



Universidad de Valladolid

E.T.S. Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

DressApp: Aplicación Web para la asistencia de inventario y personal shopping

Autor:
Pablo Manso González
Tutor:
Cesar Vaca Rodríguez

Agradecimientos

Quiero aprovechar estas líneas para agradecer a todas las personas que han hecho posible que este momento fuera posible y en especial a las cuatro personas que han estado ahí sin cesar, empujando y dándome su apoyo incondicional.

La primera representa los frenos de mi vida, esos que normalmente no necesitas hasta que tienes que detenerte en seco o bajar la velocidad para hacer un giro y cambiar la dirección de esta.

La segunda supone el corazón de mi vida, la que en todo momento continuó bombeando la sangre necesaria para que no me detuviera y llegara a alcanzar mis metas. Ella nunca me dejó, y por muchas alegrías o disgustos que reciba, estará a mi lado hasta su último día.

La tercera ha sido el guía de mi vida, la que me siempre me ha mostrado el camino a través de sus pasos, la que ha allanado el terreno para que todo fuera más fácil y seguro.

La cuarta es la que menos tiempo lleva a mi lado, pero quizás la que más tiempo se quede. Su belleza hizo que nos uniéramos, pero su bondad y dedicación han hecho que esa unión se haga cada día más fuerte.

Por último, quisiera agradecer a mi tutor César Vaca Rodríguez y a la Universidad de Valladolid por todos los conocimientos que me han ofrecido y en consecuencia, todas las puertas que me han abierto.

Resumen

Este trabajo de Fin de Grado describe el desarrollo de una aplicación web de asistencia de inventario y personal shopping utilizando el conjunto de subsistemas MEAN y siguiendo la metodología SCRUM. El proyecto ha sido finalizado mediante iteraciones retroalimentadas gracias a las cuales, se han podido corregir pequeños errores de diseño o escenarios no contemplados al comienzo del mismo.

Índice de contenido

Índice de Figuras	10
Índice de Tablas	11
1. Introducción	2
1.1. Contexto	2
1.2. Motivación Personal	4
1.3. Objetivos	5
1.4. Estructura de la memoria	5
2. Entorno Tecnológico	8
2.1. Entorno de desarrollo	8
2.2. Tecnologías utilizadas	8
2.2.1. JavaScript	8
2.2.2. Typescript.....	9
2.2.3. HTML	9
2.2.4. CSS.....	10
2.2.5. Angular	10
2.2.6. Material Design	10
2.2.7. NodeJs.....	11
2.2.8. MongoDB	11
2.3. Librerías externas.....	11
3. Gestión del Proyecto.....	13
3.1. Metodología Ágil	13
a) Versatilidad, orientada a la consecución de objetivos	14
b) Reestructuración de acciones.....	14
c) Transparencia	14
3.2. Planificación inicial	15
3.3. Identificación de riesgos.....	16
3.4. Gestión de riesgos.....	18
3.5. Seguimiento de riesgos	22
3.6. Costes	23
3.6.1. Presupuesto inicial	23
3.6.2. Reajuste de presupuesto.....	23
4. Requisitos.....	26
4.1. Requisitos Funcionales.....	26
4.2. Requisitos No Funcionales	27
4.3. Requisitos de Información	28
5. Análisis y diseño	31
5.1. Arquitectura de la aplicación.....	31
5.2. Flujo de datos	31

5.3.	Diagrama de Casos de Uso	33
5.4.	Casos de Uso	34
6.	Implementación	42
6.1.	Sprint 0	42
6.1.1.	Iteration Planning.....	42
6.1.2.	Desarrollo	42
6.1.3.	Retrospective	43
6.2	Sprint 1	43
6.2.1	Iteration Planning.....	43
6.2.2	Desarrollo	43
6.2.3	Retrospective	43
6.3	Sprint 2	43
6.3.1	Iteration Planning.....	43
6.3.2	Desarrollo	43
6.3.3	Retrospective	44
6.4	Sprint 3	44
6.4.1	Iteration Planning.....	44
6.4.2	Desarrollo	44
6.4.3	Retrospective.....	44
6.5	Sprint 4	44
6.5.1	Iteration Planning.....	44
6.5.2	Desarrollo	45
6.5.3	Retrospective	45
6.6	Sprint 5	45
6.6.1	Iteration Planning.....	45
6.6.2	Desarrollo	45
6.6.3	Retrospective	46
6.7	Sprint 6	46
6.7.1	Iteration Planning.....	46
6.7.2	Desarrollo	46
6.7.3	Retrospective	46
6.8	Sprint 7	46
6.8.1	Iteration Planning.....	46
6.8.2	Desarrollo	47
6.8.3	Retrospective	47
6.9	Sprint 8	47
6.9.1	Iteration Planning.....	47
6.9.2	Desarrollo	47
6.9.3	Retrospective	47
6.10	Sprint 9	47
6.10.1	Iteration Planning	47
6.10.2	Desarrollo	48
6.10.3	Retrospective	48
6.11	Sprint 10	48
6.11.1	Iteration Planning	48
6.11.2	Desarrollo	48
6.11.3	Retrospective	48
6.12	Sprint 11	48

6.12.1	Iteration Planning	48
6.12.2	Desarrollo	49
6.12.3	Retrospective	49
6.13	Sprint 12	49
6.13.1	Iteration Planning	49
6.13.2	Desarrollo	49
6.13.3	Retrospective	49
7	Pruebas	51
7.1	Casos de pruebas	51
7.2	Resultado de las pruebas	55
7.3	Descripción de Fallos.....	56
8	Conclusiones.....	58
8.1	Cumplimiento de objetivos iniciales.....	58
8.2	Experiencia personal.....	58
8.3	Líneas de trabajo futuro.....	59
9	Bibliografía.....	61
Anexo A: Manual de Usuario.....		66
1.	Página de Identificación de Usuario	66
2.	Página de Registro de Usuario	67
3.	Página de Inventario personal	67
4.	Pestaña de Gastos.....	72
5.	Pestaña de Sugerencias.....	73
6.	Pestaña de Historial	76
7.	Ajustes de cuenta.....	77
8.	Página de ayuda.....	79

Índice de Figuras

Figura 1: Metodología Agile.....	13
Figura 2: Metodología Scrum.....	14
Figura 3: Planning Poker	15
Figura 4: Arquitectura del Stack MEAN.....	31
Figura 5: Flujo de datos en Stack MEAN	31
Figura 6: Diagrama de secuencia del Stack MEAN	32
Figura 7: Diagrama de casos de Uso	33
Figura 8 Interfaz de Identificación de Usuario	66
Figura 9 Interfaz de Registro de Usuario.....	67
Figura 10 Interfaz de inventario personal.....	67
Figura 11 Filtros del inventario del Usuario	68
Figura 12 Carrusel de artículos de nuestro inventario.....	69
Figura 13 Fotos en miniatura en el carrusel de imágenes.	69
Figure 14 Pestañas de navegación.....	70
Figura 15 Botón para añadir nuevos artículos.	70
Figura 16 Formulario para añadir nuevos artículos.	71
Figura 17 Interfaz de Gastos	72
Figure 18 Diagrama de líneas.....	72
Figura 19 Diagrama de sectores.....	73
Figura 20 Pestaña de sugerencias.....	73
Figura 21 Asistente personal.	74
Figura 22 Bocadillo de diálogos.	74
Figura 23 Ejemplo de sugerencia de compra.....	75
Figura 24 Ejemplo de sugerencia de vestimenta.	75
Figura 25 Pestaña del Historial.	76
Figura 26 Sección de Filtros.	76
Figura 27 Ajustes de cuenta.....	77
Figura 28 Temas de la aplicación.	77
Figura 29 Apartado de idiomas.....	78
Figura 30 Página de ayuda.	79
Figura 31 Contacto con soporte técnico.	79

Índice de Tablas

Tabla 1: Identificación de riesgos	17
Tabla 2: R001 - Falta de presupuesto.	18
Tabla 3: R002 - Planificación deficiente.	18
Tabla 4: R003 - Falta de requisitos.	19
Tabla 5: R004 - Implementación retrasada.	19
Tabla 6: R005 - Errores comunes durante el periodo de prueba	19
Tabla 7: R006 - Pruebas insuficientes.	20
Tabla 8: R007 - Incertidumbre técnica.	20
Tabla 9: R008 - Falta de robustez.	20
Tabla 10: R009 - Utilidad.	20
Tabla 11: R010 - Competencia.	21
Tabla 12: R011 - Fallo de diseño.	21
Tabla 13: R012 - Adaptabilidad.	21
Tabla 14: Seguimiento de riesgos.	23
Tabla 15: UC001 - Registro de Usuario	34
Tabla 16: UC002 - Identificación de Usuario	34
Tabla 17: UC003 - Consultar Inventario	35
Tabla 18: UC004 - Registrar Prenda	35
Tabla 19: UC005 - Borrar Prenda	36
Tabla 20: UC006 - Editar Prenda	36
Tabla 21: UC007 - Consultar Gastos	37
Tabla 22: UC008 - Consulta de Compras	37
Tabla 23: UC009 - Cambiar Tema	37
Tabla 24: UC010 - Cambiar Idioma	38
Tabla 25: UC011 - Sugerencia de Compra	39
Tabla 26: UC012 - Sugerencia de Vestimenta	39
Tabla 27: UC013 - Logout	40
Tabla 28: UC014 - Borrar Usuario	40
Tabla 29 Batería de pruebas en versiones Beta y final.	56
Tabla 30 Descripción de fallos y sus correspondientes soluciones en la versión Final.	56

Capítulo 1

Introducción

1. Introducción

1.1. Contexto

Desde el principio de la especie humana, siempre ha sido importante la cuantificación de las cosas o propiedades de cada individuo, familia o tribu. Civilizaciones como la egipcia, maya o vikinga, acostumbraban a hacer inventarios de sus propiedades, ya fueran bienes materiales, alimentos o armas.

Los motivos de este almacenamiento de mercancías eran las posibles adversidades que dichas civilizaciones podían afrontar en futuros cercanos, como por ejemplo, sequías, guerras o simplemente periodos de escasez.

El tener almacenado esas grandes cantidades, traía problemas de espacio o incluso de trabajo, en el caso de bienes almacenados que requirieran algún tipo de mantenimiento como por ejemplo los animales. Por esa razón, era importante conocer exactamente qué se estaba almacenando y que se tenía ya almacenado, pudiendo así elegir entre bienes que no se poseían ya o bienes que trajeran demasiados costes de almacenamiento. Con la llegada del comercio, la necesidad de realizar inventarios se vio aún mas agravada.

El campo de la informática se ha enfrentado con el mismo problema desde sus comienzos. Herman Hollerith creó la máquina automática de tarjetas perforadas (1884), siendo nombrado así el primer ingeniero estadístico de la historia. En esta época, los censos se realizaban de forma manual.

Posteriormente, en la década de los cincuenta se da origen a las cintas magnéticas, para automatizar la información y hacer respaldos. Esto sirvió para suplir las necesidades de información de las nuevas industrias. Y a través de este mecanismo se empezaron a automatizar información, con la desventaja de que solo se podía hacer de forma secuencial.

Más tarde, en la época de los sesenta, las computadoras bajaron los precios para que las compañías privadas las pudiesen adquirir; dando paso a que se popularizara el uso de los discos, cosa que fue un adelanto muy efectivo en la época, debido a que a partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos.

Aunque no fue hasta el año 1963 donde se mencionó por primera vez el termino de bases de datos, la idea de registrar información acerca de cosechas y censos viene desde tiempos

inmemorables. Sin embargo, su búsqueda era lenta y poco eficaz y no se contaba con la ayuda de máquinas que pudiesen reemplazar el trabajo manual.

En esta misma época se dio inicio a las primeras generaciones de bases de datos de red y las bases de datos jerárquicas, ya que era posible guardar estructuras de datos en listas y arboles.

Otro de los principales logros de los años sesenta fue la alianza de **IBM** y American Airlines para desarrollar SABRE, un sistema operativo que manejaba las reservas de vuelos, transacciones e informaciones sobre los pasajeros de la compañía American Airlines.

Por lo que respecta a la década de los setenta, Edgar Frank Codd, científico informático inglés conocido por sus aportaciones a la teoría de bases de datos relacionales, definió el modelo relacional a la par que publicó una serie de reglas para los sistemas de datos relacionales a través de su artículo "Un modelo relacional de datos para grandes bancos de datos compartidos". Este hecho dio paso al nacimiento de la segunda generación de los Sistemas Gestores de Bases de Datos.

Como consecuencia de esto, en la década de los 70, Lawrence J. Ellison, desarrolló el Relational Software System, lo que en la actualidad se conoce como Oracle Corporation, desarrollando así un sistema de gestión de bases de datos relacional con el mismo nombre que dicha compañía.

Durante los ochenta, comenzó el auge de la comercialización de los sistemas relacionales, y SQL comenzó a ser el estándar de la industria

En la década de 1990 la investigación en bases de datos giró en torno a las bases de datos orientadas a objetos.

Las cuales han tenido bastante éxito a la hora de gestionar datos complejos en los campos donde las bases de datos relacionales no han podido desarrollarse de forma eficiente. Así se desarrollaron herramientas como Excel y Access del paquete de Microsoft Office que marcan el inicio de las bases de datos orientadas a objetos. Así se creó la tercera generación de sistemas gestores de bases de datos.

En la actualidad, existen muchos tipos diferentes de bases de datos, como MYSQL, Oracle...etc., que utilizan el modelo relacional y principalmente utilizadas para datos consistentes y bien estandarizados. Otro tipo son las bases de datos no relacionales, como por ejemplo: MongoDB, CouchDB...etc., conocidas como NO-SQL. Este será el tipo de bases de datos utilizadas en este Trabajo de Fin de Grado por motivos de flexibilidad y escalabilidad.

Uno de los problemas a los que se enfrenta la informática en el día de hoy, es el tratamiento de la cantidad de datos generados día a día. Dichos datos pueden proceder de cualquier ámbito, económicos, históricos o incluso datos médicos. La era del internet de las cosas ya ha llegado, por lo que sin las medidas necesarias para el almacenamiento y tratamiento de toda la información generada, esto se convierte en un problema que actúe como cuello de botella y haga que la tecnología no avance con la rapidez que debería.

¿Alguien podía imaginar hace 10 años que un frigorífico sería capaz de “conocer” los alimentos almacenados en su interior o incluso estar conectado con el supermercado y poder realizar la compra de los alimentos que faltan a través de él? Hoy en día no suena tan descabellado, ¿verdad? Ahora supongamos que hablamos de un armario en vez de un frigorífico y de ropa en vez de alimentos. La moda es uno de los grandes motores de la economía mundial y solo en nuestro país, supone una aportación al PIB por encima del 3% del sector primario (sólo algo por debajo al 3,5% de la banca y los seguros).

En definitiva, en este TFG se tratará de mezclar escalabilidad y modularidad en la aplicación “DressApp” que actuará como personal shopper y también como inventario “inteligente” que nos ayudará a la hora de saber que ropa tenemos y que estilo nos gusta para futuras compras.

1.2. Motivación Personal

Los motivos principales que han supuesto el motor principal de esta idea han sido dos:

- Motivos personales: desde que era un niño, siempre he prestado mucha atención a la limpieza y a la apariencia, por lo que, ¿Qué mejor lugar en el que centrar dichas aficiones que en el mundo de la moda? Una anécdota personal que siempre me gusta recordar, era el momento en que llegaba a casa del colegio con la edad de 5 o 6 años, sin que nadie me dijera nada, lo primero que hacía era limpiar mis zapatos y dejarlos preparados para el día siguiente. Espero que el día de mañana sea la tecnología quien me ayude a tener mi ropa bien clasificada, ordenada y preparada.
- Motivos económicos: todo el mundo sabe que el dinero es tristemente uno de los grandes pilares de nuestras vidas. Mirando la revista Forbes, vemos que la persona más rica del mundo es Bill Gates, considerado el magnate número 1 del mundo de la informática. Pero si nos fijamos en nuestro propio país, la persona con la fortuna mas grande es Amancio Ortega, empresario número 1 del sector textil. Por lo que, a simple vista, cabría pensar que la unión de dichos mundos debería ser una apuesta segura.

1.3. Objetivos

El objetivo principal de este trabajo de fin de grado consiste en la creación de una aplicación web que permita realizar una serie de filtros a la hora de ordenar el inventario, tener un registro de todos los tipos de prendas de cada uno de los usuarios y basándose en una serie de estadísticas, esta aplicación sea capaz de “aprender” de los gustos de estos y ayudarles con una serie de sugerencias a la hora de comprar nuestra ropa o elegir, entre la previamente adquirida, para nuestra vestimenta.

Una vez citados los objetivos principales, estos se descomponen en los siguientes objetivos secundarios:

- **Registro de usuarios:** Cada individuo será capaz de registrarse y tener una cuenta personal en la que tendrá almacenada toda su información.
- **Consulta de armario:** La aplicación tendrá almacenados todos los artículos en posesión del usuario. Además, esos artículos serán modificables (añadir/borrar) en cualquier momento.
- **Sugerencias de compra:** A medida que el usuario tenga más prendas registradas en la aplicación, esta será capaz de sugerir productos para comprar, basándose en los gustos que dicha aplicación haya podido aprender a lo largo del tiempo. Como es de esperar, cuantas más prendas haya procesado, las sugerencias serán más acertadas. La idea no es que la aplicación elija por el usuario, sino que le ayude a tomar la decisión, pero siempre teniendo en cuenta que esta siempre será tomada por el cliente.
- **Sugerencias de vestimenta:** Si el usuario no solo “enseña” a la aplicación sus gustos a la hora de comprar sino que también proporciona la información de cómo combinar dichos productos, es decir, como le gusta ir vestido, esta será capaz de sugerir diferentes vestimentas basadas en la ropa de disponible en el armario.
- **Control de gastos:** El sistema será capaz de realizar un seguimiento de los gastos del usuario con el fin de proporcionarle la información necesaria para saber en qué meses gasta más o cuánto lleva gastado en ropa en un mes o año.

1.4. Estructura de la memoria

Este documento está estructurado en ocho secciones, un apéndice y una bibliografía.

