

Grado en Ingeniería Informática  
2022-2023

*Trabajo Fin de Grado*

## “Sistema de Recomendación de Moda basado en Atributos Multimodales”

---

Gonzalo Llosá Cea

Tutores

Miguel Ángel Patricio Guisado

Carlos Rodríguez - Pardo

Madrid, septiembre



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**



## RESUMEN

La industria de la moda es una de las más potentes en nuestro país y esta se ha ido adaptando al auge tecnológico que hemos vivido en las últimas décadas. Un gran porcentaje de las ventas de este sector se realizan de manera online y es por ello la importancia de un sistema de recomendación.

Un sistema de recomendación de moda para las tiendas online resulta muy beneficioso, tanto para propietarios como para clientes. Los primeros aumentan sus ventas mientras que los segundos encuentran más fácil y rápido productos que, en muchas ocasiones, ni siquiera hubieran podido encontrar.

Para solucionar este problema se hace uso de una amplia base de datos en donde las prendas vienen etiquetadas por sus diferentes atributos. Después de realizar un estudio sobre las tecnologías que pueden resolver este problema se eligen diferentes arquitecturas para resolver las diferentes partes que componen el objetivo principal. Para la segmentación de imágenes, con el fin de conseguir sus máscaras, se hace uso de la arquitectura U2NET. Para el algoritmo de recomendación, se extraen los vectores de características de las imágenes haciendo uso de la arquitectura VGG16, para posteriormente calcular la similitud coseno entre estos vectores. Además se añade la posibilidad de elegir los atributos que se quiere que estén presentes en la recomendación de la prenda.

Una vez se realiza este modelo se desarrolla una aplicación web que es capaz de servirlo para que los usuarios puedan utilizarlo de una manera fácil e intuitiva. El modelo se sirve a través de una API, haciendo uso de *Flask*, a la aplicación web hecha con *React*.

También se lleva a cabo un estudio de la efectividad del sistema, analizando los porcentajes de similitud entre recomendaciones y la calidad de estas.

Este proyecto tiene gran relevancia en el mundo real. Puede llegar a causar un gran impacto tanto en clientes de negocios de venta online de ropa, como en los propios clientes.

**Palabras clave:** Red convolucional, visión por computadora, segmentación de imágenes, máscara, VGG16, U2NET.



## ABSTRACT

The fashion industry is one of the most powerful sectors in our country, and it has been adapting to the technological boom that we have experienced in recent decades. A significant percentage of sales in this sector occur online, underscoring the importance of a recommendation system.

A fashion recommendation system for online stores proves highly advantageous, benefiting both proprietors and customers. The former witness an increase in sales, while the latter find it easier and quicker to discover products that, in many instances, they might not have come across otherwise.

To address this issue, an extensive database is employed, wherein garments are labeled with their distinct attributes. After conducting a study on technologies capable of resolving this problem, diverse architectures are selected to tackle the various components constituting the primary objective. For image segmentation, aimed at obtaining their masks, the U2NET architecture is utilized. Concerning the recommendation algorithm, feature vectors are extracted from the images using the VGG16 architecture, subsequently computing the cosine similarity between these vectors. Furthermore, the option to select the attributes to be present in the garment recommendation is incorporated.

Once this model is developed, a web application is created to facilitate user-friendly and intuitive utilization. The model is served through an API, utilizing *Flask*, for the web application built with *React*.

An assessment of the system's effectiveness is also conducted, analyzing similarity percentages between recommendations and their quality.

This project holds significant relevance in the real world, potentially yielding a substantial impact on both online clothing business clientele and the customers themselves.

**Keywords:** Convolutional network, computer vision, images segmentation, mask, VGG16, U2NET.



# ÍNDICE GENERAL

1. INTRODUCCIÓN . . . . .	1
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Marco regulador . . . . .	2
1.4. Entorno socioeconómico . . . . .	2
1.5. Estructura del documento . . . . .	3
2. ESTADO DEL ARTE . . . . .	5
2.1. Visión por computadora . . . . .	5
2.2. Redes de Neuronas Convolucionales . . . . .	6
2.3. Extracción de características . . . . .	7
2.4. Segmentación de imágenes . . . . .	7
2.5. Trabajos similares . . . . .	7
3. DISEÑO DEL SISTEMA . . . . .	8
3.1. Conjunto de datos . . . . .	8
3.2. Modelo . . . . .	9
3.2.1. Segmentación. . . . .	9
3.2.2. Algoritmo de recomendación . . . . .	12
3.3. API . . . . .	13
3.4. Interfaz . . . . .	15
3.5. Tecnologías empleadas . . . . .	20
4. RESULTADOS . . . . .	22
4.1. Sin máscaras . . . . .	22
4.2. Con máscaras . . . . .	23
4.3. Uso de atributos . . . . .	26
5. CONCLUSIONES . . . . .	29
6. TRABAJOS FUTUROS . . . . .	31
7. PLANIFICACIÓN Y PRESUPUESTOS . . . . .	32
7.1. Planificación . . . . .	32

7.2. Presupuestos . . . . .	33
7.2.1. Coste del personal . . . . .	33
7.2.2. Coste del equipo . . . . .	33
7.2.3. Costes directos . . . . .	34
7.2.4. Costes indirectos . . . . .	34
7.2.5. Costes totales . . . . .	34
BIBLIOGRAFÍA . . . . .	35



## ÍNDICE DE FIGURAS

1.1	Peso de las ventas moda online sobre el total de los ingresos generados por las ventas de ropa en España de 2012 a 2022 . . . . .	1
2.1	Detección de caras con el software OpenCV . . . . .	5
2.2	Red de Neuronas . . . . .	6
3.1	Ejemplo de imagen en el conjunto de datos . . . . .	8
3.2	Ejemplo de imagen antes y después de aplicar la máscara . . . . .	9
3.3	Arquitectura U2-Net . . . . .	11
3.4	Ejemplo de creación de máscaras con la arquitectura U2-Net . . . . .	11
3.5	Estructura algoritmo VGG16 . . . . .	12
3.6	Diagrama del funcionamiento del modelo . . . . .	13
3.7	Ejemplo de petición al <i>endpoint</i> “/recommendations” . . . . .	14
3.8	Ejemplo de petición al <i>endpoint</i> “/image/<id>” . . . . .	15
3.9	Estructura del proyecto . . . . .	16
3.10	Componente <b>attributesSearch</b> . . . . .	16
3.11	Componente <b>header</b> . . . . .	16
3.12	Componente <b>imageUploader</b> . . . . .	17
3.13	Pantalla inicial, subida de imagen . . . . .	18
3.14	Selección del número de recomendaciones . . . . .	18
3.15	Selección de atributos . . . . .	19
3.16	Barra de carga mientras la API procesa la solicitud . . . . .	19
3.17	Resultado del proceso. Recomendaciones. . . . .	20
4.1	Recomendaciones sin uso de máscaras . . . . .	22
4.2	Recomendaciones sin uso de máscaras . . . . .	23
4.3	Recomendaciones con uso de máscaras . . . . .	24
4.4	Recomendaciones con uso de máscaras . . . . .	24
4.5	Gráfica de Similitud Coseno . . . . .	25
4.6	Recomendaciones con selección del atributo “blazer” . . . . .	26

4.7 Recomendaciones con selección de los atributos “blazer” y “loose (fit)” .	27
4.8 Recomendaciones con selección de los atributos “blazer”, “loose (fit)” y “geometric textile pattern” . . . . .	27



## ÍNDICE DE TABLAS

3.1	Especificaciones del hardware utilizado . . . . .	20
4.1	Tabla de similitud de recomendaciones . . . . .	22
4.2	Tabla de similitud de recomendaciones . . . . .	23
4.3	Tabla de similitud de recomendaciones . . . . .	24
4.4	Tabla de similitud de recomendaciones . . . . .	25
4.5	Medias de similitud coseno de imágenes . . . . .	26
4.6	Tabla de similitud de recomendaciones . . . . .	26
4.7	Tabla de similitud de recomendaciones . . . . .	27
4.8	Tabla de similitud de recomendaciones . . . . .	28
7.1	Planificación del proyecto . . . . .	33
7.2	Planificación del proyecto . . . . .	33
7.3	Costes materiales . . . . .	34
7.4	Costes directos . . . . .	34
7.5	Costes directos . . . . .	34



## **LISTA DE ACRÓNIMOS**

<b>API</b>	Application Programming Interface
<b>PIB</b>	Producto Interior Bruto
<b>HTML</b>	HyperText Markup Language
<b>CSS</b>	Cascading Style Sheets
<b>CPU</b>	Central Processing Unit
<b>GPU</b>	Graphics Processing Unit
<b>IDE</b>	Integrated development environment
<b>UI</b>	User Interface



# 1. INTRODUCCIÓN

## 1.1. Motivación

La industria de la moda siempre ha sido una de las más potentes en nuestra sociedad. Esta industria se ha ido adaptando al auge que han tenido las tecnologías informáticas y es que, en el año 2022, las ventas de ropa online supusieron el 21,1 % de las ventas totales en nuestro país.

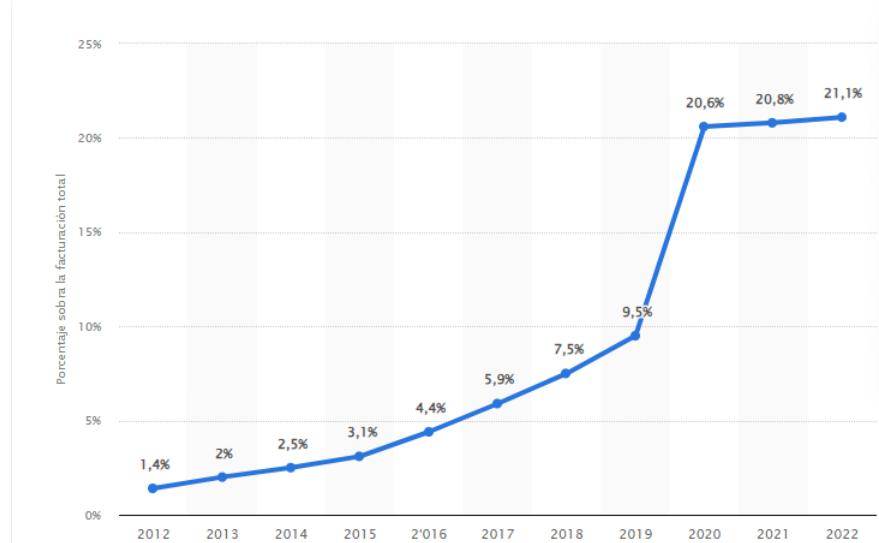


Fig. 1.1. Peso de las ventas moda online sobre el total de los ingresos generados por las ventas de ropa en España de 2012 a 2022 [1]

Es aquí donde entran en juego los sistemas de recomendación. Un sistema de recomendación de moda permite ayudar a las personas a encontrar prendas que se ajusten a sus gustos. Los clientes pueden mejorar su experiencia de compra y estar satisfechos con sus elecciones al recibir recomendaciones personalizadas y precisas. Esto resulta en un aumento de las ventas de los sitios web que se dedican a la moda. Un sistema de recomendación que esté basado únicamente en imágenes es útil, pero el hecho de introducir la posibilidad de especificar los atributos que deben estar en las prendas lo mejora aún más. Permitir a los usuarios obtener recomendaciones les ahorra mucho tiempo en búsquedas que, en muchas ocasiones, pueden resultar tediosas.

