



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Vestuario Digital con Clasificador de
Prendas Basado en Clasificadores
Estadísticos**

Autora: Yessenia Nicole Cacuango Chirán

Tutor: Vicente Martínez Orga

Madrid, Julio 2021

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Vestuario Digital con Clasificador de Prendas Basado en Clasificadores Estadísticos

Julio 2021

Autora: Yessenia Nicole Cacuango Chirán

Tutor:

Vicente Martínez Orga

Inteligencia artificial

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

Este Trabajo de Fin de Grado consiste en la creación de un vestuario virtual, que permita la visualización de prendas de vestir y con el que se puedan crear conjuntos de ropa, con la característica de que emplea un clasificador de formas para establecer la categoría de una prenda.

Para ello se ha implementado un sistema que permite a un usuario registrado visualizar las prendas que ha guardado por categorías, crear con ellas distintos conjuntos de prendas, agregar la imagen de una nueva prenda al sistema o eliminar una existente, además de crear un nuevo perfil de usuario.

Debido a las funciones que se desea que realice el sistema, su desarrollo e implementación se ha dividido en tres partes: en la primera parte se lleva a cabo la implementación del clasificador de formas, que será el encargado de dar una etiqueta a la prenda de una imagen; la segunda parte es el desarrollo de un sistema de gestor de base de datos en el que se almacenarán todos los usuarios registrados, las prendas que se han clasificado y los conjuntos creados; y por último, la interfaz gráfica, gracias a la cual el usuario puede realizar distintas acciones en el sistema. Además, será quién interactúe con las otras dos partes que componen el sistema.

La implementación tanto de la interfaz como del clasificador se ha realizado en lenguaje Python y se han empleado múltiples librerías de Machine Learning, además de MySQL para gestionar la base de datos.

Abstract

This Final Degree Project consists of the creation of a virtual wardrobe, which allows the visualisation of garments and with which clothes sets can be created, with the characteristic that it uses a shape classifier to establish the category of a garment.

For this purpose, a system has been implemented that allows a registered user to visualise the garments he/she has saved by categories, create different sets of garments with them, add the image of a new garment to the system or delete an existing one, as well as create a new user profile.

Due to the functions that the system is intended to perform, its development and implementation has been divided into three parts: in the first part, the implementation of the shape classifier is carried out, which will be in charge of giving a label to the garment in an image; the second part is the development of a database management system in which all registered users, the garments that have been classified and the sets created will be stored; and finally, the graphic interface, thanks to which the user can carry out different actions in the system. Moreover, it will be the one who interacts with the other two parts that make up the system.

The implementation of both the interface and the classifier has been done in Python language and multiple Machine Learning libraries have been used, as well as MySQL to manage the database.

Tabla de contenidos

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Características y objetivos del sistema.....	2
2	Estado del Arte	3
2.1	Sistema de reconocimiento de patrones	3
2.1.1	Construcción de un clasificador	3
2.1.2	Universo de trabajo del proyecto.	5
2.1.3	Modelo del clasificador del proyecto.	9
2.1.4	Rendimiento del clasificador del proyecto.....	10
2.2	Entorno de programación	15
3	Desarrollo del Clasificador	16
3.1	Implementación del algoritmo de clasificación	16
3.2	Entrenamiento y evaluación del clasificador	19
3.3	Comprobación del funcionamiento del clasificador	21
3.3.1	Obtención de la imagen.....	21
3.3.2	Binarización.....	22
3.3.3	Detección de la silueta de la prenda en la imagen.	23
3.3.4	Llamada al clasificador	25
4	Desarrollo de la Base de datos.....	27
4.1	Desarrollo del modelo	27
4.2	Implementación de la base de datos	29
4.3	Comunicación entre la base de datos y la interfaz	32
5	Desarrollo de la Interfaz.....	35
5.1	Ventana de inicio de sesión	36
5.2	Ventana para crear un nuevo usuario	37
5.3	Ventana principal.....	38
5.3.1	Ventana de las prendas.....	39
5.3.2	Ventana de los conjuntos	40
5.3.3	Ventana añadir una nueva prenda.....	43
5.3.4	Ventana eliminar una prenda	46
6	Desarrollo de las pruebas y ejecuciones	48
6.1	Crear un nuevo perfil	48
6.1.1	Nombre y contraseñas correctas	48
6.1.2	Se dejan campos vacíos.....	49

6.1.3	Se introduce un nombre de usuario existente	50
6.2	Acceder con un nombre de usuario y contraseña	51
6.2.1	Se accede correctamente	51
6.2.2	No se introducen correctamente los parámetros.....	52
6.3	Añadir una nueva prenda al armario.....	53
6.3.1	Se añade una prenda correctamente	53
6.3.2	No se selecciona ninguna imagen.....	55
6.4	Visualizar las prendas de una categoría	55
6.5	Crear de un nuevo conjunto	56
6.6	Modificar de un conjunto existente.....	58
6.7	Eliminar prendas del armario.....	60
6.7.1	Se elimina una prenda que no conforma un conjunto	60
6.7.2	Se elimina una prenda que conforma un conjunto	61
6.7.3	No se selecciona ninguna prenda	62
7	Resultados y conclusiones	63
8	Análisis de Impacto	64
9	Bibliografía	65