- Introducción: contextualización, temática y objetivos para el desarrollo del Trabajo Fin de Grado.
- Gestión del proyecto: en este punto se puede ver el alcance, los costes, y la planificación del presente proyecto.
- Requisitos: descripción de los requisitos y los correspondientes casos de uso.
- Análisis y diseño: se describen los modelos de análisis que contiene el TFG. Se puede ver los diagramas de casos de uso y los diagramas de secuencia. Además se explica el modelo arquitectónico seguido y la realización en diseño de los casos de uso descritos con anterioridad.
- Implementación: se detalla el entorno de desarrollo y un seguimiento en la ejecución de la aplicación.
- Pruebas: Incluye la batería de pruebas realizadas a la aplicación desarrollada y los resultados obtenidos en cada una de ellas.
- Conclusiones: Incluye las conclusiones a las que se ha llegado, tras los conocimientos adquiridos durante todo el proceso de aprendizaje y los objetivos logrados. A su vez, se señalan líneas y trabajos futuros de la aplicación.
- Bibliografía: Se enumeran las diferentes fuentes y referencias bibliográficas empleadas para el desarrollo de este TFG.
- Apéndices: Se adjunta un Manual de Uso de la aplicación desarrollada.

2. Entorno Tecnológico

2.1. Entorno de desarrollo

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto.

2.2. Tecnologías utilizadas

A lo largo de esta sección, se describen una a una las tecnologías empleadas para este trabajo de fin de grado, desde los lenguajes de programación hasta las plataformas utilizadas para ubicar esta aplicación, pasando por los entornos y dispositivos utilizados para la fase de pruebas.

Empezaremos explicando esta sección hablando de nuestro lenguaje de programación base, que es JavaScript, y avanzando hacia capas mas altas de abstracción como son los diferentes “Frameworks” y Librerías

2.2.1. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.



2.2.2. Typescript



TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. Anders Hejlsberg, diseñador de C# y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de TypeScript. TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js).

TypeScript extiende la sintaxis de JavaScript, por lo tanto, cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

TypeScript soporta ficheros de definición que contengan información sobre los tipos de librerías JavaScript existentes, similares a los ficheros de cabeceras de C/C++ que describen la estructura de ficheros de objetos existentes. Esto permite a otros programas usar los valores definidos en los ficheros como si fueran entidades TypeScript de tipado estático. Existen cabeceras para librerías populares como jQuery, MongoDB y D3.js, y los módulos básicos de Node.js.

El compilador de TypeScript está escrito asimismo en TypeScript, compilado a JavaScript y con Licencia Apache 2.

TypeScript se incluye como lenguaje de programación de primer nivel en Microsoft Visual Studio 2013 Update 2 y posteriores, junto a C# y otros lenguajes de Microsoft. Una extensión oficial permite a Visual Studio 2012 soportar también TypeScript.

2.2.3. HTML

HTML, siglas en inglés de HyperText Markup Language (“lenguaje de marcas de hipertexto”), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium

(W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que



se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos, tales como
, no tienen contenido ni llevan una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en HTML.

2.2.4. CSS



Firefox OS).

CSS (siglas en inglés de **Cascading Style Sheets**), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etc. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como

2.2.5. Angular

Angular (comúnmente llamado **Angular 2+** o **Angular 2**) es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.



2.2.6. Material Design



Material design es una normativa de diseño enfocado en la visualización del sistema operativo Android, además en la web y en cualquier plataforma. Fue desarrollado por Google y anunciado en la conferencia Google I/O celebrada el 25 de junio de 2014. Ampliando la interfaz de tarjetas vista por primera vez en Google Now.

Material se integró en Android Lollipop como reemplazo de Holo, anteriormente utilizado desde Android 4 y sucesores. La filosofía también se aplicó en Google Drive y Google Docs, Sheets y Slides, y se irá extendiendo progresivamente a todos los

productos de Google (incluyendo Google Search, Gmail y Google Calendar), proporcionando una experiencia consistente en todas las plataformas. Google también lanzó APIs para que los desarrolladores externos incorporaran Material Design a sus aplicaciones.

2.2.7. NodeJs

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables.



Pueden atenderse muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada o *callback*, pero si no hay trabajo que hacer, Node.js se dormirá.

Esto contrasta con el modelo de concurrencia más común de hoy en día, en el que se emplean hilos del Sistema Operativo. Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Además, los usuarios de Node.js están libres de preocuparse por el bloqueo del proceso, ya que no existe. Casi ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea. Por ello, es muy propicio desarrollar sistemas escalables en Node.js.

2.2.8. MongoDB

MongoDB (del inglés *humongous*, "enorme") es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.



En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

2.3. Librerías externas

Se han utilizado una serie de librerías externas que han ayudado a conseguir el propósito global con pequeñas funcionalidades añadidas. Las librerías externas principales, utilizadas en la aplicación son:

- *Ngx-gallery*: librería utilizada para la visualización de imágenes en forma de carrusel.
- *Ngx-charts*: librería utilizada para la visualización de los datos en forma de gráficos.
- *Ngx-Translate*: librería de internacionalización de Angular utilizada para la traducción a los diferentes idiomas.
- *Ngx-mat-file-input*: librería utilizada para la entrada de imágenes.
- *TypeIt*: librería utilizada para la simulación de escritura por teclado.

Gestión y planificación del proyecto

3. Gestión del Proyecto

Debido a la experiencia personal con la Metodología Agile, se ha decidido seguir esta a lo largo del desarrollo de este TFG.

3.1. Metodología Ágil

La metodología "Agile" envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto organizados y multidisciplinares, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Cada iteración del ciclo de vida incluye planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Adquiere una gran importancia el concepto de "finalizado" (done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).



Figura 1: Metodología Agile

Los principales beneficios de esta metodología son:

- a) Versatilidad, orientada a la consecución de objetivos
- b) Reestructuración de acciones
- c) Transparencia

Bajo los principios de las Metodologías Agile han surgido diferentes ramas que presentan ventajas específicas. Una de las más utilizadas es la llamada "Scrum".

La metodología Scrum se diferencia de las demás por la flexibilidad, productividad e innovación que trae al día a día del equipo. Se basa en una estructura de desarrollo incremental. El proyecto se desgrana en pequeñas tareas que se dividen en tres etapas: análisis, desarrollo y testing.

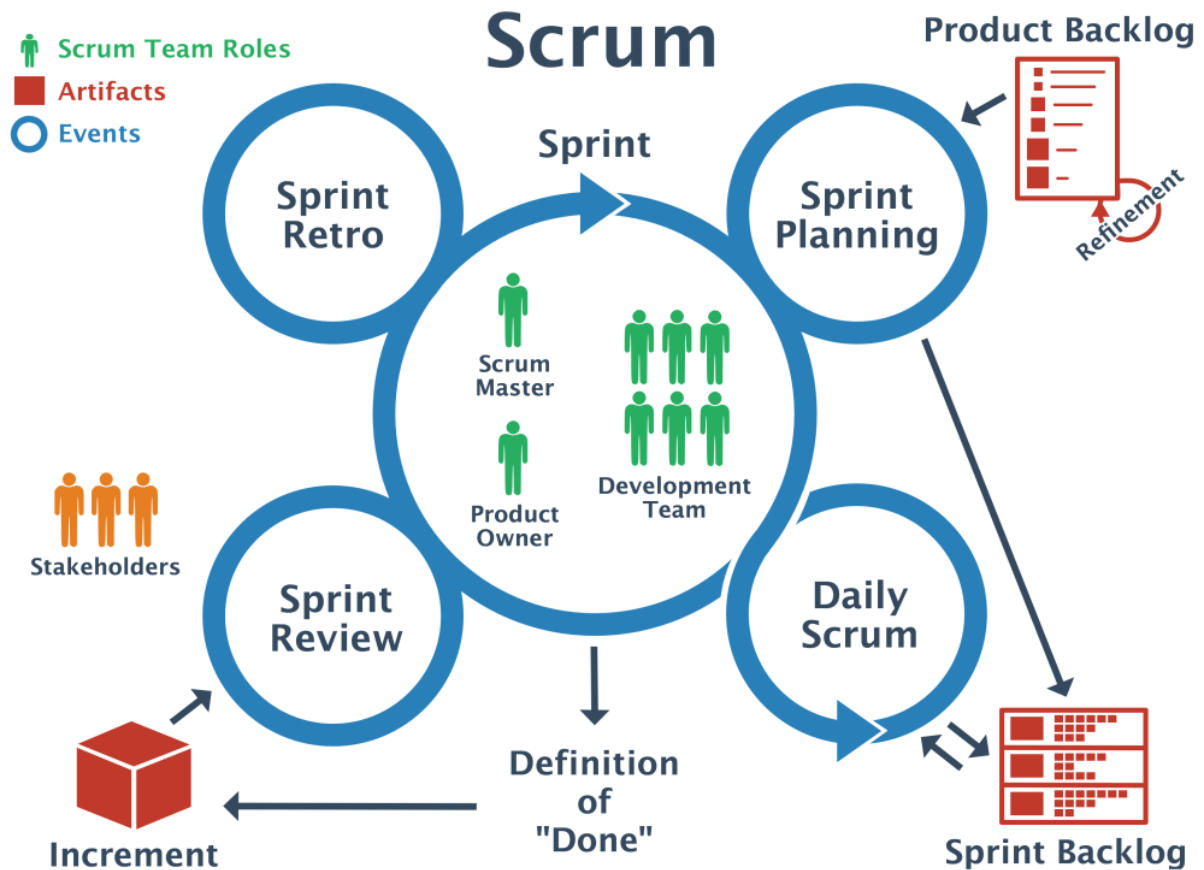


Figura 2: Metodología Scrum

Aunque este tipo de metodologías estén pensadas para equipos enteros de desarrollo, la característica principal por la que se ha decidido trabajar bajo los principios de estas es la transparencia y la flexibilidad. Se supondrá desde ahora que mis roles principales serán el equipo de desarrollo así como como el Scrum Master y que los "Stakeholders" o clientes será la Universidad de Valladolid.

Además, la realización de estas ceremonias como Sprint Planning y Sprint Retro permitirán la división en subtareas y una visión externa de las mismas para un refinamiento en futuras iteraciones o Sprints.

3.2. Planificación inicial

Como se ha explicado anteriormente, la primera etapa es la de análisis en la cual se realizará una planificación inicial.

Se puede decir de manera muy general que el objetivo principal de este TFG es el desarrollo de nuestra aplicación web. Obviamente se debe desgranar este objetivo en las partes principales de dicha aplicación. En este momento se identifican las siguientes tareas principales:

- Configuración inicial de la base de datos.
- Configuración inicial del entorno de desarrollo.
- Configuración del servidor web
- Conexión del Front-end y Back-end

Una vez se tengan estas tareas listas, se podrá comenzar con el desarrollo de la aplicación. Por ahora no se pensará en la realización de tareas más específicas, sino que se deberá hacer una estimación en términos de complejidad de las mismas y posteriormente se verán reflejadas en tiempo de implementación.

Para dicha estimación, se utilizará otra de las ceremonias de la metodología Scrum llamada "Planning Poker".

La dinámica de esta ceremonia es utilizar un conjunto de cartas o tarjetas, normalmente mostrando la secuencia de Fibonacci incluyendo un cero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. A esta secuencia se suelen añadir los símbolos "∞" y "?" que representan respectivamente que una tarea es demasiado grande o que no se sabe con exactitud el tamaño de la misma.

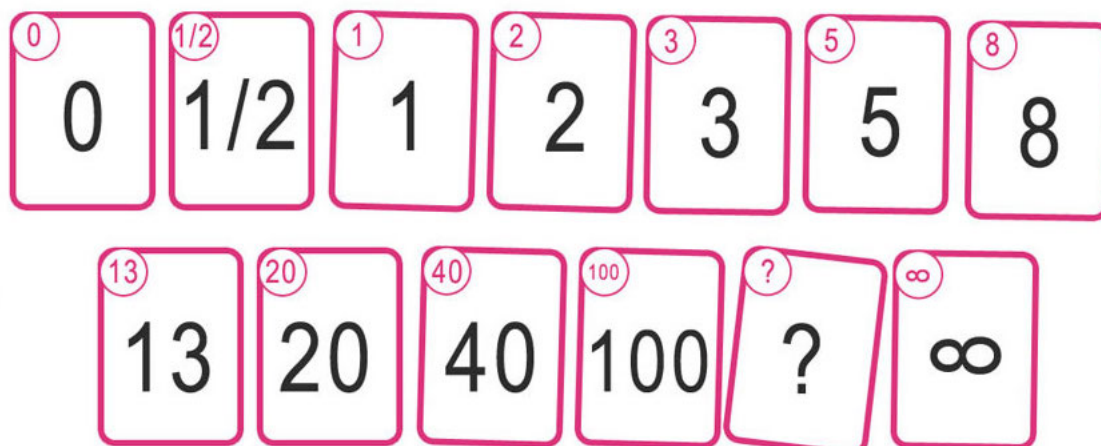


Figura 3: Planning Poker

Una vez explicado esto, se procederá a la estimación de las tareas previamente citadas:

- Configuración inicial de la base de datos → 3
- Configuración inicial del entorno de desarrollo → 1
- Configuración del servidor web → 5
- Conexión del Front-end y Back-end → 3

Suponiendo que, por ahora el equipo de desarrollo (Yo) es capaz de finalizar aproximadamente 21 story points a lo largo de un mes, esto significa que la puesta en marcha inicial llevará alrededor de medio mes.

Hasta ahora se ha supuesto que la situación es la ideal y no hay contratiempos, aunque en realidad, bien es sabido que estos son inevitables por lo que habrá que identificar los posibles riesgos así como el coste que estos supondrían si llegaran a suceder.

3.3. Identificación de riesgos

Un riesgo se puede definir como una situación incierta que, en caso de ocurrir, tendría un efecto positivo o negativo sobre el proyecto.

Existen varios tipos de riesgos, estos son:

- **Riesgos del proyecto:** ponen en peligro al plan. Si estos se cumplen, el proyecto requerirá mayor esfuerzo y dinero.
- **Riesgos técnicos:** ponen en peligro la calidad resultante. Si estos se cumplen, el proyecto es más complejo de lo estimado.
- **Riesgos del negocio:** Ponen en peligro la realización del proyecto. Si estos se cumplen, el proyecto se cancelará

Se clasifican los diferentes tipos de riesgos en función a su probabilidad:

- Muy alta (>75%)
- Alta (50-75%)
- Media (30-50%)
- Baja (10-30%)

Por último, no todos los riesgos tienen la misma repercusión en caso de ocurrencia. Consecuencia de los riesgos en orden descendiente:

- Fatal
- Crítico
- Aceptable
- Mínimo

Una vez explicado todo lo anterior, se procede a la identificación de riesgos para la aplicación:

#	Descripción	Tipo	Probabilidad	Repercusión
R001	Falta de presupuesto: se necesitan más recursos de los esperados.	Proyecto	Alta	Mínimo
R002	Planificación deficiente: se necesita más tiempo del esperado.	Proyecto	Muy Alta	Aceptable
R003	Falta de requisitos: se encuentran escenarios no contemplados durante la planificación.	Proyecto	Muy Alta	Aceptable
R004	Implementación retrasada: se necesita más tiempo del esperado durante la implementación.	Técnico	Muy Alta	Aceptable
R005	Errores comunes durante el periodo de prueba.	Técnico	Media	Mínimo
R006	Pruebas insuficientes: se requieren más pruebas de las esperadas.	Proyecto	Baja	Mínimo
R007	Incertidumbre técnica: se necesita mayor conocimiento sobre el área.	Técnico	Media	Aceptable
R008	Falta de robustez: se necesita una mayor verificación de los datos de entrada.	Proyecto y Técnico	Media	Mínimo
R009	Utilidad: el producto final no cubre todas las necesidades planificadas.	Proyecto y Técnico	Baja	Fatal
R010	Competencia: existen productos en el mercado con una aceptación ya alta.	Negocio	Media	Crítico
R011	Fallo de diseño: el estilo inicial tiene deficiencias inesperadas.	Proyecto y Técnico	Media	Aceptable
R012	Adaptabilidad: la aplicación no se adapta de manera correcta a los diferentes tipos de dispositivos.	Técnico	Baja	Mínimo

Tabla 1: Identificación de riesgos

3.4. Gestión de riesgos

Saber como gestionar los riesgos previamente identificados es clave para evitar sobrecostos así como retrasos sobre el producto final. Esta gestión no solo implica las medidas a tomar si dichos riesgos suceden (plan de contingencia) sino también aquellas medidas que nos ayuden a evitar estos riesgos lleguen a darse (plan de mitigación).

R001 - Falta de presupuesto: se necesitan más recursos de los esperados.	
Tipo	Proyecto
Probabilidad	Alta
Repercusión	Mínimo
Fases afectadas	Todas
Plan de Mitigación	A la hora de hacer una estimación de costes, se contará con el peor caso posible y al finalizar se hará un reajuste de cuentas.
Plan de Contingencia	Si incluso las peores estimaciones fueron incorrectas, se dispondrá de un presupuesto de emergencia. En caso de no ser utilizado será devuelto durante el reajuste de cuentas.

Tabla 2: R001 - Falta de presupuesto.

R002 - Planificación deficiente: se necesita más tiempo del esperado.	
Tipo	Proyecto
Probabilidad	Muy Alta
Repercusión	Aceptable
Fases afectadas	Todas excepto planificación
Plan de Mitigación	Igual que durante la estimación de costes, se tendrá una planificación pesimista, es decir, siempre contaremos con el peor caso posible.
Plan de Contingencia	Se intentaría reducir el tiempo extra en cada una de las iteraciones (Sprints) hasta recuperar el tiempo extra utilizado en la planificación.

Tabla 3: R002 - Planificación deficiente.

R003 - Falta de requisitos: se encuentran escenarios no contemplados durante la planificación.	
Tipo	Proyecto
Probabilidad	Muy Alta
Repercusión	Aceptable
Fases afectadas	Implementación y pruebas

Plan de Mitigación	Se estudiarán todos los escenarios posibles en busca de casos imprevistos y se confiará en la experiencia y los conocimientos previos en las tecnologías utilizadas.
Plan de Contingencia	Se utilizará el tiempo sobrante durante alguno de los Sprints posteriores o en el peor de los casos se añadirá otro Sprint más.

Tabla 4: R003 - Falta de requisitos.