En conclusión, el hecho de crear un sistema de recomendación de moda basado en atributos multimodales resulta muy interesante tanto para propietarios de tiendas online, ya que mejoran sus ventas, como para usuarios, que tienen más facilidad a la hora de encontrar las prendas que buscan, ahorrando así una gran cantidad de tiempo.

## **1.2. Objetivos**

El objetivo de este trabajo es el de crear una interfaz web que sea capaz de recomendar imágenes de ropa, con una cierta similitud, a partir de otra previamente dada por el usuario. Además se podrán especificar los atributos que se deseen en las imágenes de salida (por ejemplo: manga larga, ajuste suelto, etc.).

Siendo este el principal objetivo podemos diferenciar los siguientes subobjetivos:

- Desarrollar un modelo de inteligencia artificial que sea capaz de recomendar imágenes en base a atributos multimodales y otra imagen.
- Crear una API que permita la comunicación entre el modelo previamente mencionado y una interfaz.
- Diseñar una interfaz intuitiva que permita a los usuarios cargar una imagen, seleccionar los atributos de interés y visualizar los resultados proporcionados por el modelo.

Estos objetivos pretenden resolver el problema planteado, derivando en una interfaz gráfica web que sea capaz de brindar a los usuarios la capacidad de obtener recomendaciones de prendas de ropa a partir de ropa que les guste y de los atributos que consideren.

## **1.3. Marco regulador**

Para la realización del trabajo se hace uso de un conjunto de imágenes de la competición “iMaterialist (Fashion) 2020 at FGVC7” [2]. Estas imágenes contienen rostros de personas reales por lo que hay que tener en cuenta la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales [3], aún así el conjunto de datos ha sido citado de manera apropiada. Por la misma razón se ha de prestar atención a la Ley de Propiedad Intelectual [4]. Además si consideramos este como un trabajo de investigación habría que contemplar la Ley 14/2011, de 1 de junio, de la Ciencia, la Tecnología y la Innovación [5].

Las herramientas utilizadas para el desarrollo de este proyecto son de uso libre y gratuito, como son *Python*, Visual Studio Code, Tensorflow, Keras o React.

## **1.4. Entorno socioeconómico**

Como ya se mencionó anteriormente en el apartado de la motivación, la moda es una de las mayores industrias a nivel global. Esta genera mucho interés y supone el 2,8 % del PIB español y el 4 % del mercado laboral [6]. Es por ello que un sistema de recomendación de moda tiene un gran interés e impacto a nivel social. Algunos de los beneficios son

los siguientes, según el artículo de Alma Muñoz “Cómo los sistemas de recomendación pueden ayudarte a conseguir más ventas” [7].

- “Entre 15 % y 45 % de aumento de las conversiones”
- “25 % de media de incremento del valor promedio de compra.”
- “Alargamiento del ciclo de vida del cliente.”
- “Generación de fidelización del comprador.”

Además, Alma, menciona que “Amazon declara que alrededor del 35 % de sus ingresos se debe a recomendaciones de producto.”

En concreto, este producto pretende ser un sistema de recomendación de moda. Es por ello que podría ser útil para cualquier empresa que se dedique a la venta de ropa vía online por todos los beneficios que este aporta y que ya han sido mencionados. Además, implementar este sistema en una web de venta de ropa supone una gran ventaja sobre el resto de los competidores.

Un sistema de recomendación también tiene una gran relevancia a nivel social. Los usuarios de estos pueden, no solo llegar a ahorrar grandes cantidades de tiempo, sino también conseguir encontrar prendas más afines a su gusto y personalidad. Esto resulta en una mayor satisfacción por parte del cliente, que necesitará invertir menos tiempo en encontrar mejores productos.

En conclusión, este proyecto aporta muchas ventajas económicas tanto a vendedores como a compradores, ahorrando tiempo y consiguiendo una mayor satisfacción con la compra. Más adelante, en su correspondiente apartado, se detallará la planificación y presupuesto.

## 1.5. Estructura del documento

El documento se presenta con una portada, a continuación podemos encontrar un resumen junto con las palabras clave. Después, podemos encontrar el índice general, el índice de figuras y la lista de acrónimos. De aquí en adelante el documento está dividido en siete marcados capítulos:

- **Introducción:** aquí podemos encontrar tanto la motivación para la realización del trabajo como los objetivos que este pretenden cumplir. Además se incluye el marco regulador y el entorno socioeconómico.
- **Estado del arte:** se elabora un análisis sobre el estado actual del objetivo del proyecto. Se analizan trabajos parecidos y el estado de las tecnologías empleadas. Además de argumentar como este proyecto se puede diferenciar de los trabajos ya realizados.

- **Diseño del sistema:** se explica como se ha llevado a cabo la solución del problema propuesto. Dividiéndolo en sus respectivos subobjetivos ya planteados.
- **Resultados:** se presentan los resultados obtenidos, con ejemplos, de una manera visual. Además se analiza la calidad de estos a través de los porcentajes de similitud.
- **Conclusiones:** reflexión que valora si se han cumplido los objetivos propuestos.
- **Trabajos futuros:** se analizan las posibles continuaciones y mejoras que se pueden llevar a cabo a partir de este trabajo.
- **Planificación y presupuestos:** resumen a cerca de como se ha planificado el proyecto y explicación del cálculo de los presupuestos.

## 2. ESTADO DEL ARTE

En este capítulo se analizará el estado actual del proyecto, haciendo un recorrido por las tecnologías necesarias para su realización. Además se realizará un análisis de los proyectos similares junto con una argumentación a cerca de como este puede llegar a diferenciarse de ellos.

### 2.1. Visión por computadora

La visión por computadora es la disciplina del aprendizaje automático que trata con imágenes. A través de ella se intenta entender de forma computacional las imágenes. Entre otras cosas, la visión por computadora, se utiliza para el reconocimiento de objetos, la restauración de imágenes, el reconocimiento facial y un sin fin de tareas más. Dado que se trabaja con imágenes la importancia de esta disciplina en este proyecto es máxima.

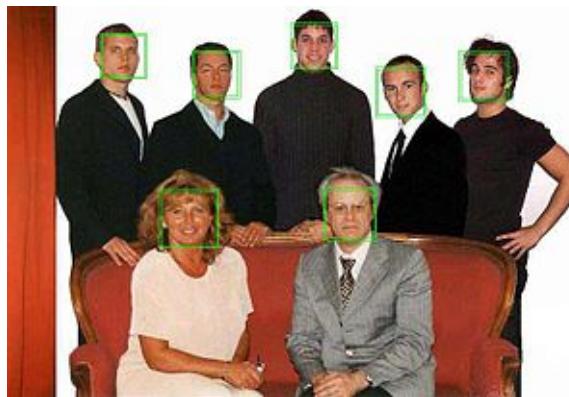


Fig. 2.1. Detección de caras con el software  
OpenCV [8]

La visión por computadora ha tenido un gran avance durante los últimos años debido, principalmente, a las mejoras en hardware. Somos ahora capaces de realizar una mayor cantidad de cálculos en una cantidad mayor de datos gracias a un significativo aumento de la capacidad de cómputo. Estos progresos permiten que se investigue y profundice más en este campo.

Con la llegada de las redes neuronales se ha hecho posible la realización de tareas complejas como la clasificación de imágenes, la detección de objetos o la segmentación. Las redes de neuronas convolucionales se han convertido en el estándar para este campo de la ciencia. Estas son capaces de ver patrones visuales que a un humano, se le harían complicados, si no imposibles, de ver.

## 2.2. Redes de Neuronas Convolucionales

Las redes neuronales están compuestas por diferentes capas, que a su vez están compuestas por diferentes neuronas. Todas las redes tienen una capa de entrada, una o más capas intermedias y una capa de salida. Los nodos de las capas están conectados entre sí y estos pueden activar o no, dependiendo del resultado de su función asociada, el nodo al que están conectados.

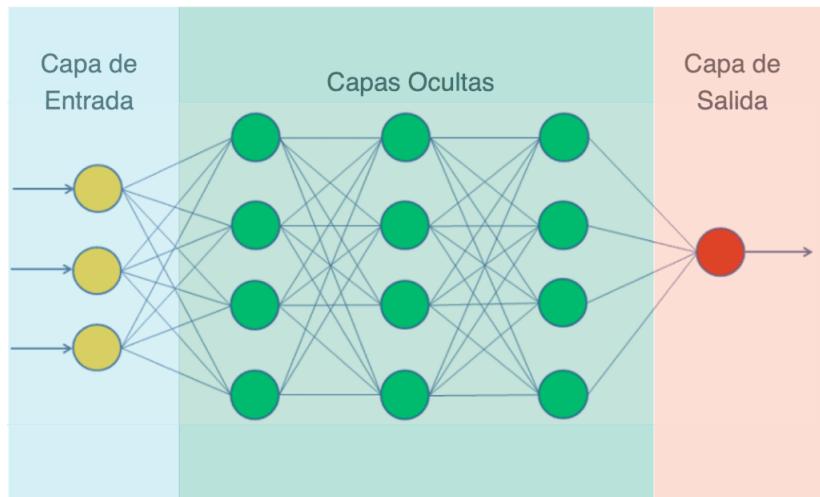


Fig. 2.2. Red de Neuronas [9]

Las redes de neuronas convolucionales son las que se utilizan para tareas de visión por computadora. Estas redes son mucho más eficientes a la hora de tratar con imágenes y esta es la razón por la que se usan en este campo del aprendizaje automático. En las primeras capas se identifican los elementos de la imagen a una escala grande, como pueden ser los bordes o los colores. A medida que se va profundizando en la red se van reconociendo partes más grandes de la imagen. Una vez se llega al final, la red es capaz de identificar el tipo de imagen. Las redes convolucionales se componen de tres capas:

- **Capa convolucional:** en esta capa se aplica el proceso conocido como convolución. Se aplica un filtro (kernel) con el fin de detectar las características que se buscan en la imagen.
- **Capa de agrupación:** en esta capa se aplica una reducción de dimensionalidad. Esta capa es muy importante ya que, aunque se pierde información, se gana en eficiencia.
- **Capa totalmente conectada:** en esta capa, a diferencia de las demás, todos los nodos están conectados a algún nodo de la capa anterior.

