontrada en la aplicación.

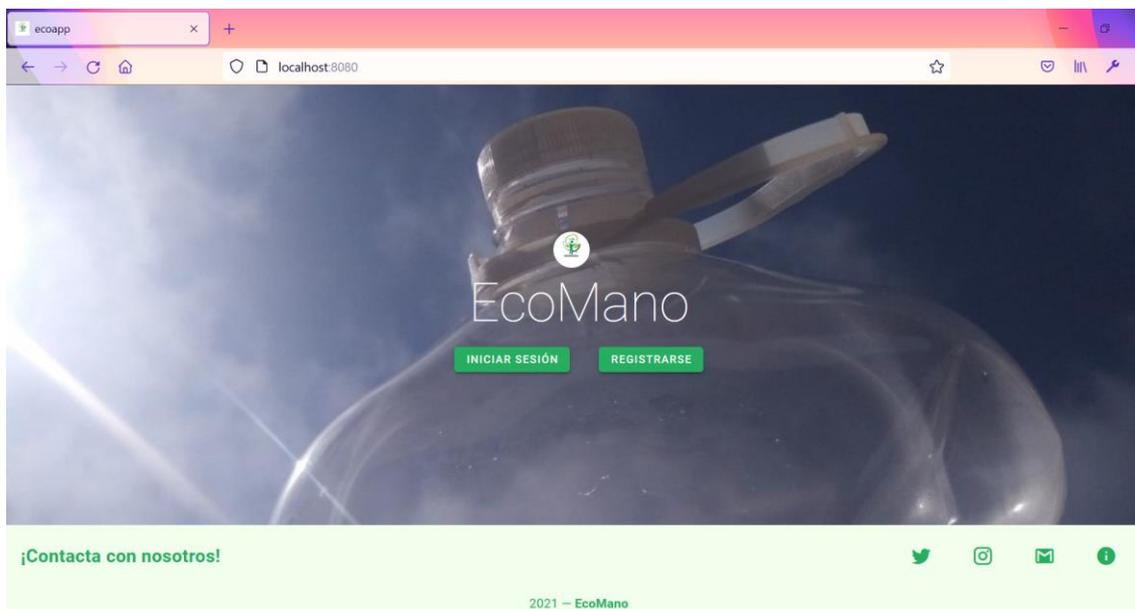
Para la explicación de este apartado se va a realizar una división en tres partes: Interfaz de usuario no identificado, interfaz usuario identificado, e interfaz administrador/empleada.

### 7.1 Interfaz Usuario no identificado

Aquí se encuentran las partes de iniciar sesión y registro en la aplicación, en los siguientes puntos se va a explicar cada una de estas.

#### 7.1.1 Interfaz principal

En esta página se va a mostrar la interfaz inicial que se va a encontrar una vez accede a la aplicación el usuario, dependiendo de si le das al botón “Iniciar sesión” o “Registrarse”, se abrirá una ventana modal para iniciar sesión en la aplicación o el usuario se redirigirá al apartado de registro respectivamente. En la **Figura 32** se puede ver una ilustración de esta interfaz.



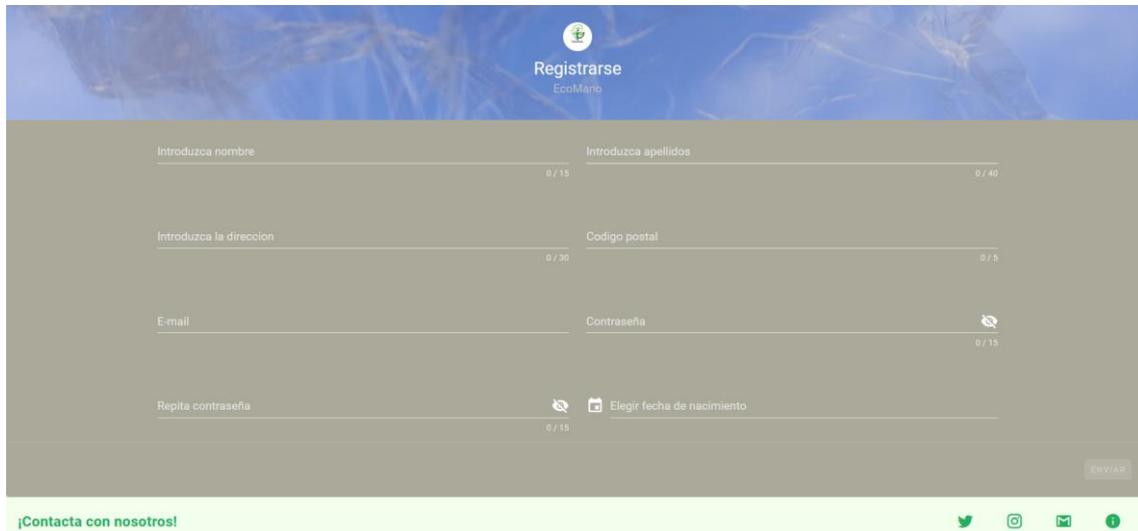
**Figura 32: Interfaz principal**

Si el usuario accede a iniciar sesión se presentará una ventana, en esta se pueden introducir las credenciales email y contraseña, necesarias para el acceso a la zona privada de la aplicación. Si los datos introducidos son incorrectos el sistema lo señalará.

Por último, en esta ventana si se pulsa sobre “HE OLVIDADO LA CONTRASEÑA”, se abrirá otra ventana modal la cual nos ofrece la oportunidad de solicitar una nueva contraseña introduciendo el Email

### 7.1.2 Interfaz Registro

En este apartado el usuario se podrá registrar para quedar autenticado en la aplicación y así poder acceder a las funcionalidades de la zona privada de la aplicación. En la **Figura 33** se puede observar un ejemplo de esta interfaz.



**Figura 33: Interfaz registro**

Como se puede observar en este apartado hay una serie de campos, los cuales son necesarios que se rellenen conforme a las restricciones que conllevan cada uno de ellos, si es así se habilitará el botón de enviar y pulsando sobre este el usuario quedaría registrado en la aplicación.