R004 - Implementación retrasada: se necesita más tiempo del esperado durante la implementación.	
Tipo	Técnico
Probabilidad	Muy Alta
Repercusión	Aceptable
Fases afectadas	Implementación y pruebas
Plan de Mitigación	En todos los Sprints se añadirá un tiempo extra para imprevistos.
Plan de Contingencia	Se haría una entrega parcial del producto con todas las funcionalidades principales y las funcionalidades secundarias restantes se entregarían en una segunda fase.

Tabla 5: R004 - Implementación retrasada.

R005 - Errores comunes durante el periodo de prueba.	
Tipo	Técnico
Probabilidad	Media
Repercusión	Mínimo
Fases afectadas	Implementación y pruebas
Plan de Mitigación	Gracias a la metodología Agile esto no supondría un gran problema ya que, al trabajar por iteraciones, los errores pueden ser solventados en iteraciones siguientes.
Plan de Contingencia	Como se ha mencionado en el plan de mitigación, cada iteración alimenta a la siguiente. Esto significa que en la iteración x se podrán arreglar problemas de la iteración x-1

Tabla 6: R005 - Errores comunes durante el periodo de prueba

R006 - Pruebas insuficientes: se requieren más pruebas de las esperadas.	
Tipo	Proyecto
Probabilidad	Baja
Repercusión	Mínimo
Fases afectadas	Pruebas

Plan de Mitigación	Se realizarán pruebas después de cada funcionalidad implementada desde varios puntos de vista (pruebas con más de un usuario).
Plan de Contingencia	Añadir una nueva iteración (Sprint) solo con las partes a probar.

Tabla 7: R006 - Pruebas insuficientes.

R007 - Incertidumbre técnica: se necesita mayor conocimiento sobre el área.	
Tipo	Técnico
Probabilidad	Media
Repercusión	Aceptable
Fases afectadas	Implementación
Plan de Mitigación	Se debe contemplar esta posibilidad en la planificación inicial así como en cada una de las iteraciones posteriores.
Plan de Contingencia	Se realizará la ceremonia del “Planning Poker” en cada iteración (Sprint) para estimar la complejidad de cada funcionalidad con relación al conocimiento previo de la tecnología utilizada.

Tabla 8: R007 - Incertidumbre técnica.

R008 - Falta de robustez: se necesita una mayor verificación de los datos de entrada.	
Tipo	Proyecto y Técnico
Probabilidad	Media
Repercusión	Mínimo
Fases afectadas	Implementación y pruebas
Plan de Mitigación	Se deberán verificar todas las entradas de datos al sistema aceptando solo la información esperada y mostrando mensajes de error en caso de recibir alguna información errónea o inesperada.
Plan de Contingencia	Mismo que en R005.

Tabla 9: R008 - Falta de robustez.

R009 - Utilidad: el producto final no cubre todas las necesidades planificadas.	
Tipo	Proyecto y Técnico
Probabilidad	Baja
Repercusión	Fatal
Fases afectadas	Todas
Plan de Mitigación	Se priorizarán las funcionalidades principales para que en caso de faltar alguna funcionalidad, sea alguna secundaria.
Plan de Contingencia	Mismo que en R004.

Tabla 10: R009 - Utilidad.

R010 - Competencia: existen productos en el mercado con una aceptación ya alta.	
Tipo	Negocio
Probabilidad	Media
Repercusión	Crítico
Fases afectadas	Todas
Plan de Mitigación	Se hará un estudio de mercado para ver las aplicaciones similares y sus puntos débiles.
Plan de Contingencia	Se deberá hacer una inversión extra en una campaña de marketing que permita vender las funcionalidades extra en las que la aplicación es superior a la competencia.

Tabla 11: R010 - Competencia.

R011 - Fallo de diseño: el estilo inicial tiene deficiencias inesperadas.	
Tipo	Proyecto y Técnico
Probabilidad	Media
Repercusión	Aceptable
Fases afectadas	Implementación y pruebas
Plan de Mitigación	Se harán pruebas de diseño antes de empezar la programación utilizando “mockups”* que nos permitan ver visualmente el producto final.
Plan de Contingencia	Se harán los cambios necesarios durante el tiempo extra de la iteración correspondiente o si son cambios significativos, se añadirá una iteración (Sprint) extra.

Tabla 12: R011 - Fallo de diseño.

*Mockup: un modelo o réplica de una máquina o estructura, utilizada con fines educativos o experimentales.

R012 - Adaptabilidad: la aplicación no se adapta de manera correcta a los diferentes tipos de dispositivos.	
Tipo	Técnico
Probabilidad	Baja
Repercusión	Mínimo
Fases afectadas	Implementación y pruebas
Plan de Mitigación	Se deberá probar cada cambio en la interfaz de usuario en diferentes tipos de dispositivos, desde monitores más grandes a pantallas de teléfonos móviles.
Plan de Contingencia	Se harán los cambios necesarios durante el tiempo extra de la iteración correspondiente o si son cambios significativos, se añadirá una iteración (Sprint) extra.

Tabla 13: R012 - Adaptabilidad.

3.5. Seguimiento de riesgos

Una vez se han identificado los riesgos y se ha explicado como serán gestionados, se hará un seguimiento de los mismos durante las fases posteriores y se verá si el plan de mitigación fue necesario o en su defecto, se explicarán como afectarán al desarrollo de este proyecto.

Riesgo	Sucedió	Repercusión
R001 - Falta de presupuesto: se necesitan más recursos de los esperados.	No	
R002 - Planificación deficiente: se necesita más tiempo del esperado.	No	
R003 - Falta de requisitos: se encuentran escenarios no contemplados durante la planificación.	Si	No se ha pensado en alguna funcionalidad que ha habido que añadir sobre la marcha. Ha causado el retraso de algunas iteraciones
R004 - Implementación retrasada: se necesita más tiempo del esperado durante la implementación.	Si	Debido a funcionalidades no contempladas o problemas a la hora de desarrollar algunas de las esperadas, ha habido que añadir nuevas iteraciones.
R005 - Errores comunes durante el periodo de prueba.	No	
R006 - Pruebas insuficientes: se requieren más pruebas de las esperadas.	No	
R007 - Incertidumbre técnica: se necesita mayor conocimiento sobre el área.	Si	El almacenamiento de imágenes en la base de datos así como la entrada de ellas en la aplicación ha supuesto un reto y ha supuesto retrasos en la implementación.
R008 - Falta de robustez: se necesita una mayor verificación de los datos de entrada.	No	
R009 - Utilidad: el producto final no cubre todas las necesidades planificadas.	No	
R010 - Competencia: existen productos en el mercado con una aceptación ya alta.	No	
R011 - Fallo de diseño: el estilo inicial tiene deficiencias inesperadas.	No	

R012 - Adaptabilidad: la aplicación no se adapta de manera correcta a los diferentes tipos de dispositivos.	No	
---	----	--

Tabla 14: Seguimiento de riesgos.

3.6. Costes

Después de haber explicado las tecnologías utilizadas y haber pensado en todo el hardware y software a utilizar se puede hacer un presupuesto inicial que será reajustado en el momento de la entrega del producto final en el caso de haber sufrido cambios en términos de tiempo o costes inesperados. Se añadirá un presupuesto extra para cubrir los mencionados costes inesperados.

3.6.1. Presupuesto inicial

Concepto		Uds.	Base Ud.	Base Total
Honorarios (130 días x 4h = 520h)		520	17,045€	8800,63€
Material	Microsoft 365 Personal	1	69€	69€
	MacBook Pro (15-inch, 2017)	1	2490,09€	2490,09€
	Samsung Galaxy S8	1	809€	809€
	Galaxy Tab S6 Lite (10.4", Wi-Fi) con S Pen	1	399€	399€
Licencias	Visual Studio Code	1	0€	0€
Hosting	Heroku	1	0€	0€
Storage	MongoDB	1	0€	0€
Extra	Presupuesto para imprevistos	1	3000€	3000€

Base Imponible	15630,73€
Tipo impuesto	21%
TOTAL	18913,18€

3.6.2. Reajuste de presupuesto

Debido a que el proyecto se ha visto alargado por los motivos previamente expuestos, se ha tenido que hacer un reajuste de presupuesto.

Concepto		Uds.	Base Ud.	Base Total
Honorarios (200 días x 4h = 800h)		800	17,045€	13636€

Material	Microsoft 365 Personal	1	69€	69€
	MacBook Pro (15-inch, 2017)	1	2490,09€	2490,09€
	Samsung Galaxy S8	1	809€	809€
	Galaxy Tab S6 Lite (10.4", Wi-Fi) con S Pen	1	399€	399€
Licencias	Visual Studio Code	1	0€	0€
Hosting	Heroku	1	0€	0€
Storage	MongoDB	1	0€	0€

Base Imponible	17403,09€
Tipo impuesto	21%
TOTAL	21057,74€

Este reajuste no ha sido demasiado significativo gracias al presupuesto extra para imprevistos. En total, se ha tenido que añadir 2144,56€ al presupuesto inicial.

4. Requisitos

4.1. Requisitos Funcionales

A continuación, se identifican los principales servicios con los que la aplicación contará a través de los siguientes Requisitos funcionales:

<i>FR 001</i>	<i>Registro de Usuario</i>
<i>Descripción</i>	El Sistema permitirá el registro de nuevos usuarios.
<i>FR 002</i>	<i>Identificación de Usuario</i>
<i>Descripción</i>	El Sistema será capaz de identificar usuarios previamente registrados.
<i>FR 003</i>	<i>Mostrar Inventario</i>
<i>Descripción</i>	El Sistema será capaz de mostrar el inventario del usuario identificado.
<i>FR 004</i>	<i>Mostrar Gastos</i>
<i>Descripción</i>	El Sistema será capaz de mostrar todos los gastos registrado por el usuario.
<i>FR 005</i>	<i>Sugerir Compra</i>
<i>Descripción</i>	El Sistema será capaz de dar sugerencias de compra al usuario identificado.
<i>FR 006</i>	<i>Sugerir Vestimenta</i>
<i>Descripción</i>	El Sistema permitirá al usuario identificado recibir sugerencias de vestimenta.
<i>FR 007</i>	<i>Registrar prendas</i>
<i>Descripción</i>	El Sistema permitirá registrar nuevas prendas una vez el usuario está identificado.
<i>FR 008</i>	<i>Eliminar prendas</i>
<i>Descripción</i>	El Sistema permitirá el borrado de prendas previamente registradas.
<i>FR 009</i>	<i>Moldeabilidad</i>
<i>Descripción</i>	El Sistema permitirá al usuario moldear la apariencia visual de la aplicación en cualquier momento después de haberse identificado.

4.2. Requisitos No Funcionales

Para acotar las restricciones de la aplicación que permitirán al usuario actuar de una manera concreta a lo largo del uso de la misma y responder a la pregunta del “Cómo” se implementan las principales funcionalidades, se definen los siguientes requisitos No Funcionales:

<i>NFR 001</i>	<i>Navegador Web</i>
<i>Descripción</i>	La aplicación será accesible desde cualquier navegador web como por ejemplo Google Chrome o Firefox.
<i>NFR 002</i>	<i>User friendly</i>
<i>Descripción</i>	La aplicación deberá ser lo más intuitiva posible haciendo que en mas del 90% de las actividades que el usuario pueda realizar, no tenga la necesidad de consultar el manual de usuario.
<i>NFR 003</i>	<i>Conexión a internet</i>
<i>Descripción</i>	La aplicación solo será accesible con conexión a internet y no tendrá ningún modo “sin conexión”.
<i>NFR 004</i>	<i>Eficiente</i>
<i>Descripción</i>	El Sistema deberá ser lo suficientemente efectivo para responder a cualquier petición del usuario en un tiempo menor a 5s.
<i>NFR 005</i>	<i>Concurrente</i>
<i>Descripción</i>	El Sistema permitirá que un número ilimitado de usuarios esté conectado al mismo tiempo sin experimentar problemas de eficiencia o fallos en la respuesta.
<i>NFR 006</i>	<i>Respaldo de datos</i>
<i>Descripción</i>	Las bases de datos utilizadas por el sistema deberán contar con copias de seguridad para garantizar la consistencia de los datos ante cualquier tipo de imprevisto.
<i>NFR 007</i>	<i>Seguridad de los datos</i>
<i>Descripción</i>	El Sistema deberá encriptar todo tipo de información sensible a la hora de ser transmitida o recibida por el usuario.

<i>NFR 008</i>	<i>Manual de usuario</i>
<i>Descripción</i>	El Sistema deberá contar con manuales de usuario documentando cualquier tipo de actividad permitida al usuario.
<i>NFR 009</i>	<i>Mensajes de error</i>
<i>Descripción</i>	Ante cualquier fallo del Sistema, este deberá responder con mensajes de error lo suficientemente claros como para que el usuario entienda cual fue el problema ocurrido.
<i>NFR 010</i>	<i>Visualización</i>
<i>Descripción</i>	EL Sistema deberá contar con un diseño “Responsive” para garantizar la adecuada visualización desde cualquier tamaño de monitor, tableta o teléfono móvil.
<i>NFR 011</i>	<i>Tema agradable</i>
<i>Descripción</i>	La aplicación deberá contar con temas agradables visualmente con los que el usuario sienta plena comodidad a lo largo de su uso.
<i>NFR 012</i>	<i>Multi-Lenguaje</i>
<i>Descripción</i>	La aplicación deberá contar con al menos dos lenguas (español e inglés) para poder tener un alcance internacional.

4.3. Requisitos de Información

Además de los requisitos previamente detallados, un apartado esencial de cualquier aplicación es la información almacenada y el tratamiento de la misma. Por este motivo, se declaran los siguientes requisitos de Información:

<i>IFR 001</i>	<i>Usuario</i>
<i>Descripción</i>	La aplicación deberá almacenar información relativa al usuario registrado para su correcta identificación y un trato personalizado.
<i>Datos</i>	<ul style="list-style-type: none"> ○ Nombre ○ Apellidos ○ Sexo ○ Contraseña ○ Correo electrónico ○ Preferencias <ul style="list-style-type: none"> ○ Idioma ○ Tema principal

IFR 002

Prenda

<i>Descripción</i>	La aplicación deberá almacenar información relativa a las prendas registradas con el fin de ser lo más preciso posible a la hora de su sugerencia así como su debida identificación.
<i>Datos</i>	<ul style="list-style-type: none">○ Propietario○ Coste○ Fecha de compra○ Tipo○ Color○ Estación○ Género○ Imagen<ul style="list-style-type: none">○ Nombre de la imagen○ Tipo de Imagen○ Fichero

5. Análisis y diseño

5.1. Arquitectura de la aplicación

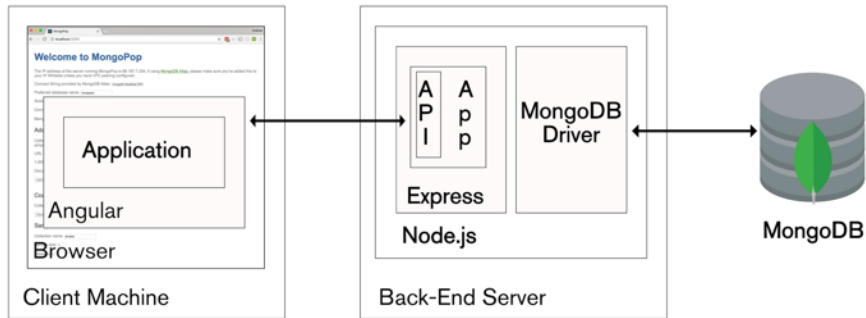


Figura 4: Arquitectura del Stack MEAN

Como se ha descrito anteriormente, la aplicación consta de dos partes principales que son: el Frontend o cliente y el Backend. Dentro del cliente, la aplicación hecha en Angular es accesible a través de un navegador web. El Backend está desarrollado en NodeJs y engloba dos partes principales: Los “endpoints”, en express, los cuales son llamados por el cliente, y el driver de MongoDB que se encargara de la conexión con la base de datos.

5.2. Flujo de datos

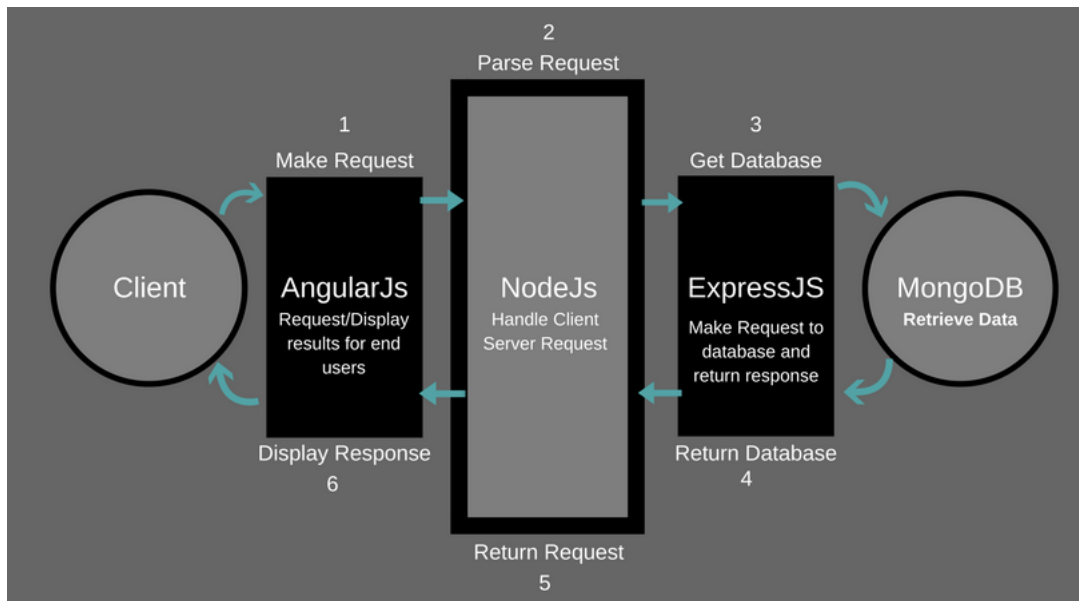


Figura 5: Flujo de datos en Stack MEAN

Los datos en una aplicación desarrollada con el Stack MEAN, fluyen de la manera descrita en la figura 5. El cliente se comunica con la aplicación Angular y ciertos datos son calculados y devueltos por el propio Angular. Algunos datos necesitan ser calculados o devueltos por el

Backend por motivos de seguridad o complejidad. Si esos cálculos requieren información almacenada en la base de datos, entonces el ciclo llegaría hasta su punto más profundo que es la base de datos MongoDB. Los datos que son requeridos por el Backend se utilizan para los cálculos oportunos y se muestran al cliente a través de la aplicación Angular.