Existen varios tipos de redes de neuronas convolucionales, entre ellas: AlexNet, GoogleNet, ResNet, ZFNet o VGGNet. Para la realización de este proyecto se ha optado por usar la última.

### **2.3. Extracción de características**

VGG16

### **2.4. Segmentación de imágenes**

U2NET

### **2.5. Trabajos similares**

Hay muchos trabajos previos similares a este. Entre ellos destacan:

- “Image-based Product Recommendation System with Convolutional Neural Networks” [10]. En este artículo se describen las técnicas utilizadas para realizar un sistema de recomendación para tiendas online basado en la similitud entre imágenes. Se prueban los algoritmos SVG, AlexNet y VGG. Tras evaluar los tres este último resulta ser el más preciso. Una vez obtenidos los vectores de las imágenes se calcula la distancia coseno entre estos para calcular la similitud entre ambas.
- “Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach” [11]. Este artículo consta de dos fases. En la primera tratan de clasificar las imágenes según el tipo de producto mientras que en la segunda se crea un sistema de recomendación que trata de encontrar los productos más similares a otro dado. Para la segunda fase, que es la que aplica en nuestro caso, se usan coeficientes JPEG para sacar las características de las imágenes. Después se calcula la distancia euclíadiana entre estas.

Este trabajo se diferencia en varios aspectos de los anteriores. No solo se busca una recomendación en base a una imagen inicial, sino que también se permite al usuario seleccionar los atributos que quiere que tengan las imágenes recomendadas. Además de esta principal diferencia este proyecto no se queda solo en un modelo, también cuenta con una interfaz gráfica que permite a los usuarios hacer uso del modelo.

### 3. DISEÑO DEL SISTEMA

#### 3.1. Conjunto de datos

Como se mencionó en el apartado “Marco Reguador” en este trabajo se hace uso del conjunto de datos “iMaterialist (Fashion) 2020 at FGVC7” [2]. Este conjunto de datos cuenta con varios archivos que se explican a continuación.

En primer lugar tenemos dos conjuntos de imágenes, “train” y “test”. El conjunto “train” cuenta con 43.793 imágenes y “test” con 3200. Ambos conjuntos contienen estas imágenes nombradas por un identificador.



Fig. 3.1. Ejemplo de imagen en el conjunto de datos

Además hay un archivo llamado “train.csv” que contiene los siguientes campos:

- **imageId**: identificador único para las imágenes. Para cada imagen hay varias entradas en el archivo, tantas como segmentaciones tenga la imagen
- **encodedPixels**: estos pixeles son la segmentación de la imagen
- **height**: altura de la imagen
- **width**: ancho de la imagen
- **classId**: identificador de la clase a la que pertenecen
- **attributeIds**: lista de identificadores de atributos que contiene este segmento de la imagen

La segmentación es de gran utilidad, ya que sirve para realizar las máscaras de las imágenes. También se destaca el campo **attributesIds** ya que gracias a este podemos filtrar las imágenes a gusto del usuario. Estos atributos vienen descritos en un archivo llamado “label\_descriptions.json”. Cada atributo tiene una descripción y una supercategoria asociadas a su identificador único.

### 3.2. Modelo

#### 3.2.1. Segmentación

En primer lugar se deben preprocesar las imágenes, el objetivo del trabajo es hacer recomendaciones de prendas similares, por lo tanto todo lo que no sea una prenda en la imagen puede influir negativamente en la recomendación. Por ejemplo, se podrían recomendar prendas con el fondo de la imagen similar, imágenes que tengan una misma marca de agua o imágenes en las que los modelos que las visten tengan parecidos físicos, entre otras cosas. Estos aspectos no deberían influir en la recomendación por lo que lo primero que se realiza son las máscaras de las imágenes. Para ello se utiliza el atributo **encoded-Pixels**. Con estos pixeles, que son una segmentación de las prendas, se puede obtener una imagen que consista únicamente en ropa. A continuación se muestra un ejemplo de una imagen antes y después de aplicarle la máscara.



Fig. 3.2. Ejemplo de imagen antes y después de aplicar la máscara

Como se puede observar en la figura, la máscara ha quitado todo excepto la prenda. Todas estas máscaras son preprocesadas y almacenadas, ya que serán las que se usen para el modelo.

Esto no es suficiente, ya que el usuario proporciona una imagen que no está en la base de datos, por lo que no se tiene la segmentación de esta. Debido a este motivo es necesario hacer uso de un modelo de segmentación para generar la máscara de la imagen dada por el usuario. Hay un gran número de arquitecturas que podrían resolver este problema. A continuación se describen algunas de ellas:

- **U-Net:** su nombre viene dado porque su arquitectura tiene forma de U, combina capas de convolución y deconvolución para conseguir la segmentación de las imágenes.
- **SegNet:** utiliza capas de **pooling** para conseguir el objetivo de segmentar las imágenes.
- **Mask R-CNN:** esta arquitectura es una combinación de **Faster R-CNN**, que sirve para la detección de objetos, con la generación de máscaras.
- **DeepLab:** esta arquitectura utiliza la convolución dilatada para lograr el objetivo de segmentar las imágenes.
- **U2-Net:** esta arquitectura es una extensión de **U-Net** que incluye la generación de máscaras.

Finalmente se elige hacer uso de la arquitectura **U2-Net**, que cuenta con todas las ventajas de **U-Net** y además está adaptada para la creación de máscaras. Esta arquitectura tiene una gran precisión aún cuando actúa sobre imágenes de alta resolución, lo que la convierte en una gran candidata para la segmentación de imágenes de moda.

**U2-Net** nació en el año 2020 en la universidad de Alberta, Canadá, con el artículo “U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection” [12]. Este artículo fue galardonado en 2020 como el mejor en reconocimiento de patrones.

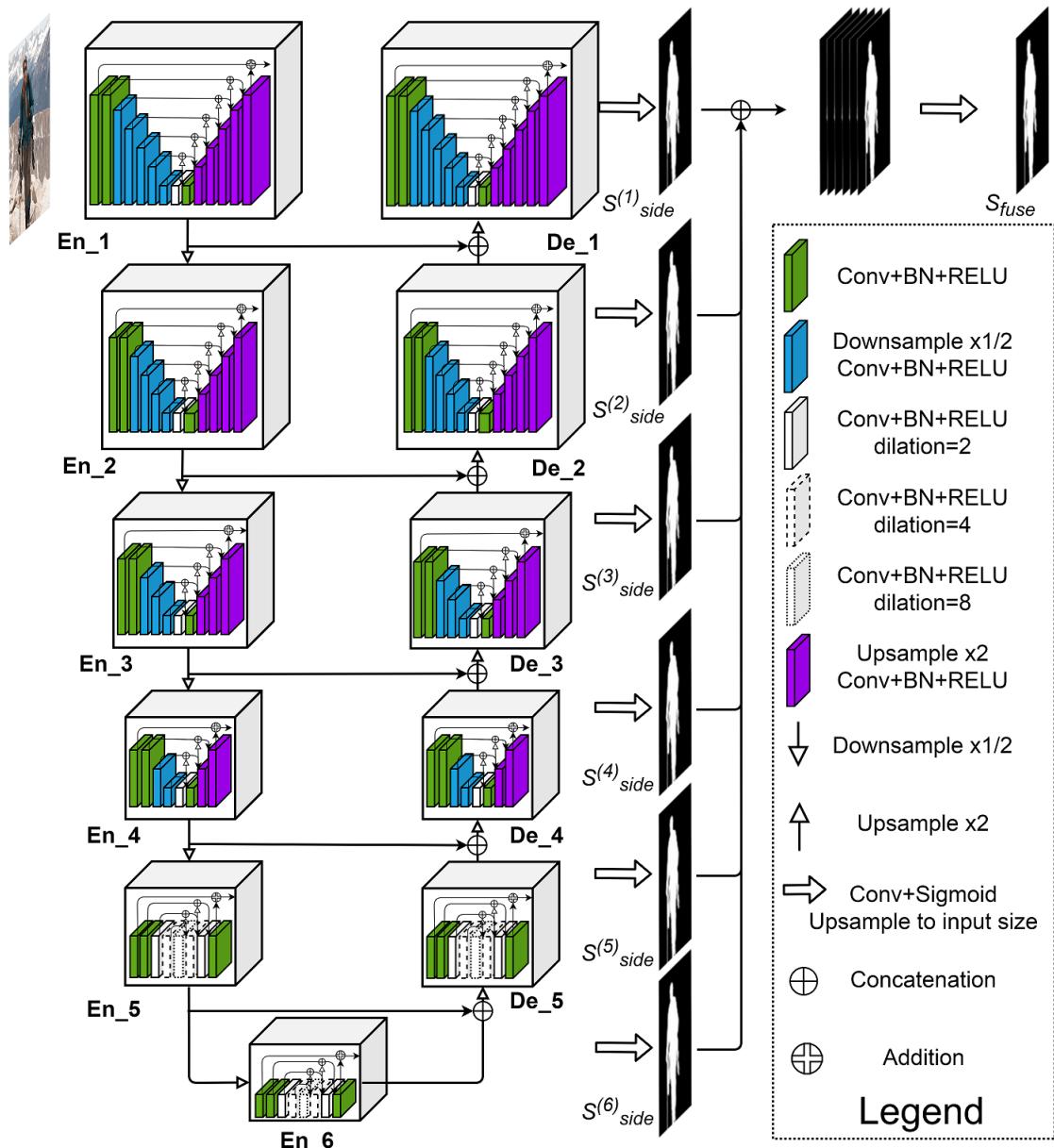


Fig. 3.3. Arquitectura U2-Net [13]



Fig. 3.4. Ejemplo de creación de máscaras con la arquitectura U2-Net [13]

### 3.2.2. Algoritmo de recomendación

El siguiente paso es extraer las características de todas las imágenes. Para ello se hace uso de una red de neuronas convolucional. Como ya se mencionó anteriormente se ha optado por elegir el modelo VGG16. Este modelo tiene una estructura fácil de entender y es un modelo más profundo que sus predecesores, como puede ser AlexNet, lo que permite extraer características de mayor complejidad.