Por último, en este apartado hay que destacar la elección de fecha de nacimiento “Elegir fecha de nacimiento”, dicho campo abrirá una ventana donde el usuario podrá escoger una fecha de nacimiento limitada hasta 2003 debido a que no se pueden registrar personas de menos de 18 años.

### 7.2 Interfaz de usuario identificado

Las primeras figuras que se deben describir son las del navbar, y barra de ayuda y apartados personales. En la **Figura 34** se puede observar el navbar.

En el navbar se encuentran siete apartados,  abrirá la barra de ayuda y apartados, “HOME” a la interfaz de home, “BUSCAR PRODUCTOS” a la interfaz de buscar productos, “PROPORCIONAR PLÁSTICO” a la interfaz proporcionar plástico, “CHAT” a la interfaz de chat, “NOTIFICACIONES” a la interfaz de notificaciones y, por último,  a la interfaz de cesta.



**Figura 34: Navbar**

## Memoria del proyecto

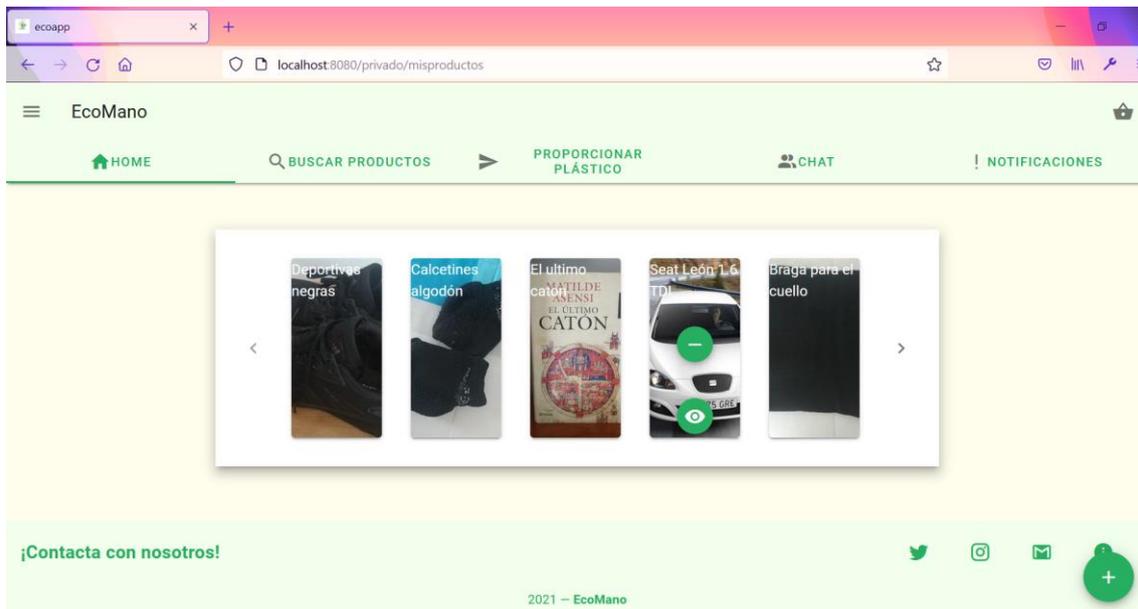
En la **Figura 35** se puede observar la barra de ayuda y apartados personales quedaría de la siguiente forma, donde al principio muestra información del usuario identificado luego se encuentran una serie de apartados que nos redirigirán a otras pantallas, “Mis productos” a la interfaz de mis productos, “Mi perfil” a la interfaz de mi perfil, “Ayuda plástico” a la interfaz de ayuda plástico y por último “Cerrar sesión” que cerrara sesión de la zona privada y redirigirá a la Interfaz principal.



**Figura 35: Toolbar**

### 7.2.1 Interfaz mis productos

Aquí se muestran los productos que el usuario ha ido subiendo para poner en venta, aquí se pueden ver, eliminar o subir nuevos productos. Se muestra una lista de productos donde como vista previa se pueden observar las fotos y nombres de los diferentes productos, si se pulsa sobre alguno de ellos se mostrarán dos nuevos botones, el de icono “-” sirve para eliminar, mientras que el de icono de un ojo sirve para ver el producto. Y por último abajo a la derecha se muestra el botón con icono “+” que sirve para añadir un nuevo producto. Esta interfaz se puede observar en la **Figura 36**.



**Figura 36: Interfaz mis productos**

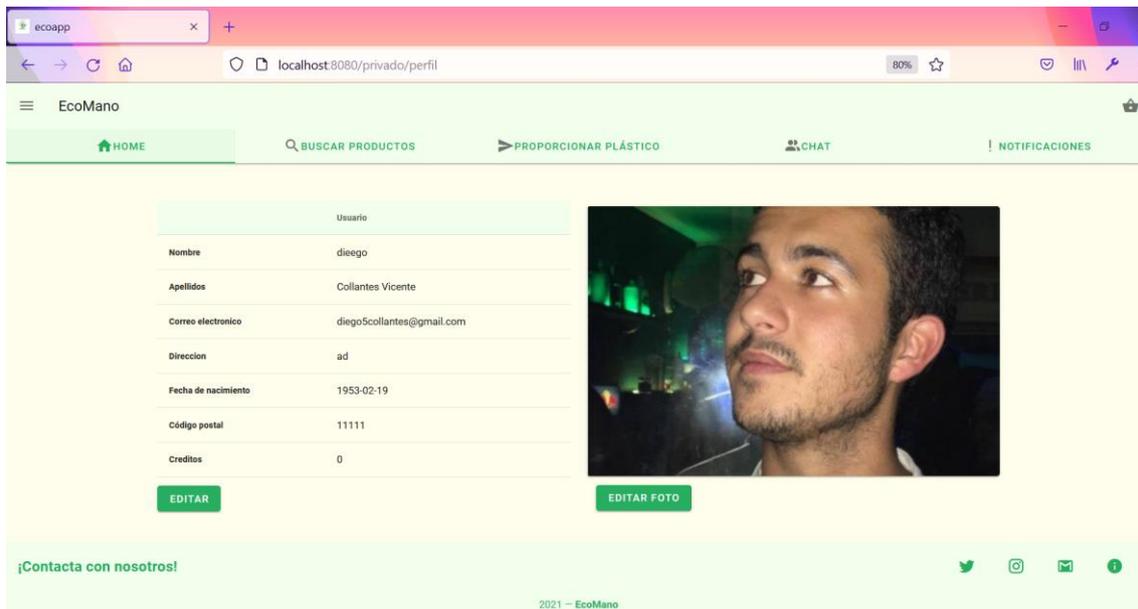
Pulsando el botón menos se mostrará una nueva lista sin el producto sobre el que se ha pulsado ese botón.

