

**UNIVERSIDAD EUROPEA DE MADRID**

**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**GRADO EN INGENIERÍA INFORMÁTICA**

PROYECTO FIN DE GRADO

**EASYCOM: plataforma para comparar precios de productos**

**MARIO UCEDA YEVES**

**Dirigido por**

**[INGENIERO] BORJA MONSALVE PIQUERAS**

**CURSO 2022-2023**

**TÍTULO**: EASYCOM: plataforma para comparar precios de productos

**AUTOR**: MARIO UCEDA YEVES

**TITULACIÓN**: GRADO EN INGENIERÍA INFORMÁTICA

**DIRECTOR/ES DEL PROYECTO**: [INGENIERO] BORJA MONSALVE PIQUERAS

**FECHA**: MAYO de 2023

# RESUMEN

El comercio on-line se está imponiendo poco a poco al tradicional. El porcentaje de transacciones on-line realizadas en España durante el año 2021 fue del 24% 1 y objetivamente, estos datos se han disparado durante el pasado año por los efectos del COVID que ha potenciado el uso de las nuevas tecnologías en todos los ámbitos (el 80% de internautas reconocen comprar on-line habitualmente).

La principal consecuencia de esta tendencia, es la proliferación de marketplaces y sitios web que ofrecen los mismos productos o servicios. Desafortunadamente, este incremento en la oferta no se traduce, en la mayoría de ocasiones, en un beneficio para el comprador, debido a que la mayoría de buscadores incluyen algoritmos que muestran la información según criterios propios.

El objetivo inicial de este proyecto, es plantear un sistema multiplataforma para que los usuarios puedan realizar un seguimiento personalizado de productos en los marketplaces y tiendas más populares. Entre sus principales funcionalidades cabe destacar la posibilidad de obtener información adicional sobre los productos y monitorizar la evolución de precios, permitiendo a los usuarios tomar decisiones de compra aún más informadas.

**Palabras clave:** on-line, marketplaces, web, productos, precios

# ABSTRACT

Ecommerce is more relevant than ever. The evolution in Spain of this type of transactions has been impressive. In 2021, the 24% 1 of all the operations were made by ecommerce. Later, as a consecuence of the COVID, users and enterprises were forced to use new technologies (80% of internet users claim to buy using ecommerce).

The dramatic increase of ecommerce has raised the creation of a big number of marketplaces and websites offering the same products and services. Unfortunatelly, rising supply doesn’t meant a great opportunity for customer because web browsers use native algorithms focus, mainly, on their business growth.

The main goal for this project is developing a multi-platform system to allow users the traceability of products in the most relevant marketplaces and web sites. Some key features include the ability to obtain additional product information and monitor price trends, allowing users to make even more informed purchasing decisions.

**Keywords:** on-line, marketplaces, web, products, price

**AGRADECIMIENTOS**

En ocasiones se incluye este apartado para agradecer a aquellos que han ofrecido su ayuda en el desarrollo del trabajo, ya sea técnica o de otro tipo.

**Cita - frase célebre / Dedicatoria**

Esta página es del todo opcional, pero resulta una muy buena forma de presentar el trabajo académico más importante de todo el grado.

# TABLA RESUMEN

Tabla 1: Resumen

|  |  |
| --- | --- |
|  | **DATOS** |
| **Nombre y apellidos:** | Mario Uceda Yeves |
| **Título del proyecto:** | EASYCOM: plataforma para comparar precios de productos |
| **Directores del proyecto:** | Borja Monsalve Piqueras |
| **El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:** | No |
| **El proyecto ha implementado un producto:** | Si |
| **El proyecto ha consistido en el desarrollo de una investigación o innovación:** | NO |
| **Objetivo general del proyecto:** | Crear una plataforma para comparar precios de productos a través del código de barras. |

**Índice**

[RESUMEN 3](#_Toc135734364)

[ABSTRACT 4](#_Toc135734365)

[TABLA RESUMEN 7](#_Toc135734366)

[Capítulo 1. RESUMEN DEL PROYECTO 13](#_Toc135734367)

[1.1 Contexto y justificación 13](#_Toc135734368)

[1.2 Planteamiento del problema 13](#_Toc135734369)

[1.3 Objetivos del proyecto 13](#_Toc135734370)

[1.4 Resultados obtenidos 13](#_Toc135734371)

[1.5 Estructura de la memoria 13](#_Toc135734372)

[Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE 15](#_Toc135734373)

[2.1 Estado del arte 15](#_Toc135734374)

[2.2 Contexto y justificación 15](#_Toc135734375)

[2.3 Planteamiento del problema 15](#_Toc135734376)

[Capítulo 3. OBJETIVOS 17](#_Toc135734377)

[3.1 Objetivos generales 17](#_Toc135734378)

[3.2 Objetivos específicos 17](#_Toc135734379)

[3.3 Beneficios del proyecto 17](#_Toc135734380)

[Capítulo 4. DESARROLLO DEL PROYECTO 18](#_Toc135734381)

[4.1 Planificación del proyecto 18](#_Toc135734382)

[4.2 Descripción de la solución, metodologías y herramientas empleadas 19](#_Toc135734383)

[4.2.1 Incremento 1 21](#_Toc135734384)

[4.2.2 Incremento 2 21](#_Toc135734385)

[4.2.3 Incremento 3 27](#_Toc135734386)

[4.2.4 Incremento 4 31](#_Toc135734387)

[4.3 Recursos requeridos 34](#_Toc135734388)

[4.4 Presupuesto 35](#_Toc135734389)

[4.5 Viabilidad 36](#_Toc135734390)

[4.6 Resultados del proyecto 37](#_Toc135734391)

[4.6.1 Pruebas y validación 38](#_Toc135734392)

[Capítulo 5. MANUAL DE INSTALACIÓN Y USUARIO 44](#_Toc135734393)

[5.1 Manual de instalación 44](#_Toc135734394)

[5.1.1 Crear base de datos 44](#_Toc135734395)

[5.1.2 Migración de la base de datos 45](#_Toc135734396)

[5.1.3 Iniciar servidor web 46](#_Toc135734397)

[5.1.4 Configurar app 46](#_Toc135734398)

[5.2 Manual de usuario 47](#_Toc135734399)

[Capítulo 6. CONCLUSIONES 48](#_Toc135734400)

[6.1 Conclusiones del trabajo 48](#_Toc135734401)

[6.2 Conclusiones personales 48](#_Toc135734402)

[Capítulo 7. FUTURAS LÍNEAS DE TRABAJO 49](#_Toc135734403)

[Capítulo 8. REFERENCIAS 50](#_Toc135734404)

[Capítulo 9. ANEXOS 51](#_Toc135734405)

**Índice de Figuras**

[Ilustración 1: Cronograma anteproyecto 18](#_Toc135666849)

[Ilustración 2: Cronograma final 18](#_Toc135666850)

[Ilustración 3: Estadística commits Github 19](#_Toc135666851)

[Ilustración 4: Esquema proyecto 19](#_Toc135666852)

[Ilustración 5: Diagrama casos de uso app 20](#_Toc135666853)

[Ilustración 6: Diagrama casos de uso web 20](#_Toc135666854)

[Ilustración 7: Modelo realacional de la BBDD 21](#_Toc135666855)

[Ilustración 8: Estructura Frontend 22](#_Toc135666856)

[Ilustración 9: Navbar crear sección 22](#_Toc135666857)

[Ilustración 10: Iniciar sección en una vista 23](#_Toc135666858)

[Ilustración 11: Cerrar sección en una vista 23](#_Toc135666859)

[Ilustración 12: Gráfica de precios 23](#_Toc135666860)

[Ilustración 13: Lista de migraciones 24](#_Toc135666861)

[Ilustración 14: Ejemplo de migración 24](#_Toc135666862)

[Ilustración 15: Lista de modelos 25](#_Toc135666863)

[Ilustración 16: Ejemplo de un modelo 25](#_Toc135666864)

[Ilustración 17: Lista de controladores 25](#_Toc135666865)

[Ilustración 18: Lista de ficheros de endpoints 26](#_Toc135666866)

[Ilustración 19: Ejemplo fichero de endpoints 26](#_Toc135666867)

[Ilustración 20: Lista de vistas 27](#_Toc135666868)

[Ilustración 21: Lista de controladores 27](#_Toc135666869)

[Ilustración 22: Crear vista desde controlador 28](#_Toc135666870)

[Ilustración 23: Función con corrutinas para iniciar sesión 28](#_Toc135666871)

[Ilustración 24: Ejemplo de peticiones al servidor 29](#_Toc135666872)

[Ilustración 25: Lista de modelos y respuestas de peticiones 29](#_Toc135666873)

[Ilustración 26: Ejemplo de respuesta del servidor 30](#_Toc135666874)

[Ilustración 27: Configuración e iniciar el escáner 30](#_Toc135666875)

[Ilustración 28: Recuperar datos del escáner 30](#_Toc135666876)

[Ilustración 29: Lista de ficheros para web scraping 31](#_Toc135666877)

[Ilustración 30: Esquema de clase para hacer scraping a una tienda 31](#_Toc135666878)

[Ilustración 31: Fichero que lanza peticiones a las tiendas 32](#_Toc135666879)

[Ilustración 32: Fichero que busca un producto nuevo 32](#_Toc135666880)

[Ilustración 33: Clase para actualizar los precios de los productos 33](#_Toc135666881)

[Ilustración 34: Crontab para ejecutar el fichero que actualiza los precios 33](#_Toc135666882)

[Ilustración 35: Coste servidor 35](#_Toc135666883)

**Índice de Tablas**

[Tabla 1: Resumen 7](#_Toc135666884)

[Tabla 2: Presupuesto 35](#_Toc135666885)

[Tabla 3: Viabilidad 36](#_Toc135666886)

[Tabla 4: Pruebas 37](#_Toc135666887)

# RESUMEN DEL PROYECTO

## Contexto y justificación

En la actualidad, el comercio electrónico está posicionándose como un elemento diferenciador en la estrategia de marketing y ventas de todas las compañías con independencia de su tamaño. Esta tendencia seguirá en los próximos años hasta convertirse en el principal canal de ventas. La paradoja de este modelo es que la sobreoferta de productos y servicios, no repercute en un claro beneficio para los usuarios por la cantidad abrumadora de datos que se manejan y la dudosa calidad de los mismos (resultado de algoritmos externos).