Tabla de ilustraciones

Ilustración 1	Diseño de un sistema de reconocimiento de formas [7]	4
Ilustración 2	Muestra del contenido del dataset Fashion-MNIST [8].....	6
Ilustración 3	Distribución del dataset Fashion-MNIST [8]	7
Ilustración 4	Agrupación de las clases del dataset Fashion-MNIST	8
Ilustración 5	Ejemplo sesgo-varianza [7]	12
Ilustración 6	Evolución varianza – sesgo - error [12].....	13
Ilustración 7	Evolución sesgo-varianza en función de los datos de entrenamiento y evaluación [12].....	13
Ilustración 8	Método de partición	14
Ilustración 9	Método de exclusión.....	14
Ilustración 10	Método de validación cruzada	15
Ilustración 11	Representación de la matriz que contiene los valores de la función de discriminación	18
Ilustración 12	Prenda de original.....	22
Ilustración 13	Prenda en escala de grises	22
Ilustración 14	Resultado de aplicar distintos umbrales [20]	23
Ilustración 15	Prenda binarizada.....	23
Ilustración 16	Contornos detectados en la prenda	24
Ilustración 17	Contornos seleccionado de la prenda	25
Ilustración 18	Prenda con etiqueta	26
Ilustración 19	Diagrama entidad-relación de la base de datos	29
Ilustración 20	Esquema comunicación Usuario-Interfaz-BBDD-Clasificador	35
Ilustración 21	Ventana de inicio de sesión.....	36

Ilustración 22 Ventana para crear un nuevo usuario.....	37
Ilustración 23 Ventana principal	38
Ilustración 24 Ventana de las prendas	40
Ilustración 25 Ventana de los conjuntos.....	41
Ilustración 26 Ventana de los conjuntos.....	42
Ilustración 27 Ventana de los conjuntos.....	43
Ilustración 28 Ventana añadir nueva prenda.....	44
Ilustración 29 Ventana añadir nueva prenda.....	45
Ilustración 30 Ventana eliminar prenda	46
Ilustración 31 Nuevo perfil - Nombre y contraseña correctas	48
Ilustración 32 Nuevo perfil – Mensaje del sistema.....	48
Ilustración 33 Nuevo perfil - Tabla usuarios	49
Ilustración 34 Nuevo perfil - Nombre vacío	49
Ilustración 35 Nuevo perfil - Contraseña vacía	49
Ilustración 36 Nuevo perfil - Mensaje del sistema	50
Ilustración 37 Nuevo perfil - Usuario existente	50
Ilustración 38 Nuevo perfil - Mensaje del sistema	50
Ilustración 39 Acceder - Acceso correcto.....	51
Ilustración 40 Acceder - Ventana principal.....	51
Ilustración 41 Acceso - Nombre erróneo	52
Ilustración 42 Acceso - Contraseña errónea.....	52
Ilustración 43 Acceder - Mensaje del sistema	52
Ilustración 44 Nueva prenda - Incio.....	53
Ilustración 45 Nueva prenda - Contenido de la carpeta Resources.....	54
Ilustración 46 Nueva prenda - elección de etiqueta.....	54
Ilustración 47 Nueva prenda - Mensaje del sistema	55
Ilustración 48 Nuevo prenda - Tabla prendas	55
Ilustración 49 Nueva prenda - Mensaje del sistema	55
Ilustración 50 Visualización de las sudaderas	56
Ilustración 51 Nuevo conjunto - Tabla Prendas	57
Ilustración 52 Nuevo conjunto - Conjunto 1	57
Ilustración 53 Nuevo conjunto - Mensaje del sistema	58
Ilustración 54 Nuevo conjunto - Tabla Conjuntos	58
Ilustración 55 Modificar conjunto – Conjunto 1	59
Ilustración 56 Nuevo conjunto - Mensaje del sistema	59
Ilustración 57 Modificar conjunto - Tabla Conjuntos	59
Ilustración 58 Eliminar prenda – Prenda no perteneciente a un conjunto....	60
Ilustración 59 Eliminar prenda - Mensaje del sistema	60
Ilustración 60 Eliminar prenda - Tabla Prendas	61
Ilustración 61 Eliminar prenda – Prenda perteneciente a un conjunto.....	61
Ilustración 62 Eliminar prenda - Mensaje del sistema	62
Ilustración 63 Eliminar prenda - Mensaje del sistema	62

Tabla de tablas

Tabla 1 Clasificador - Porcentaje de aciertos.....	20
---	----

1 Introducción

1.1 Motivación

La moda tiene una alta importancia en nuestra sociedad dado que siempre ha sido una de las maneras en las que la sociedad ha podido expresarse y mostrar la pertenencia a un grupo social.

Hoy en día esto no ha cambiado, dado que el gasto medio en ropa por persona cada año, desde 2013 hasta 2019, ha aumentado [1] y tiene previsiones de seguir creciendo.

En función de los datos proporcionados por el INE [2] se puede determinar que las mujeres comprendidas entre 16 y 29 años son las que más dinero han invertido en vestimenta durante el año 2019. Además, según datos de Eurostat [3], la ropa encabeza la lista de bienes comprados por internet durante el año 2020, donde el rango de edad que predomina es el comprendido entre los 16 y 25 años, seguido por las personas que tienen entre 24 y 54 años.