Si se pulsa el botón con el icono de un ojo, se mostrará más información sobre el producto subido.

Por último, para añadir un nuevo producto una vez pulsado el botón con icono “+”, se abrirá una nueva ventana que permitirá al usuario rellenar una serie de campos para añadir el producto que dese el usuario.

### 7.2.2 Interfaz mi perfil

En este apartado se mostrarán los datos pertenecientes al usuario identificado, estos datos son, nombre, apellidos, correo electrónico, dirección, fecha de nacimiento, código postal o créditos conseguidos, estos se obtienen al proporcionar plástico cuya función se verá más adelante en el apartado de proporcionar plástico. En esta pantalla también se muestra la foto elegida por el usuario para su perfil, por último, hay dos botones “EDITAR” y “EDITAR FOTO”, para editar o bien los campos nombre, dirección y código postal o la foto respectivamente. Esta interfaz se puede observar en la **Figura 37**.



**Figura 37: Interfaz mi perfil**

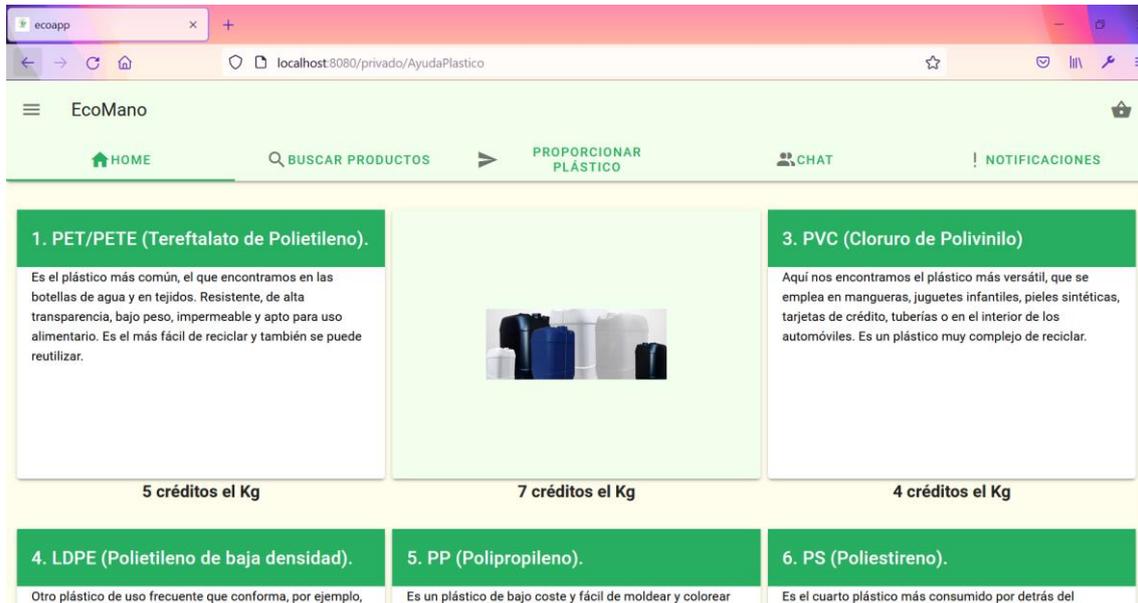
Si se elige el botón de “EDITAR”, se abrirá una nueva pantalla, aquí se encuentran los datos anteriores, donde se pueden editar una serie de datos o bien se puede solicitar el cambio de la contraseña pulsando sobre “CAMBIAR CONTRASEÑA” enviando así un correo al usuario.

Por último, en este apartado de “EDITAR FOTO” se puede editar la foto de perfil, se abrirá una ventana en la que podemos seleccionar una foto del explorador de archivos del usuario.

### 7.2.3 Interfaz ayuda plástico

En este apartado podemos acceder a información relacionada con el tipo de plástico, dicha información es de gran ayuda para que el usuario acceda a ella cuando dude del tipo de plástico que desea proporcionar.