## Planteamiento del problema

Por desgracia, la inmensa cantidad de marketplaces y tiendas que venden los mismos productos provoca que en muchas ocasiones estemos pagando de más al realizar nuestras compras.

Para solucionar este problema, se ha creado Easycom, una plataforma formada por una página web y una aplicación móvil para dispositivos Android que permite obtener información sobre los productos y sus precios de una forma rápida e intuitiva a sus usuarios.

## Objetivos del proyecto

Los principales objetivos del proyecto son:

1. Crear página web y aplicación móvil (Android) que permita interactuar con los usuarios.
2. Integración de servicio de lectura por código de barras y/o QR.
3. Crear un servidor web para gestión de la lógica de negocios. El principal reto será la obtención de datos de páginas web externas usando ‘web scraping’.
4. Crear un servidor de base de datos. Base de datos Relacional (MVC).

## Resultados obtenidos

El proyecto final se ajusta al diseño técnico y requerimientos planteados. De esta forma tenemos una solución accesible mediante página web (http://easycom.sytes.net/) o app para móviles (Easycom) que permite a los usuarios seleccionar mediante lector de código de barras cualquier producto, ejecutar búsquedas en tiempo real en las marketplaces predefinidas, realizar seguimiento de precios por producto o gestión de favoritos.

El principal reto ha sido la obtención de datos externos. Dentro de una misma página hay diferentes formas de presentación de la información, o incluso redireccionamientos a otras páginas web. Otro problema que se ha identificado es el bloqueo de acceso por los firewalls al realizar un gran número de accesos.

## Estructura de la memoria

Describe muy brevemente los capítulos y su contenido, con tus propias palabras.

# ANTECEDENTES / ESTADO DEL ARTE

## Estado del arte

La idea de crear un comparador de precios mediante un escáner en el móvil surge de la necesidad de facilitar el proceso de compra para los consumidores. Actualmente, la mayoría de las personas utilizan sus dispositivos móviles para buscar información y realizar compras en línea, por lo que contar con una herramienta de este tipo en el móvil, resulta sumamente conveniente.

## Contexto y justificación

Desde la pandemia provocada por el Covid-19, nos encontramos en un momento económico complicado, en el que manda la inflación y la subida de los precios de los productos. Debido a esto, a día de hoy es muy importante saber en qué lugar se puede encontrar el producto deseado, al menor precio posible.

Easycom es una plataforma que ofrece una aplicación para dispositivos móviles que facilita a los usuarios la posibilidad de escanear el código de barras de un artículo mientras se encuentran en una tienda física. Esto les permite comparar instantáneamente el precio de dicho producto con los precios disponibles en otras tiendas, ya sea en línea o en tiendas físicas.

Además, Easycom permite a los usuarios obtener información adicional sobre los productos, como la descripción y especificaciones técnicas. Esto les brinda la oportunidad de tomar decisiones de compra aún más informadas, al tener acceso a una amplia gama de información sobre el producto antes de comprarlo.

La web ‘<http://easycom.sytes.net/>’ permite a los usuarios encontrar los productos que deseen utilizando un buscador. Además, ofrece un seguimiento más exhaustivo sobre los productos, ya que permite la visualización gráficos de precios para determinar si es el momento ideal para comprar el producto.

Easycom posee un enorme potencial para ayudar a los consumidores a tomar decisiones de compra informadas, ahorrándoles tiempo y dinero en sus adquisiciones. En consecuencia, se puede argumentar que existe una justificación sólida para el desarrollo de este proyecto, ya que busca mejorar la experiencia de compra de los consumidores y brindarles el apoyo necesario para tomar decisiones informadas acerca de sus compras.

## Planteamiento del problema

La inmensa cantidad de marketplaces y tiendas que ofrecen los mismos productos provoca que, en numerosas ocasiones, los consumidores terminen pagando de más al realizar sus compras. Para solucionar este problema, se ha creado Easycom, una plataforma basada en una app para el móvil y en una web, que permita la gestión y seguimiento de productos.

Además, incluirá un sistema de notificaciones para alertar de los cambios de precios de los productos. Todo esto funcionará gracias al sistema de gestión de cuentas que permitirá al usuario gestionar los productos desde el dispositivo que desee.

Actualmente existen varias aplicaciones y sitios web que permiten comparar precios, pero pocos utilizan un escáner en el teléfono para leer los códigos de barras de los productos y ofrecer información en tiempo real. La más conocida es Idealo [1], un comparador online el cual tiene una app móvil que permite escanear el código de barras del producto. Tras testear y ver reseñas de otros usuarios, se han reconocido algunos aspectos en los que esta plataforma flaquea, como precios que no son actualizados en tiempo real y no siempre coinciden con los precios en las tiendas, la recomendación de tiendas de dudosa fiabilidad y el desmedido consumo en segundo plano de la app móvil. Estos puntos negativos son suficientemente importantes como para utilizarlos como fortalezas de este proyecto.

# OBJETIVOS

## Objetivos generales

El objetivo general del presente proyecto consiste en crear una plataforma que permita a los usuarios realizar una comparación de precios por producto en páginas webs predefinidas. La carga de productos se realizará mediante el uso de un scanner de código de barras y/o un buscador.

Adicionalmente, la plataforma ofrece la posibilidad de identificar productos como favoritos, de esta forma el sistema realizará un seguimiento automático de los mismos, y proporcionará información actualizada relativa a la evolución de su precio.

## Objetivos específicos

Los objetivos específicos del proyecto son:

* [OE1]: Permitir al usuario obtener información de un producto escaneando su código de barras (app).
* [OE2]: Permitir al usuario obtener información de un producto utilizando un buscador (web).
* [OE3]: Extraer información en internet de los productos para mostrar los datos al usuario.
* [OE4]: Automatizar la extracción de datos para generar un historial de precios.
* [OE5]: Permitir al usuario acceder a su historial de búsquedas desde ambas plataformas (web y app)
* [OE6]: Mostrar al usuario una gráfica con histórico de precios de un producto (web).
* [OE7]: Permitir al usuario tener un control de productos favoritos.
* [OE8]: Crear un sistema de notificaciones/alertas del precio de productos.

## Beneficios del proyecto

Incluye una descripción del beneficio que aporta tu proyecto, en relación a los objetivos.

# DESARROLLO DEL PROYECTO

## Planificación del proyecto

En el anteproyecto se realizó el siguiente cronograma en el que se estimaban unos plazos para las diferentes tareas a realizar.

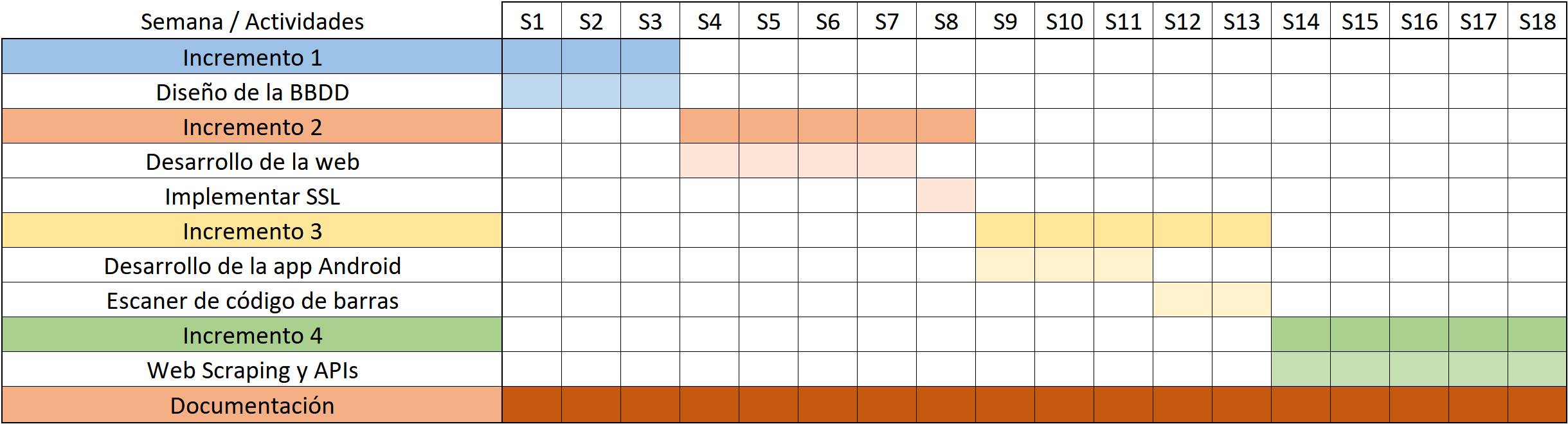


Ilustración 1: Cronograma anteproyecto

El desarrollo del proyecto ha variado ligeramente del estimado durante el anteproyecto, por ello hay algunas diferencias entre el cronograma estimado y el real.