En consecuencia, se puede deducir que las personas más interesadas en la ropa se encuentran entre los 16 y 29 años. Esto puede estar directamente relacionado con el hecho de que la segunda red social más usada por gente comprendida entre 18 y 29 años sea Instagram [4] y, además, la segunda con más publicidad, por detrás de Facebook. Hay que añadir que el producto/servicio por el que más se decantaron comprar a raíz de la publicidad en las redes sociales fue ropa, calzado o complementos [5].

En vista de la importancia que se le da a la vestimenta, el repertorio de prendas que tiene cada persona tiende a ir en aumento, dando lugar a fondos de armarios muy grandes. Además, sumado a esto, cada cierto tiempo aparece una nueva moda que hace que se adquieran nuevas prendas.

El volumen de la ropa puede ocasionar que no se conozcan bien toda las prendas y complementos que se poseen haciendo que, en ocasiones, se compre ropa parecida a la que ya se tiene o prendas poco combinables. Incluso, que se creen los mismos conjuntos en bucle al no tener una visión general de todo, dando lugar a la clásica frase de “No tengo nada que ponerme”.

A partir de estas situaciones, surgió la motivación para la realización del diseño e implementación de un sistema que ayude a evitarlas. Siendo más específicos, un sistema que permita consultar en cualquier momento las prendas que un usuario tiene en su armario de una manera muy cómoda y en la que se pueda planificar conjuntos.

Actualmente existen aplicaciones similares en el mercado, pero en lo que se diferencia esta propuesta es que hace uso de un clasificador para realizar el etiquetado de prendas, en lugar de un etiquetado manual. Con esto se busca dar un paso en dirección al automatizado de las tareas, pensando en que en un futuro hasta la creación de los conjuntos se pueda realizar de manera automática.

1.2 Características y objetivos del sistema

El objetivo de este trabajo es la implementación de un sistema que emule un vestuario o armario, permitiendo la clasificación y el almacenamiento de imágenes de prendas de vestir.

El sistema será capaz de determinar el tipo de prenda que hay en una imagen, empleando un clasificador, y almacenarla, permitiendo de esta forma que el usuario pueda consultar todas las prendas que ha etiquetado y crear conjuntos con elementos de distintas categorías.

La aplicación podrá etiquetar cualquier imagen, siempre y cuando, el elemento que represente pertenezca a una de las siguientes categorías: camiseta de manga corta, pantalón, sudadera, vestido, abrigo, zapatillas, botas o bolso.

Para ser más concretos, el usuario podrá realizar desde la interfaz del sistema las acciones que se detallan a continuación:

- Acceder al sistema con un nombre de usuario.
- Dada la imagen de una prenda perteneciente a una de las categorías mencionadas anteriormente, obtener la categoría a la que pertenece y almacenarla en su perfil.
- Visualizar las imágenes etiquetadas y almacenadas en función de la categoría a la que pertenecen
- Agrupar imágenes de distintas categorías creando así conjuntos de ropa, que se almacenarán para su posterior consulta.

De modo que, en función de las acciones que el sistema quiere proporcionar, la lista de objetivos concretos de este proyecto son los siguientes:

- Desarrollo de un algoritmo de clasificación óptimo para el clasificador.
- Desarrollo de una base de datos
- Desarrollo de una interfaz gráfica para el usuario
- Implementación de la comunicación entre la interfaz del usuario con la base de datos y el clasificador.
- Ejecución, correcciones de errores y validación de las pruebas convenientes.

El desarrollo del sistema se va a llevar a cabo mediante el uso del lenguaje de programación Python. Se emplearán librerías de Machine Learning para el desarrollo del clasificador, para la gestión de la base de datos se empleará MySQL y para el desarrollo de la interfaz, la librería Tkinter.

2 Estado del Arte

En este apartado se van a comentar los conceptos implicados en el trabajo, así como las tareas que han sido necesarias llevar a cabo antes de poder empezar con la implementación del sistema.

2.1 Sistema de reconocimiento de patrones

Las personas se cruzan diariamente con otras personas por la calle a la cuáles no siempre saludan, en un parking con múltiples coches se puede diferenciar cuál es el que nos pertenece y cuál no, cuando se busca algo en un bolso, sin necesidad de mirar, se sabe si el objeto que se toca es el que se quiere o no, e incluso en la frutería se puede intuir cuál de todas las manzanas tiene más probabilidades de saber mejor.

La explicación de que se pueda hacer esto es que todo a nuestro alrededor está formado por una serie de características únicas, que hacen que, en función de la información previa de la que se dispone, se pueda asignar una etiqueta o categoría a ese elemento.

Por ejemplo, una persona sabe perfectamente que la persona con la que se acaba de cruzar no es conocida porque tiene asociado al conjunto de todos sus rasgos una etiqueta. Sin embargo, le puede resultar parecida porque su altura, color de pelo y forma de vestir coincide con alguien que conoce, pero existirá alguna característica que haga que la etiquete como desconocido.

Este tipo de reconocimiento de patrones es el que se quiere que sea capaz de realizar un sistema de reconocimiento de patrones o clasificador de formas.