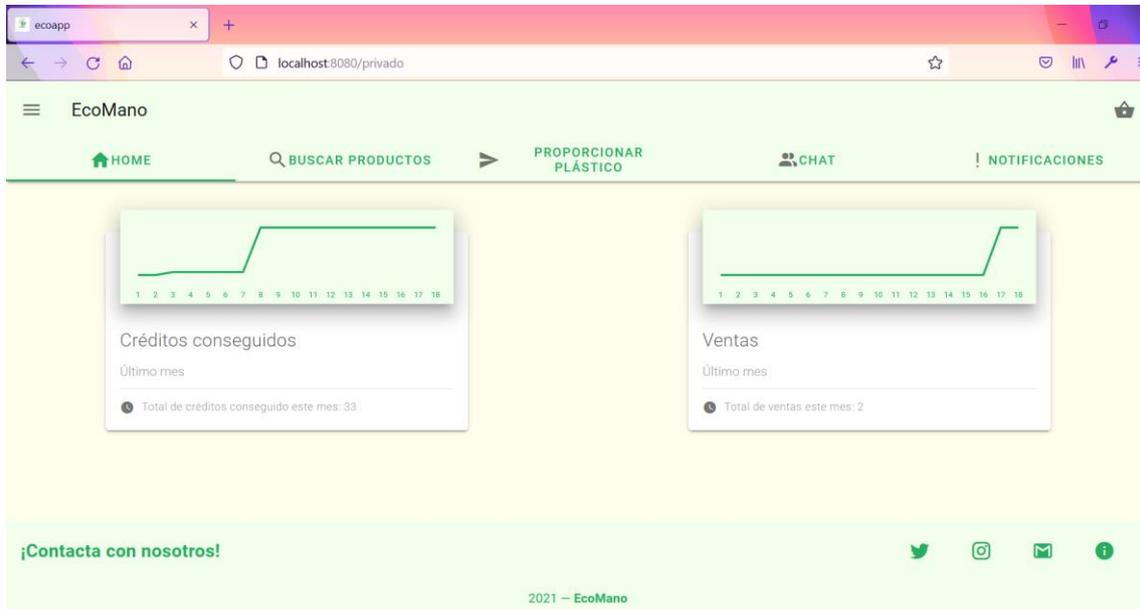
Se van a mostrar “tarjetas”, cada una de estas va a explicar un tipo de plástico, si se pasa el cursor por encima de alguna de estas tarjetas se mostrará una imagen de ejemplo del tipo de plástico. Esta interfaz se puede observar en la **Figura 38**.



**Figura 38: Interfaz ayuda plástico**

### 7.2.4 Interfaz home

En este apartado se muestran las estadísticas relacionadas con los créditos conseguidos el mes en curso y también el total de créditos conseguido, y también muestra las estadísticas relacionadas con las ventas conseguidas el mes en curso, así como las ventas totales. Este apartado se puede observar en la **Figura 39**.

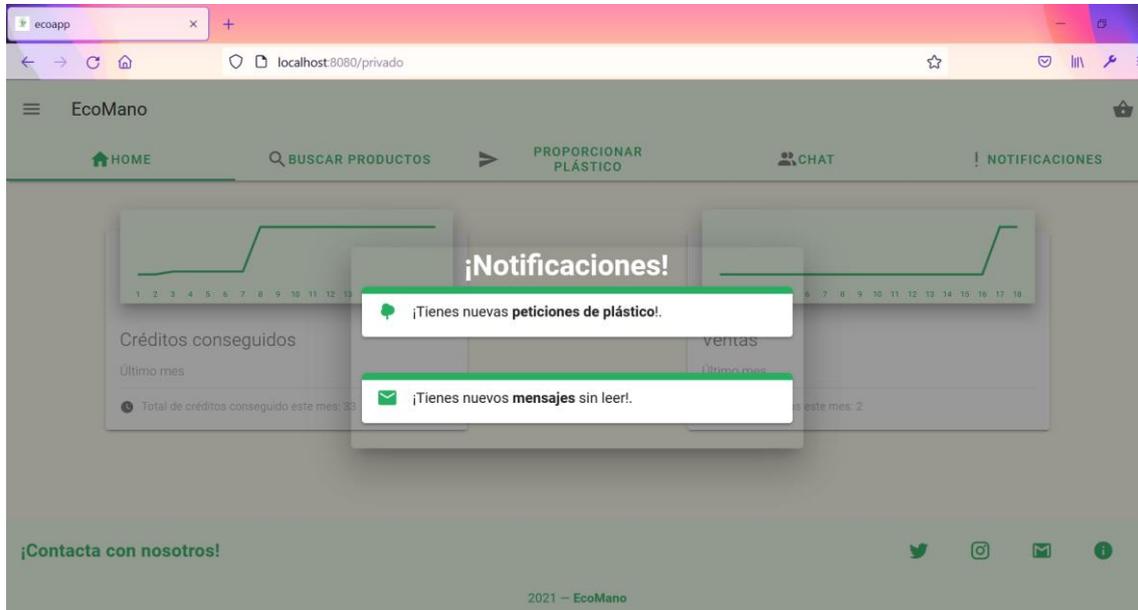


**Figura 39: Interfaz home**

## Memoria del proyecto

Si el usuario tiene notificaciones nuevas, bien sean notificaciones de mensajes no leídos por otros usuarios o bien las peticiones que ha realizado de plástico las cuales han sido rechazadas o aceptadas por parte del empleado, se mostrará lo siguiente, si se pulsa sobre una de las dos notificaciones el usuario se redirigirá al apartado de notificaciones. Se puede observar en la