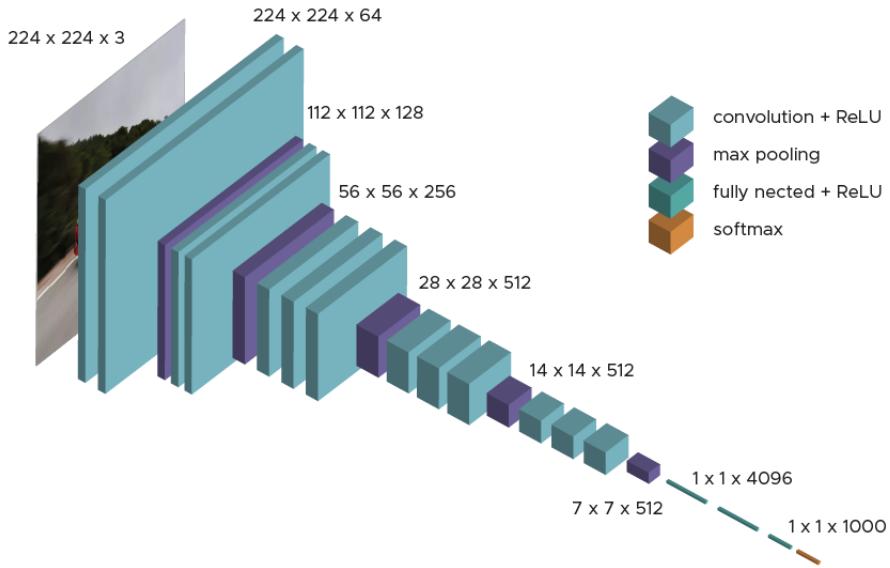


Fig. 3.5. Estructura algoritmo VGG16 [14]

El modelo hace uso de la arquitectura VGG16 y extrae el vector de características de todas las máscaras. Estos vectores son almacenados en una matriz. Cuando el usuario proporciona una imagen, se genera su máscara con el procedimiento mencionado anteriormente. Después se obtiene también su vector de características y se calcula la similitud coseno con los vectores de todas las demás imágenes. Finalmente se filtra el conjunto de datos por las prendas que contengan los atributos seleccionados por el usuario y se devuelven las  $N$  imágenes con la mayor similitud, donde  $N$  es el número de imágenes que el usuario desea obtener. El siguiente diagrama resume el funcionamiento del modelo.

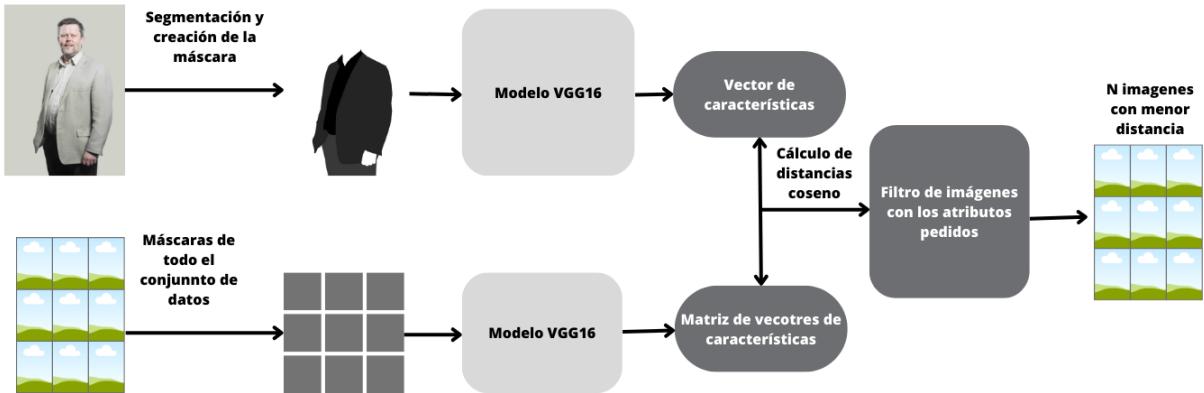


Fig. 3.6. Diagrama del funcionamiento del modelo

### 3.3. API

Con el fin de poder crear una interfaz gráfica fácil de usar para un usuario común es necesario tener una API que sirva el modelo. Una API es una forma de conectar diferentes aplicaciones para lograr que trabajen juntas. La API permite a la interfaz gráfica poder consumir el modelo de inteligencia artificial previamente descrito.

Se ha optado por hacer esta API en *Python*, ya que facilita el desarrollo al estar el modelo también hecho con este mismo lenguaje. La librería elegida para construir esta API es *Flask*, un *framework* diseñado para poder hacer aplicaciones CRUD (*Create, Read, Update, Delete*). Una aplicación CRUD puede tener varios *endpoint*, rutas a las que enviar una solicitud. Cada una de estas rutas puede realizar las acciones de crear, leer, actualizar o borrar información en la API.

En el caso de nuestra API solo han sido necesarios dos *endpoints*.

- “/recommendations”:

Uso el método *POST*. Este *endpoint* espera recibir una imagen, un número de recomendaciones y una lista de identificadores de atributos opcional. Una vez recibe estos argumentos, la API hace uso del modelo y devuelve la lista de identificadores de las imágenes que, teniendo los atributos requeridos, tengan la mayor similitud.

KEY	VALUE
<input checked="" type="checkbox"/> selectedAttributes	{"17": true}
<input checked="" type="checkbox"/> recomendationsNumber	3
<input checked="" type="checkbox"/> file	000c9b4926cd78edd4c19cbc6beba111.jpg <span style="color: red;">X</span>

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ≡

```

1
2   "images": [
3     "409d0263a72abdad2af97d85131f9761.jpg",
4     "ba3bb072bf67724461ae89c2368c8be.jpg",
5     "aaf1eb0b7d382cf8bc102e3e9e0e883f.jpg"
6   ]
7

```

Fig. 3.7. Ejemplo de petición al *endpoint* “/recommendations”

- “/image/<id>”:

La API permite recuperar imágenes a partir de su identificador gracias a este *endpoint*, donde “id” es el identificador de la imagen que se desea obtener. Solo es necesario hacer una petición de tipo *GET* con un identificador válido en la ruta.

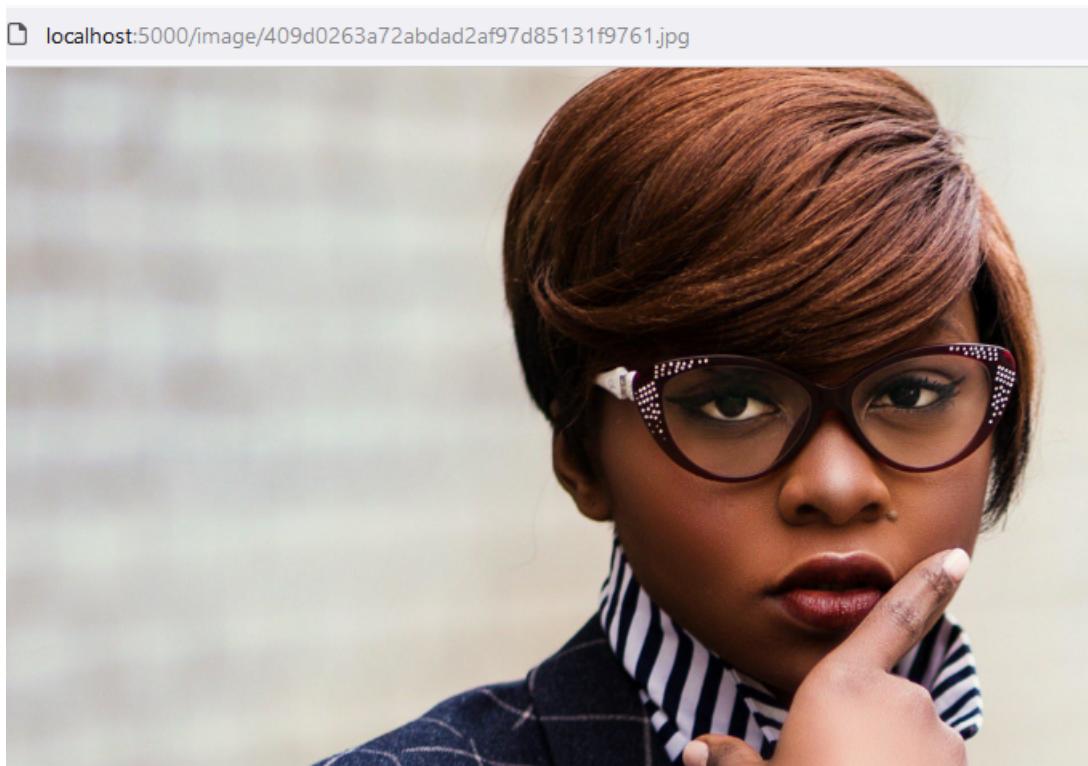


Fig. 3.8. Ejemplo de petición al *endpoint* “/image/<id>”

### 3.4. Interfaz

Para la interfaz gráfica se tuvieron en cuenta varias opciones. La primera de ellas fue *StreamLit*, una plataforma para desarrollar interfaces para proyectos de inteligencia artificial escrita en *Python*. Permite desplegar este tipo de aplicaciones de una manera muy sencilla, escribiendo muy pocas líneas de código. Por otro lado se pensó en usar *Gradio*, una plataforma que funciona de una manera muy similar a *StreamLit*. Ambas opciones son buenas pero al final se decidió hacer uso del *framework* de desarrollo *React*, del lenguaje *JavaScript*. Aunque las dos primeras herramientas eran interesantes para conseguir un desarrollo rápido, las opciones de interfaz quedaban más limitadas.