De manera genérica se puede decir que cualquier interacción del usuario con la aplicación, seguirá de total o parcialmente la secuencia descrita en el siguiente diagrama.

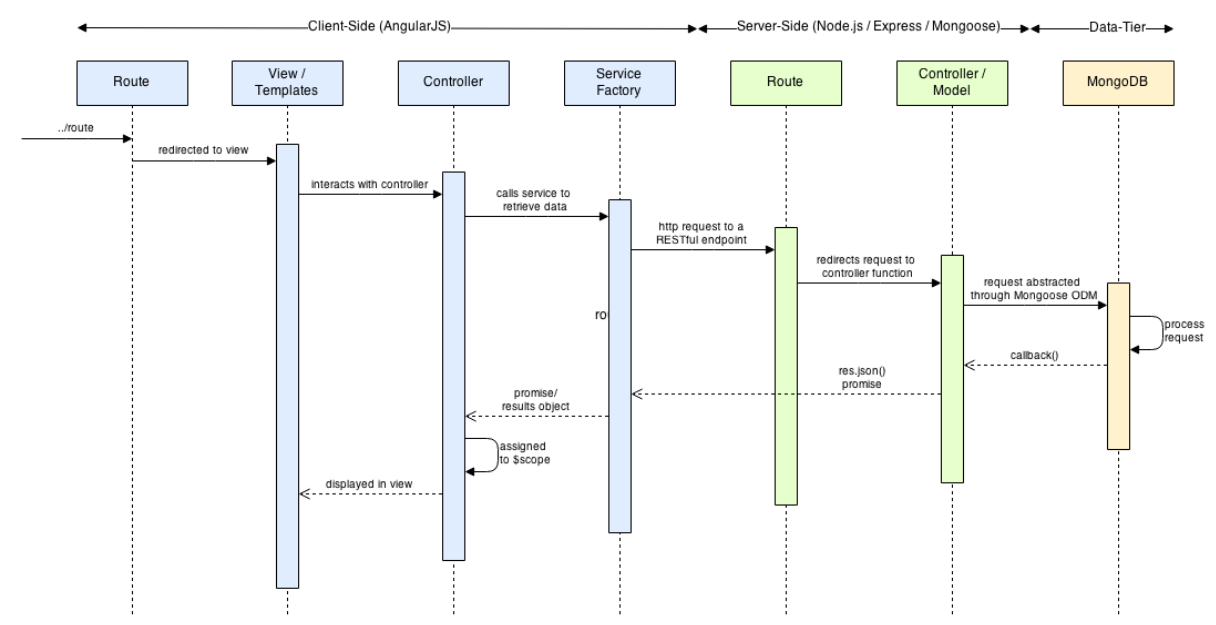


Figura 6: Diagrama de secuencia del Stack MEAN

5.3. Diagrama de Casos de Uso

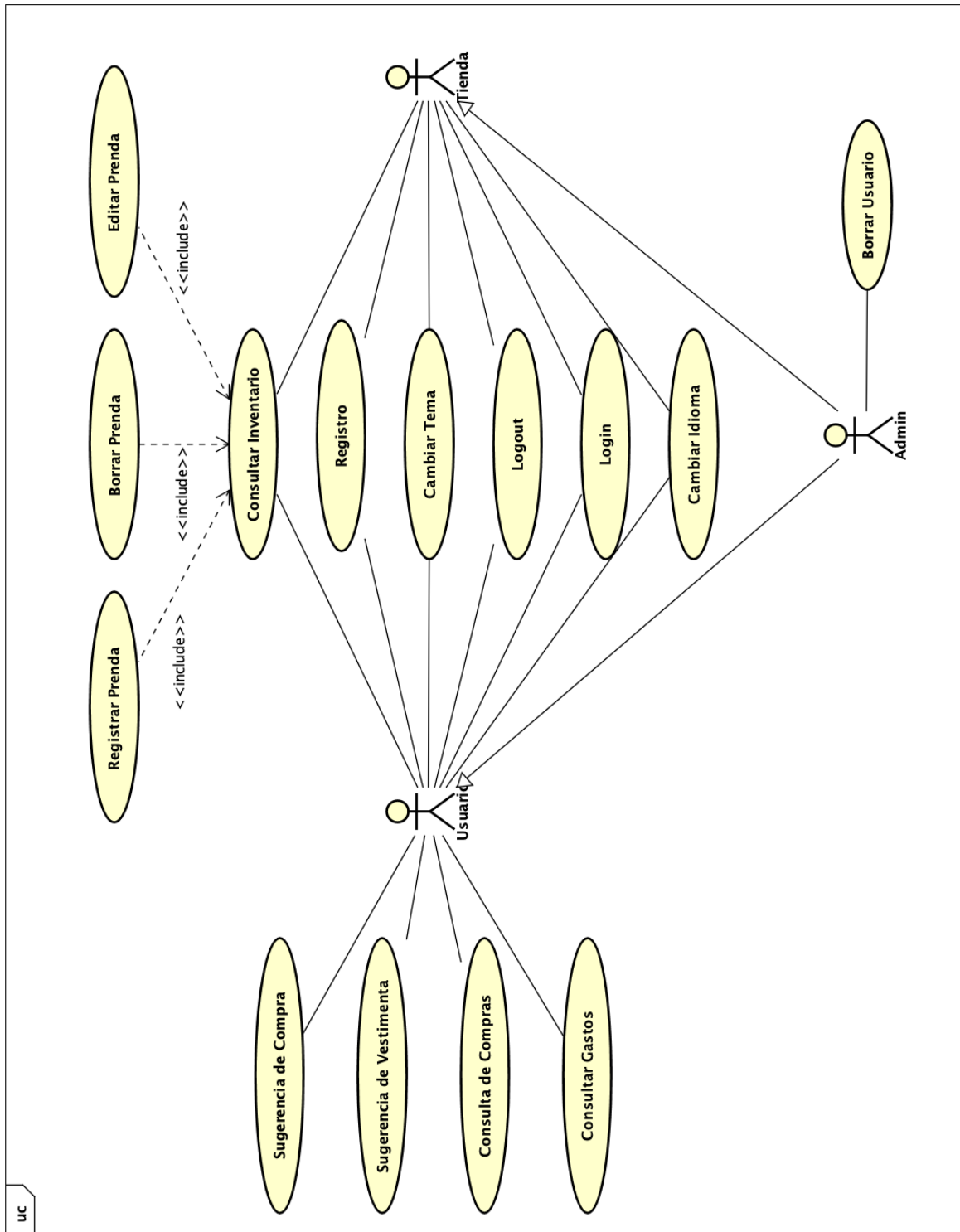


Figura 7: Diagrama de casos de Uso

5.4. Casos de Uso

UC001 - Registro de Usuario	
Descripción	Usuario se registra en el sistema.
Precondición	El usuario no está registrado previamente.
Secuencia	<ol style="list-style-type: none"> 1 El Usuario selecciona "Registro" en la pantalla principal. 2 El Sistema muestra los campos necesarios para el registro. 3 El Usuario introduce los campos necesarios para el registro. 4 El Sistema valida que todos los campos introducidos son correctos y habilita el botón "enviar". 5 El Usuario presiona el botón "enviar". 6 El Sistema notifica al usuario el registro satisfactorio y redirige a la pantalla de Login
Postcondición	El Usuario no podrá registrarse de nuevo con las mismas credenciales.
Excepciones	<ol style="list-style-type: none"> 4* El Sistema identifica un error en alguno de los campos introducidos, notifica el error y continua en el paso 3. 6* El Sistema tiene un fallo a la hora de registrar al usuario, notifica del error y redirige a la pantalla de login.

Tabla 15: UC001 - Registro de Usuario

UC002 - Identificación de Usuario	
Descripción	Usuario se identifica en el sistema para acceder a sus datos.
Precondición	El Usuario esta previamente registrado en el Sistema
Secuencia	<ol style="list-style-type: none"> 1 El Usuario rellena los campos "Usuario" y "Contraseña". 2 El sistema valida que los datos introducidos son validos y habilita el botón "enviar". 3 El Usuario presiona el botón "enviar". 4 El Sistema verifica los datos recibidos y carga la información relativa al Usuario.
Postcondición	El Usuario no podrá identificarse de nuevo hasta cerrar la sesión.
Excepciones	<ol style="list-style-type: none"> 2* El Sistema identifica algún carácter inválido, notifica del error y continua en el paso 1. 4* El Sistema no encuentra al Usuario dentro de los perfiles registrados, notifica del error y continua en el paso 1

Tabla 16: UC002 - Identificación de Usuario

UC003 - Consultar Inventario	
Descripción	Usuario desea ver los ítems almacenados en su inventario.
Precondición	El Usuario está identificado en el sistema.

Secuencia	1 El Usuario se sitúa en la pestaña principal "Inventario".
	2 El Sistema carga todo el inventario y habilita los diferentes filtros.

Tabla 17: UC003 - Consultar Inventario

UC004 - Registrar Prenda	
Descripción	Usuario desea añadir a su inventario un nuevo elemento.
Precondición	El Usuario está identificado en el Sistema y se encuentra en la pestaña "Inventario".
Secuencia	1 El Usuario presiona el botón "+" situado en la parte inferior derecha de la ventana.
	2 El Sistema muestra un formulario con los campos necesarios para añadir la nueva prenda.
	3 El Usuario rellena dicho formulario.
	4 El sistema valida los campos introducidos y habilita el botón "Añadir".
	5 El Usuario presiona el botón "Añadir".
	6 El Sistema registra el cambio en la base de datos y redirige al usuario a su inventario.
Excepciones	2* El Sistema no es capaz de redireccionar al Usuario al formulario debido a un error interno, notifica del error y desconecta al Usuario de la sesión.
	4* El Sistema identifica algún campo inválido, notifica del error y continua en el paso 3.
	6* El Sistema no es capaz de registrar el cambio en la base de datos, notifica del error y continua en el paso 3.

Tabla 18: UC004 - Registrar Prenda

UC005 - Borrar Prenda	
Descripción	Usuario desea borrar de su inventario un elemento.
Precondición	El Usuario está identificado en el Sistema y se encuentra en la pestaña "Inventario".
Secuencia	1 El Usuario selecciona el elemento a borrar.
	2 El Sistema muestra la información de la prenda seleccionada.
	3 El Usuario presiona el botón "Eliminar".
	4 El sistema muestra un diálogo de confirmación del borrado.
	5 El Usuario presiona el botón "Confirmar".
	6 El Sistema registra el cambio en la base de datos y redirige al usuario a su inventario actualizado.
Postcondición	La prenda eliminada no será recuperable.

Excepciones	6* El Sistema no es capaz de registrar el cambio en la base de datos, notifica del error y continua en el paso 1.
--------------------	---

Tabla 19: UC005 - Borrar Prenda

UC006 - Editar Prenda	
Descripción	Usuario desea editar la información de alguno de los elementos de su inventario.
Precondición	El Usuario está identificado en el Sistema y se encuentra en la pestaña "Inventario".
Secuencia	<ol style="list-style-type: none"> 1 El Usuario selecciona el elemento a editar. 2 El Sistema muestra la información de la prenda seleccionada. 3 El Usuario presiona el botón "Editar". 4 El Sistema redirige al usuario al formulario de añadir prenda pero con los campos rellenos con la información correspondiente a la prenda a editar. 5 El Usuario edita el o los campos a cambiar. 6 El Sistema valida los campos del formulario y habilita el botón "Actualizar". 7 El Usuario presiona el botón "Actualizar". 8 El Sistema registra el cambio en la base de datos y redirige al usuario a su inventario.
Postcondición	El Sistema almacena una única información de la prenda y no el historial de cambios.
Excepciones	<ol style="list-style-type: none"> 4* El Sistema identifica algún error al rellenar los campos con la información previamente almacenada, notifica del error, deja el campo vacío y continua en el paso 5. 6* El Sistema identifica algún error en alguno de los campos actualizados, notifica del error y continua en el paso 5. 8* El Sistema no es capaz de registrar el cambio en la base de datos, notifica del error y continua en el paso 5.

Tabla 20: UC006 - Editar Prenda

UC007 - Consultar Gastos	
Descripción	Usuario desea consultar sus gastos.
Precondición	El Usuario está identificado en el Sistema.
Secuencia	<ol style="list-style-type: none"> 1 El Usuario presiona la pestaña "Gastos". 2 El Sistema carga los gastos del Usuario en forma de lista y gráfico ordenados cronológicamente.
Excepciones	2.1 El Sistema no es capaz de cargar los gastos, notifica del error y muestra la lista o gráfico vacíos.

Tabla 21: UC007 - Consultar Gastos

UC008 - Consulta de Compras	
Descripción	Usuario desea consultar sus compras.
Precondición	El Usuario está identificado en el Sistema.
Secuencia	1 El Usuario presiona la pestaña Compras.
	2 El Sistema carga las compras del Usuario en forma de lista y ordenadas cronológicamente.
Excepciones	2* El Sistema no es capaz de cargar las compras, notifica del error y muestra la lista vacía.

Tabla 22: UC008 - Consulta de Compras

UC009 - Cambiar Tema	
Descripción	Usuario desea cambiar el tema principal de la Aplicación
Precondición	El Usuario está identificado en el Sistema.
Secuencia	1 El Usuario presiona su avatar situado en la parte superior derecha de la ventana.
	2 El Sistema muestra una serie de opciones en forma de menú desplegable.
	3 El Usuario selecciona el icono de cambio de tema situado dentro del menú.
	4 El Sistema muestra los diferentes colores principales de los temas a escoger en forma de submenú desplegable.
	5 El Usuario selecciona el tema deseado.
	6 El Sistema carga el tema seleccionado, guarda el cambio en las preferencias del Usuario y cierra los menús desplegables.
Postcondición	El Usuario mantendrá el tema seleccionado hasta que vuelva a cambiarlo.
Excepciones	6* El Sistema no es capaz de cargar el tema seleccionado, notifica del error y continua en el paso 5.
	6** El Sistema no es capaz de guardar el cambio en las preferencias del Usuario, notifica del error, cambia el tema y el caso de uso finaliza.

Tabla 23: UC009 - Cambiar Tema

UC010 - Cambiar Idioma	
Descripción	Usuario desea cambiar el idioma principal de la Aplicación
Precondición	El Usuario está identificado en el Sistema.
Secuencia	1 El Usuario presiona su avatar situado en la parte superior derecha de la ventana.

	2 El Sistema muestra una serie de opciones en forma de menú desplegable.
	3 El Usuario selecciona el icono de cambio de idioma situado dentro del menú.
	4 El Sistema muestra los diferentes idiomas a escoger en forma de submenú desplegable.
	5 El Usuario selecciona el idioma deseado.
	6 El Sistema carga el idioma seleccionado, guarda el cambio en las preferencias del Usuario y cierra los menús desplegados.
Postcondición	El Usuario mantendrá el idioma seleccionado hasta que vuelva a cambiarlo.
Excepciones	6* El Sistema no es capaz de cargar el idioma seleccionado, notifica del error y continua en el paso 5.
	6** El Sistema no es capaz de guardar el cambio en las preferencias del Usuario, notifica del error, cambia el idioma y el caso de uso finaliza.

Tabla 24: UC010 - Cambiar Idioma

UC011 - Sugerencia de Compra	
Descripción	Usuario desea recibir una sugerencia de un nuevo elemento a comprar.
Precondición	El Usuario está identificado en el Sistema
Secuencia	1 El Usuario presiona la pestaña "Sugerencias".
	2 El Sistema carga el asistente personal y habilita los dos tipos de sugerencias disponibles (compra y vestimenta).
	3 El Usuario selecciona "Sugerencia de compra".
	4 El Sistema analiza el inventario personal del usuario y sugiere un tipo de prenda a comprar.
	5 El Usuario acepta la sugerencia presionando el botón "Aceptar sugerencia".
	6 El Sistema registra la sugerencia como satisfactoria y notifica al usuario.
Postcondición	La sugerencia aceptada no podrá ser borrada del historial de sugerencias satisfactorias.
Secuencia alternativa 1	5.1 El Usuario no acepta la sugerencia de compra y el caso de uso continua en el paso 4.
Excepciones	2* El Sistema no es capaz de cargar el asistente personal, notifica del error, redirecciona al Usuario a su inventario y el caso de uso finaliza.
	4* El Sistema no es capaz de analizar el inventario personal, lo notifica mediante excusa*, y el caso de uso finaliza.

	5.1* El Usuario no acepta 3 sugerencias de compra seguidas, el Sistema notifica que se rinde*, y el caso de uso finaliza.
	6* El Sistema no es capaz de registrar la sugerencia como satisfactoria, notifica al usuario y el caso de uso finaliza.

Tabla 25: UC011 - Sugerencia de Compra

*excusa: mensaje de error camuflado, por ejemplo, “No estoy muy motivado en estos momentos, prueba más tarde”.

*rendirse: mensaje de finalización camuflado, por ejemplo, “No he sido capaz de ayudarte con esta compra, espero poder hacerlo para la próxima”.

UC012 - Sugerencia de Vestimenta	
Descripción	Usuario desea recibir una sugerencia de vestimenta.
Precondición	El Usuario está identificado en el Sistema
Secuencia	<p>1 El Usuario presiona la pestaña “Sugerencias”.</p> <p>2 El Sistema carga el asistente personal da los dos tipos de sugerencias disponibles (compra y vestimenta).</p> <p>3 El Usuario selecciona las sugerencias de vestimenta.</p> <p>4 El Sistema analiza el inventario personal del usuario y sugiere un conjunto de prendas a llevar.</p> <p>5 El Usuario acepta la sugerencia presionando el botón “Aceptar sugerencia”.</p> <p>6 El Sistema registra la sugerencia como satisfactoria y notifica al usuario.</p>
Postcondición	La sugerencia aceptada no podrá ser borrada del historial de sugerencias satisfactorias.
Secuencia alternativa 1	5.1 El Usuario no acepta la sugerencia de vestimenta y el caso de uso continua en el paso 4.
Excepciones	<p>2* El Sistema no es capaz de cargar el asistente personal, notifica del error, redirecciona al Usuario a su inventario y el caso de uso finaliza.</p> <p>4* El Sistema no es capaz de analizar el inventario personal, lo notifica mediante alguna excusa*, y el caso de uso finaliza.</p> <p>5.1* El Usuario no acepta 3 sugerencias de vestimenta seguidas, el Sistema notifica que se rinde*, y el caso de uso finaliza.</p> <p>6* El Sistema no es capaz de registrar la sugerencia como satisfactoria, notifica al usuario y el caso de uso finaliza.</p>

Tabla 26: UC012 - Sugerencia de Vestimenta

*excusa: mensaje de error camuflado, por ejemplo, “No estoy muy motivado en estos momentos, prueba más tarde”.