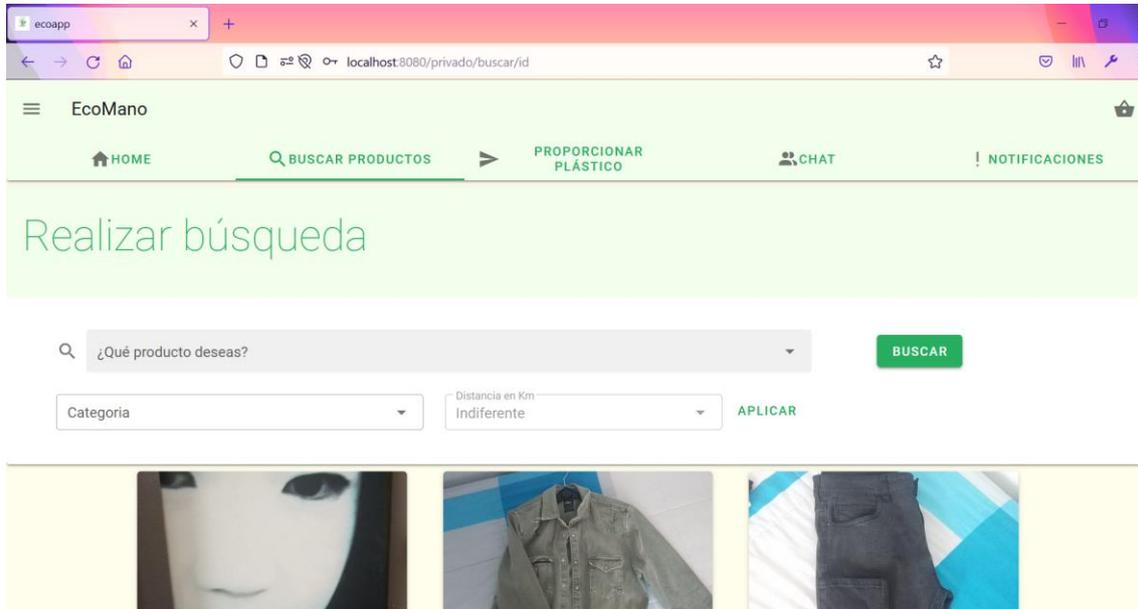
**Figura 40:**



**Figura 40: Interfaz home con notificaciones**

### 7.2.5 Interfaz búsqueda

En este apartado el usuario interpreta el papel de comprador, podrá realizar una búsqueda introduciendo letras y los productos cuyos nombres sean coincidentes con estas saldrán para ser elegidos, si posteriormente se pulsa el botón de “BUSCAR” saldrá el producto buscado, por último, hay unos tags que son categoría y distancia para realizar una búsqueda dependiendo de estos dos campos, al pulsar aplicar saldrán los productos correspondientes con estos. Esta interfaz se puede observar en la **Figura 41**.



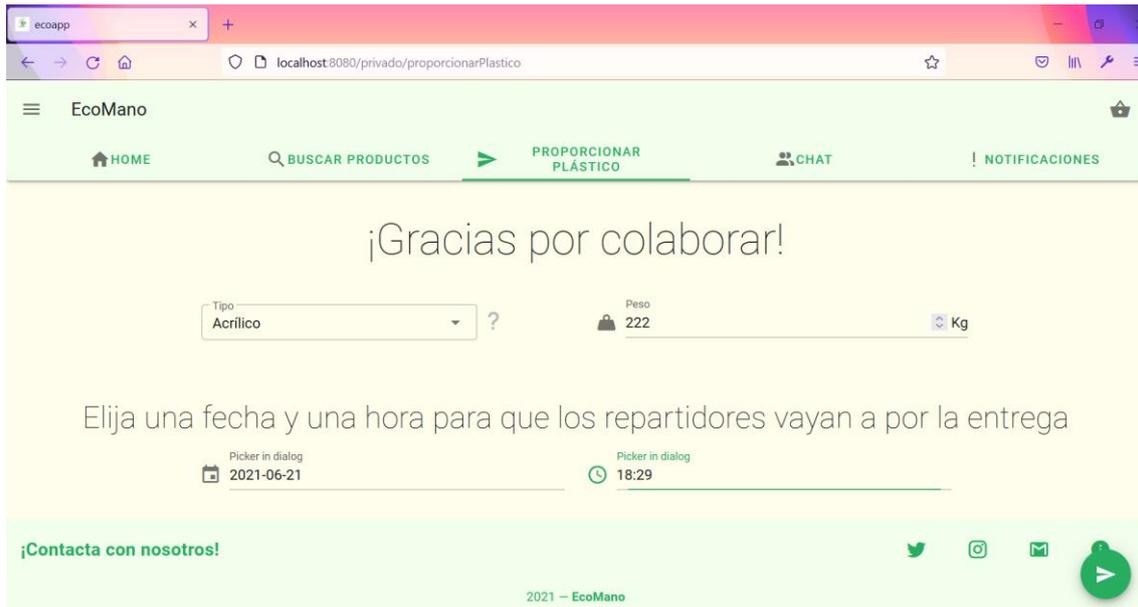
**Figura 41: Interfaz búsqueda**

Una vez realizada la búsqueda se muestran “tarjetas” correspondientes con cada producto, estas contienen información como el precio nombre y si pasas el cursor por encima se verá una descripción del producto. Por último, se explican los botones que hay en cada una de las tarjetas que se corresponden con cada producto, el botón “VER FOTO” muestra una foto a tamaño completo del producto, el botón “CHAT” redirige al usuario al apartado de chat con el propietario del producto y el botón “CESTA” añadirá el producto a la cesta si este aún no lo está.

### 7.2.6 Interfaz proporcionar plástico

Este apartado desempeña una funcionalidad muy importante en la aplicación, aquí se puede elegir el tipo de plástico que se va a proporcionar, si el usuario tiene alguna duda con este, se recomienda pulsar el botón “?” que redirigirá al apartado de ayuda plástico. También se puede introducir la cantidad de plástico que se va a proporcionar, así como la opción de elegir fecha y hora disponible para que el empleado vaya a buscar y validar el plástico que se va a

proporcionar. Si los datos introducidos son correctos se habilitará el botón . En La **Figura 42** se puede observar esta interfaz.