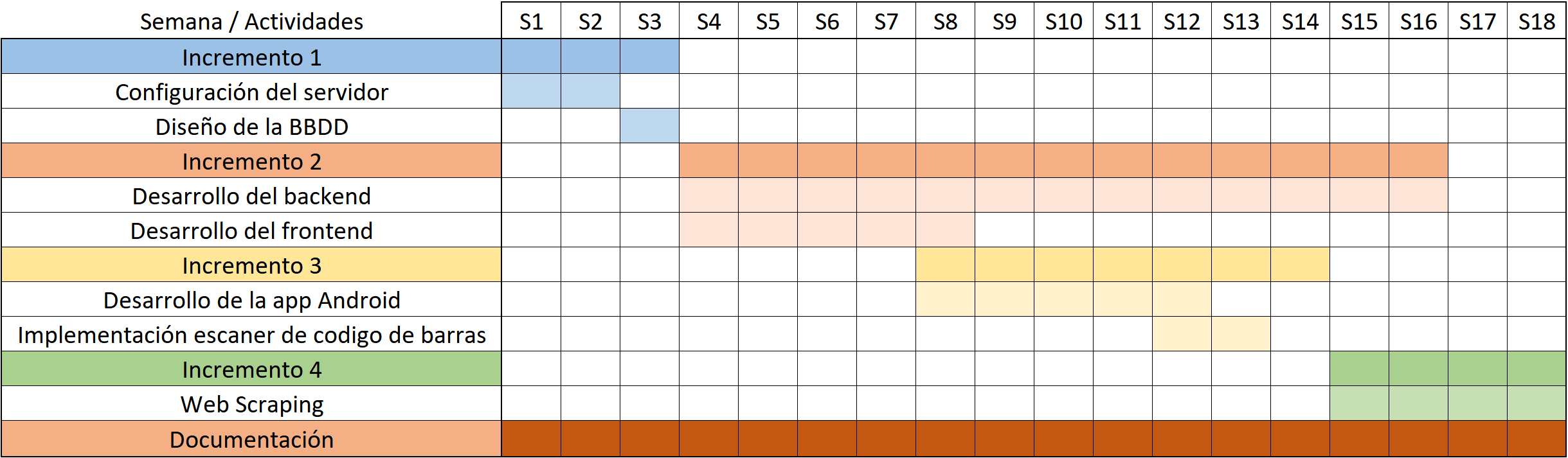


Ilustración 2: Cronograma final

La diferencia más notable en el cronograma real es el aumento de la duración del segundo incremento, más concretamente del desarrollo backend, lo cual se explicará en los siguientes apartados.

Para la gestión del código se creó un repositorio en Github , el cual me permite ver el número de comits a la rama principal del proyecto he realizado a lo largo del tiempo, justificando así el desarrollo continuo del proyecto.

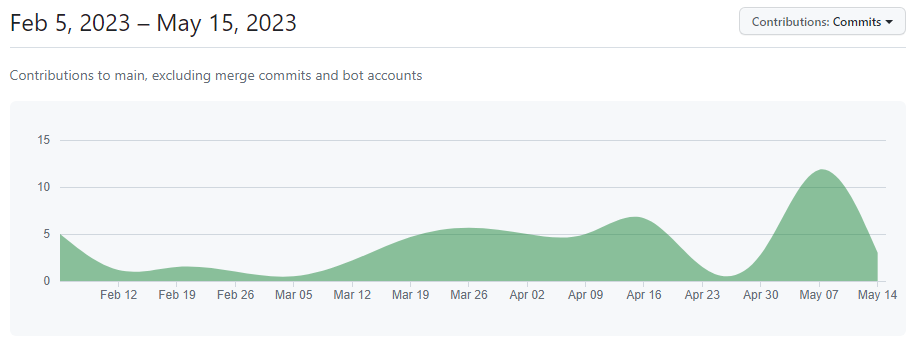


Ilustración 3: Estadística commits Github

## Descripción de la solución, metodologías y herramientas empleadas

Para llevar a cabo este proyecto, se realiza un esquema con el objetivo de entender cómo va a funcionar y establecer las diferentes áreas de las que se va a componer la plataforma.

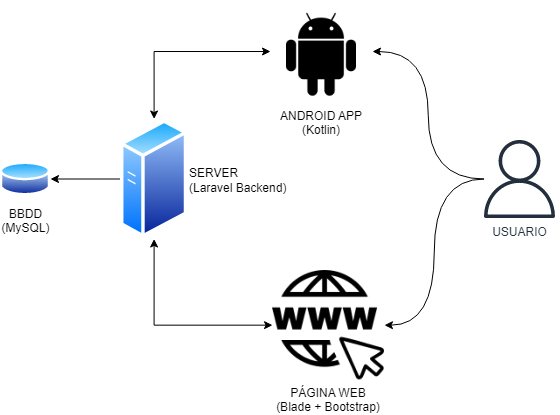


Ilustración 4: Esquema proyecto

El usuario final de la plataforma dispone de dos formas de acceder a ella, una aplicación Android y una página web, ambas conectadas a un servidor central que es el encargado de dar servicio y de la gestión del almacenamiento.

La página web y la aplicación móvil disponen de diferentes funcionalidades, por ello se debe realizar un diagrama de caso de uso diferente para cada una.

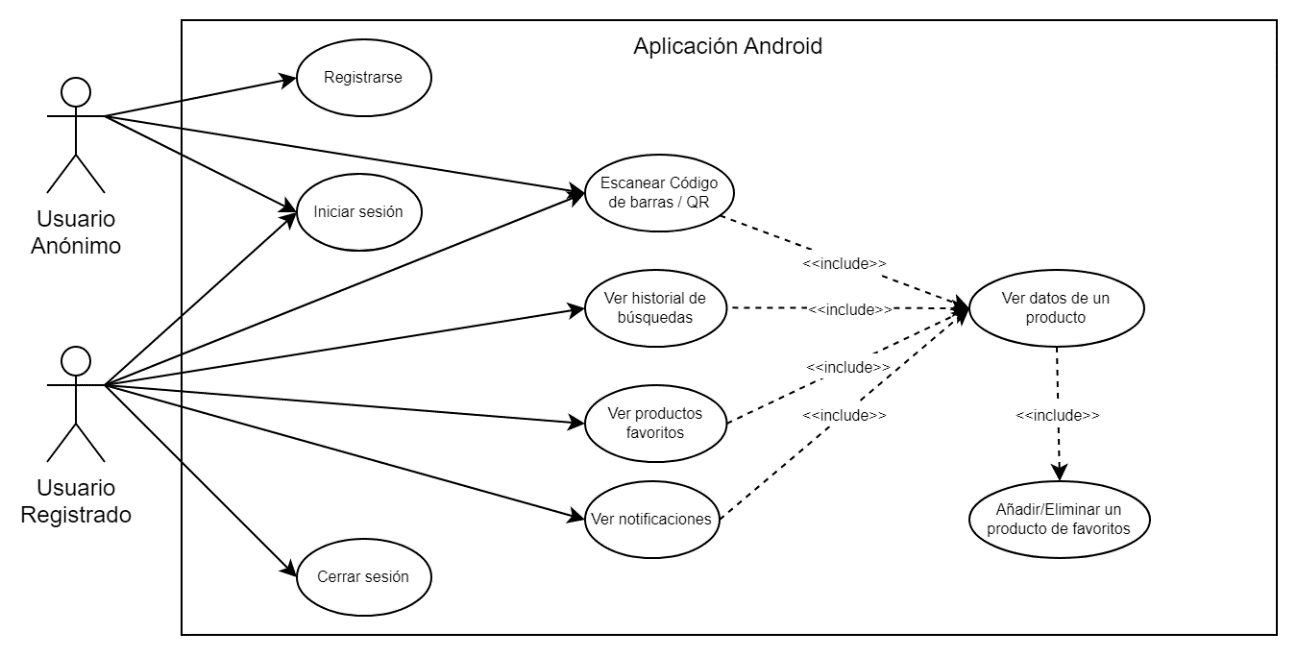


Ilustración 5: Diagrama casos de uso app

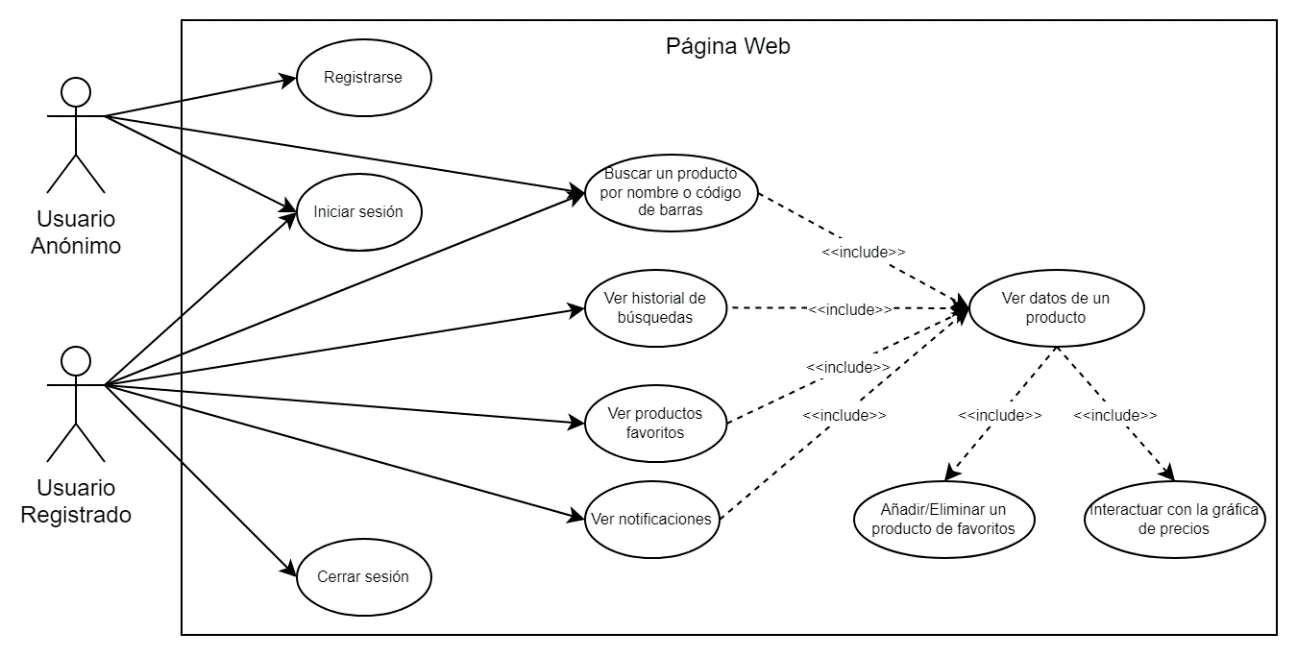


Ilustración 6: Diagrama casos de uso web

Para el desarrollo de este proyecto se ha utilizado una metodología de trabajo incremental. Esta metodología permite trabajar en funciones concretas de la plataforma e ir juntándolas según estén completas.