2.1.1 Construcción de un clasificador

Algunos de los términos que se emplearán durante los siguientes apartados relacionados con los sistemas de reconocimientos de patrones son los siguientes:

- Universo de trabajo (UT): conjunto de objetos disponibles para la construcción del clasificador. Ej: Todos los alumnos pertenecientes al grado de ingeniería Informática de la UPM.
- Clase: agrupación de objetos del UT. Ej: Alumnos que solo cursan asignaturas de primero, alumnos que solo cursan asignaturas de segundo, ...
- Objeto: un elemento del UT: Ej: un alumno perteneciente al grado de ingeniería Informática de la UPM.
- Vector de características discriminantes: conjunto de descriptores que representan un objeto y que permiten clasificarlo. Ej: Lista de asignaturas en las que se ha matriculado un alumno.
- Función discriminante $f(x)$: función que decide la pertenencia de un objeto a una clase al combinar las componentes del vector de características. Pueden ser de dos tipos: basadas en la regionalización o en la distancia. Ej: Si un alumno esta matriculado en más de 5 asignaturas de un curso (Primero), ese alumno pertenecerá a ese curso (alumno de primero) [6].

Un clasificador de formas se construye de la siguiente manera:

- En primer lugar, se obtiene un universo de trabajo representativo para el tipo de clasificador que se quiere construir. Para este trabajo en concreto, dado que lo que se busca es un clasificador que, en función de la imagen de una prenda diga a qué categoría de ropa pertenece. Lo ideal es buscar un conjunto de datos (data set) que contenga múltiples imágenes de prendas etiquetadas.
- Una vez se tiene un UT, se eligen qué características del vector de características son significativas para el problema. Por ejemplo, si entre las características del UT que se elija para este proyecto se encuentra el color de la prenda, se eliminaría dado que no es un dato relevante.
- El siguiente paso consiste en elegir un modelo que se adapte a las características del UT elegido y a las clases en las que se quiere dividir el universo de trabajo, dado que con ese modelo se construirán las funciones discriminantes.
- Para finalizar, se elige el modo de entrenamiento y evaluación del clasificador. Tras la evaluación de este, se decide si el porcentaje de aciertos que proporciona es adecuado para el problema o no. En el caso de que no lo sea, se reevalúa el vector de características y los parámetros internos del clasificador. Si los resultados son los suficientemente bueno, se puede usarlo para la clasificación de datos no etiquetados.

En la siguiente imagen se representa de manera esquemática lo comentado anteriormente:

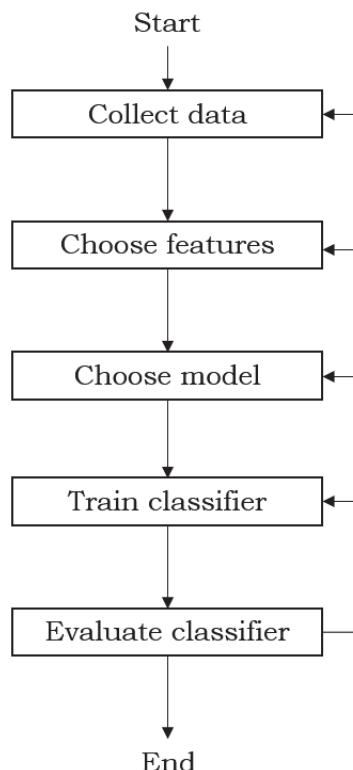


Ilustración 1 Diseño de un sistema de reconocimiento de formas [7]

Como se puede ver en la ilustración 1, la construcción de un clasificador es cíclica, dado que, una vez se obtienen los resultados de la evaluación del clasificador, se puede plantear la elección de datos, características o modelo para mejorar los resultados.

Hay que destacar que no existe un algoritmo o modelo de clasificador superior a otro, la mejor elección dependerá de la información previa de la que se disponga (tipo de problema a resolver), la distribución de los datos del UT, la cantidad de datos de los que se disponga para el entrenamiento del clasificador y los costes que implique.

Partiendo de esto, en los siguientes subapartados se hará un estudio de los componentes del clasificador para construir el más adecuado.

2.1.2 Universo de trabajo del proyecto.

Como se ha mencionado en el apartado 2.1.1, el primer paso es la recolección de datos.

Las prendas que se quiere que se clasifiquen, como mínimo, en este trabajo son camisetas de manga corta, pantalones y sudaderas, por lo que el UT que se utilice debe tener, por lo menos, estas prendas.

Se puede bien construir un UT recopilando imágenes de estas prendas, recogiendo sus características y etiquetándolas manualmente u optar por un conjunto de datos ya construido.

En este proyecto se descartó la primera opción por varios motivos: el primero es que con cuantas más imágenes se construya y entrene el clasificador, mayor rendimiento se obtendrá, por lo que se tendrían que recopilar no menos de 500 fotos, preferiblemente de bancos de fotos ya que son imágenes sin derechos de autor, y etiquetarlas. Esto es una tarea que, dado la gran cantidad de imágenes requeriría demasiado tiempo; el segundo motivo es que, para realizar el entrenamiento y la evaluación habría que reutilizar las imágenes al tener un UT tan limitado, lo que puede dar lugar a una baja tasa de aciertos sobre imágenes desconocidas; por último, se estarían limitando las capacidades del clasificador ya que, si en un futuro se quisiese que el sistema clasifique en más clases, sería necesario realizar de nuevo una recopilación manual de imágenes, tarea con un alto coste en tiempo.