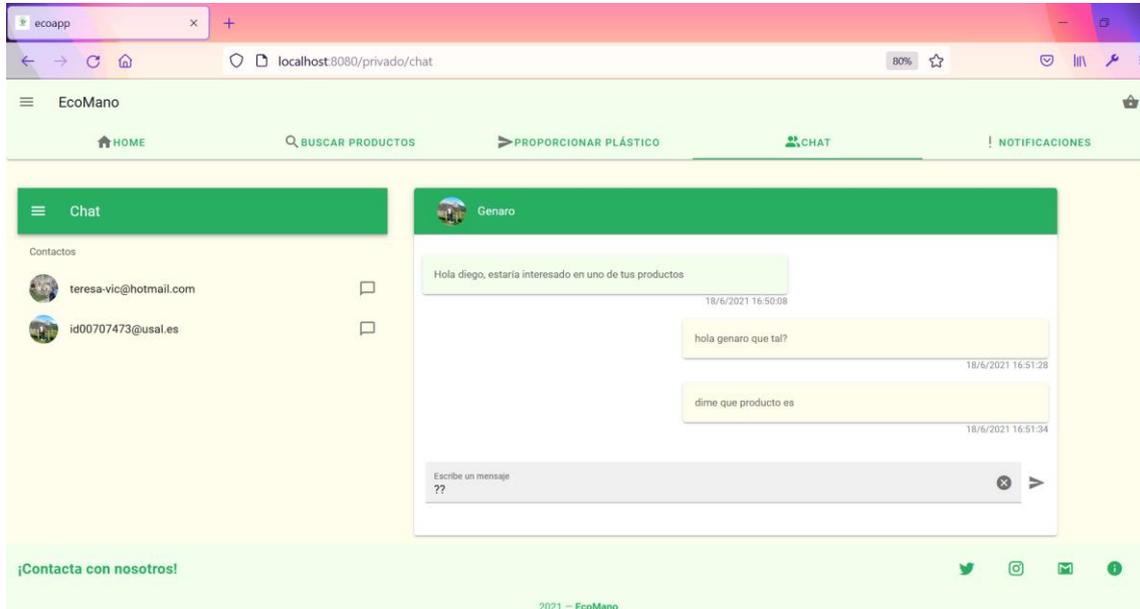


**Figura 42: Interfaz proporcionar plástico**

Las fechas disponibles van desde el día de hoy en adelante, de lunes a jueves y las horas van de 8 de la mañana a 8 de la tarde.

### 7.2.7 Interfaz chat

En este apartado los usuarios se pueden comunicar con otros a través de un chat, a la izquierda se mostrará la lista de usuarios disponibles para chatear, y a la derecha se muestra la ventana de la conversación cuyo contenido son los mensajes intercambiados entre ambos usuarios. Esta interfaz se puede ver en la **Figura 43**.

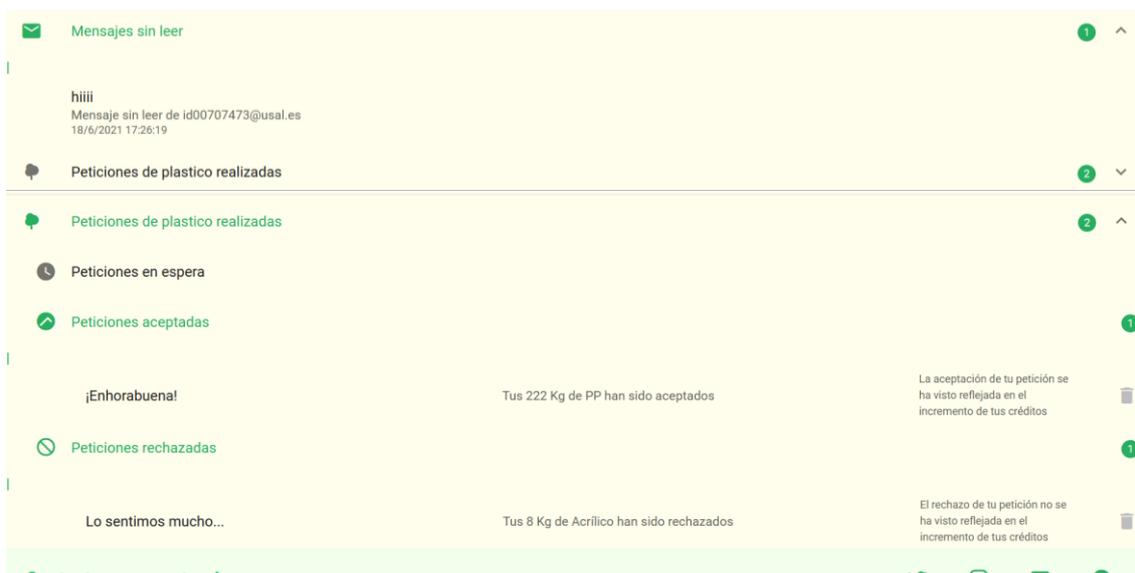


**Figura 43: Interfaz chat**

### 7.2.8 Interfaz notificaciones

En este apartado se mostrará la opción al usuario de visualizar la lista de los mensajes que tiene sin leer y las peticiones de plástico que ha realizado y el estado en el que se encuentran estas, puede mostrar una lista de peticiones de plástico aún en espera, lista de peticiones rechazadas por parte del empleado o por el contrario una lista de peticiones aceptadas por el empleado.

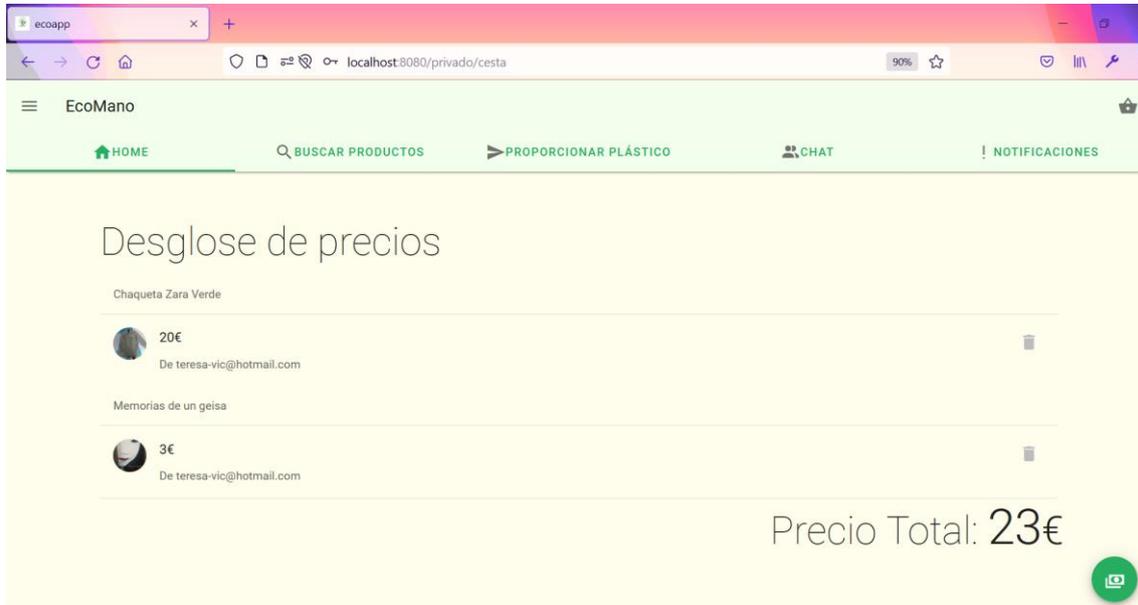
El usuario puede realizar una limpieza de estas notificaciones, si lo que desea es eliminar notificaciones de mensajes no leídos, lo que debe hacer es entrar en la conversación donde se encuentra dicho mensaje, si lo que desea es eliminar una petición de plástico ya cursada por el empleado, se muestra un icono de una papelera al lado de cada petición para poder eliminarla de esta lista. Esta interfaz se muestra en la **Figura 44**.