El hecho de no poder hacer la interfaz completamente a mi gusto fue lo que me hizo decantarme por *React*, con el cual se puede hacer una web completa de principio a fin, con todas sus características. Además se hace uso de *TypeScript*, en lugar de *JavaScript*, ya que ofrece una mayor solidez gracias a su tipado estático. Esto ayuda a no cometer errores difíciles de depurar durante el desarrollo, aunque a veces pueda resultar tedioso el hecho de tener que marcar todos los tipos. El hecho de utilizar *React* resulta muy útil, ya que se puede organizar el código por componentes reutilizables en donde se junta el *HTML* con el *JavaScript*.

El proyecto tiene la siguiente estructura:

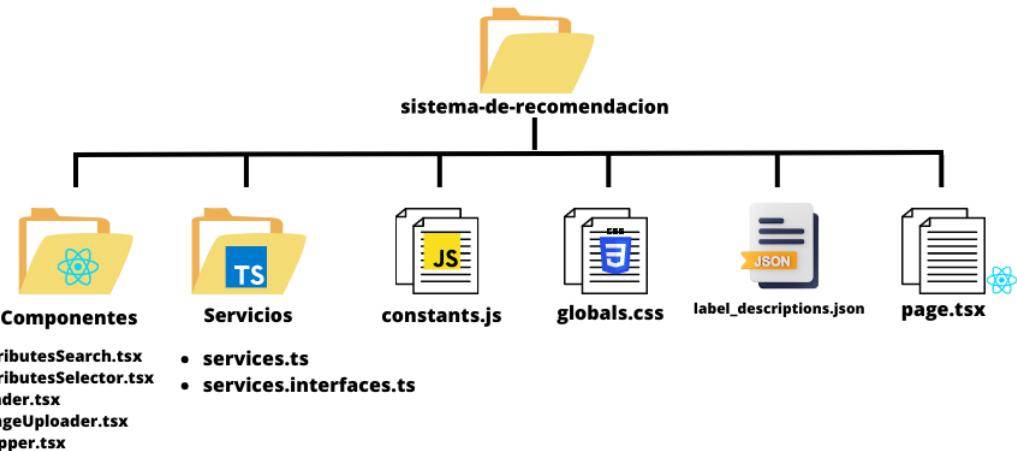


Fig. 3.9. Estructura del proyecto

- **attributtesSearch.tsx**: este componente sirve para buscar los atributos que el usuario quiera seleccionar para que aparezcan en sus recomendaciones. Cuenta con la lista completa de atributos (que son elementos seleccionables) y con un buscador, ya que hay 341 atributos diferentes.
- **attributtesSelector.tsx**: este componente contiene la búsqueda de atributos. Además cuenta con un botón para que, en caso de que el usuario así lo desee, sirve para obtener recomendaciones de prendas con todo tipo de atributos.

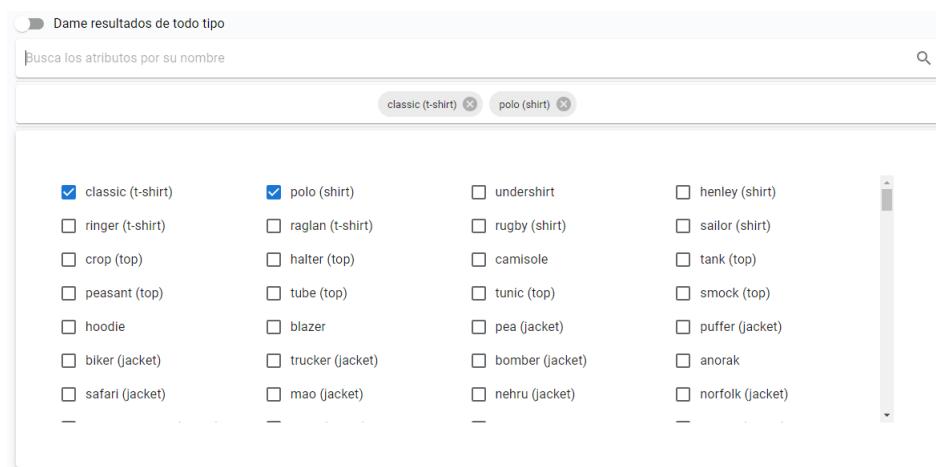


Fig. 3.10. Componente **attributesSearch**

- **header.tsx**: este es el encabezado de la página.



Fig. 3.11. Componente **header**

- **imageUploader**: este componente es una caja que permite al usuario subir la imagen que desea utilizar para obtener recomendaciones. Esta permite explorar sus archivos locales y subir un archivo que debe ser de tipo imagen.

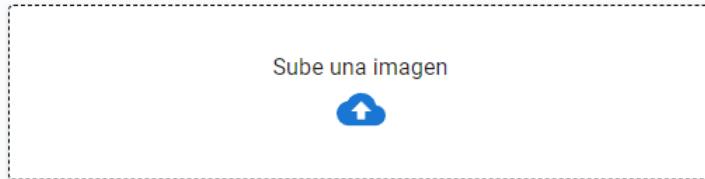


Fig. 3.12. Componente **imageUploader**

- **stepper**: este componente sirve para pasar las pantallas del proceso de dar toda la información, ya que la aplicación consiste en una sola página.
- **services.ts**: este archivo se encarga de realizar las peticiones a la API.
- **services.interfaces.ts**: este archivo contiene las interfaces necesarias para los servicios.
- **constants.js**: en este fichero se almacenan las variables que son constantes a lo largo del proyecto.
- **globals.css**: en este fichero se guarda la configuración CSS que afecta a todo el proyecto.
- **label\_descriptions.json**: este archivo se usa para asociar los identificadores de los atributos a sus descripciones.
- **page.tsx**: esta es la página principal, donde todos los componentes se unen para componer la aplicación.

Nada más cargar la página podemos ver una ventana que nos pide subir una imagen. Una vez subida la imagen se puede presionar el botón de “siguiente”, el cual nos lleva a seleccionar el número de recomendaciones, a través de una barra de selección de números. Una vez más es posible presionar el botón de “siguiente”, esta vez nos llevará a la ventana final, la selección de atributos. Esta contiene la lista completa de los atributos seleccionables. Los atributos seleccionados quedan marcados en la parte superior, desde donde se puede eliminar la selección. Además hay un buscador dada el gran número de atributos que hay. Una vez ya está todo listo se puede pulsar sobre el botón “procesar”. Este botón manda una petición a la API que, haciendo uso del modelo, devuelve los identificadores de las imágenes con mayor similitud. Con estos identificadores se vuelve a hacer una petición por cada uno de ellos para obtener sus respectivas imágenes. Estas imágenes se muestran en una galería que se puede poner a tamaño completo. Si se desea repetir el proceso basta con pulsar el botón “resetear”.

A continuación se muestra, a través de imágenes, el funcionamiento descrito anteriormente.