El conjunto de dato elegido para construir el clasificador en este trabajo es el denominado Fashion-MNIST [8]. Se trata de un conjunto de imágenes de artículos de la página Zalando cuyas etiquetas se dividen en las categorías de camiseta de manga corta, pantalón, sudadera, vestido, abrigo, sandalias, camisa, zapatillas, bolso y botas. En total contiene 70000 muestras divididas en 60000 imágenes para el entrenamiento de un clasificador y 10000 para su evaluación.

El número de muestras que proporciona es muy superior al que se podría conseguir manualmente y, además, dispone de categorías que en un principio no se habían considerado, por lo que se podría ampliar fácilmente las categorías en la que se clasifican las prendas. Además, se trata de un conjunto de datos que emplea una licencia MIT, lo que permite usarla y retocarla libremente.

Adicionalmente, en la página web en la que se encuentra el dataset [8], se proporcionan instrucciones de como descargar los datos e incluirlos en un programa, lo cual facilita en gran medida el trabajo.

A continuación, se muestra una imagen con algunos de los elementos que conforman el dataset:

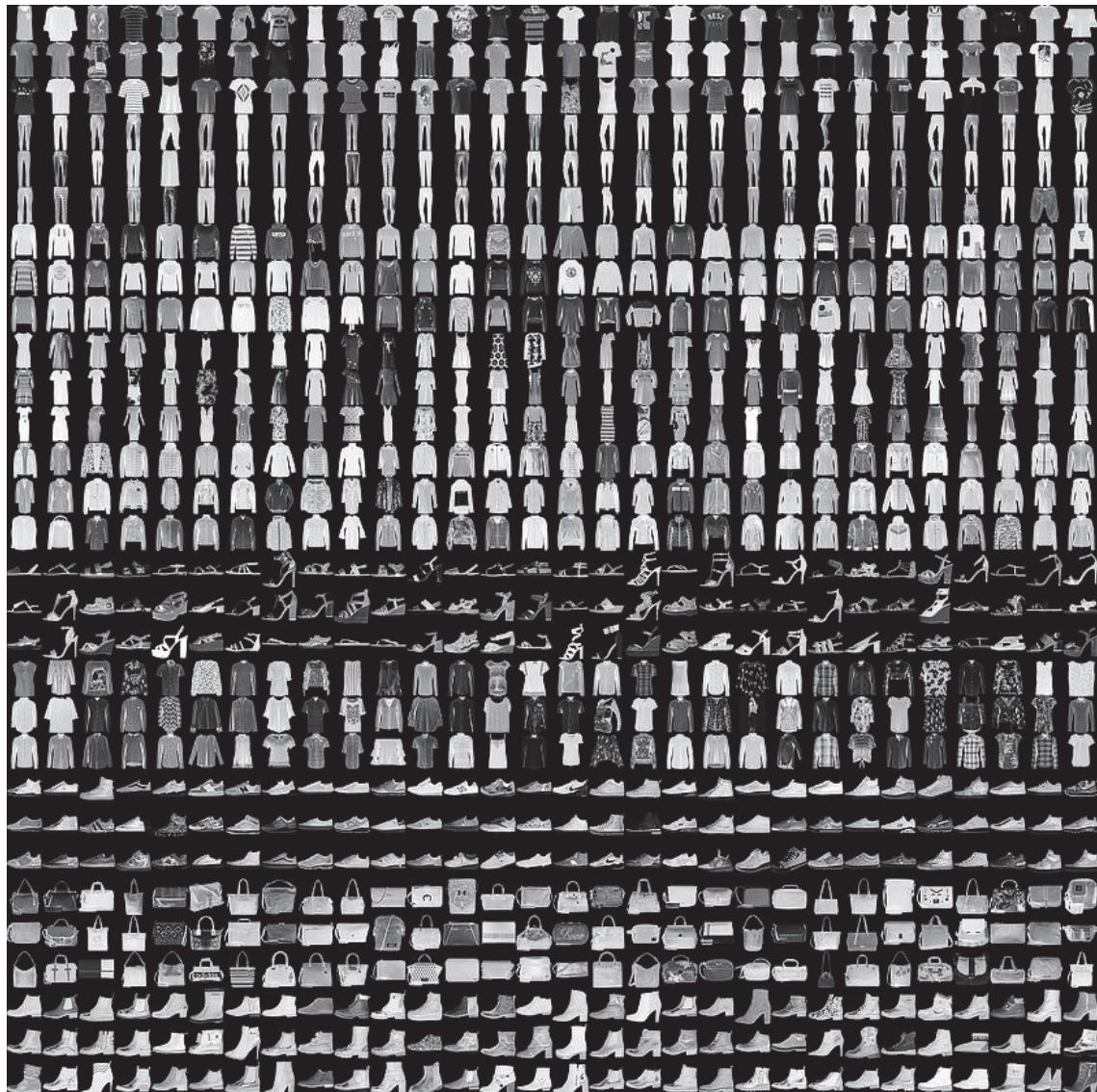


Ilustración 2 Muestra del contenido del dataset *Fashion-MNIST* [8]

La ilustración 2 tiene varias filas de fotos, donde durante tres filas se tienen imágenes pertenecientes a una categoría. Como bien se puede apreciar, dentro de cada clase se tiene una serie de imágenes con distintas formas. Esto hará que el tipo de prenda de una clase que se pueda clasificar sea mayor, por ejemplo, en el caso de las camisetas de manga corta, una camiseta de tirante también se considerá manga corta.