Al iniciar el desarrollo del proyecto se divide el trabajo en cuatro incrementos diferentes:

### Incremento 1

En el primer incremento se realiza la configuración del servidor y el diseño de la base de datos. Antes de configurar el servidor se establece la arquitectura del proyecto.

Una vez conocida la arquitectura del proyecto se empieza a configurar el servidor. El servidor que da vida a este proyecto es una Raspberry Pi 3, con sistema operativo Raspbian GNU/Linux 11 (bullseye). En ella se ha instalado y configurado:

* Apache/2.4.54 (Raspbian)
* Laravel Framework 10.5.1
* DNS No-Ip
* Python 3.9.2
* Mariadb 15.1

Tras la configuración del servidor, el siguiente objetivo de este incremento es el diseño de la base de datos:

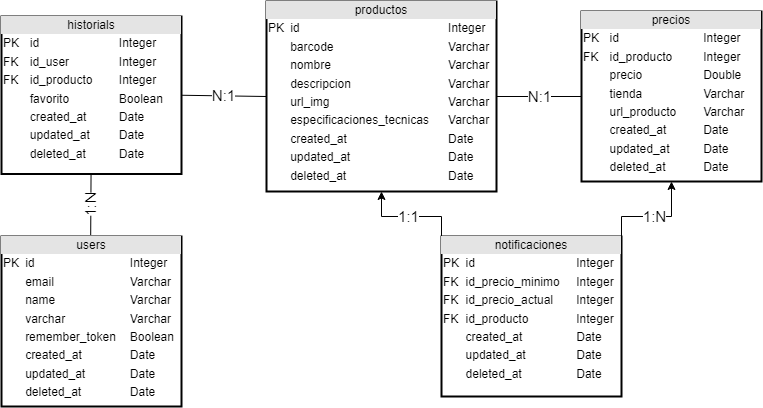


Ilustración 7: Modelo realacional de la BBDD

Esta es la arquitectura de la base de datos utilizada en el proyecto.

### Incremento 2

Este es el incremento con mayor extensión en el tiempo. Este incremento se centra en desarrollar completamente en Laravel, y se puede dividir en dos partes, la parte de frontend y la parte de backend. Esto es posible gracias a que Laravel implementa la arquitectura MVC (modelo, vista, controlador),

#### Frontend

El apartado de frontend se ha desarrollado utilizando Blade, un gestor de vistas incluido dentro de Laravel. El frontend se ha desarrollado utilizando la siguiente estructura:

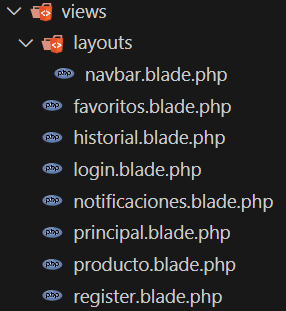


Ilustración 8: Estructura Frontend

El fichero ‘navbar.blade.php’ es elemento raíz de todas las vistas, este incluye la barra de navegación y debajo se crea una sección, a la que se llama desde las demás vista para incluir su código.

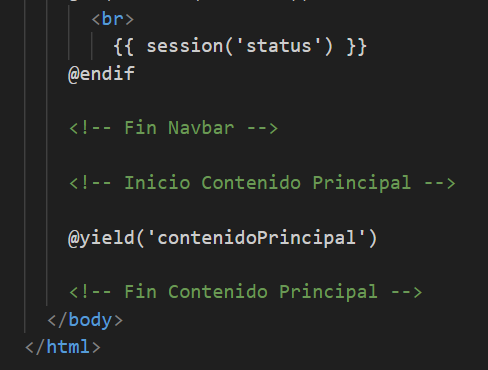


Ilustración 9: Navbar crear sección

En esta ilustración[] se puede ver como al finalizar la barra de navegación se crea una sección llamada ‘contenidoPrincipal’ y después finaliza el html.

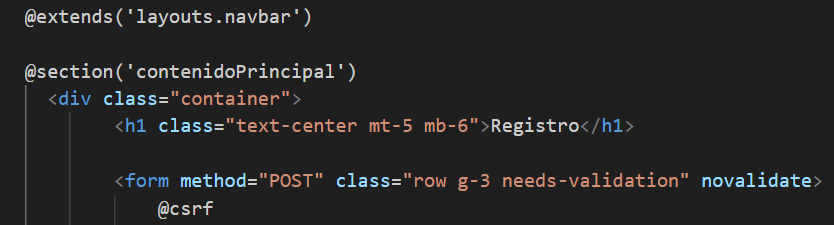


Ilustración 10: Iniciar sección en una vista

En esta ilustración[] podemos ver el inicio del fichero ‘register.blade.php’ y como primero se extiende la clase ‘navbar.blade.php’ y posteriormente se abre la sección ‘contenidoPrincipal’.

Al terminar el código propio de la clase se debe cerrar la sección.

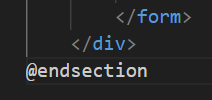


Ilustración 11: Cerrar sección en una vista

Se ha hecho uso de Bootstrap 5 para el diseño de la web, de FontAwesome 5.15 para algunos iconos y del fichero ‘estilos.css’ para personalizar la estética de la web.

Por último, en la parte de frontend, se ha utilizado la librería Plotly 2.20 para la realización de gráficas de precios.

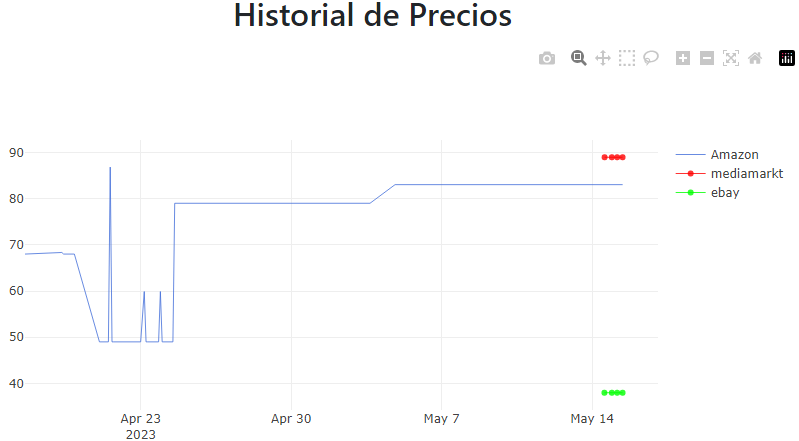


Ilustración 12: Gráfica de precios

#### Backend

El backend del proyecto se ha desarrollado en Laravel. El primer paso fue la creación de la base de datos, para ello, Laravel ofrece las migraciones, que permiten gestionar la creación/modificación de la base de datos de manera sencilla.

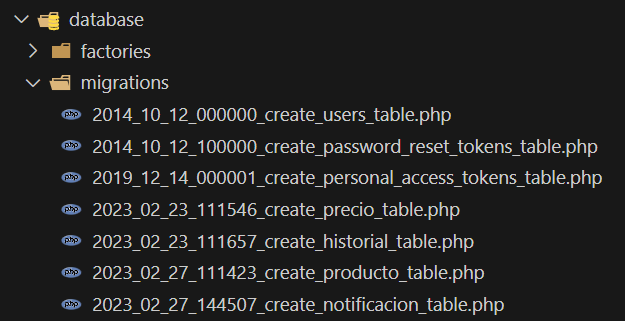


Ilustración 13: Lista de migraciones

Como se puede observar en la ilustración[], existen los ficheros ‘password\_reset\_tokens’ y ‘personal\_access\_tokens’, estos ficheros se crean por defecto en Laravel, pero no se han utilizado en este proyecto. El resto de ficheros representan cada uno una tabla del diseño realizado en el primer incremento.

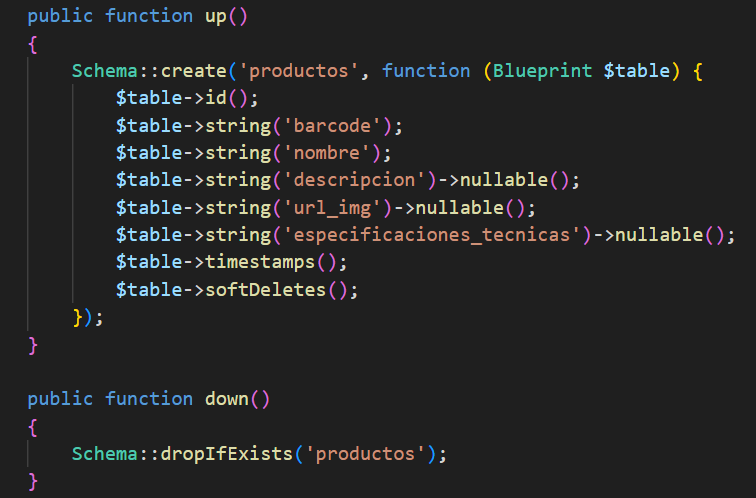


Ilustración 14: Ejemplo de migración

Con todos los ficheros creados, se realiza la migración mediante el comando ‘php artisan migrate’ creando todas las tablas en la base de datos.