*rendirse: mensaje de finalización camuflado, por ejemplo, “No he sido capaz de ayudarte con esta compra, espero poder hacerlo para la próxima”.

UC013 - Logout	
Descripción	Usuario desea cerrar la sesión.
Precondición	El Usuario está identificado en el Sistema.
Secuencia	1 El Usuario presiona su avatar situado en la parte superior derecha de la ventana. 2 El Sistema muestra una serie de opciones en forma de menú desplegable. 3 El Usuario selecciona la opción "cerrar sesión". 4 El Sistema cierra la sesión del Usuario y muestra la ventana de inicio de sesión.
Postcondición	La sesión permanecerá cerrada

Tabla 27: UC013 - Logout

UC014 - Borrar Usuario	
Descripción	Administrador desea borrar un Usuario definitivamente del Sistema.
Precondición	El usuario Administrador está identificado en el Sistema.
Secuencia	1 El usuario Administrador presiona la pestaña "Usuarios". 2 El Sistema muestra todos los usuarios, en forma de lista y ordenados alfabéticamente por nombre de usuario. 3 El usuario Administrador selecciona el usuario deseado. 4 El Sistema muestra los datos registrados del usuario seleccionado. 5 El usuario Administrador presiona el botón "Eliminar". 6 El Sistema muestra un diálogo de confirmación. 7 El usuario Administrador confirma el borrado presionando el botón "Confirmar". 8 El Sistema borra el usuario seleccionado, notifica el borrado y actualiza la lista de usuarios.
Postcondición	El usuario borrado no será recuperable.
Secuencia alternativa 1	5.1 El usuario Administrador presiona el botón "Cerrar". 5.2 El Sistema cierra la información del usuario seleccionado y el caso de uso continua en el paso 2.
Secuencia alternativa 2	7.1 El usuario Administrador presiona el botón "Cancelar". 7.2 El Sistema cierra el diálogo de confirmación y el caso de uso continua en el paso 4.
Excepciones	2* El Sistema no es capaz de cargar la lista de usuarios, notifica del error, muestra la lista vacía y el caso de uso finaliza. 8* El Sistema no es capaz de eliminar al usuario, notifica del error y el caso de uso continua en el paso 2.

Tabla 28: UC014 - Borrar Usuario

6. Implementación

Como se ha explicado anteriormente, se seguirá una metodología Ágil en la cual se trabajará por iteraciones. Cada iteración durará 2 semanas y constará de 3 partes:

- I. **Iteration Planning:** se trata de la planificación para la iteración actual o posteriores, en la cual se marcarán objetivos en términos de tareas y tiempo.
- II. **Desarrollo:** es la fase en la que se implementan las tareas planificadas durante el Iteration Planning.
- III. **Retrospective:** al finalizar la iteración se hará una retrospectiva para ver si se han cumplido los objetivos marcados y en caso de no ser así, determinar la carga de trabajo que debe ser movida a la siguiente iteración.

El proceso de implementación comienza el día 15/12/2019.

6.1. Sprint 0

6.1.1. Iteration Planning

Como se describió en la planificación inicial, las tareas principales que se comenzarán en esta iteración son:

- Configuración inicial de la base de datos → 3puntos
- Configuración inicial del entorno de desarrollo → 1punto
- Configuración del servidor web → 5puntos
- Conexión del Front-end y Back-end → 3puntos

Esto supone que se estima, que dichas tareas ($3+1+5+3 = 12$ puntos) estén acabadas en aproximadamente 1 iteración.

6.1.2. Desarrollo

ACTIVIDAD	15/12	18/12	22/12	25/12	29/12
Configuración del entorno de desarrollo					
Configuración de la base de datos					
Configuración del servidor web					
Conexión Frontend-Backend					

6.1.3. Retrospective

Después de configurar una base de datos en “Firebase” y empezar con la configuración del servidor web se ha decidido desechar la base de datos y configurar de nuevo una base de datos en “Cloudant” debido a la familiaridad con la misma. Por lo tanto, la configuración de la base de datos y la conexión Frontend-Backend deberá ser implementada en la siguiente iteración.

6.2 Sprint 1

6.2.1 Iteration Planning

Debido al retraso en la iteración anterior, se continuarán las tareas que han sido arrastradas desde la iteración anterior.

6.2.2 Desarrollo

ACTIVIDAD	01/01	05/01	08/01	12/01	14/01
Configuración del servidor web					
Conexión Frontend-Backend					

6.2.3 Retrospective

Se ha completado la configuración inicial del servidor web así como la conexión entre el servidor y el cliente.

6.3 Sprint 2

6.3.1 Iteration Planning

En las siguientes iteraciones se realizarán las tareas descritas a continuación:

- Búsqueda de hosting para el despliegue de la aplicación → ?Puntos

6.3.2 Desarrollo

ACTIVIDAD	15/01	19/01	22/01	26/01	31/01
Búsqueda de Hosting					

6.3.3 Retrospective

Se ha decidido desplegar la aplicación en “Heroku”, una plataforma como servicio que soporta diferentes lenguajes de programación como NodeJs en nuestro caso. Surge un problema debido a que la base de datos gratuita NoSQL es MongoDB por lo que se deberá volver a configurar la base de datos y desechar el trabajo previamente realizado.

6.4 Sprint 3

6.4.1 Iteration Planning

De la iteración anterior, hemos concluido que se debe volver a configurar la base de datos. Además de esta tarea se comenzará con la creación de las interfaces de login y registro, estimadas con 5 y 6 puntos respectivamente.

6.4.2 Desarrollo

ACTIVIDAD	01/02	05/02	09/02	12/02	15/02
Configuración MongoDB					
Creación de interfaz de login					
Creacion de interfaz de registro					

6.4.3 Retrospective

Se han completado las tareas planificadas para esta iteración.

6.5 Sprint 4

6.5.1 Iteration Planning

Durante esta iteración se han identificado las siguientes tareas con su estimación:

- Interfaz gastos → 4Puntos
- Interfaz historial → 3Puntos
- Interfaz inventario → 5Puntos
- Interfaz Añadir Prenda → 2Puntos

En total son 14 puntos por lo que es posible que alguna de las tareas se arrastre a la iteración siguiente.

6.5.2 Desarrollo

ACTIVIDAD	16/02	20/02	23/02	26/02	29/02
Interfaz de Gastos					
Interfaz de Historial					
Interfaz de Inventario					
Interfaz añadir prenda					

6.5.3 Retrospective

La decisión de la librería para los gráficos de gastos y la animación en los filtros para el historial han hecho que la interfaz de inventario aún no este acabada y la última tarea ni tan siquiera este empezada

6.6 Sprint 5

6.6.1 Iteration Planning

Dos tareas vienen arrastradas de la iteración anterior y además se han identificado las siguientes funcionalidades a implementar en la actual iteración con su correspondiente estimación:

- Interfaz sugerencias → 10Puntos
- Interfaz ayuda → 3Puntos

En total son 13 Puntos más aproximadamente 4 de las tareas arrastradas de la iteración anterior por lo que es muy probable que alguna de estas funcionalidades deba ser movida a la siguiente iteración.

6.6.2 Desarrollo

ACTIVIDAD	01/03	04/03	08/03	11/03	15/03
Interfaz de Inventario					
Interfaz Añadir Prenda					
Interfaz Sugerencias					
Interfaz de ayuda					

6.6.3 Retrospective

Como se esperaba, las dos últimas tareas no han sido acabadas y por lo tanto, se continuara con ellas en la siguiente iteración.

6.7 Sprint 6

6.7.1 Iteration Planning

Como en la anterior iteración, algunas tareas vienen siendo arrastradas. Además de estas, se han identificado las siguientes funcionalidades con su correspondiente estimación:

- Routing → 2Puntos
- Cambio de tema → 2Puntos
- Cambio de idioma → 2 Puntos

6.7.2 Desarrollo

ACTIVIDAD	16/03	19/03	22/03	26/03	29/03
Interfaz de Sugerencias					
Interfaz de ayuda					
Routing					
Cambio de tema					
Cambio de idioma					

6.7.3 Retrospective

Se han finalizado todas las tareas planificadas para la iteración y no ha habido contratiempos inesperados.

6.8 Sprint 7

6.8.1 Iteration Planning

Una vez finalizado con las interfaces de usuario, se empezará a programar la funcionalidad de las mismas. Se ha considerado que las dos funcionalidades siguientes son las primeras a implementar y se han estimado de la siguiente manera:

- Implementar Registro → 5Puntos
- Implementar Login → 6Puntos

6.8.2 Desarrollo

ACTIVIDAD	30/03	03/04	07/04	10/04	14/04
Funcionalidad Registro					
Funcionalidad Login					

6.8.3 Retrospective

Las tareas se han implementado en el tiempo estimado.

6.9 Sprint 8

6.9.1 Iteration Planning

Una vez implementada la funcionalidad de login y de registro, se deberán almacenar y recibir los datos reales en la base de datos por lo que se han identificado las siguientes funcionalidades con su correspondiente estimación:

- Conexión base de datos → 4Puntos
- Búsquedas en base de datos → 4Puntos

6.9.2 Desarrollo

ACTIVIDAD	15/04	19/04	22/04	26/04	30/04
Conexión base de datos					
Búsquedas en base de datos					

6.9.3 Retrospective

Aunque se estimaba que en esta iteración se tendría tiempo para implementar otra funcionalidad, las búsquedas en la base de datos han sido mas laboriosas de lo esperado.

6.10 Sprint 9

6.10.1 Iteration Planning

Una vez es posible el envío o la recogida de datos de nuestra base de datos MongoDB, se ha considerado oportuno continuar con las siguientes funcionalidades y su correspondiente estimación:

- Funcionalidad inventario → 4Puntos
- Funcionalidad gastos → 3Puntos
- Funcionalidad historial → 3Puntos

6.10.2 Desarrollo

ACTIVIDAD	01/05	04/05	08/05	12/05	15/05
Funcionalidad Inventario					
Funcionalidad Gastos					
Funcionalidad Historial					

6.10.3 Retrospective

No se esperaba que la implementación de la funcionalidad del inventario requiriera tanto tiempo, pero el almacenamiento y recogida de las imágenes de la base de datos se ha alargado durante toda la iteración.

6.11 Sprint 10

6.11.1 Iteration Planning

Debido a los imprevistos sufridos en la anterior iteración, se procederá a una estimación precavida y solo se continuará con las tareas arrastradas del anterior sprint.

6.11.2 Desarrollo

ACTIVIDAD	16/05	19/05	23/05	27/05	31/05
Funcionalidad Gastos					
Funcionalidad Historial					

6.11.3 Retrospective

Gracias a la estimación precavida tomada en esta iteración, ha supuesto concluir con todas las tareas identificadas hasta el momento.

6.12 Sprint 11

6.12.1 Iteration Planning

Se ha considerado oportuno, el desarrollo de una funcionalidad de soporte técnico. Esta funcionalidad se trata de una forma de contacto de los usuarios en caso de identificar algún problema. Este sistema de tickets junto con la funcionalidad de sugerencias tienen la siguiente estimación:

- Funcionalidad ayuda → 5Puntos
- Funcionalidad sugerencias → 10Puntos

6.12.2 Desarrollo

ACTIVIDAD	01/06	04/06	07/06	11/06	14/06
Funcionalidad Ayuda					
Funcionalidad Sugerencias					

6.12.3 Retrospective

Como ya era esperado, la funcionalidad de sugerencias se trata de la más compleja de la aplicación por lo que se arrastrará a la siguiente iteración y se intentará finalizar dentro de la misma.

6.13 Sprint 12

6.13.1 Iteration Planning

Siguiendo lo anteriormente citado en la última retrospective, se intentará acabar con la funcionalidad de las sugerencias y se añadirán las traducciones. Estas traducciones ya habían sido previamente configuradas y lo único que falta es un fichero con todos los textos traducidos a los diferentes idiomas. La aplicación contará con los idiomas inglés, español, eslovaco, griego y finlandés y dichas traducciones serán realizadas por personas nativas en cada lengua.

6.13.2 Desarrollo

ACTIVIDAD	15/06	19/06	23/06	27/06	30/06
Funcionalidad Sugerencias					
Traducciones					

6.13.3 Retrospective

Se ha conseguido finalizar todas las tareas hasta ahora identificadas en los plazos estimados. En ocasiones se han arrastrado tareas de unas iteraciones a las siguientes o incluso se ha tenido que volver a implementar algunas de estas funcionalidades, pero gracias a la continua retroalimentación de esta metodología se han cumplido con los plazos previstos.

7 Pruebas

Durante todo el periodo de desarrollo de la aplicación se han ido realizando diferentes tipos de pruebas. Las mas comunes y utilizadas han sido las **pruebas Unitarias**. En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo, en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correcta y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido, entonces el estado final es válido también.

La idea es escribir casos de prueba para cada función no trivial o método en el módulo, de forma que cada caso sea independiente del resto. Luego, con las **pruebas de Integración**, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

Además de este tipo de pruebas, la aplicación, después de cada iteración o después de ser acabada por completa, ha sido probada con los casos de pruebas descritos en el siguiente apartado.

7.1 Casos de pruebas

CP-001	Identificación de Usuario Registrado
Descripción de la prueba	Un usuario previamente registrado intenta iniciar sesión con sus credenciales.
Resultado esperado	La aplicación identifica correctamente al usuario y le redirige a su pagina de "home".

CP-002	Identificación de Usuario sin Registrar
Descripción de la prueba	Un usuario sin registrar intenta iniciar sesión con unas credenciales aleatorias.
Resultado esperado	La aplicación no reconoce las credenciales e informa de ellos a través de un mensaje de error.

CP-003	Identificación con credenciales erróneas
Descripción de la prueba	Un usuario previamente registrado intenta iniciar sesión con unas credenciales incorrectas.
Resultado esperado	La aplicación no reconoce las credenciales y le informa a través de un mensaje de error.

CP-004	Registro de nuevo usuario
Descripción de la prueba	Un usuario sin registrar intenta registrarse en el sistema a través de la página de registro.
Resultado esperado	La aplicación almacena las credenciales del usuario e informa que estas fueron registradas satisfactoriamente.

CP-005	Consulta de inventario en carrusel
Descripción de la prueba	El usuario identificado en el sistema intenta consultar sus prendas previamente almacenadas a través de la galería de imágenes.
Resultado esperado	La aplicación muestra las imágenes deseadas y va cambiando de una a otra con cada click del usuario.

CP-006	Filtro de inventario a través de imágenes
Descripción de la prueba	El usuario identificado en el sistema intenta filtrar sus prendas a través de las imágenes de filtro.
Resultado esperado	La aplicación identifica el tipo de prenda seleccionado y muestra las imágenes en la galería en concordancia.

CP-007	Añadir nueva prenda
Descripción de la prueba	El usuario intenta añadir una nueva prenda en el apartado "Añadir nuevo elemento", rellenando el formulario y adjuntando una imagen de la prenda.
Resultado esperado	La aplicación notifica de la nueva prenda registrada y actualiza la galería de imágenes con la nueva imagen guardada.

CP-008	Añadir nueva prenda con foto de mas de 16MB
Descripción de la prueba	El usuario intenta registrar una nueva prenda adjuntando una imagen de la misma de un tamaño mayor de 16MB.
Resultado esperado	La aplicación informa a través de un mensaje de error, que el tamaño máximo de las imágenes es de 16MB y ha sido excedido.

CP-009	Consulta de gastos en diferentes gráficos
Descripción de la prueba	El usuario accede al apartado de gastos y cambia los gráficos para una visualización deseada.
Resultado esperado	La aplicación actualiza el gráfico al tipo seleccionado con los datos del usuario almacenados en la base de datos.

CP-010	Sugerencia de Vestimenta
Descripción de la prueba	El usuario selecciona una sugerencia de vestimenta dentro del apartado de sugerencias.
Resultado esperado	La aplicación calcula la nueva sugerencia y la muestra a través de varias imágenes y un mensaje en pantalla.

CP-011	Sugerencia de Compra
Descripción de la prueba	El usuario selecciona una sugerencia de compra dentro del apartado de sugerencias.
Resultado esperado	La aplicación calcula la nueva sugerencia y la muestra a través de una imagen y un mensaje en pantalla.

CP-012	Aceptar sugerencia
Descripción de la prueba	El usuario que ha recibido una sugerencia selecciona la opción "Aceptar sugerencia".
Resultado esperado	La aplicación registra la sugerencia como satisfactoria y muestra un mensaje por pantalla de confirmación.

CP-013	Rechazar sugerencia
Descripción de la prueba	El usuario que ha recibido una sugerencia selecciona la opción "Rechazar sugerencia".
Resultado esperado	La aplicación cuenta el número de sugerencias rechazadas y en caso de ser menor de 3, calcula y muestra una diferente. Si se hubieran rechazado 3 sugerencias, la aplicación no mostraría más sugerencias por pantalla.

CP-014	Consulta de Historial
Descripción de la prueba	EL usuario identificado en el sistema intenta revisar el historial a través del apartado "Historial".
Resultado esperado	La aplicación muestra el historial de compras del usuario a través de una tabla paginada con 5 entradas por página.

CP-015	Filtrado de Historial
Descripción de la prueba	El usuario intenta filtrar su historial de compras a través de alguno de los filtros.
Resultado esperado	La aplicación pide los datos filtrados a la base de datos y muestra la tabla con los nuevos datos.

CP-016	Consulta de ayuda
Descripción de la prueba	Un usuario, identificado o no, intenta visitar el apartado de ayuda.
Resultado esperado	La aplicación redirecciona al usuario a la URL de ayuda y muestra su contenido.

CP-017	Contactar soporte técnico
Descripción de la prueba	Un usuario, identificado o no, contacta al servicio técnico a través del formulario que aparece al hacer click en “Abrir un ticket”
Resultado esperado	La aplicación registra el ticket creado en github y notifica a través de un mensaje de que la creación del ticket fue satisfactoria.