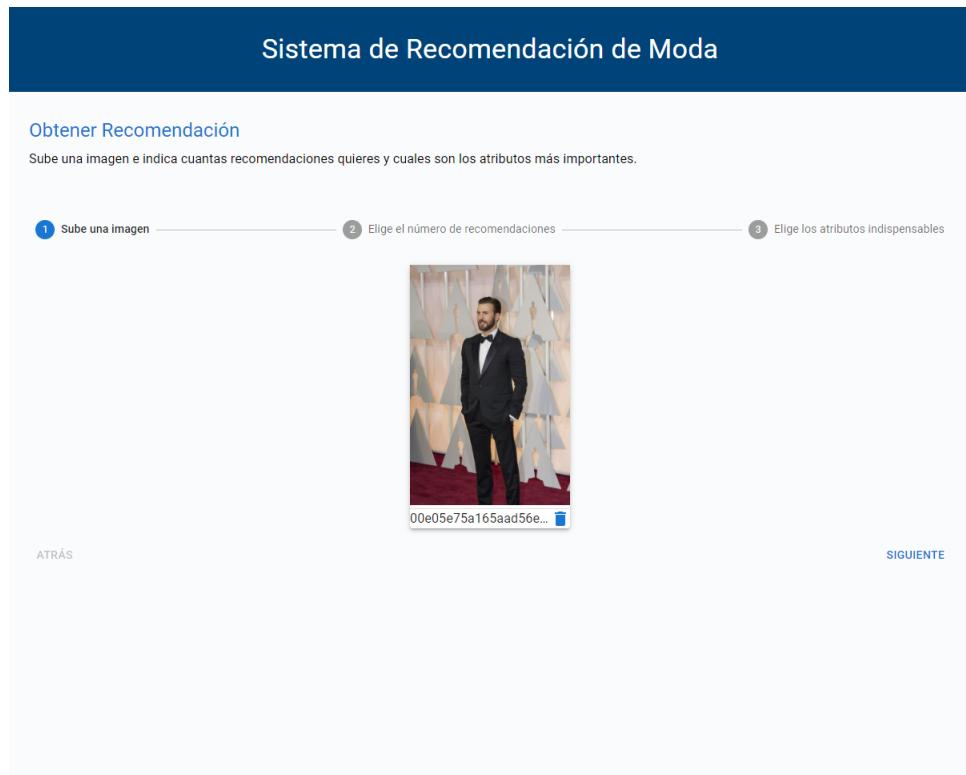


Fig. 3.13. Pantalla inicial, subida de imagen

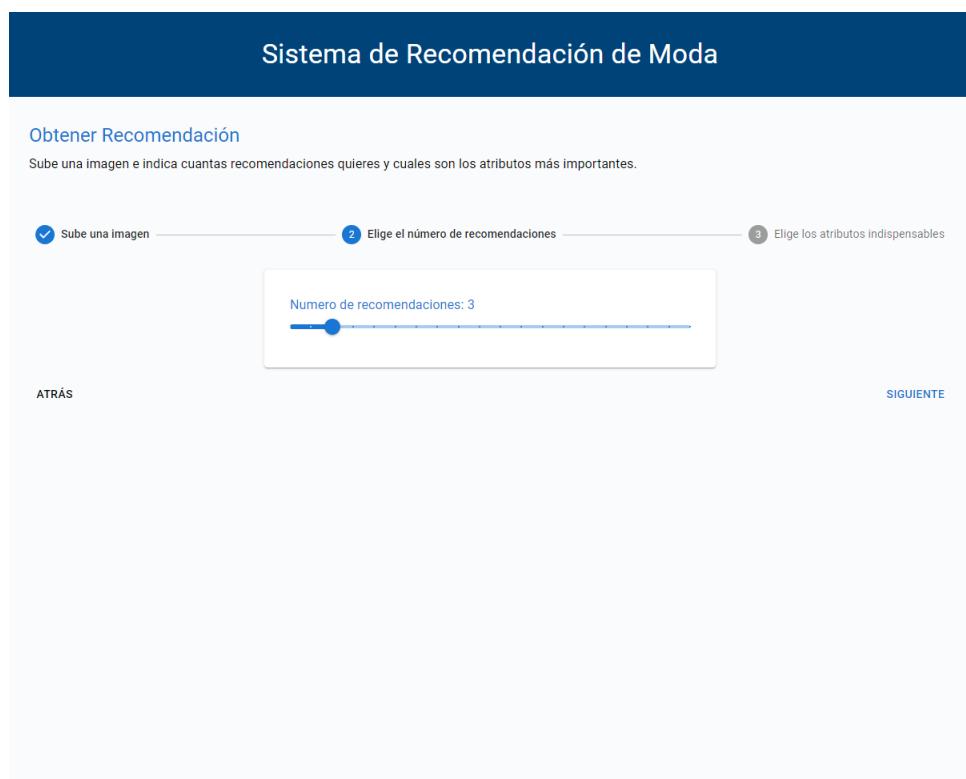


Fig. 3.14. Selección del número de recomendaciones

**Sistema de Recomendación de Moda**

**Obtener Recomendación**

Sube una imagen e indica cuantas recomendaciones quieres y cuales son los atributos más importantes.

1 Sube una imagen      2 Elige el número de recomendaciones      3 Elige los atributos indispensables

Dame resultados de todo tipo

Busca los atributos por su nombre

blazer

<input type="checkbox"/> classic (t-shirt)	<input type="checkbox"/> polo (shirt)	<input type="checkbox"/> undershirt	<input type="checkbox"/> henley (shirt)
<input type="checkbox"/> ringer (t-shirt)	<input type="checkbox"/> raglan (t-shirt)	<input type="checkbox"/> rugby (shirt)	<input type="checkbox"/> sailor (shirt)
<input type="checkbox"/> crop (top)	<input type="checkbox"/> halter (top)	<input type="checkbox"/> camisole	<input type="checkbox"/> tank (top)
<input type="checkbox"/> peasant (top)	<input type="checkbox"/> tube (top)	<input type="checkbox"/> tunic (top)	<input type="checkbox"/> smock (top)
<input type="checkbox"/> hoodie	<input checked="" type="checkbox"/> blazer	<input type="checkbox"/> pea (jacket)	<input type="checkbox"/> puffer (jacket)
<input type="checkbox"/> biker (jacket)	<input type="checkbox"/> trucker (jacket)	<input type="checkbox"/> bomber (jacket)	<input type="checkbox"/> anorak
<input type="checkbox"/> safari (jacket)	<input type="checkbox"/> mao (jacket)	<input type="checkbox"/> nehru (jacket)	<input type="checkbox"/> norfolk (jacket)
—	—	—	—

ATRÁS      PROCESAR

Fig. 3.15. Selección de atributos

**Sistema de Recomendación de Moda**

**Obtener Recomendación**

Sube una imagen e indica cuantas recomendaciones quieres y cuales son los atributos más importantes.

1 Sube una imagen      2 Elige el número de recomendaciones      3 Elige los atributos indispensables

RESETEAR

Fig. 3.16. Barra de carga mientras la API procesa la solicitud

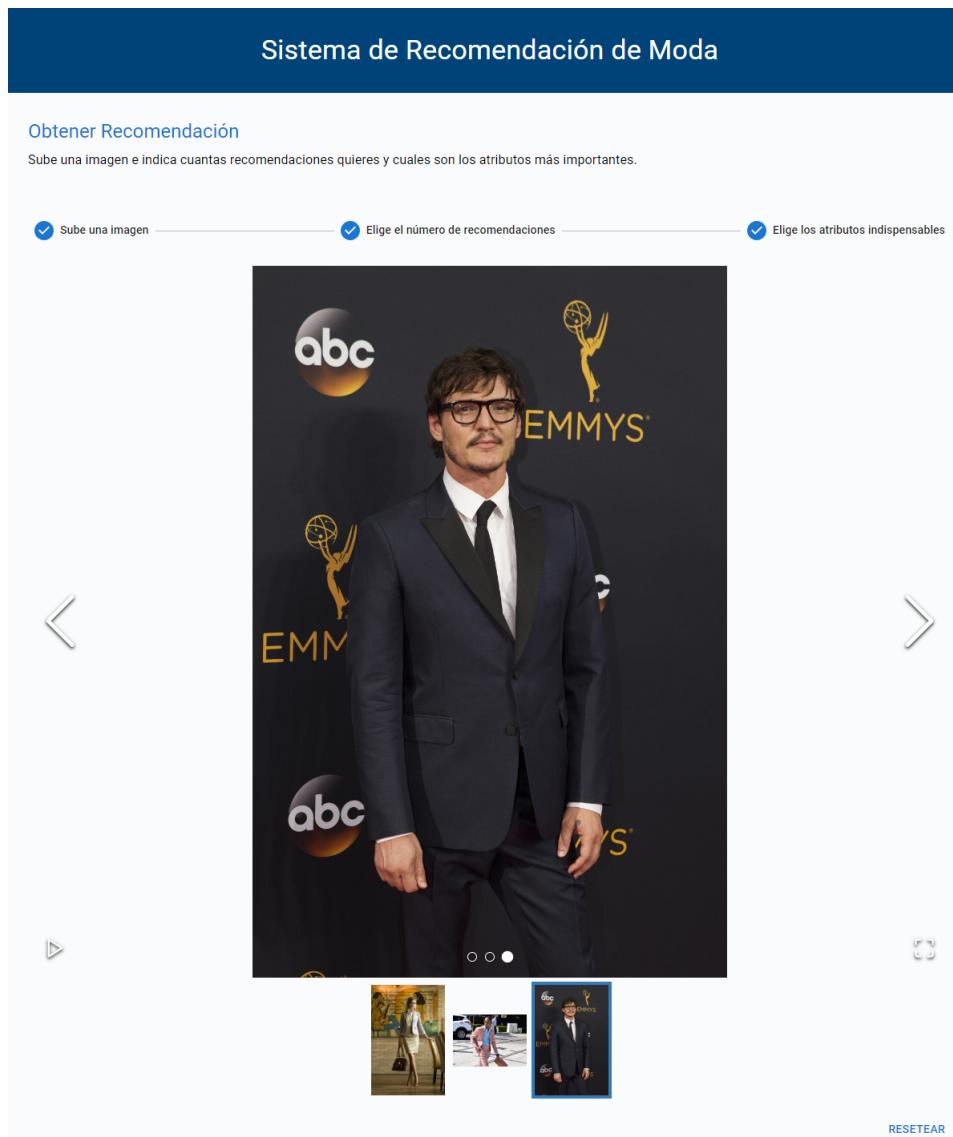


Fig. 3.17. Resultado del proceso. Recomendaciones.

### 3.5. Tecnologías empleadas

Para la realización de este trabajo ha sido necesario el uso tanto de hardware como de software. En la siguiente tabla se muestra el hardware utilizado.

Equipo	RAM	CPU	GPU
<b>Ordenador de sobremesa</b>	16 GB	AMD RYZEN 53500X	AMD Radeon RX 5500 XT
<b>HP ENVY 13</b>	8 GB	INTEL i7	NVIDIA GeForce MX150

TABLA 3.1. ESPECIFICACIONES DEL HARDWARE UTILIZADO

Ambos ordenadores tienen como sistema operativo *Windows*, 10 y 11 respectivamente. Se ha hecho uso del IDE *VScode*, de código abierto, para el desarrollo de tanto la API

como la interfaz gráfica.

En el caso de la API se ha usado el lenguaje de programación *Python* junto con las siguientes librerías:

- **Flask**: *framework* minimalista que sirve para desarrollar aplicaciones web. En este caso sirve para comunicar el modelo con la web.
- **Pickle**: utilizada para cargar a memoria el modelo con la matriz de características extraídas de las imágenes.
- **OS**: utilizada para trabajar con los archivos del sistema. Recuperar, crear y eliminar imágenes en él.
- **JSON**: manejo de objetos de tipo JSON. Utilizada para comunicarse con la aplicación web.
- **Numpy**: cálculos optimizados con las matrices que representan las imágenes. Utilización de algunas de sus funciones matemáticas.
- **Keras**: utilizado para hacer uso de la arquitectura VGG16.
- **Tensorflow**: utilizado para cargar imágenes y transformarlas a listas.
- **Scipy**: uso de su función optimizada para calcular la distancia coseno entre vectores, ya que ofrece un rendimiento superior y se reducen los tiempos de espera.
- **Pandas**: utilizado para usar la estructura *Dataframe* que aporta una gran velocidad a la hora de trabajar con grandes cantidades de datos, como es el caso.
- **Matplotlib**: utilizado para generar gráficas que sirvieron para analizar el conjunto de datos.

La interfaz gráfica por otra parte ha sido desarrollada con *TypeScript* junto con las siguientes librerías:

- **React**: este es el *framework* principal utilizado en la interfaz gráfica. Aporta una gran facilidad para poder reutilizar código y no repetirlo. Además cuenta con multitud de librerías que resultan de gran utilizad.
- **Material UI**: librería para el *framework React* que contiene elementos gráficos de UI. Además aporta una gran usabilidad a la web.
- **Axios**: librería que sirve para realizar las peticiones web a la API.
- **React image gallery**: componente de *React* que renderiza una galería de imágenes.

## 4. RESULTADOS

En este apartado se analizan los resultados obtenidos de la recomendación a través de ejemplos de entrada y salida. Además se comparan los resultados con y sin máscara y con y sin el uso de atributos.

### 4.1. Sin máscaras

En un inicio el sistema de recomendación funcionaba sin máscaras. Este sistema funcionaba correctamente pero tenía ciertas desventajas:

- En ocasiones la recomendación se fijaba en el color de fondo de las imágenes en lugar de las prendas en sí.
- Tenía en cuenta los modelos que vestían las prendas.
- Recomendaba prendas con el mismo color.
- Tenía en cuenta marcas de agua en las imágenes.