**Figura 44: Interfaz notificaciones**

### 7.2.9 Interfaz cesta

En este apartado se muestra una lista de productos que el usuario ha ido añadiendo previamente a la cesta con el motivo de efectuar la compra de estos productos, se ofrece la oportunidad de quitar de la cesta el producto que no se quiera dándole al icono de la papelera que sale a la derecha de cada producto en la lista. La información que se muestra de cada producto es el precio de cada uno de estos, la foto, de quien es el producto y el nombre. Esta interfaz se puede ver en la **Figura 45**.



**Figura 45: Interfaz cesta**

También se muestra el precio total a pagar. Si se pulsa sobre el botón , se abrirá una ventana para efectuar el pago.

En esta ventana se pueden añadir los créditos si es que tiene, que el usuario considere aplicar para que el precio de la compra sea más barato.

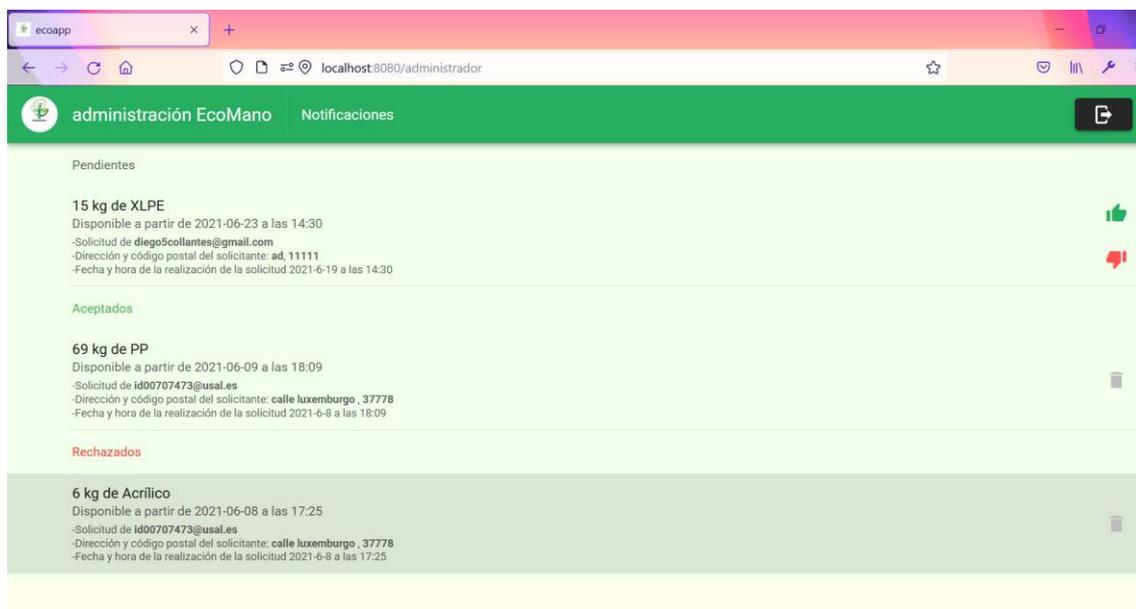
### 7.3 Interfaz administrador/empleador

A este apartado accederán los empleados de EcoMano, tiene como cometido aceptar o rechazar las diferentes peticiones de los usuarios. La credencial con la que los empleados accederán a este apartado es [admin@ecomano.com](mailto:admin@ecomano.com) tal y como lo hace un usuario normal.

Este apartado está dividido en tres partes:

- Pendientes: Peticiones que tiene pendiente el empleado por aceptar o rechazar.
- Aceptados: Peticiones aceptadas, al ser aceptadas el usuario obtendrá los créditos correspondientes a su aportación. Estas pueden ser eliminadas de la lista pulsando la papelera que sale a la derecha de cada petición.
- Rechazados: Peticiones rechazadas. Estas pueden ser eliminadas de la lista pulsando la papelera que sale a la derecha de cada petición.

La información que contienen cada una de estas peticiones son, cantidad y tipo de plástico, la disponibilidad del usuario para que el empleado vaya a ejecutar la validación, correo, dirección y código postal del solicitante, y la fecha y hora a la que han realizado dicha solicitud. Esta interfaz se muestra en la **Figura 46**.



**Figura 46: Interfaz empleado/administrador**

## 8. Conclusiones

Una vez finalizado el desarrollo del proyecto, se pueden concluir una serie de objetivos los cuales se han ido consiguiendo durante el transcurso del desarrollo. El objetivo principal era la creación de una aplicación web utilizando las tecnologías marcadas, esta aplicación debía ofrecer diversas funcionalidades dependiendo el papel que desempeñase el usuario.