El siguiente paso consiste en la creación de los modelos para estas tablas.

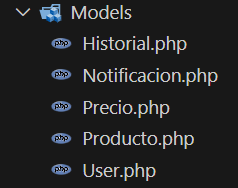


Ilustración 15: Lista de modelos

Se crea un modelo para cada tabla, y cada uno tiene una forma similar a la siguiente:

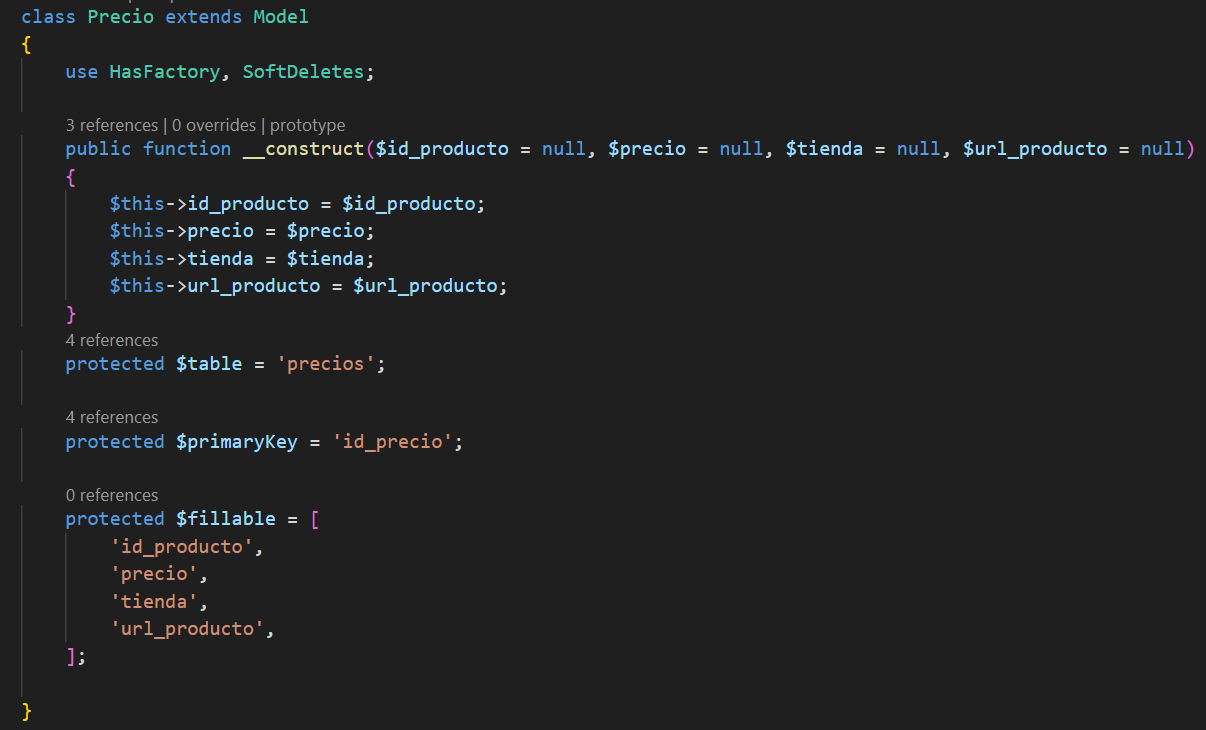


Ilustración 16: Ejemplo de un modelo

En cada modelo se asigna a que tabla de la base de datos referencia, sus campos y un constructor.

Finalmente se crean los controladores, y con el objetivo de conseguir la máxima limpieza y calidad del código se ha creado un controlador para cada modelo.

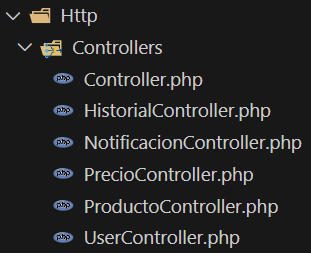


Ilustración 17: Lista de controladores

Los controladores son los encargados de que responder a las peticiones de los usuarios, ya sea a través de las vistas, o a través de peticiones a endpoints. En este proyecto, el backend en Laravel es el servidor central tanto para frontend como para la aplicación en Android. Por ello, este objetivo se alarga en el tiempo durante el tercer incremento, ya que en el desarrollo de la app móvil se crean nuevos endpoints y se modifican los controladores.

Para gestionar los endpoints de la web y de la app se separan en dos ficheros distintos:

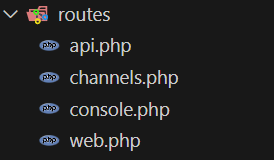


Ilustración 18: Lista de ficheros de endpoints

En ‘api.php’ se encuentran los endpoints correspondientes a la app Android y en ‘web.php’ los endpoints de la web.



Ilustración 19: Ejemplo fichero de endpoints

Ambos ficheros son similares, aunque en ‘web.php’ se utilizan middlewares para limitar el acceso a rutas a usuarios no autorizados.

### Incremento 3

En este incremento se ha realizado el desarrollo de la aplicación Android utilizando el lenguaje de programación Kotlin.

Para realizar la parte visual de la aplicación se ha utilizado una estrategia similar al frontend de la web. La aplicación consta de una actividad principal llamada ‘Activity\_Main’ el cual está formado por un contenedor de fragmentos y la barra de navegación.

Por defecto, en el contenedor de fragmentos se inicializa el ‘fragment\_scan’ el cual permite abrir la vista que inicia el escáner de códigos de barras, pero dependiendo de la interacción del usuario, el fragmento activo en el contenedor cambia.

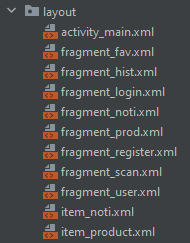


Ilustración 20: Lista de vistas

Estos son los diferentes fragmentos que dispone la aplicación, al igual que en la web, también tienen sus propios controladores.

Texto

Descripción generada automáticamente

Ilustración 21: Lista de controladores

En Android los controladores también gestionan que se hace antes de cargar el fragmento, y una vez cargado.

Texto

Descripción generada automáticamente

Ilustración 22: Crear vista desde controlador

Para comunicar la aplicación con el servidor, se han utilizado las librerías okhttp3 y retrofit2. En la clase `peticiones.kt’ en la se inicializan todos los métodos para hacer las peticiones, manejando los hilos mediante corrutinas.

Texto

Descripción generada automáticamente

Ilustración 23: Función con corrutinas para iniciar sesión

Y cada método definido en la clase `peticiones.kt’ realiza una petición a un endpoint del servidor. Los endpoints a los que se realiza la petición están definidos en la clase ‘APIService.kt’ donde se especifica el tipo, la dirección, los parámetros que se envían y la respuesta esperada.

Texto

Descripción generada automáticamente

Ilustración 24: Ejemplo de peticiones al servidor

Las recuperar las respuestas de las peticiones se han creado modelos de las clases, y modelos para cada respuesta.

Imagen que contiene Texto

Descripción generada automáticamente

Ilustración 25: Lista de modelos y respuestas de peticiones

Texto

Descripción generada automáticamente

Ilustración 26: Ejemplo de respuesta del servidor

Durante este incremento también se implementa el lector de códigos de barra y de QR. Para ello se hace uso de la librería zxing. El funcionamiento es el siguiente:

Texto

Descripción generada automáticamente

Ilustración 27: Configuración e iniciar el escáner

Se crea una función que configura e inicializa el escáner. Este método es llamado al pulsar un botón, y al ejecutarse se inicia una nueva actividad con la cámara iniciada. Una vez escaneado un código de barras o un QR, se cierra la actividad, y para recuperar la información se utiliza el método de Android ‘onActivityResult’.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ilustración 28: Recuperar datos del escáner

### Incremento 4

Este incremento está enfocado en la obtención de información de distintas fuentes a través de internet. Para ello he realizado Web Scraping en diferentes plataformas.

La elección de fuentes de las que obtener información para la plataforma está compuesta por Amazon, Mediamarkt y Ebay. Durante este incremento también se ha intentado añadir más fuentes de datos como: El Corte Inglés, Carrefour, Alcampo, Fnac, Worten, Game, PcComponentes, Wallapop, MilAnuncios. Estas fuentes de datos no han podido incorporarse al producto final, ya que las webs bloquean las peticiones realizadas desde scripts, por lo que no ha sido posible incorporarlas.

Para este incremento se ha creado una carpeta dentro del servidor llamada ‘WebScraping’, que contiene los diferentes programas escritos en Python para la obtención de la información.