En la ilustración 3 se muestra una imagen de la distribución de datos que sigue el data set:

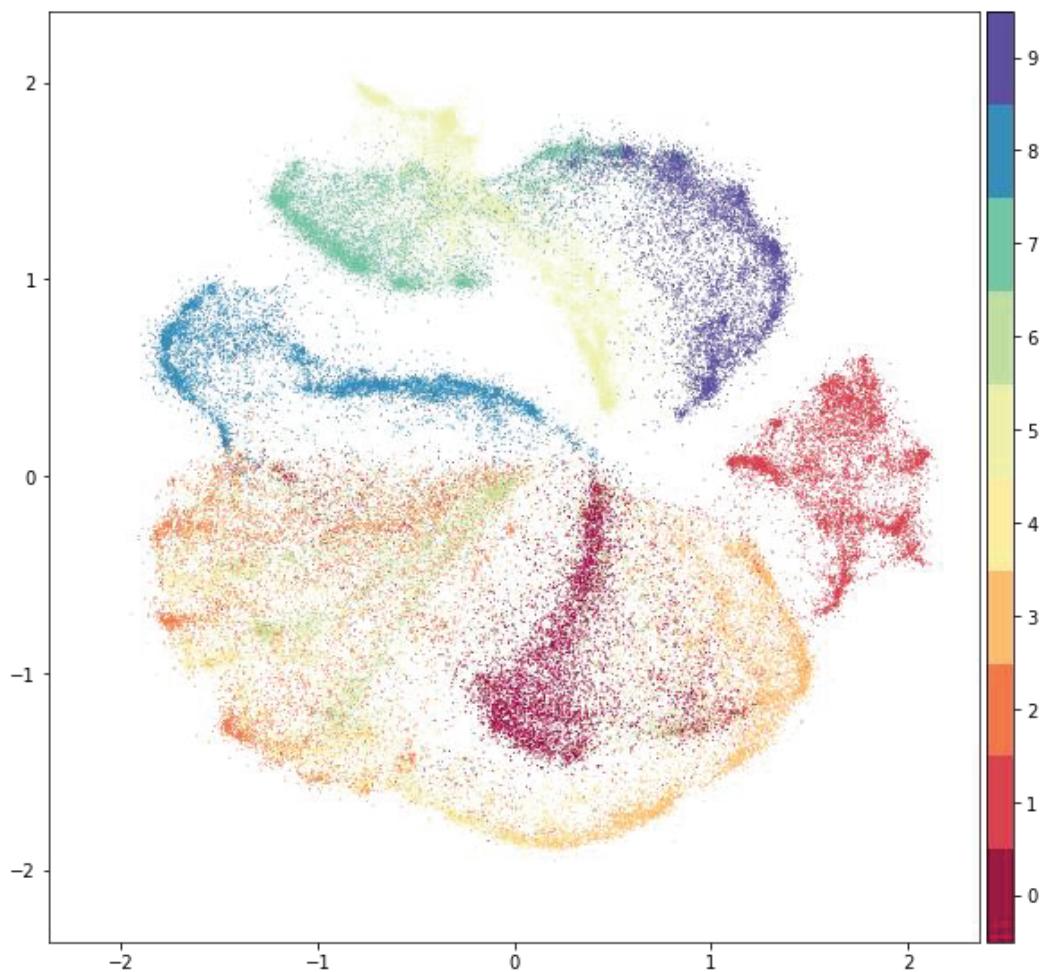


Ilustración 3 Distribución del dataset Fashion-MNIST [8]

La ilustración esta compuesta por numerosos puntos de distintos colores, donde cada uno de ellos representa el objeto de una clase. Cada color se asocia a un número y este a una clase. La asociación de número-clase es la siguiente:

- 0: camiseta de manda corta.
- 1: pantalón.
- 2: sudadera.
- 3: vestido.
- 4: abrigo.
- 5: sandalias.
- 6: camisa.
- 7: zapatillas.
- 8: bolso.
- 9: botas.

Como bien se puede apreciar, los objetos de cada clase se agrupan en una zona concreta pero también se mezclan unos entre otros.

En la ilustración 4 se han marcado las zonas en las que más elementos de una clase se agrupan.