Aun así se obtienen recomendaciones con un alto nivel de similitud coseno. A continuación se muestran algunos ejemplos de entrada y salida con sus correspondientes porcentajes de similitud.



Fig. 4.1. Recomendaciones sin uso de máscaras

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	75'75
Segunda recomendación	75'42
Tercera recomendación	74'65

TABLA 4.1. TABLA DE SIMILITUD DE RECOMENDACIONES

Imagen original



Recomendaciones



Fig. 4.2. Recomendaciones sin uso de máscaras

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	79'51
Segunda recomendación	77'84
Tercera recomendación	77'23

TABLA 4.2. TABLA DE SIMILITUD DE RECOMENDACIONES

Estas recomendaciones ejemplifican el problema presentado anteriormente. Todas ellas tienen una marca de agua en la esquina inferior derecha, todas ocurren en un desfile de moda, todas están vistiendo el color negro. Este hecho no impide que se obtengan porcentajes de similitud bastante altos, aun así el hecho de no hacer uso de las máscaras presenta los problemas descritos anteriormente.

## 4.2. Con máscaras

En este apartado se llevará a cabo un análisis similar al realizado en el anterior, pero esta vez haciendo uso de las máscaras. El hecho de utilizar máscaras reporta varios beneficios, entre ellos:

- Eliminar las recomendaciones basadas en el fondo de la imagen.
- Eliminar las recomendaciones basadas en el color de la prenda.
- Eliminar las recomendaciones basadas en el modelo que viste las prendas.
- Eliminar recomendaciones basadas en marcas de agua que pueda tener la imagen.

Es decir, las máscaras permiten al sistema recomendar únicamente en base a las prendas, sin tener en cuenta otros factores propios de las imágenes. A continuación, algunos ejemplos.



Fig. 4.3. Recomendaciones con uso de máscaras

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	88'56
Segunda recomendación	88'54
Tercera recomendación	86'57

TABLA 4.3. TABLA DE SIMILITUD DE RECOMENDACIONES



Fig. 4.4. Recomendaciones con uso de máscaras

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	88'99
Segunda recomendación	88'03
Tercera recomendación	87'66

TABLA 4.4. TABLA DE SIMILITUD DE RECOMENDACIONES

Es fácil observar que los resultados de hacer uso de la máscara son claramente beneficiosos para el modelo de recomendación de imágenes. En primer lugar, las prendas recomendadas no tienen el mismo color que la original, tampoco la misma marca de agua o fondo. Además, los modelos no tienen ningún parecido entre sí. Otro indicador de la mejora de la recomendación es el porcentaje de similitud, que sube en torno al 10 %. Y, aunque puede tratarse de un aspecto subjetivo, las prendas son más parecidas entre sí, aunque no tengan el mismo color.

### Similitud Coseno Media (%)

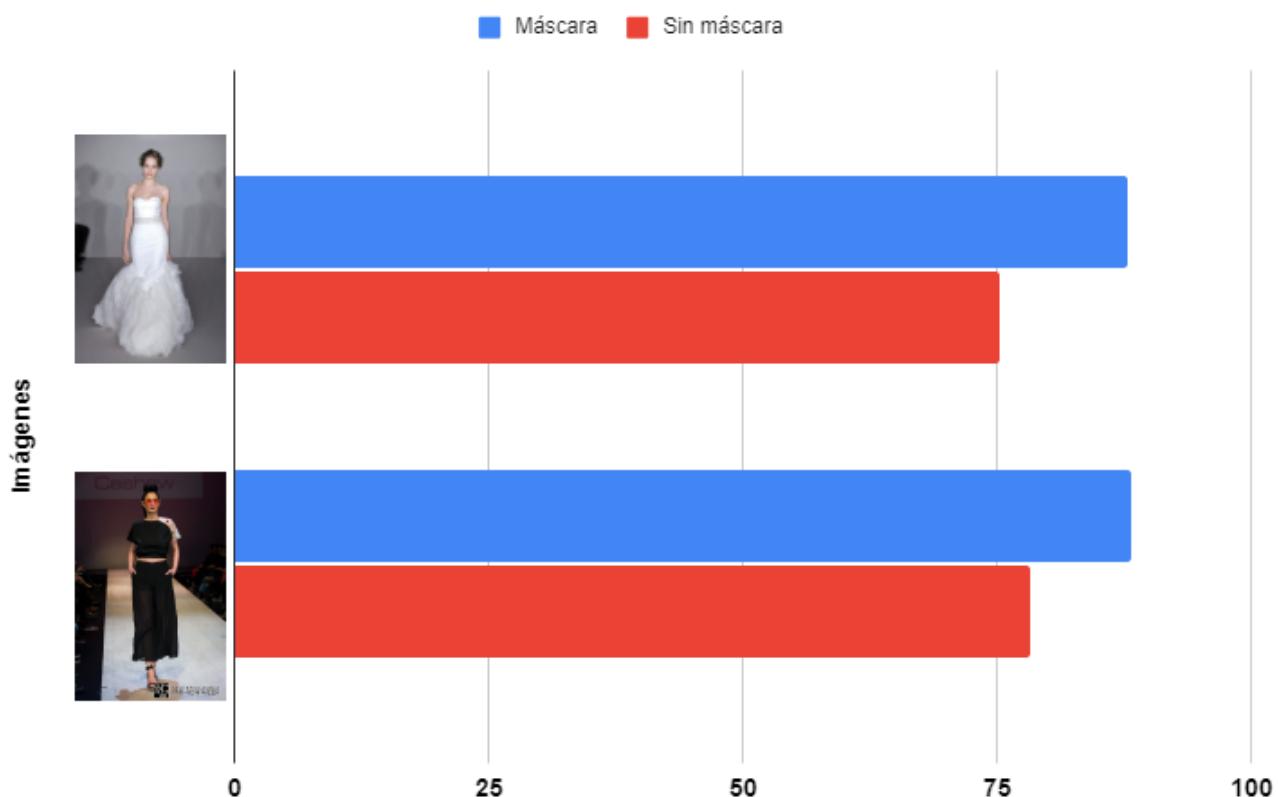


Fig. 4.5. Gráfica de Similitud Coseno

Similitud Coseno Media ( %)	
Con Máscara	Sin Máscara
	87.89
	75.27
	88.23
	78.19

TABLA 4.5. MEDIAS DE SIMILITUD COSENO DE IMÁGENES

### 4.3. Uso de atributos

En este apartado se analiza el uso de los atributos en las recomendaciones y como estos afectan a la calidad de estas. En el siguiente ejemplo se selecciona el atributo “blazer” (chaqueta) para ejemplificar le uso de estos.



Fig. 4.6. Recomendaciones con selección del atributo “blazer”

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	87'79
Segunda recomendación	87'23
Tercera recomendación	86'79

TABLA 4.6. TABLA DE SIMILITUD DE RECOMENDACIONES

Se puede observar que los porcentajes de similitud coseno siguen siendo buenos, a la altura de los anteriores. Además podemos comprobar que todas las recomendaciones tienen el atributo “blazer” solicitado.

Ahora vamos a probar a añadir otro atributo a la consulta. Este atributo será “loose (fit)”.



Fig. 4.7. Recomendaciones con selección de los atributos “blazer” y “loose (fit)”

Recomendación	Porcentaje de Similitud Coseno ( %)
Primera recomendación	84'06
Segunda recomendación	84'05
Tercera recomendación	83'57

TABLA 4.7. TABLA DE SIMILITUD DE RECOMENDACIONES

Por último vamos a añadir el atributo “geometric textile pattern”.



Fig. 4.8. Recomendaciones con selección de los atributos “blazer”, “loose (fit)” y “geometric textile pattern”

<b>Recomendación</b>	<b>Porcentaje de Similitud Coseno ( %)</b>
Primera recomendación	67.12
Segunda recomendación	55.89

TABLA 4.8. TABLA DE SIMILITUD DE RECOMENDACIONES

Como podemos comprobar a través de estos resultados, el uso de los atributos es un arma de doble filo. Resulta muy útil, pero se sacrifica porcentaje de similitud. Esto se debe a que cuantos más atributos queramos menos imágenes hay en el *dataset* que contengan estos atributos, de hecho en el último ejemplo solo hay dos de ellas. Si tuvieramos un *dataset* de un tamaño mucho mayor, como así sería en un ejemplo práctico en la vida real, este problema no se daría tan pronto. Aun así la selección de atributos resulta de gran utilidad.

## 5. CONCLUSIONES

En este capítulo se llevarán a cabo las conclusiones generales del trabajo realizado. Se hará un análisis que consistirá en comprobar si los objetivos iniciales han sido cumplidos.

El objetivo principal de este trabajo era crear una interfaz gráfica que fuera capaz de recomendar imágenes de ropa a partir de, no solo una imagen dada por el usuario, sino también una serie de atributos deseados en esta recomendación. Este objetivo se ha podido cumplir por completo. Además se planteaban los siguientes subobjetivos:

- Desarrollar un modelo de inteligencia artificial que sea capaz de recomendar imágenes en base a atributos multimodales y otra imagen.
- Etiquetar las imágenes a utilizar con sus respectivos atributos.
- Crear una API que permita la comunicación entre el modelo previamente mencionado y una interfaz.
- Diseñar una interfaz intuitiva que permita a los usuarios cargar una imagen, seleccionar los atributos de interés y visualizar los resultados proporcionados por el modelo.

Se ha conseguido construir una aplicación web que lleva a cabo las tareas descritas anteriormente, por lo que se concluye que el trabajo ha sido exitoso.

La arquitectura VGG16 ha resultado ser útil para extraer las características de las imágenes. Esta ha sido efectiva y eficaz. El uso de máscaras, aunque haya ocasionado una solución más compleja debido a la tarea de segmentación, ha resultado aportar claras mejoras al modelo de recomendación. Las imágenes recomendadas dejan de estar influidas por aspectos ajenos a las prendas, como pueden ser el fondo de la imagen, marcas de agua o el aspecto físico de los modelos. Gracias a la combinación de la segmentación de las imágenes para la realización de las máscaras y la extracción de características de las imágenes, para el posterior cálculo de su similitud coseno, se ha podido llegar a obtener unos porcentajes de similitud muy buenos para las recomendaciones.