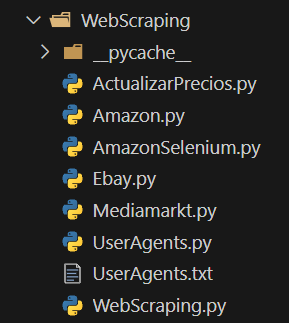


Ilustración 29: Lista de ficheros para web scraping

Los ficheros ‘Amazon.py’, ‘Ebay.py’, ‘Mediamarkt.py’ contienen los métodos necesarios para la obtención de la información de esa web. Todas siguen un esquema similar al siguiente:

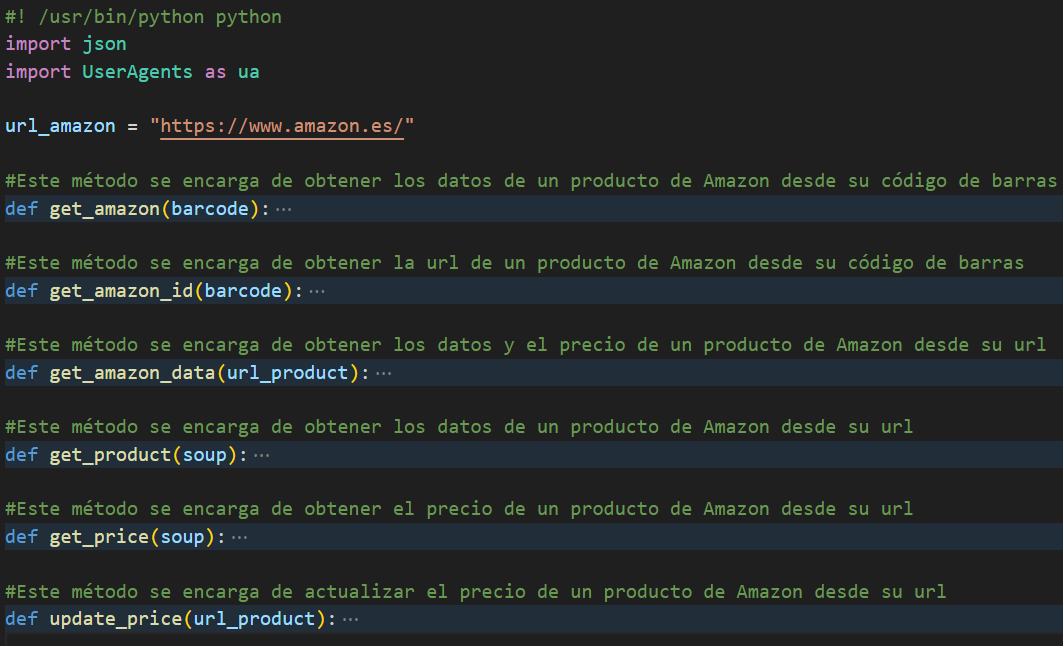


Ilustración 30: Esquema de clase para hacer scraping a una tienda

Los ficheros ‘userAgents.py’ y ‘userAgents.txt’ son los encargados de hacer la petición y devolver a los ficheros anteriores la respuesta de la petición usando BeautifulSoup. El fichero ‘userAgents.txt’ está formado por una lista de 1000 encabezados distintos, esto se utiliza para que las peticiones se envíen como si fuesen realizadas por diferentes dispositivos.

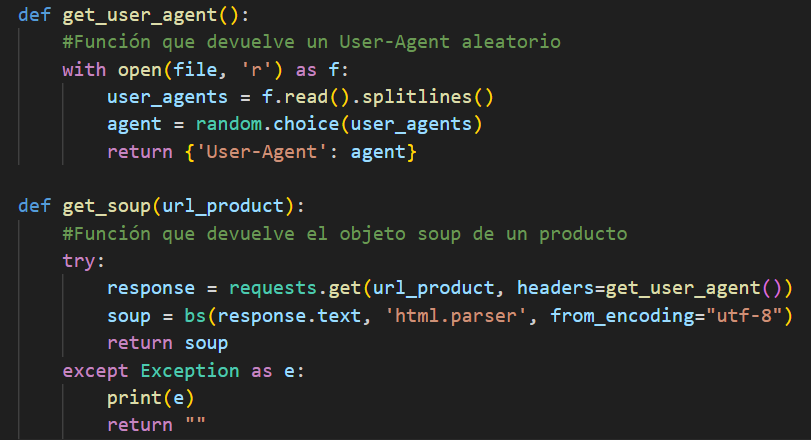


Ilustración 31: Fichero que lanza peticiones a las tiendas

Cuando un usuario realiza una búsqueda de un producto, se ejecuta el fichero ‘WebScraping.py’, El cual obtiene el código de barras o el nombre de producto como parámetro y devuelve la información al servidor.

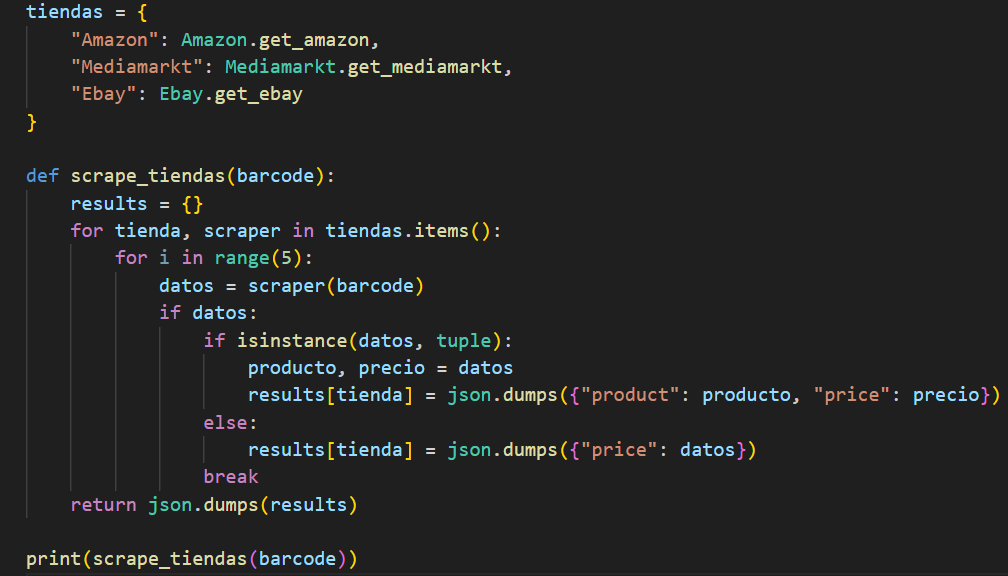


Ilustración 32: Fichero que busca un producto nuevo

Para generar el actualizar los precios de los productos se ejecuta el fichero ‘ActualizarPrecios.py’. Este fichero tiene un funcionamiento similar al anterior, pero en este caso se lanza primero una petición para obtener un listado de precios, después se recorre cada precio, se actualiza, y se envía una petición con el nuevo valor al servido para guardarlo en la base de datos.

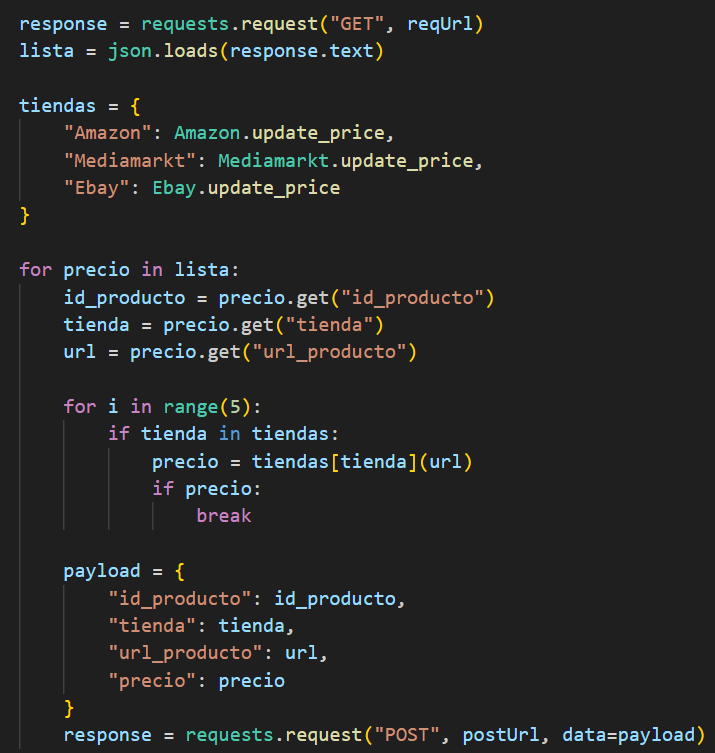


Ilustración 33: Clase para actualizar los precios de los productos

Es importante que los precios se actualicen de manera constante, para generar un historial de precios, para ello se ha utilizado crontab. Mediante esta herramienta, cada 6 horas se ejecuta el fichero ‘ActualizarPrecios.py’.



Ilustración 34: Crontab para ejecutar el fichero que actualiza los precios

## Recursos requeridos

Para utilizar este proyecto es necesario el siguiente software:

* Programas
  + Visual Studio Code
  + Android Studio
* Lenguajes de programación
  + Kotlin
  + Python
  + PHP
  + JavaScript
* Librerías
  + Servidor
    - Mariadb
    - Laravel Framework
  + App
    - com.google.zxing:core:3.3.0
    - com.journeyapps:zxing-android-embedded:4.1.0
    - com.squareup.okhttp3:okhttp:4.9.3
    - com.github.bumptech.glide:glide:4.12.0
    - com.github.bumptech.glide:compiler:4.12.0
    - com.google.code.gson:gson:2.8.9
    - com.squareup.retrofit2:retrofit:2.9.0
    - com.squareup.retrofit2:converter-gson:2.9.0
    - com.squareup.retrofit2:adapter-rxjava2:2.9.0
    - org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.6
  + Web
    - Bootstrap:5.0.2
    - Fontawesome:5.15.3
    - Axios
    - Plotly:2.20.0
  + Web Scraping
    - Requests
    - Json
    - BeautifulSoup
    - Lxml

## Presupuesto

Tabla 2: Presupuesto

|  |  |  |
| --- | --- | --- |
| **Tipo de coste** | **Valor** | **Comentarios** |
| Horas de trabajo en el proyecto | 300 horas | Horas de trabajo en el proyecto + documentación |
| 50 horas | Formación y aprendizaje en Kotlin, Laravel y Bootstrap |
| Equipo técnico utilizado | Portátil – 750 € | El equipo utilizado para el desarrollo del proyecto es un [portátil MSI](https://www.pccomponentes.com/msi-modern-14-b4mw-032xes-amd-ryzen-7-4700u-16gb-512gb-ssd-14). |
| Servidor – 100 € | El servidor es una [Raspberry pi 3](https://www.amazon.es/Raspberry-Pi-Modelo-Quad-Core-Cortex-A53/dp/B01CD5VC92/ref=sr_1_3?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=16633DK4YKEZM&keywords=raspberry+pi+3&qid=1684419321&sprefix=reaspberry+pi+3%2Caps%2C101&sr=8-3&ufe=app_do%3Aamzn1.fos.5e544547-1f8e-4072-8c08-ed563e39fc7d). |
| Software utilizado | 0 € | Todo el software utilizado es de uso libre y gratuito, por lo que no hay ningún coste. |
| Estudios e informes | 0 € | No se ha necesitado ningún estudio y/o informe |
| Materiales empleados | 0 € | No se ha necesitado ningún material |

## 

## Viabilidad

En la tabla del presupuesto, se puede observar como los costes del proyecto son muy bajos. Para implantarlo como un proyecto real, sería interesante trabajar en las mejoras propuestas en el Capítulo 7.