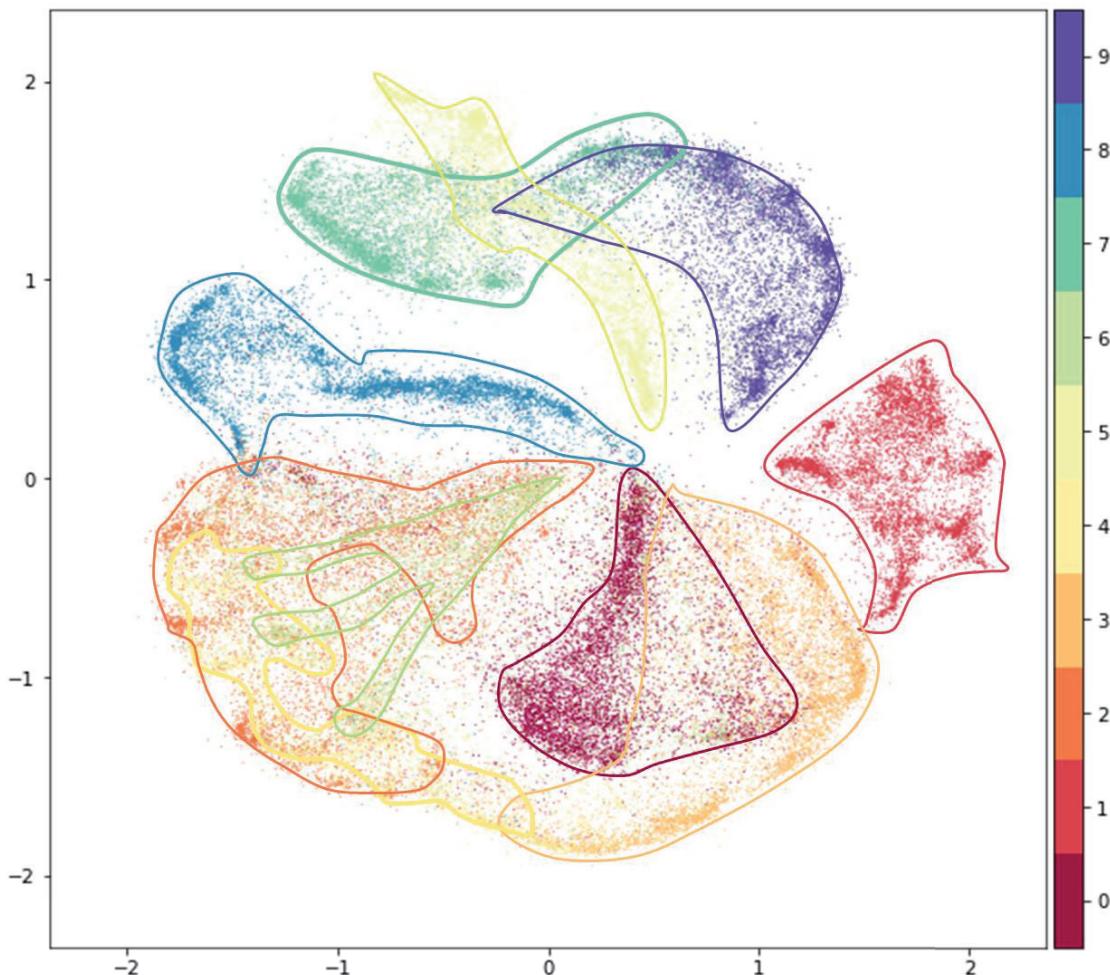


Ilustración 4 Agrupación de las clases del dataset Fashion-MNIST

En esta ilustración se puede ver que hay clases considerablemente bien definidas ya que sus objetos apenas interfieren con otras clases, como puede ser la clase 1 (pantalón) o la clase 8 (bolso), mientras que por otro lado hay clases cuyos objetos se encuentran mezclados unos con otros, como ocurre con las clases 2 (sudadera), 4 (abrigo) y 6 (camisa).

En base a esto, se ha decidido prescindir de la clase 6 ya que sus elementos interfieren con más de una clase y no se diferencia ninguna zona con una agrupación densa de elementos. Esto puede indicar que las características de esta clase son muy variadas, a diferencia de la clase 2, que se pueden ver ciertas zonas con mayor densidad.

Por otro lado, también se ha eliminado la clase de las sandalias (5) dado que interfiere entre dos clases y no es una de las categorías que como mínimo se quería que tuviese el clasificador.

Tras estas modificaciones, se obtiene el dataset con el que se va a construir el clasificador del este proyecto.

2.1.3 Modelo del clasificador del proyecto.

En este apartado se van a comentar los diversos tipos de clasificadores que existen y cuál es el que se ha elegido para este proyecto.

En primer lugar, las clasificaciones se dividen en supervisada o no supervisadas.

- Clasificación supervisada: se caracteriza en que el número de clases se conoce a priori y a qué clase de esas pertenece cada elemento del UT. Dentro de este modelo, a la hora de decidir la clase a la que pertenece un nuevo objeto se pueden hacer dos cosas: construir una función de pertenencia a la clase o una función discriminante que divida a las clases [6].

Es altamente probable que los objetos de una clase estén cerca o se mezclen con los objetos de otra, lo que da lugar a límites de decisión en los que una clasificación errónea es muy probable, de ahí que, si un objeto que se está clasificando se encuentra en una de estas fronteras, se pueda decidir rechazarlo o retenerlo hasta tener más información que haga que se decante por una clase u otra [9].

- Clasificación no supervisada: en este tipo de clasificación no se conoce el número de clases que existen y los datos que conforman el UT no están etiquetados. La forma en la que trabaja un clasificador de este estilo es encontrando grupos en los datos cuyas características distinga a un grupo de otro [9]. Para realizar estas agrupaciones se pueden usar distintos algoritmos de agrupación que darán lugar a distintas agrupaciones. A menudo el usuario establece un número aproximado de agrupaciones que cree que pueden surgir [7], pero no sabrá las clases que existen con certeza hasta que se realicen las agrupaciones y habrá ocasiones en las que este número diste de la realidad.

En base a lo comentado anteriormente, el tipo de clasificador que se empleará es un clasificador con aprendizaje supervisado dado que se sabe con certeza el número de clases que se tiene y el UT seleccionado está etiquetado.