El uso de los atributos también ha sido un éxito. El modelo es capaz de filtrar el conjunto de datos y recomendar únicamente las prendas que contienen estos. Aun así, cuando se usan excesivos atributos juntos o se seleccionan algunos poco comunes, el modelo puede tener problemas recomendando debido al tamaño, no demasiado grande, del conjunto de datos.

En conclusión, este trabajo ha sido exitoso, ya que ha cumplido a la perfección con los objetivos propuestos, brindando una interfaz limpia que es capaz de servir un sistema de recomendación de imágenes con un gran porcentaje de similitud en las recomendaciones

que aporta. Además, se ha podido introducir el uso de atributos, lo cual permite a los usuarios obtener recomendaciones más precisas y ajustadas a sus deseos. Si se contara con un conjunto de datos más grande esta última funcionalidad sería aún más precisa.

## 6. TRABAJOS FUTUROS

El campo de la visión por computadora podría decirse que es relativamente nuevo. Es por ello que hay mucho margen de mejora y se pueden plantear una variedad de futuros trabajos a partir de este. Algunos de ellos son:

- Ampliar la base de datos sobre la que se trabaja, ya que a mayor cantidad de datos disponibles mayor será la calidad de las recomendaciones.
- Implementación de una red neuronal más enfocada en la ropa y no tan generalista como la arquitectura VGG16. Entrenar una red que se enfoque en prendas en específico podrá ayudar a obtener mejores recomendaciones.
- Añadir *feedback* de usuarios reales mientras se entrena la red podría ser de gran utilidad, ya que la calidad de las recomendaciones es muchas veces subjetiva al gusto del usuario.
- Posibilidad de utilizar la cámara del dispositivo desde el que se acceda a la aplicación web con el fin de hacer más fácil la recomendación
- Permitir al usuario dar más de una imagen para obtener resultados conjuntos
- Permitir al usuario elegir los colores de la recomendación

## **7. PLANIFICACIÓN Y PRESUPUESTOS**

### **7.1. Planificación**

En este apartado se presenta la planificación que se ha llevado a cabo para la realización de este proyecto. La planificación ha consistido en las siguientes etapas:

- **Análisis del conjunto de datos.**

Esta etapa consistió en analizar el conjunto de datos, prepararlo para su uso y ver como se podía hacer uso de este. Se llevaron a cabo estadísticas que ayudaron al posterior diseño de la solución. Esta fase duró siete días.

- **Análisis de las técnicas posibles para la implementación del modelo y de la aplicación web.**

En esta etapa se analizan las diferentes tecnologías posibles para la resolución del problema. Se investiga cuales son las mejores técnicas para el modelo de segmentación, de extracción de características y se exploran las diferentes tecnologías web. Esta fase duró siete días.

- **Diseño de la solución.**

Se diseña la mejor solución a partir del análisis de tecnologías previamente hecho. Esta fase duró once días

- **Desarrollo, pruebas y mejoras.**

Esta es la etapa más larga del proyecto, que consiste en el desarrollo. Una vez se finaliza el desarrollo del proyecto se itera a través de pruebas y se mejora la solución. Esta fase duró cincuenta días.

- **Elaboración de la memoria.**

Esta etapa consiste en documentar el trabajo realizado y duró treinta días.

Dadas estas etapas se puede estimar que el proyecto se ha elaborado en un total de 105 días. Cada uno de estos días se ha trabajado un total de ocho horas, lo que supone que el proyecto ha requerido 840 horas de trabajo. En la siguiente tabla se muestran las fechas detalladas de la planificación del proyecto.

Fase	Duración (días)	Inicio	Fin
<b>Análisis del conjunto de datos</b>	7	15/05/2023	22/05/2023
<b>Análisis de técnicas y algoritmos</b>	7	23/05/2023	30/05/2023
<b>Diseño de la solución</b>	11	31/05/2023	11/06/2023
<b>Desarrollo, pruebas y mejoras</b>	50	12/06/2023	31/07/2023
<b>Elaboración de la memoria</b>	30	01/08/2023	31/08/2023

TABLA 7.1. PLANIFICACIÓN DEL PROYECTO

## 7.2. Presupuestos

### 7.2.1. Coste del personal

Este trabajo ha sido realizado por un solo ingeniero. Un ingeniero informático junior tiene un sueldo promedio en España de 2.208 €/mes [15], lo que supone un sueldo de 13,8 €/hora. Se han trabajado ocho horas diarias por lo que el sueldo diario sería de 110,4 €.

Empleado	Días de trabajo	Sueldo diario	Coste total
Ingeniero junior	105	110,4 €	11.592 €

TABLA 7.2. PLANIFICACIÓN DEL PROYECTO

### 7.2.2. Coste del equipo

El coste del equipo consiste en un ordenador de sobremesa y un portátil, teniendo en cuenta su uso. Para el cálculo del coste estimado teniendo en cuenta su uso se aplica la siguiente fórmula:

$$\text{Coste imputable} = \frac{\text{Tiempo de uso}}{\text{Tiempo de vida}} * \text{Coste total} \quad (7.1)$$

- **Ordenador de sobremesa.** Uso de ochenta horas con un coste total de 450 €
- **Portátil.** Uso de veinticinco horas con un coste total de 849,25 €
- **Teclado.** Uso de ochenta horas con un coste de 95 €
- **Monitor.** Uso de ochenta horas con un coste 240 €

<b>Concepto</b>	<b>Coste total (€)</b>	<b>Horas de uso</b>	<b>Tiempo de vida</b>	<b>Coste imputable (€)</b>
Ordenador de sobremesa	450	80	1.200	30
Portátil	849,25	25	800	26,54
Teclado	95	No aplica	No aplica	95
Monitor	240	No aplica	No aplica	240
<b>Total</b>	<b>1634,25</b>			<b>390,54</b>

TABLA 7.3. COSTES MATERIALES

#### 7.2.3. Costes directos

Los costes directos incluyen los costes del personal y los costes del equipo.

<b>Descripción</b>	<b>Coste (€)</b>
Costes de personal	11.592
Costes de equipo	390,54
<b>Total</b>	<b>11.982,54</b>

TABLA 7.4. COSTES DIRECTOS

#### 7.2.4. Costes indirectos

Los costes indirectos incluyen gastos como agua, luz, gas, alquiler. Se estima que los costes indirectos son un 15 % del total de los costes directos, lo que supone un total de 1797,38 €.

#### 7.2.5. Costes totales

Los costes totales son la suma de todos los costes descritos anteriormente.

<b>Descripción</b>	<b>Coste (€)</b>
Costes directos	11.982,54
Costes indirectos	1.797,38
<b>Total</b>	<b>13.779,92</b>

TABLA 7.5. COSTES DIRECTOS

Por lo tanto se concluye que el coste total del proyecto estimado es de 13.779,92 €.

## BIBLIOGRAFÍA

- [1] A. Orús, *Peso de las ventas moda online sobre el total de los ingresos generados por las ventas de ropa en España de 2012 a 2022*, 2023. [En línea]. Disponible en: <https://es.statista.com/estadisticas/1027850/cuota-de-venta-de-prendas-de-ropa-por-internet-en-espana/>.
- [2] C. Kaeser-Chen et al., *iMaterialist (Fashion) 2020 at FGVC7*, 2020. [En línea]. Disponible en: <https://www.kaggle.com/competitions/imaterialist-fashion-2020-fgvc7/data>.
- [3] *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. 2018. [En línea]. Disponible en: <https://www.boe.es/eli/es/lo/2018/12/05/3>.
- [4] *Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia*. 1996. [En línea]. Disponible en: <https://www.boe.es/eli/es/rdlg/1996/04/12/1/con>.
- [5] *Ley 14/2011, de 1 de junio, de la Ciencia, la Tecnología y la Innovación*. 2011. [En línea]. Disponible en: <https://www.boe.es/eli/es/l/2011/06/01/14/con>.
- [6] M. Xirau, *El sector de la moda en España, en cifras*, 2020. [En línea]. Disponible en: <https://forbes.es/empresas/78279/el-sector-de-la-moda-en-espana-en-cifras/>.
- [7] A. Muñoz, *Cómo los sistemas de recomendación pueden ayudarte a conseguir más ventas*, 2021. [En línea]. Disponible en: <https://blog.saleslayer.com/es/sistemas-recomendacion-ecommerce>.
- [8] *Detección de caras con el software OpenCV*, 1998. [En línea]. Disponible en: [https://commons.wikimedia.org/wiki/File:Face\\_detection\\_example\\_opencv.jpg](https://commons.wikimedia.org/wiki/File:Face_detection_example_opencv.jpg).
- [9] L. González, *¿Qué son las Redes Neuronales Artificiales?* 2021. [En línea]. Disponible en: <https://aprendeia.com/que-son-las-redes-neuronales-artificiales/>.
- [10] L. Chen, F. Yang y H. Yang, “Image-based Product Recommendation System with Convolutional Neural Networks,” 2017. [En línea]. Disponible en: <http://cs231n.stanford.edu/reports/2017/pdfs/105.pdf>.
- [11] F. Ullah, B. Zhang y R. U. Khan, “Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach,” *IEEE Access*, vol. 8, pp. 3308-3318, 2020. doi: [10.1109/ACCESS.2019.2962315](https://doi.org/10.1109/ACCESS.2019.2962315).

- [12] X. Qin et al., “U-2-Net: Going Deeper with Nested U-Structure for Salient Object Detection,” 2022. [En línea]. Disponible en: <https://arxiv.org/pdf/2005.09007.pdf>.
- [13] X. Qin et al., “U-2-Net,” 2022. [En línea]. Disponible en: <https://github.com/xuebinqin/U-2-Net>.
- [14] “VGG: ¿Qué es este modelo? ¡Daniel te lo cuenta todo!,” 2022. [En línea]. Disponible en: <https://datascientest.com/es/vgg-que-es-este-modelo-daniel-te-lo-cuenta-todo>.
- [15] “Salario medio para Ingeniero Informático en España, 2023,” 2023. [En línea]. Disponible en: <https://es.talent.com/salary?job=ingeniero+inform%C3%A1tico>.
- [16] *¿Qué son las redes neuronales convolucionales?* [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>.

## **ANEXO A: ENGLISH SUMMARY**