CP-018	Cambiar tema principal
Descripción de la prueba	Un usuario, registrado o no, intenta cambiar el tema principal de la aplicación, cambiándolo a alguno de los temas disponibles.
Resultado esperado	La aplicación registra el cambio como preferencias del usuario y cambia la apariencia de la aplicación en concordancia al tema seleccionado.

CP-019	Cambiar Idioma principal
Descripción de la prueba	Un usuario, registrado o no, intenta cambiar el lenguaje principal de la aplicación a cualquiera de los idiomas disponibles.
Resultado esperado	La aplicación cambia todos los textos en tiempo de ejecución sin mostrar ningún mensaje de confirmación.

CP-020	Prueba del Routing
Descripción de la prueba	Un usuario, registrado o no, cambia manualmente la URL de la aplicación.
Resultado esperado	La aplicación valida que el usuario tiene el acceso permitido a la URL deseada y en caso afirmativo, redirecciona a la URL seleccionada. En caso de no tener acceso a la URL deseada, redirecciona a la página de inicio de sesión si el usuario no ha iniciado sesión o a su página de home.

CP-021	Cerrar sesión
Descripción de la prueba	El usuario identificado en el sistema intenta cerrar sesión haciendo click en “Cerrar sesión”.
Resultado esperado	La aplicación cierra la sesión del usuario y le redirecciona a la pagina de inicio de sesión.

CP-022	Uso de aplicación en diferente Navegador Web
Descripción de la prueba	Se prueba la aplicación desde Mozilla Firefox y Google Chrome.
Resultado esperado	La aplicación responde sin ninguna diferencia desde cualquiera de los navegadores.

7.2 Resultado de las pruebas

Una vez identificadas las pruebas a realizar junto con los resultados esperados, se procederá a su ejecución y su consecuente resultado. En caso de que alguno de los resultados no sea el esperado, se deberá estudiar si el resultado esperado no estaba bien calculado o si se trata de un error a arreglar. Si existe un error en la versión actual de la aplicación, se debe arreglar y realizar las mismas pruebas en una versión siguiente.

Caso de Prueba	Versiones Beta	Versión Final
CP-001	OK	OK
CP-002	OK	OK
CP-003	OK	OK
CP-004	OK	OK
CP-005	OK	OK
CP-006	Fallo-001	OK
CP-007	Fallo-002	OK
CP-008	OK	OK
CP-009	OK	OK
CP-010	Fallo-003	OK
CP-011	OK	OK
CP-012	OK	OK
CP-013	OK	OK
CP-014	OK	OK
CP-015	Fallo-004	OK
CP-016	OK	OK
CP-017	OK	OK
CP-018	OK	OK
CP-019	Fallo-005	OK

CP-020	Fallo-006	OK
CP-021	OK	OK
CP-022	OK	OK

Tabla 29 Batería de pruebas en versiones Beta y final.

7.3 Descripción de Fallos

A continuación se describen los fallos ocurridos en más detalle y se da una explicación de la solución tomada.

#Fallo	Descripción	Solución
Fallo-001	Al seleccionar un tipo de prenda concreto, el carrusel de imágenes no se actualiza.	Llamada al Backend para la recepción de los nuevos elementos ya filtrados.
Fallo-002	Al añadir una prenda de manera satisfactoria, el carrusel de imágenes no se actualiza con la nueva prenda añadida.	Llamada al método de actualizar del carrusel para la recepción de posibles cambios sucedidos.
Fallo-003	Al recibir una sugerencia de vestimenta, se muestran más de una prenda del mismo tipo.	Actualizar la lógica de sugerencias para elegir solo un tipo de cada prenda.
Fallo-004	Al intentar filtrar por alguno de los campos, los resultados mostrados en la tabla no eran correctos.	Refinar la lógica del filtrado a la hora de hacer la llamada a la base de datos.
Fallo-005	Cuando el idioma principal no es inglés, los mensajes de error de todos los formularios aparecen sin traducir.	Etiquetar los mensajes de error como texto a traducir e insertarlos en los ficheros de traducción de los diferentes idiomas.
Fallo-006	Cuando se está identificado en el sistema y se cambia la URL manualmente para intentar acceder a la página de Login, esta aparece y permite un nuevo inicio de sesión.	Añadir "Guarda" que hace que si ya se está identificado y se intenta acceder a la página de inicio de sesión, la aplicación redirige automáticamente a la página "Home"

Tabla 30 Descripción de fallos y sus correspondientes soluciones en la versión Final.

8 Conclusiones

Una vez acabado este TFG, es oportuno recapitular y analizar los puntos previamente descritos con más profundidad con el fin de decidir si se han cumplido los objetivos iniciales, cómo ha sido el proceso de desarrollo del mismo y, por último, planificar el posible trabajo futuro.

8.1 Cumplimiento de objetivos iniciales

Si se comprueban los requisitos iniciales expuestos anteriormente puede verse que no solo se han cumplido los objetivos que se plantearon entonces, sino que, además a medida que han ido avanzando las iteraciones, se han ido refinando dichos requisitos y solucionando los problemas no previstos entonces. Personalmente considero que la aplicación desarrollada satisface las necesidades del usuario y además hace que la experiencia a lo largo de su uso sea agradable y divertida, lo cual, ayudará a su aceptación en el mercado.

Si se echa la vista atrás y se verifica el cumplimiento de, al menos, los requisitos funcionales:

- ◇ Registro de Usuario
- ◇ Identificación de Usuario
- ◇ Mostrar Inventario
- ◇ Mostrar Gastos
- ◇ Sugerir Compra
- ◇ Sugerir Vestimenta
- ◇ Registrar prendas
- ◇ Eliminar prendas
- ◇ Moldeabilidad

Todos y cada uno de ellos han sido implementados junto con otras funcionalidades como el cambio del tema principal de la aplicación para darle una apariencia mas agradable para cada tipo de usuario o como la posibilidad de cambiar el lenguaje y con ello hacer que la aplicación tenga un alcance internacional.

8.2 Experiencia personal

A lo largo del desarrollo de la aplicación he comprendido la importancia de todas las fases y aún más importante, he aprendido a valorar los diferentes roles que intervienen en el proceso de creación de software.

Aunque en la actualidad tengo más de cuatro años trabajando como desarrollador, no había sido capaz de entender que, sin una buena planificación y un estudio de todas las

necesidades del cliente, los desarrolladores, por muy buenos que sean, no cumplirían las expectativas. Creo que esta dependencia se trata de algo bidireccional, ya que por muy buena planificación que un producto tenga, si su implementación es deficiente, se volverían a dejar las expectativas incompletas.

También me gustaría remarcar que quizás no era consciente de que para cualquier producto de software, no solo se necesita una planificación o un desarrollo, además se requiere un buen proceso de marketing y un correcto estudio de mercado.

Por último, quiero destacar un apartado muchas veces menospreciado. Este es el proceso del diseño. Han sido muchas las veces que hemos oído la expresión: “Mientras funcione...” pero ¿Cuántas de las aplicaciones que usamos a diario no tienen un uso fácil y ligero, producto de muchas horas de diseño y la utilización de patrones probados en infinidad de ocasiones?

8.3 Líneas de trabajo futuro

Aunque los objetivos iniciales fueron concluidos con éxito, siempre existe la posibilidad de mejora. A lo largo de la implementación de este producto han ido surgiendo diferentes funcionalidades que serían un buen complemento y crearían una simbiosis perfecta con las capacidades ya desarrolladas.

- a) Añadir nuevas prendas y complementos.
- b) Mejorar la lógica de las sugerencias.
- c) Añadir un nuevo usuario Tienda.
- d) Capacidad de realizar compras desde la aplicación.
- e) Registro de las compras como entradas de nuestro inventario.

Si se añade un nuevo usuario “Tienda” capaz de poner sus productos a la venta, las sugerencias de compra podrían apuntar directamente a productos reales en el stock de las estas. Además, si fuera posible la compra de dichos productos a través de la aplicación, se registrarían y añadirían como nuevos elementos de nuestro inventario, nuevas entradas de nuestro historial de compras y se añadirían a nuestros gastos.

9 Bibliografía

<https://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>

Descripción: Teoría acerca de la historia de las bases de datos.

[Última visita]: 01/11/2019

<https://uniwebsidad.com/libros/javascript>

Descripción: Teoría acerca de Javascript.

[Última visita]: 01/11/2019

<https://es.wikipedia.org/>

Descripción: Consultas teóricas utilizadas para diferentes términos o tecnologías.

[Última visita]: 01/12/2019

<https://nodejs.org/es/>

Descripción: Web oficial de NodeJs.

[Última visita]: 10/01/2020

<https://www.waremarketing.com/es/blog/que-es-la-metodologia-agile-y-que-beneficios-tiene-para-tu-empresa.html>

Descripción: Teoría acerca de la metodología agile.

[Última visita]: 15/11/2019

<https://medium.com/@jw207427/how-scrum-help-turn-around-our-development-process-dac6ff7c700>

Descripción: Teoría acerca de la metodología SCRUM.

[Última visita]: 18/11/2019

<https://www.scrum.as/academy.php?show=1&chapter=10>

Descripción: Teoría acerca de la metodología SCRUM.

[Última visita]: 18/11/2019

https://es.wikiversity.org/wiki/Gesti%C3%B3n_de_riesgos_de_proyectos_software

Descripción: Teoría acerca de la gestión de riesgos.

[Última visita]: 10/12/2019

<https://www.apple.com/>

Descripción: Página oficial de Apple consultada para obtener los precios de un dispositivo.

[Última visita]: 10/12/2019

<https://programmaticponderings.com/tag/mean-stack/>

Descripción: Teoría acerca del Stack MEAN.

[Última visita]: 18:11/2019

<https://medium.com/@ariana10andrason/mean-stack-feature-and-architecture-e70409d18337>

Descripción: Teoría acerca del Stack MEAN.

[Última visita]: 12/12/2019

<https://swimlane.gitbook.io/ngx-charts/>

Descripción: Librería externa utilizada para los diagramas de gastos.

[Última visita]: 20/02/2020

<https://material.angular.io/>

Descripción: Página oficial de angular material.

[Última visita]: 22/03/2020

<https://github.com/SaturnTeam/saturn-datepicker>

Descripción: Github oficial de librería externa para la selección de fechas.

[Última visita]: 04/03/2020

<https://material.io/resources/icons/?style=baseline>

Descripción: Página oficial de material icons.

[Última visita]: 22/03/2020

<https://docs.github.com/en/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account>

Descripción: Manual para la creación de clave ssh en github.

[Última visita]: 22/03/2020

<https://rxjs-dev.firebaseapp.com/guide/observable>

Descripción: Manual de tipos observables.

[Última visita]: 10/04/2020

<https://github.com/ngx-translate/core>

Descripción: Github oficial de la librería externa para la internacionalización.

[Última visita]: 29/03/2020

<https://www.ingeniovirtual.com/tipos-de-graficos-y-diagramas-para-la-visualizacion-de-datos/>

Descripción: Teoría acerca de los diferentes tipos de diagramas para la muestra de datos.

[Última visita]: 20/02/2020

<https://angular.io/docs>

Descripción: Página oficial de angular, consultada para resolución de dudas acerca de esta tecnología.

[Última visita]: 04/06/2020

<https://typeitjs.com/>

Descripción: Página oficial de la librería externa de simulación de escritura por teclado.

[Última visita]: 27/06/2020

<https://pixabay.com/>

Descripción: Página de imágenes sin copyright.

[Última visita]: 22/03/2020

<https://tburleson-layouts-demos.firebaseio.com/#/docs>

Descripción: Demos de la utilización de flex layout para la distribución de elementos en HTML.

[Última visita]: 27/06/2020

<https://murhafsousli.github.io/ngx-gallery/#/>

Descripción: Página oficial de la librería para la visualización de imágenes en forma de carrusel.

[Última visita] 01/03/2020

<https://github.com/MurhafSousli/ngx-gallery/wiki/Gallery-API>

Descripción: Github oficial de la librería para la visualización de imágenes en forma de carrusel.

[Última visita] 01/03/2020

<https://alligator.io/js/introduction-localstorage-sessionstorage/>

Descripción: Teoría acerca de la utilización de Session storage en Angular.

[Última visita]: 14/04/2020

<https://docs.mongodb.com/drivers/node>

Descripción: Documentación del controlador para MongoDB

[Última visita]: 27/06/2020

<https://www.youtube.com/watch?v=mkn5n8Ws-HE&t=895s>

Descripción: Curso para el despliegue de aplicaciones en la plataforma Heroku.

[Última visita]: 31/01/2020

<https://www.kissclipart.com/>

Descripción: Página utilizada para la obtención de imágenes sin copyright.

[Última visita]: 22/03/2020

<https://merlosy.github.io/ngx-material-file-input/>

Descripción: Librería externa para la entrada de imágenes a la aplicación.

[Última visita]: 15/05/2020

<https://www.cleanpng.com/>

Descripción: Página para la obtención de imágenes sin copyright.

[Última visita]: 22/03/2020

<http://css3.bradshawenterprises.com/cfimg/>

Descripción: Manual para la creación de animaciones en CSS.

[Última visita]: 16/03/2020

https://tympanus.net/codrops/css_reference/opacity/

Descripción: Manual para la creación de animaciones en CSS.

[Última visita]: 16/03/2020

<https://ip-api.com/docs/api:json>

Descripción: API para la geolocalización a través de IP.

[Última visita]: 14/04/2020

<https://openweathermap.org/api>

Descripción: API para la obtención del tiempo.

[Última visita]: 27/06/2020

Anexo A: Manual de Usuario

La Aplicación está alojada en el link: <https://dressapp-project.herokuapp.com/>

Se intentarán cubrir todos los escenarios posibles de la manera más clara y concisa a través de imágenes con anotaciones.

1. Página de Identificación de Usuario

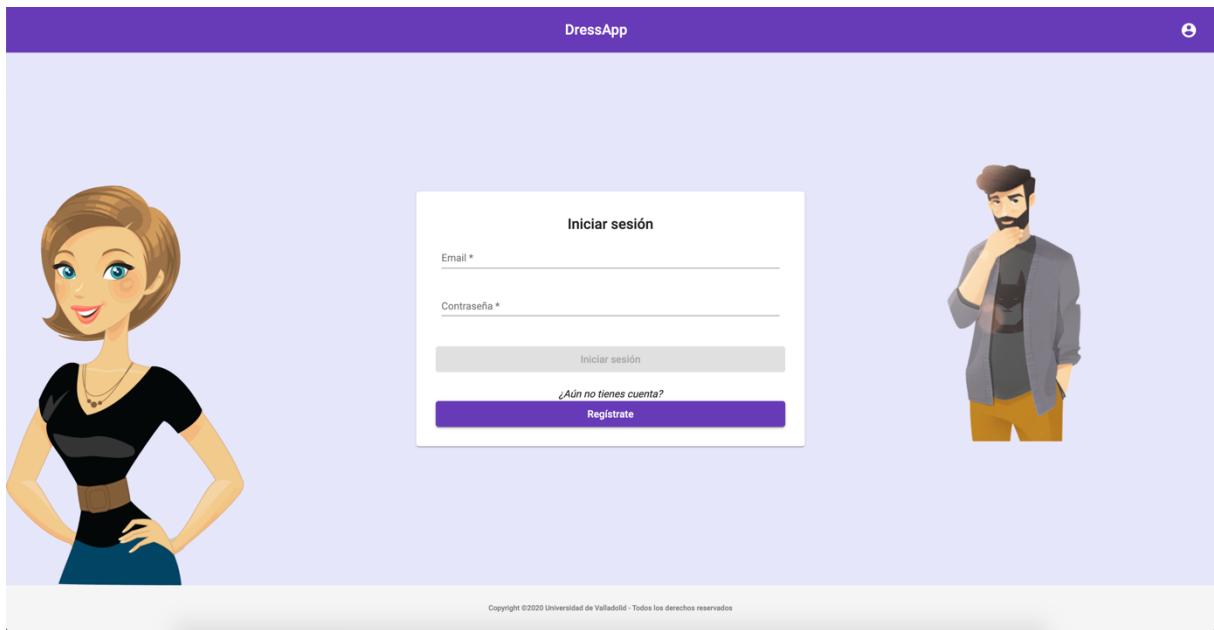


Figura 8 Interfaz de Identificación de Usuario

En el caso de estar registrado previamente, solo se deberán rellenar los campos “Email” y contraseña para acceder a la sección personal.

En caso de no estar registrado se necesita hacer click en el botón “Regístrate”.

2. Página de Registro de Usuario

Nombre * Apellidos *

Email *

Fecha de nacimiento *

Género *

Contraseña *

Confirmar contraseña *

Volver Enviar

Copyright ©2020 Universidad de Valladolid - Todos los derechos reservados

Figura 9 Interfaz de Registro de Usuario

Para un registro satisfactorio, hay que rellenar todos los campos y hacer click en el botón de “Enviar”.

En caso de fallo, el usuario será notificado a través de un snackbar en la parte inferior de la pantalla.