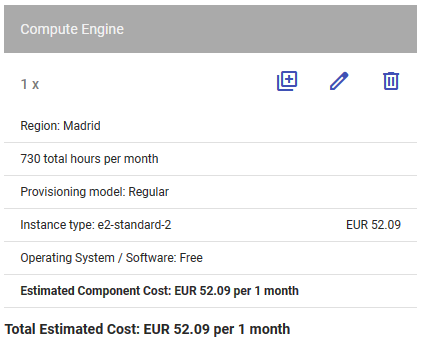


Ilustración 35: Coste servidor

Se ha utilizado la calculadora de precios de GCP[[Calculadora de precios de Google Cloud  |  Google Cloud](https://cloud.google.com/products/calculator?hl=es)] para obtener un presupuesto estimado de un servidor con las características necesarias para poder hacer que funcione este proyecto e incluso con la potencia suficiente para implantar todas las propuestas comentadas. También habría que adquirir un dominio, pero el precio de este es inferior a 30€ al año.

Tabla 3: Viabilidad

|  |  |
| --- | --- |
| **Producto** | **Coste** |
| Servidor cloud | 52.09€/mes |
| Dominio | 30€/año -> 2.50€/mes |
| **Total** | **54.59€/mes** |

Estos 54.59€ al mes serían los únicos gastos propios del proyecto, por lo que una vez implantado, esta cantidad de dinero es fácilmente recuperable mediante anuncios en la web y en la app.

Por ello, este proyecto es viable económicamente hablando.

## Resultados del proyecto

El resultado principal del proyecto es la creación de una plataforma capaz de mejorar la experiencia de compra de los usuarios. La plataforma está formada por tres partes, el servidor, la aplicación móvil y la página web.

En el servidor ha creado un sistema de extracción de datos en Internet que recopila información de productos de diversas tiendas y marketplaces. Esta información se muestra al usuario de manera organizada y actualizada en la aplicación móvil y en la página web. Además, se ha automatizado la extracción de datos de precios de productos en diferentes tiendas en línea. Esta información se utiliza para generar un historial de precios que permite a los usuarios visualizar las fluctuaciones de precios a lo largo del tiempo.

Para facilitar el acceso a estas funcionalidades, se ha implementado un sistema de registro y autenticación que permite a los usuarios acceder a su historial de búsquedas, productos favoritos y alertas de precios desde ambas plataformas, tanto en la aplicación móvil como en la página web. También se ha creado un sistema que permite a los usuarios guardar productos como favoritos, otorgándoles un control sencillo y rápido sobre los productos que desean seguir o adquirir en el futuro.

Por último, se ha implementado un sistema de alertas que informa a los usuarios sobre cambios en el precio de los productos que han marcado como favoritos. Estas alertas se pueden ver tanto en la aplicación móvil como en la página web, lo que permite a los usuarios aprovechar oportunidades de compra o estar al tanto de cualquier modificación en los precios.

Se ha desarrollado una aplicación móvil, para dispositivos Android que permite a los usuarios escanear el código de barras o el código QR de un producto y obtener información detallada sobre el mismo, incluyendo la imagen, el precio, la descripción y las características técnicas.

Se ha desarrollado una página web que ha implementado un buscador que permite a los usuarios ingresar el nombre o código de barras de un producto para obtener información sobre él. También se ha desarrollado una funcionalidad que muestra a los usuarios una gráfica con el histórico de precios de un producto en particular. Esto les permite analizar las tendencias de precios y tomar decisiones informadas al realizar compras.

En resumen, el proyecto ha logrado desarrollar una aplicación móvil y una página web que permiten a los usuarios obtener información detallada de productos, acceder a un historial de búsquedas, visualizar gráficas de precios, gestionar productos favoritos y recibir alertas de cambios de precios. Estos resultados finales cumplen con los objetivos específicos establecidos.

### Pruebas y validación

Tabla 4: Pruebas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Caso de prueba** | **Descripción** | **Fecha** | **Área funcional** | **Resultado** |
| PS1 | Base de datos correcta. | Se realiza la migración de la base de datos. | 27/02/2023 | Servidor | La migración se realizó correctamente y el resultado de la misma fue el esperado. |
| PS2 | Servidor web. | Se comprueba utilizando Google Chrome, si el servidor web funciona correctamente. | 01/03/2023 | Servidor | El servidor web respondía correctamente. |
| PB1 | Registrar datos no válidos. | Se envía una petición para registrar un usuario con datos no válidos. | 03/03/2023 | Backend | El servidor devuelve error indicando que los datos no son válidos y no se ha registrado el usuario. |
| PB2 | Registrar datos válidos. | Se envía una petición para registrar un usuario con datos válidos. | 03/03/2023 | Backend | El servidor devuelve un mensaje indicando se ha registrado el usuario. |
| PB3 | Iniciar sesión datos no válidos. | Se envía una petición para iniciar sesión con datos no válidos. | 04/03/2023 | Backend | El servidor devuelve error indicando que los datos no son válidos y no se ha iniciado sesión. |
| PB4 | Iniciar sesión datos válidos. | Se envía una petición para iniciar sesión con datos válidos. | 04/03/2023 | Backend | El servidor devuelve un mensaje indicando se ha iniciado sesión. |
| PBW5 | Cerrar sesión. | Se envía una petición para cerrar sesión. | 04/03/2023 | Backend Web | Se cierra la sesión correctamente. |
| PB6 | Buscar producto. | Se envía una petición con el nombre de un producto. | 18/03/2023 | Backend | Se devuelve la información de un producto y si no se encuentra te devuelve un mensaje diciendo que el producto no se ha encontrado. |
| PB7 | Añadir/eliminar favorito | Se envía una petición con el id del usuario y del producto. | 25/03/2023 | Backend | Se cambia en la base de datos el valor de favorito de un usuario en un producto, después te devuelve un mensaje |
| PW1 | Registrar datos no válidos. | Se utilizan datos no válidos para registrar un usuario. | 03/03/2023 | Web | No se envía petición al servidor y se indica que los valores no son válidos. |
| PW2 | Registrar datos válidos. | Se utilizan datos válidos para registrar un usuario. | 03/03/2023 | Web | Se envía una petición al servidor con los datos, se registra el usuario y te devuelve a la página principal con la sesión iniciada. |
| PW3 | Iniciar sesión datos no válidos. | Se utilizan datos no válidos para iniciar sesión. | 04/03/2023 | Web | No se envía petición al servidor y se indica que los valores no son válidos. |
| PW4 | Iniciar sesión datos válidos. | Se utilizan datos válidos para iniciar sesión. | 04/03/2023 | Web | Se envía petición al servidor y si los valores son correctos se inicia sesión y te devuelve a la página principal y si no se indica que los datos no son correctos. |
| PW5 | Cerrar sesión. | Se da al botón de cerrar sesión. | 04/03/2023 | Web | Se cierra sesión y vuelves a la página principal |
| PW6 | Buscar producto. | Se introduce el nombre de un producto en el buscador. | 18/03/2023 | Web | Si se encuentra el producto te envía a la página del producto, sino se indica que el producto no se ha encontrado. |
| PW7 | Añadir/eliminar favorito. | Se da al botón de favorito. | 25/03/2023 | Web | Si el producto era favorito, le quita.  Si el producto no era favorito, le vuelve favorito. |
| PW8 | Ver historial. | Se da al botón de historial. | 01/04/2023 | Web | Se cambia a la vista de historial con todos los productos que el usuario ha buscado/escaneado con anterioridad. |
| PW9 | Ver favoritos. | Se da al botón de favoritos. | 01/04/2023 | Web | Se cambia a la vista de favoritos con todos los productos que el usuario ha añadido a favoritos con anterioridad. |
| PW10 | Ver alertas. | Se da al botón de alertas. | 01/04/2023 | Web | Se cambia a la vista de alertas con todos los productos que el usuario ha añadido a favoritos con anterioridad, mostrándose el precio actual y mínimo de cada producto junto con su fecha de actualización. |
| PA1 | Test conexión | Se envía una petición de prueba desde el móvil al servidor | 08/04/2023 | App | La respuesta del servidor llega al móvil |
| PA2 | Registrar datos no válidos. | Se intenta registrar a un usuario desde el móvil utilizando datos no válidos | 09/04/2023 | App | La propia app valida los datos, y si no son válidos no envía la petición. |
| PA3 | Registrar datos válidos. | Se intenta registrar a un usuario desde el móvil utilizando datos válidos | 09/04/2023 | App | La app valida los datos, luego se vuelven a validar en el servidor y se registra al usuario. Por último te devuelve a la página principal de la app con la sesión iniciada. |
| PA4 | Iniciar sesión datos válidos | Se utilizan datos no válidos para iniciar sesión. | 15/04/2023 | App | La propia app valida los datos, y si no son válidos no envía la petición. |
| PA5 | Iniciar sesión datos no válidos | Se utilizan datos válidos para iniciar sesión. | 15/04/2023 | App | La app valida los datos, y se envía petición al servidor. Si los valores son correctos se inicia sesión y te devuelve a la página principal de la app con la sesión iniciada, y si no se indica que los datos no son correctos. |
| PA6 | Cerrar sesión. | Se da al botón iniciar sesión. | 15/04/2023 | App | Se cierra sesión y se vuelve a la página principal de la app. |
| PA7 | Escanear un código de barras/QR | Se da al botón de escanear y se escanea un producto con la cámara | 22/04/2023 | App | Se envía una petición al servidor con el código de barras/qr y si encuentra un producto se va a la página del producto y si no sale un mensaje de producto no encontrado. |
| PA8 | Añadir/eliminar favorito. | Se da al botón de favorito. | 23/04/2023 | App | Si el producto era favorito, le quita.  Si el producto no era favorito, le vuelve favorito. |
| PA9 | Ver historial. | Se da al botón de historial. | 23/04/2023 | App | Se cambia a la vista de historial con todos los productos que el usuario ha buscado/escaneado con anterioridad. |
| PA10 | Ver favoritos. | Se da al botón de favoritos. | 23/04/2023 | App | Se cambia a la vista de favoritos con todos los productos que el usuario ha añadido a favoritos con anterioridad. |
| PA11 | Ver alertas. | Se da al botón de alertas. | 23/04/2023 | App | Se cambia a la vista de alertas con todos los productos que el usuario ha añadido a favoritos con anterioridad, mostrándose el precio actual y mínimo de cada producto junto con su fecha de actualización. |
| PWS1 | Web scraping Amazon | Se ejecuta el script para Amazon enviando un nombre de producto | 30/04/2023 | Web Scraping | Se obtiene la información y el precio de ese producto. |
| PWS2 | Web scraping Mediamarkt | Se ejecuta el script para Mediamarkt enviando un nombre de producto | 30/04/2023 | Web Scraping | Se obtiene la información y el precio de ese producto. |
| PWS3 | Web scraping Ebay | Se ejecuta el script para Ebay enviando un nombre de producto | 30/04/2023 | Web Scraping | Se obtiene el precio más bajo de los resultados de Ebay para una búsqueda ese producto. |