Dentro de los objetivos del sistema que se han ido consiguiendo:

- El usuario puede autenticarse en la aplicación y acceder a la zona privada donde puede desempeñar las funcionalidades que esta ofrece. Previamente es necesario que este haya sido registrado correctamente.
- Se ha conseguido la funcionalidad principal que se deseaba alcanzar, esta es la compra y venta de productos de segunda mano ya que se trata de una aplicación donde los usuarios pueden poner en venta productos o la compra de estos.
  - Por un lado, se le da al usuario la oportunidad de poder ejercer la funcionalidad de vendedor, donde podrá subir sus propios productos, así como verlos o eliminarlos.
  - Por otro lado, el usuario puede desempeñar el papel de comprador, para ello se ha conseguido alcanzar la funcionalidad de búsqueda de productos deseados por el usuario.
- La funcionalidad que difiere con respecto a otras aplicaciones de segunda mano y que se ha logrado alcanzar, era la posibilidad de que el usuario pudiese proporcionar plástico a cambio de créditos, estos últimos se utilizarían en compras abaratando su coste. Esta funcionalidad se puede ver desde dos caras diferentes:
  - Una es la del usuario identificado cuyo objetivo es la obtención de créditos proporcionando plástico.
  - La otra es la del empleado el cual validaría ese plástico que ha solicitado proporcionar el usuario.
  - Otra funcionalidad alcanzada es ofrecer la posibilidad de que los usuarios dados de alta en la aplicación tengan la opción de conversar entre ellos, esta funcionalidad se ha realizado con el fin de que los usuarios puedan negociar precios o detallar más características del producto.
- Se ha conseguido la función de notificar al usuario cuando este haya alcanzado algún objetivo como por ejemplo cuando ha efectuado una compra, ha subido un producto nuevo, haya conseguido que le acepten plástico, tenga mensajes nuevos sin leer, añada productos a la cesta o no le hayan rechazado plástico.
- Se ha conseguido utilizar los servicios proporcionados por firebase, ha permitido la gestión de datos en tiempo real y una gestión de la autenticación correcta.

En cuanto a los objetivos personales alcanzados:

- Se ha conseguido crear una aplicación web por primera vez, lo cual suponía un reto con cierto grado de dificultad.
- Se ha alcanzado el aprendizaje de nuevas tecnologías, este está relacionado con el entorno utilizado por parte del desarrollador como por ejemplo aprender a utilizar firebase lo que conlleva utilizar sus servicios, a utilizar el framework VueJS o framework de diseño como puede ser vuetify.
- Se han obtenido nuevos conocimientos o nuevas percepciones de algunos conceptos que nunca se habían puesto en práctica.

## Memoria del proyecto

Por último, cabe destacar en este apartado la obtención de un proyecto completo, esto no solo implica a la aplicación como tal, sino que también implica a toda la documentación generada como son los anexos y memoria del proyecto.

## 9. Líneas de trabajo futuras

Después de haber concluido con el proyecto surgen diferentes ideas con el objetivo de poder hacer una aplicación más completa, con más funcionalidades.

Una idea sería hacer más hincapié sobre la vista de la aplicación desde el navegador de un smartphone, aunque haya componentes de la interfaz que se modifiquen dependiendo del tipo de pantalla del dispositivo utilizado, no hay una optimización para la vista de algunas pantallas de la aplicación en estos dispositivos móviles.

También se podría ofrecer la oportunidad no solo de que el usuario proporcione plástico, sino que también se podrían utilizar otros tipos de materiales reciclados como papel, mascarillas que en la actualidad están suponiendo un gran incremento en su producción, pilas, etc.

Como último aspecto que sería interesante alcanzar, sería crear en la aplicación un apartado de sorteos donde el usuario podría entrar en estos dependiendo de la cantidad de plástico que haya conseguido proporcionar.

## 10. Referencias

- [1] “¿Por qué es importante reciclar el plástico”
  - <https://www.leonardo-gr.com/es/blog/por-qu-es-importante-reciclar-el-pl-stico>
- [2] “España, el país de la UE que más plástico vierte al océano”
  - <https://www.elagoradiario.com/desarrollo-sostenible/economia-circular/espana-plasticos-oceanos-contaminacion/>
- [3] “80% de la basura de los océanos es plástico”
  - [https://www.elconfidencial.com/tecnologia/ciencia/2019-08-20/espana-plastico-mediterraneo-medioambiente\\_2184715/](https://www.elconfidencial.com/tecnologia/ciencia/2019-08-20/espana-plastico-mediterraneo-medioambiente_2184715/)
- [4] “70 años de ‘boom’ del plástico”
  - <https://es.statista.com/grafico/20441/produccion-de-plastico-a-nivel-mundial/>
- [5] “Qué es Material Design”
  - <http://vectoraula.com/material-design/>
- [6] “Diferencias entre front-end y back-end”
  - <https://www.ironhack.com/es/desarrollo-web/diferencias-entre-front-end-y-back-end>
- [7] “Code editing. Redefined”
  - <https://code.visualstudio.com/>
- [8] “Navegador Firefox. Developer edition”
  - <https://www.mozilla.org/es-ES/firefox/developer/>
- [9] “¿Qué es Vue?”
  - <https://lenguajejs.com/vuejs/introduccion/que-es-vue/>
- [10] “Vuetify”
  - <https://vuetifyjs.com/en/>
- [11] “Firebase helps you build and run succesful apps”
  - <https://firebase.google.com/?hl=es>
- [12] “Microsoft Project”
  - <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- [13] “Visual Paradigm”
  - <https://www.visual-paradigm.com/>
- [14] “NodeJS”
  - <https://nodejs.org/es/>
- [15] “Constuyendo módulos de utilidad para nuestras aplicaciones con npm”
  - <https://medium.com/startup-reef/como-construir-aplicaciones-modulares-con-m%C3%B3dulos-npm-locales-1dcba908199>
- [16] “VueSe”
  - <https://vuese.org/cli/#quick-start>
- [17] “GitHub”
  - <https://github.com/>
- [18] “Calculate distance, bearing and more between Latitude/Longitude points”
  - <https://www.movable-type.co.uk/scripts/latlong.html>
- [19] “Creador de logos”
  - <https://play.google.com/store/apps/details?id=org.contentarcade.apps.logomaker&hl=es&gl=US>

## Memoria del proyecto

- [20] “Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP)”
  - <https://web.archive.org/web/20140323082738/http://www.utvm.edu.mx/Organoinformativo/orgJul07/RUP.htm>
- [21] “Vue.js guide”
  - <https://vuejs.org/v2/guide/>
- [22] “Consola de firebase”
  - <https://console.firebase.google.com/?hl=es-419&pli=1>