3. Página de Inventario personal

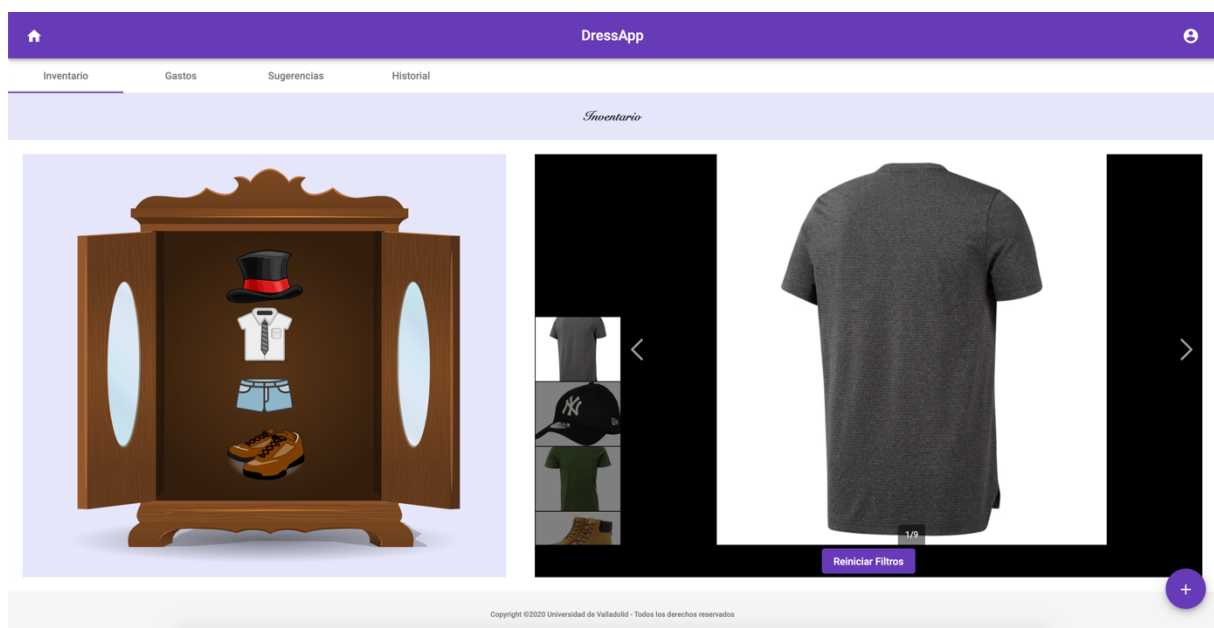


Figura 10 Interfaz de inventario personal

Una vez iniciada la sesión, el usuario será redirigido a su inventario personal.

Dentro de la página del inventario se distinguen varias secciones principales:



Figura 11 Filtros del inventario del Usuario

El armario situado en la parte izquierda de la pantalla representa los filtros de nuestro inventario. Al hacer click en cualquiera de los elementos del armario un menú desplegable aparecerá con las diferentes opciones para filtrar en correspondencia a la zona del cuerpo seleccionada.

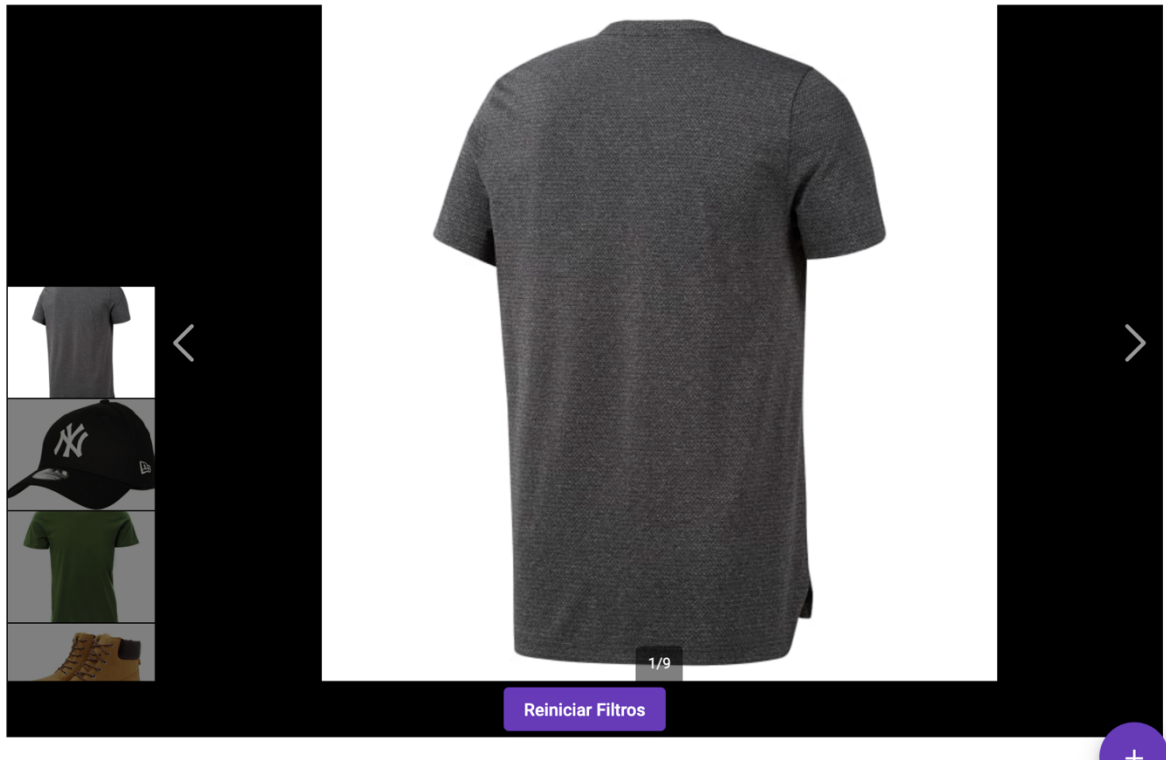


Figura 12 Carrusel de artículos de nuestro inventario.

La forma de desplazarse por el carrusel de imágenes es haciendo click en las flechas laterales o directamente seleccionado la imagen en miniatura del artículo deseado.

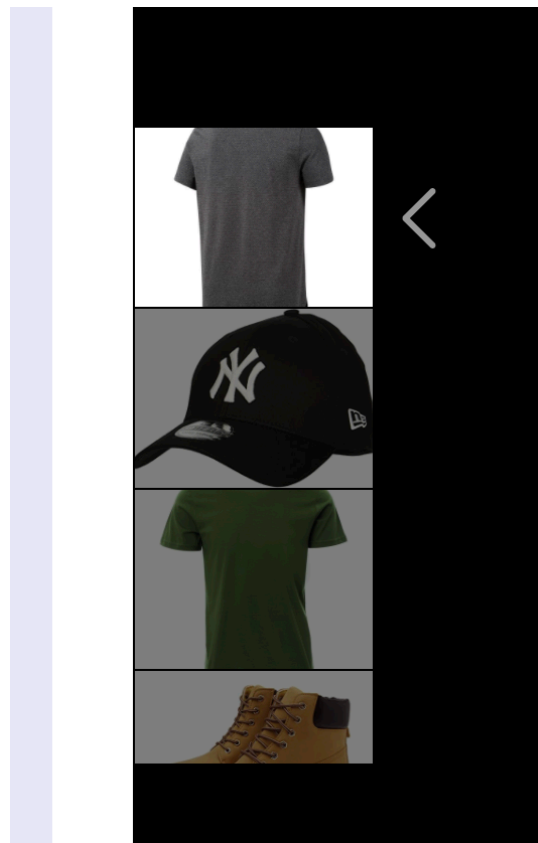


Figura 13 Fotos en miniatura en el carrusel de imágenes.

En la parte derecha de la pantalla se muestra un carrusel de imágenes con las prendas filtradas, o en su defecto con todas las prendas en posesión del usuario. La forma de reiniciar los filtros previamente marcados, es haciendo click en el botón “Reiniciar Filtros” situado bajo el carrusel.

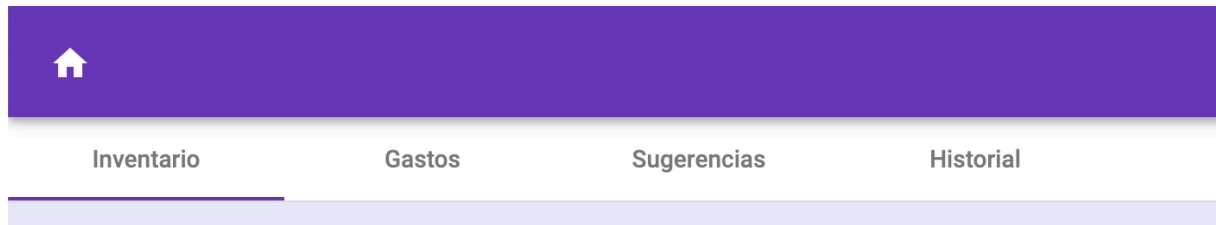


Figure 14 Pestañas de navegación.

En la parte superior de la ventana se encuentran una serie de pestañas que permitirán al usuario desplazarse por las diferentes secciones de la aplicación.

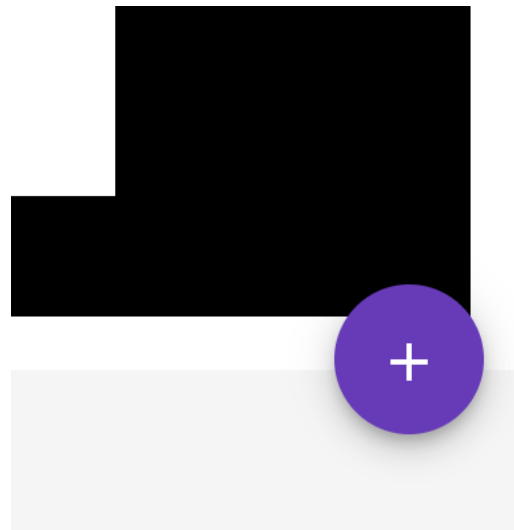


Figura 15 Botón para añadir nuevos artículos.

Por último, en la parte inferior derecha de la pantalla aparece un botón con el símbolo “+”. Este botón permitirá al usuario añadir nuevos artículos a su inventario. AL hacer click sobre este botón, un dialogo aparecerá con un formulario para añadir el nuevo artículo.

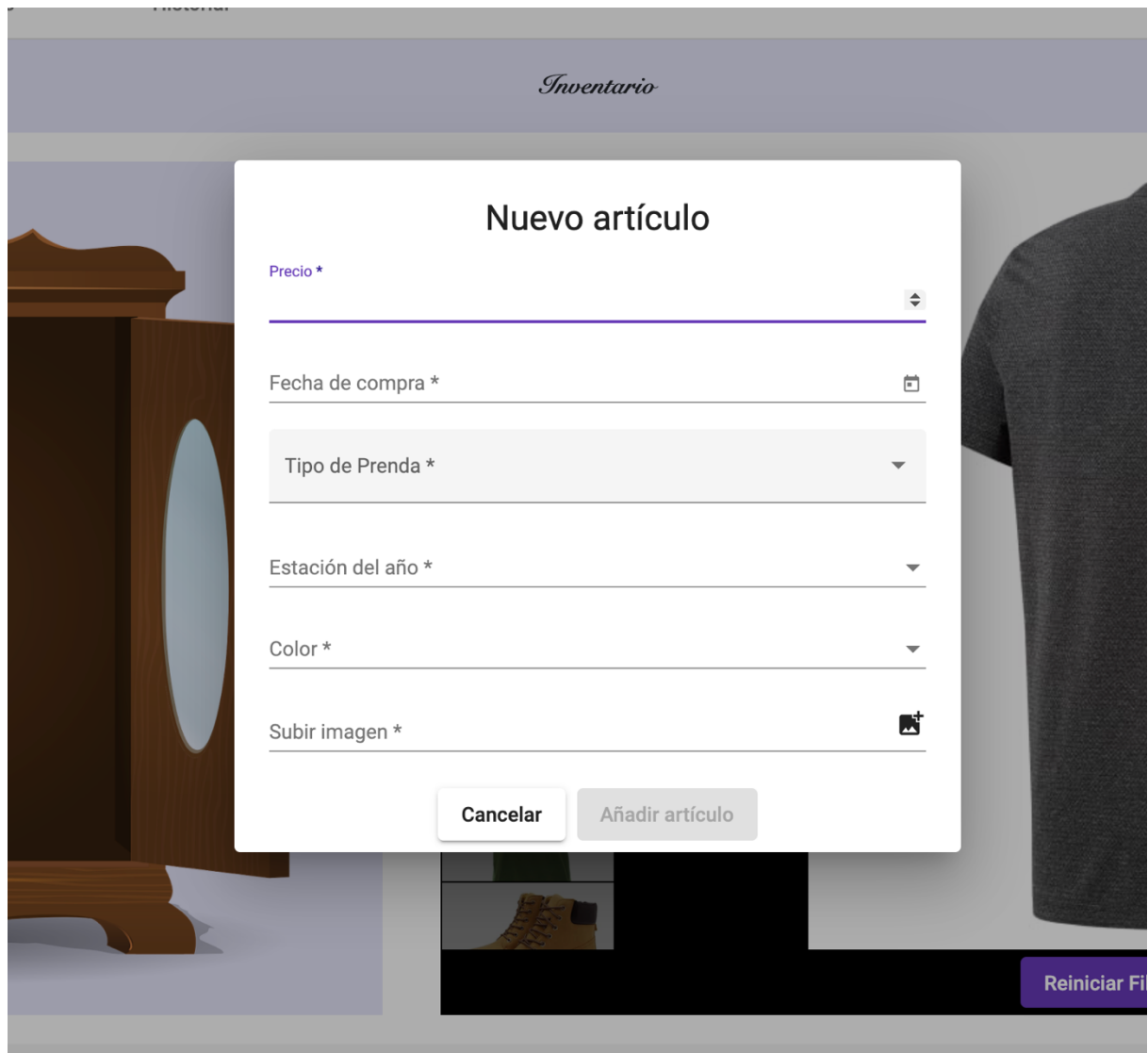


Figura 16 Formulario para añadir nuevos artículos.

Una vez rellenados todos los campos, el botón de “Añadir artículo” estará habilitado. El apartado de “Subir imagen” tiene la restricción de imágenes hasta 16MB.

4. Pestaña de Gastos

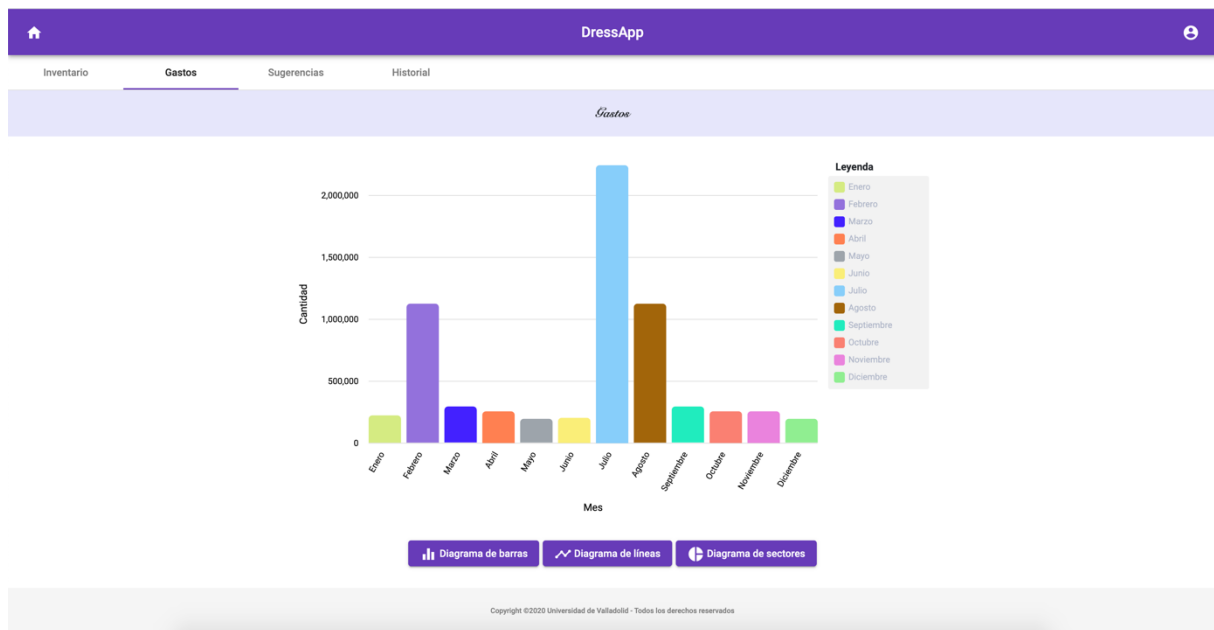


Figura 17 Interfaz de Gastos

En el apartado de gastos existen tres modos de visualización de los datos en forma de gráfico. El primero de ellos, mostrado en la figura 17, representa un diagrama de barras con los gastos producidos en el año actual.

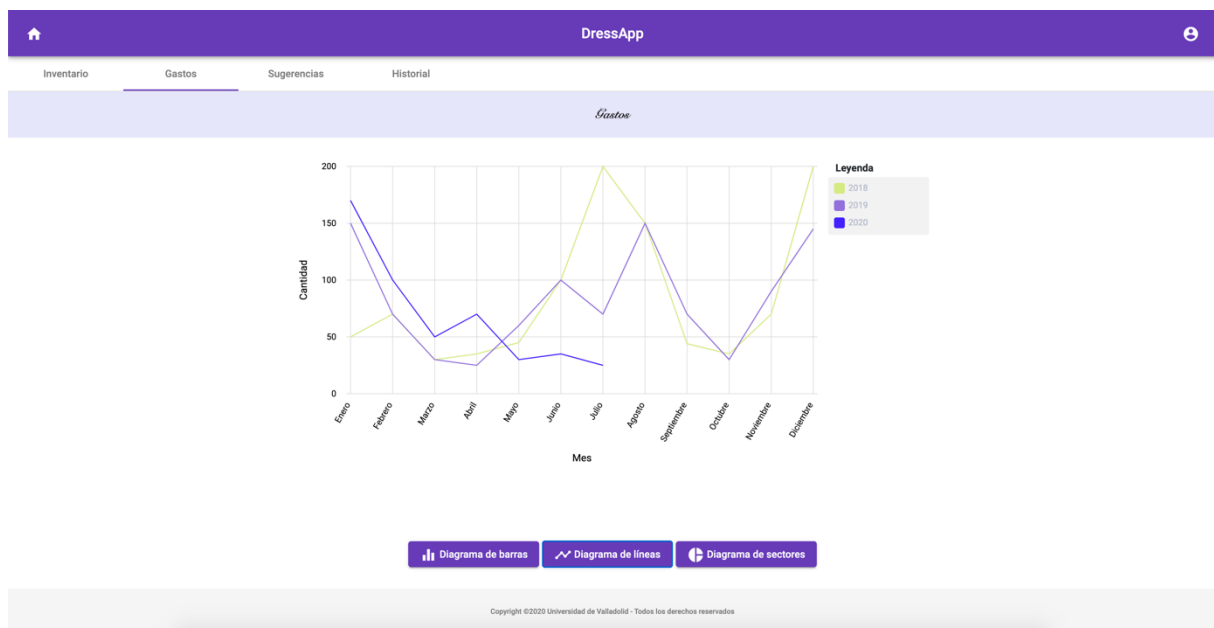


Figure 18 Diagrama de líneas.