# MANUAL DE INSTALACIÓN Y USUARIO

## Manual de instalación

Para utilizar Easycom en local, es necesario instalar [Xampp](https://www.apachefriends.org/es/index.html), [Composer](https://getcomposer.org/) y [Python](https://www.python.org/downloads/) en el equipo. Además será necesario contar con un IDE o un editor de texto, para el desarrollo se ha utilizado [Visual Studio Code](https://code.visualstudio.com/download). Por último, para la aplicación Android será necesario instalar [Android Studio](https://developer.android.com/studio).

En Python será necesario instalar una serie de librerías, para ello se ejecutarán en orden los siguientes comandos:

* py -m pip
* pip install -r requirements.txt

Una vez instalado todo lo anterior , tendremos todo lo necesario para ejecutar el proyecto.

### Crear base de datos

Para iniciar el proyecto, el siguiente paso es crear la base de datos, para ello lo primero es ejecutar Xampp e iniciar el proceso de MySQL.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 36: Xampp iniciar MySQL

Una vez iniciado puedes usar el gestor de bases de datos que se prefiera para crear la base de datos. Si no se dispone de ninguno, se puede iniciar el servicio de Apache en Xampp y usar phpMyAdmin.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 37: Xampp abrir phpMyAdmin

Esto abrirá la siguiente página, donde iremos al apartado de bases de datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 38: phpMyAdmin

En la pagina de base de datos introduciremos el nombre que queremos asignarle a la misma y le daremos al botón crear.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 39: phpMyAdmin crear base de datos

Una vez se le ha dado al botón de crear, ya tendremos la base de datos lista para seguir con la instalación.

### Migración de la base de datos

Si ya tenemos la base de datos creada el siguiente paso es crear las tablas, para ello en hay que añadir los datos de la base de datos creada en el proyecto. En el fichero ‘.env’ se indica el nombre de la base de la base de datos y del usuario y contraseña si se han modificado por el usuario.

Texto

Descripción generada automáticamente

Ilustración 40: modificar .env

Una vez añadido al proyecto la dirección de la base de datos, el siguiente paso es realizar la migración de las tablas. Para ello un terminal en la carpeta del proyecto se ejecutará el siguiente comando:

* php artisan migrate

Con este paso se tiene la base de datos instalada y configurada para funcionar.

### Iniciar servidor web

Para iniciar el servidor se abre un terminal en la carpeta del proyecto y se ejecutará el siguiente comando:

* php artisan serve

### Configurar app

El único cambio que hay que realizar en la app para que funcione es cambiar la ruta del servidor a la que se hacen las peticiones. Para ello se modifica el fichero ‘Peticiones.kt’, cambiando la url del proyecto por la dirección local. El servidor se ejecuta en el puerto 8000, un ejemplo de como quedaría una vez modificado es: “http://192.168.0.2:8000/api/”

Texto

Descripción generada automáticamente

Ilustración 41: Cambiar ruta peticiones app

## Manual de usuario

### Web

### App

Esta sección es más habitual en trabajos de tipo "científico" o de "investigación", donde uno presenta brevemente los resultados principales y los discute, pero también puede utilizarse en otro tipo de trabajos.

Puedes incluir secciones específicas para discutir cuestiones como: limitaciones del estudio, limitaciones de la tecnología empleada, cambios respecto a objetivos planteados inicialmente.

Puedes incluir respuestas a preguntas del tipo: ¿la metodología inicialmente pensada ha sido útil?, ¿he tenido que adaptarme a cambios a lo largo del proyecto?, ¿qué cambios han sido y cómo he adaptado el proyecto para poder manejar esos cambios?, ¿qué impacto ha tenido el resultado de mi proyecto?

# CONCLUSIONES

## Conclusiones del trabajo

Breve descripción objetiva del resultado en relación al objetivo general de tu proyecto.

## Conclusiones personales

Describe tus impresiones y experiencia personal durante el desarrollo del proyecto, o destacar la importancia que tiene el tema para ti, lo que has aprendido, o la trascendencia que ha tenido para ti o para otros este proyecto.

# FUTURAS LÍNEAS DE TRABAJO

El proyecto que se ha desarrollado es la base de la plataforma, pero esta tiene aún mucho margen de mejora. De cara a continuar el proyecto estas mejoras pueden ser muy interesantes:

* Adquirir un dominio para que la web se posicione mejor.
* Alojar el servidor en la nube, esto mejoraría el rendimiento global de la plataforma.
* Configurar un ‘SSL’ que permita que la conexión con la web sea completamente seguro.
* Implementar más tiendas, para ello sería necesario utilizar herramientas de web scraping más potentes, esto sería posible si se realiza el punto anterior.
* Implementar un apartado de comentarios dentro de cada producto, para que usuarios de la plataforma puedan preguntar/opinar sobre los productos.
* Implementar un sistema de recomendación de productos similares con inteligencia artificial.
* Implementar un sistema de correo electrónico, para validar el email, cambiar contraseña y recibir alertas.
* Implementar notificaciones push para Android.
* Crear app en IOS para dispositivos Apple.
* Implementación de un modo oscuro en la página web y traducción a varios idiomas.

# REFERENCIAS

Orús, Abigail. 2023. Cifras Clave del Comercio Electrónico. 7 feb 2023

https://es.statista.com/temas/3167/el-comercio-electronico-dentro-de-espana/

En este apartado figurará el conjunto de libros, revistas u otros textos que el autor considere de interés para justificar las soluciones adoptadas en el Proyecto. **Cita todas las fuentes** que has utilizado como consulta para elaborar el trabajo.

Sigue el estilo de cita que te indiquen las normas de estilo y respétalo a lo largo de todo el proyecto. Recuerda que has de citar todas las fuentes que hayas usado. Los estilos de cita más comunes son:

* ISO
* IEEE
* APPA
* Etc.

En ingeniería se suele usar el ISO o el IEEE.

Si puedes usar un gestor de citas bibliográficas te será más fácil. Si no tendrás que recurrir a las páginas web de las bibliotecas para saber cómo citar adecuadamente. Por ejemplo, lo encuentras en:

<http://biblioteca.uem.es/es/aprendizaje-y-formacion/citas-bibliograficas-documentos>

Referencias usadas en este manual de estilo:

**AENOR. 2010.** AEN/CTN 157 - PROYECTOS. *Normas y Publicaciones.* [En línea] 2010. [Citado el: 25 de abril de 2013.] http://www.aenor.es/aenor/normas/ctn/fichactn.asp?codigonorm=AEN/CTN%20157.

**Miró Julià, José. 2010.** Recursos para aprender a escribir. [En línea] 2010. http://bioinfo.uib.es/~joemiro/RecEscr/manual.pdf.

**UNE 157001. 2002.** Criterios generales para la elaboración de proyectos. *Escuela Universitaria de Ingeniería de Vitoria.* [En línea] 2002. [Citado el: 25 de abril de 2013.] http://www.coiib.es/coiib/documentos/DocumentosContenidos/Gu%C3%ADa%20de%20elaboraci%C3%B3n%20de%20proyectos/2-Electricidad/5\_PNE\_157701\_Criterios.pdf.

# ANEXOS

Sirven para incluir documentación complementaria (planos, circuitos, código, ficheros de configuración, especificaciones técnicas y hojas de características, fichas explicativas, resultado de encuestas, reglamentación y normativas requeridas, etc.).

[PÁGINA INTENCIONADAMENTE EN BLANCO]