El segundo de los diagramas es un diagrama de líneas que muestra la progresión de los gastos en los últimos años hasta el día de hoy. Este gráfico permite hacer una comparativa de los últimos años mes a mes.

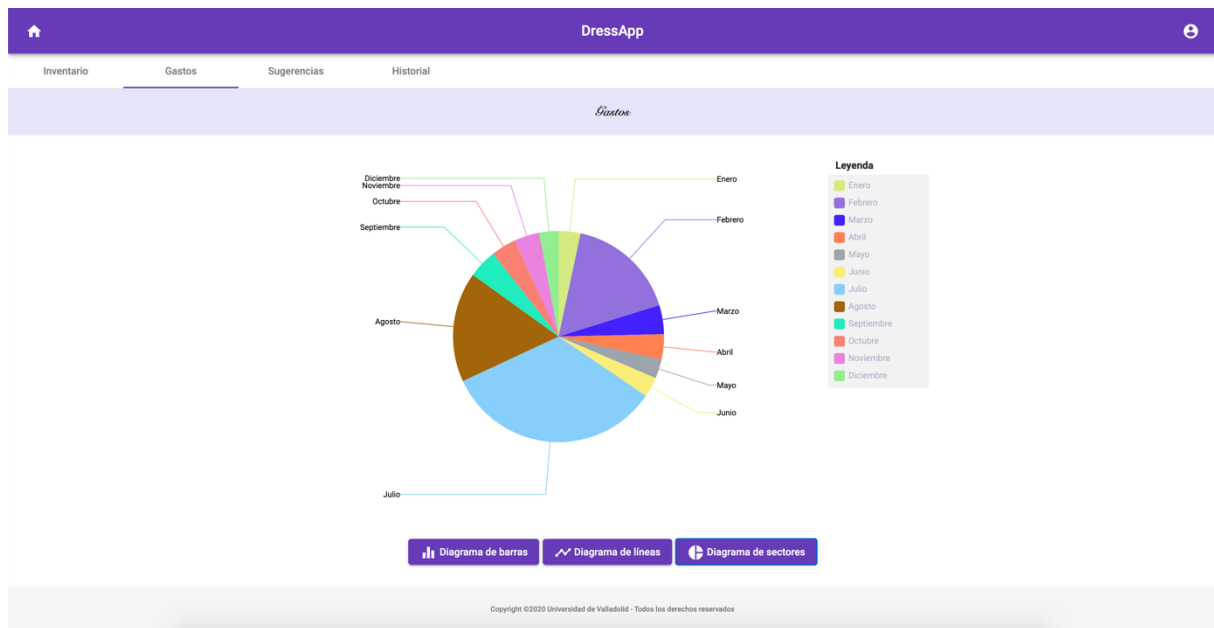


Figura 19 Diagrama de sectores.

Por último, el gráfico muestra un diagrama de sectores para hacer una comparación cuantitativa de los gastos en el año actual.

5. Pestaña de Sugerencias

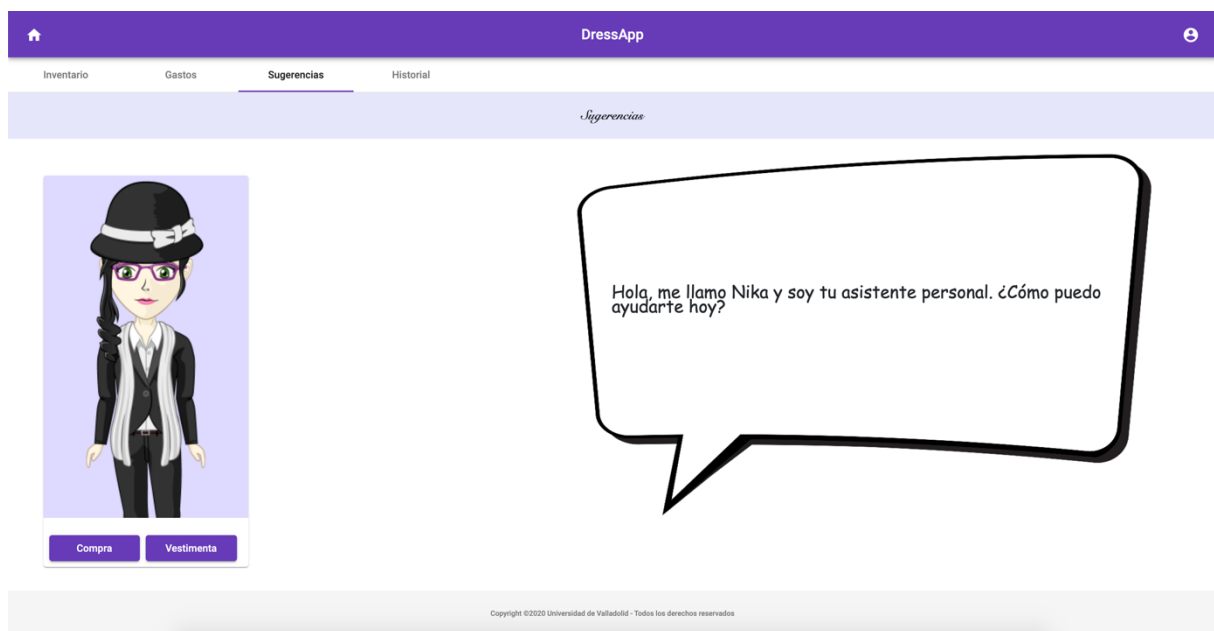


Figura 20 Pestaña de sugerencias.

La pestaña de Sugerencias cuenta con dos apartados principales, que son:

- El asistente personal
- El bocado de diálogos



Figura 21 Asistente personal.

Este asistente cuenta con dos funcionalidades básicas, las sugerencias de compra y las de vestimenta. Las sugerencias de Compra se activarán al hacer click en el botón “Compra”. Una vez hecho click, el asistente se comunicará con el usuario a través de mensajes dentro del bocadillo de diálogos, mostrado en la figura 22. Las sugerencias de Vestimenta funcionan de la misma manera y se activarán al hacer click en el botón “Vestimenta”.

Hola, me llamo Nika y soy tu asistente personal. ¿Cómo puedo ayudarte hoy?

Figura 22 Bocadillo de diálogos.

Las sugerencias de compra serán independientes unas de las otras y seguirán una lógica específica para los diferentes tipos de usuario. En el caso de las sugerencias de vestimenta, estas están relacionadas unas con las otras con un máximo de tres sugerencias por intento. Un ejemplo de sugerencia de compra es el siguiente:

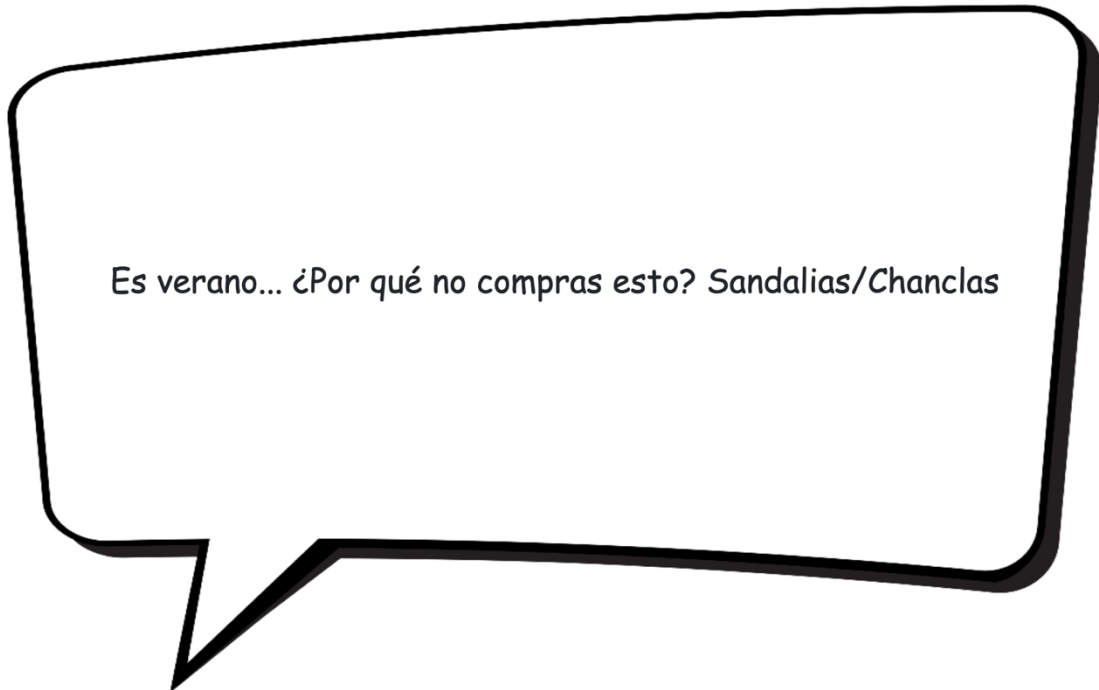


Figura 23 Ejemplo de sugerencia de compra.

Las sugerencias de vestimenta sin embargo, esperarán la respuesta del usuario de aceptar o rechazar la misma.



Figura 24 Ejemplo de sugerencia de vestimenta.

Si la sugerencia es aceptada, el asistente mostrara su conformidad y se registrarán las prendas como satisfactoriamente sugeridas. Esto ayudará al asistente a dar sugerencias más precisas con el paso del tiempo.

Sin embargo, si se rechazan 3 veces estas sugerencias, el asistente se dará por vencido y esperará a la petición de nuevas sugerencias.

6. Pestaña de Historial

Copyright ©2020 Universidad de Valladolid - Todos los derechos reservados

Figura 25 Pestaña del Historial.

En esta pestaña, se muestra un listado de todos los artículos registrados en la aplicación por el usuario con la posibilidad de filtrar a través de la sección de Filtros. Esta sección se despliega haciendo click en el botón “Mostrar filtros”. La lista cuenta con un elemento de búsqueda situado en la barra superior de la sección y los elementos filtrados o no aparecerán en la tabla paginada del inferior.

Figura 26 Sección de Filtros.

Estos filtros se activarán después de rellenar los campos que el usuario desee filtrar y haciendo click en “Aplicar Filtros” posteriormente.

7. Ajustes de cuenta

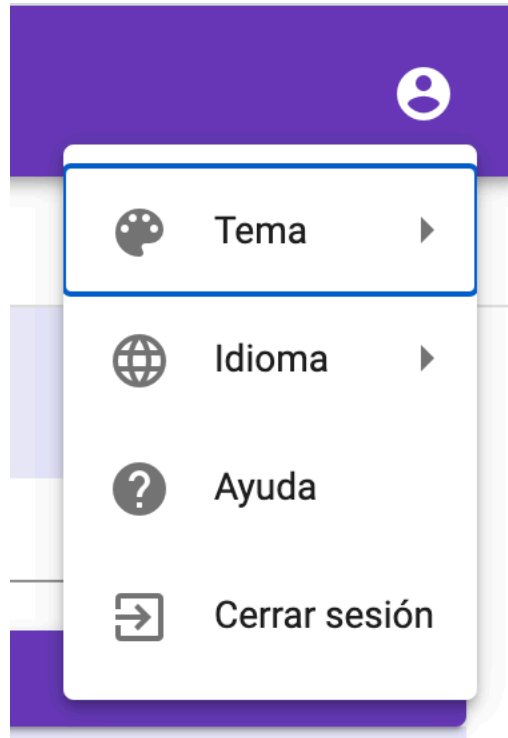


Figura 27 Ajustes de cuenta.

Estos ajustes aparecen en un menú desplegable situado en la parte superior derecha de la pantalla con forma de avatar. Dentro de este menú hay diferentes apartados:

- Tema
- Idioma
- Ayuda

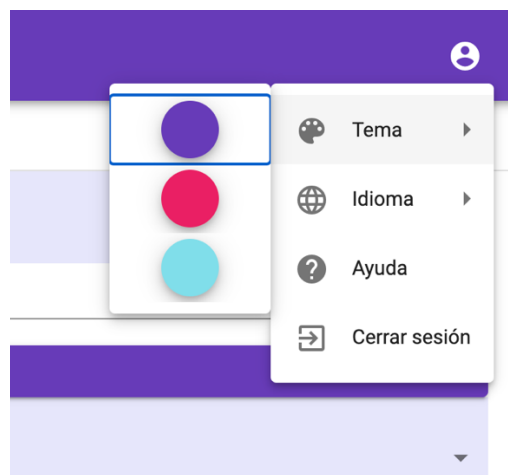


Figura 28 Temas de la aplicación.

Existen 3 temas principales que, de ser seleccionados, cambiarán por completo el estilo y diseño de las interfaces. Esta selección se almacenará como preferencias del usuario y será el estilo por defecto que verá dicho usuario en su próximo inicio de sesión.

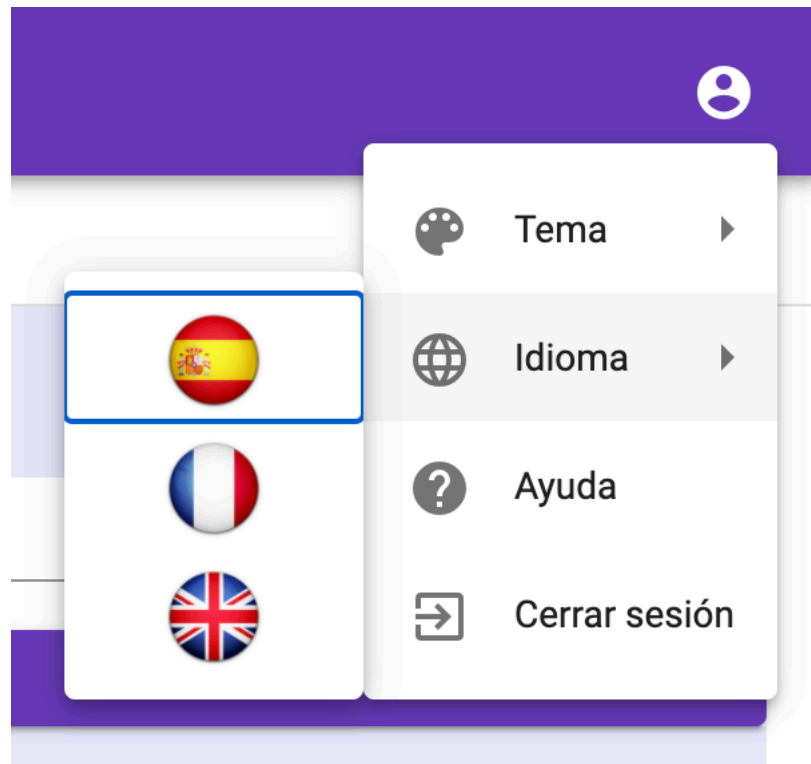


Figura 29 Apartado de idiomas.

Los diferentes idiomas reaccionan de la misma manera que los temas explicados anteriormente y también se guardarán como preferencias del usuario para futuros inicios de sesión. Al hacer click en cualquiera de las banderas de los idiomas, todos los textos de la aplicación cambiarán en tiempo de ejecución sin producir ningún tipo de retardo.

Si el usuario desea cerrar la sesión, solo tiene que seleccionar el ultimo apartado llamado “Cerrar Sesión”. Esto redireccionará la pagina a la url de inicio de sesión con el usuario ya desconectado.

8. Página de ayuda

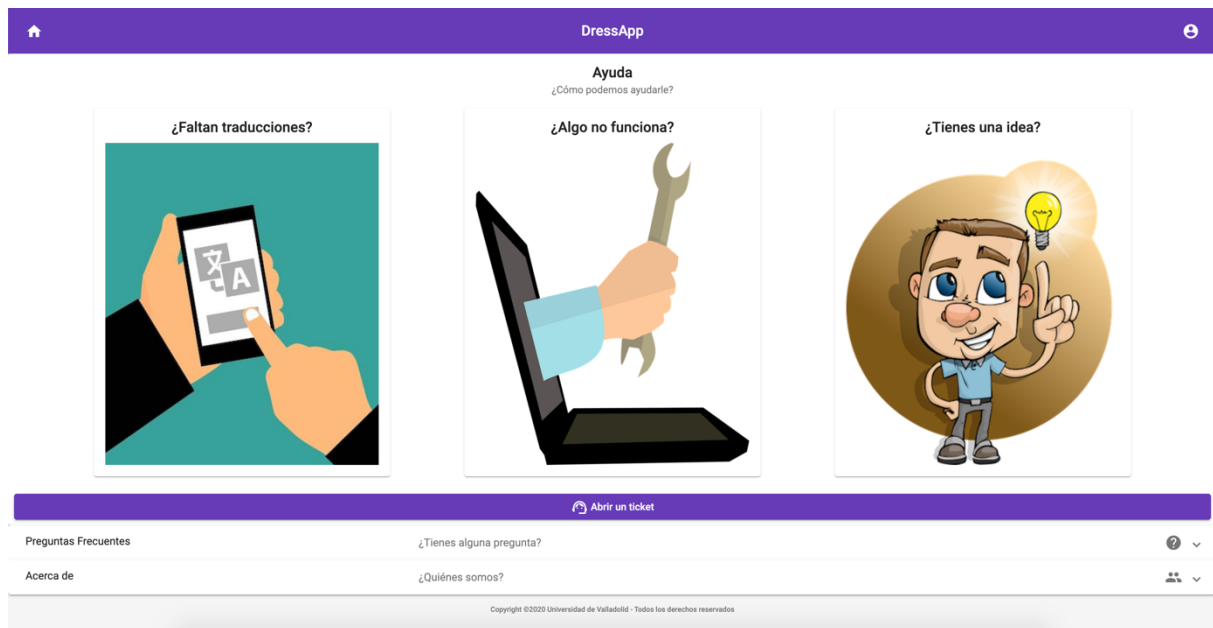


Figura 30 Página de ayuda.

La página de ayuda muestra 3 apartados principales que son:

- Preguntas frecuentes: Menú desplegable con algunas de las preguntas más frecuentes de los usuarios y su explicación.
- Acerca de: Menú desplegable con la información referente al desarrollador de la aplicación y a la Universidad de Valladolid
- Sección de contacto: botón que facilita al usuario la forma de contactar con el servicio técnico en caso de ser requerida su ayuda. Un pequeño formulario con dos apartados (Título y Descripción) deberá ser rellenado para posteriormente ser enviado.

Figura 31 Contacto con soporte técnico.