En el UT elegido existen objetos que se mezclan unos con otros, lo que hace que no se cumpla la hipótesis determinista sobre la distribución de las muestras, por lo que se necesitará un clasificador que realice el modelado estadístico de la distribución de las muestras en cada clase [10].

Existen dos grupos de técnicas que permiten modelar este tipo de distribución:

- Técnicas paramétricas: se caracterizan en que la función discriminante se puede obtener con las funciones de densidad de probabilidad de cada clase y el uso del Teorema de Bayes. Las funciones de densidad de probabilidad se estiman con los objetos del UT dado que son una representación de la distribución de los valores de los objetos de una clase.
- Técnicas no paramétricas: se emplean en los casos en los que se desconoce por completo las distribuciones por lo que no se puede calcular su distribución a priori ya que esta se calcula en base a los datos conocidos [7].

En este proyecto, las funciones discriminantes se van a obtener mediante técnicas paramétricas. A partir de ahora, funciones discriminantes se va a denotar también como $f(x)$, siendo x el vector de características de un objeto.

Se tendrán tantas funciones discriminantes como clases haya, por lo que cada función se referenciará como $f_i(x)$ siendo i un valor número comprendido entre 1 y el número total de clases. Las posibles clases se denotarán como C_i , siendo nuevamente i un valor número comprendido entre 1 y el número total de clases.

La forma en la que el clasificador asignará un vector de características x a una clase C_i será cuando el valor de $f_i(x)$ sea mayor al valor de $f_j(x)$, siendo i un valor distinto a j , y j un valor numérico comprendido entre 1 y el número total de clases.

Las funciones de discriminación se pueden calcular siguiendo la regla de decisión de Bayes para el mínimo error. Esta regla establece que el clasificador asignará un objeto a aquella clase cuya probabilidad a posteriori sea máxima:

$$P(C_i|x) > P(C_j|x) \quad j = 1, \dots, C; i \neq j$$

En otras palabras, $x \in C_i \leftrightarrow i = \arg \max_j \{f_j(x)\}$.

La función de discriminación se podría escribir como:

$$f_i(x) = P(C_i|x) = p(x|C_i) * P(C_i) = \ln p(x|C_i) + \ln P(C_i).$$

La probabilidad condicional (o densidad condicional):

$$p(x|C_i) = p(C_i) * \frac{p(x|C_i)}{p(x)}$$

Dado que se sigue un método paramétrico, se estimará la densidad condicionada siguiendo un enfoque predictivo (Bayesiano), de esta forma se tendrá en cuenta la variabilidad muestral de los datos del UT, por lo que $p(x|C_i) = p(x|C_i, \theta)$, donde θ representa a todos los parámetros. Para simplificar la notación se escribirá $p(x|\theta)$ en lugar de $p(x|C_i, \theta)$. Se supondrá que las densidades siguen una distribución normal, por lo que $p(x|\theta) \sim N(x|\mu, \Sigma)$, donde μ representa a la media de la clase y Σ a la covarianza de la clase. De esta forma se obtiene que $f_i(x) = N(x|\mu, \Sigma) * P(C_i)$.

Para evitar tener términos constantes en la fórmula, se aplica la función del \ln a todo $f_i(x)$, obteniendo finalmente la función de pertenencia a cada clase que empleará el clasificador [10]:

$$f_i(x) = -\frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (x - \mu_i)^T * \Sigma_i^{-1} * (x - \mu_i) + \ln P(C_i)$$

2.1.4 Rendimiento del clasificador del proyecto.

La mejor manera de medir el rendimiento de un clasificador es empleando el error cuadrático medio, que indica la diferencia entre los valores estimados y los que el clasificador estima, en otras palabras, como de cerca se encuentra la función de decisión del conjunto de datos y se puede expresar con la siguiente fórmula [11]:

$$ECM = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

A partir de esta fórmula y con el sesgo y la varianza, se puede evaluar un clasificador. El sesgo es distancia del modelo medio a la solución y tendrá un valor alto si el modelo no se puede ajustar a los datos; por otro lado, la varianza mide la especificidad, es decir, los cambios en el modelo de datos de entrenamiento y un valor alto implica que el modelo contiene ruido.

Ajustando la fórmula de ECM al contexto de que $F(x)$ es la función de regresión que se quiere hallar (valores que el clasificador estima); $g(x|D_i)$, la función de regresión (valores estimados) y D , el conjunto de datos, se obtiene [12]:

$$ECM = \varepsilon_{Di} [(g(x|D_i) - F(x))^2] = \underbrace{\varepsilon_{Di} [(g(x|D_i) - \bar{g})^2]}_{\text{Varianza}} + \underbrace{(\bar{g} - F(x))^2}_{\text{Sesgo}}$$

Siendo $\bar{g} = \varepsilon_{Di}[g(x|D_i)]$

Varianza

Sesgo

Con esto se deduce que un modelo con varios parámetros se ajustará mejor a un problema, dado que dará lugar a una varianza alta, mientras que, si tienen pocos, la varianza será pequeña pero el sesgo alto por lo que no se podrá ajustar bien.

A continuación, se muestra en una ilustración un ejemplo de cómo cambia la función de regresión según qué valor tengan el sesgo y varianza:

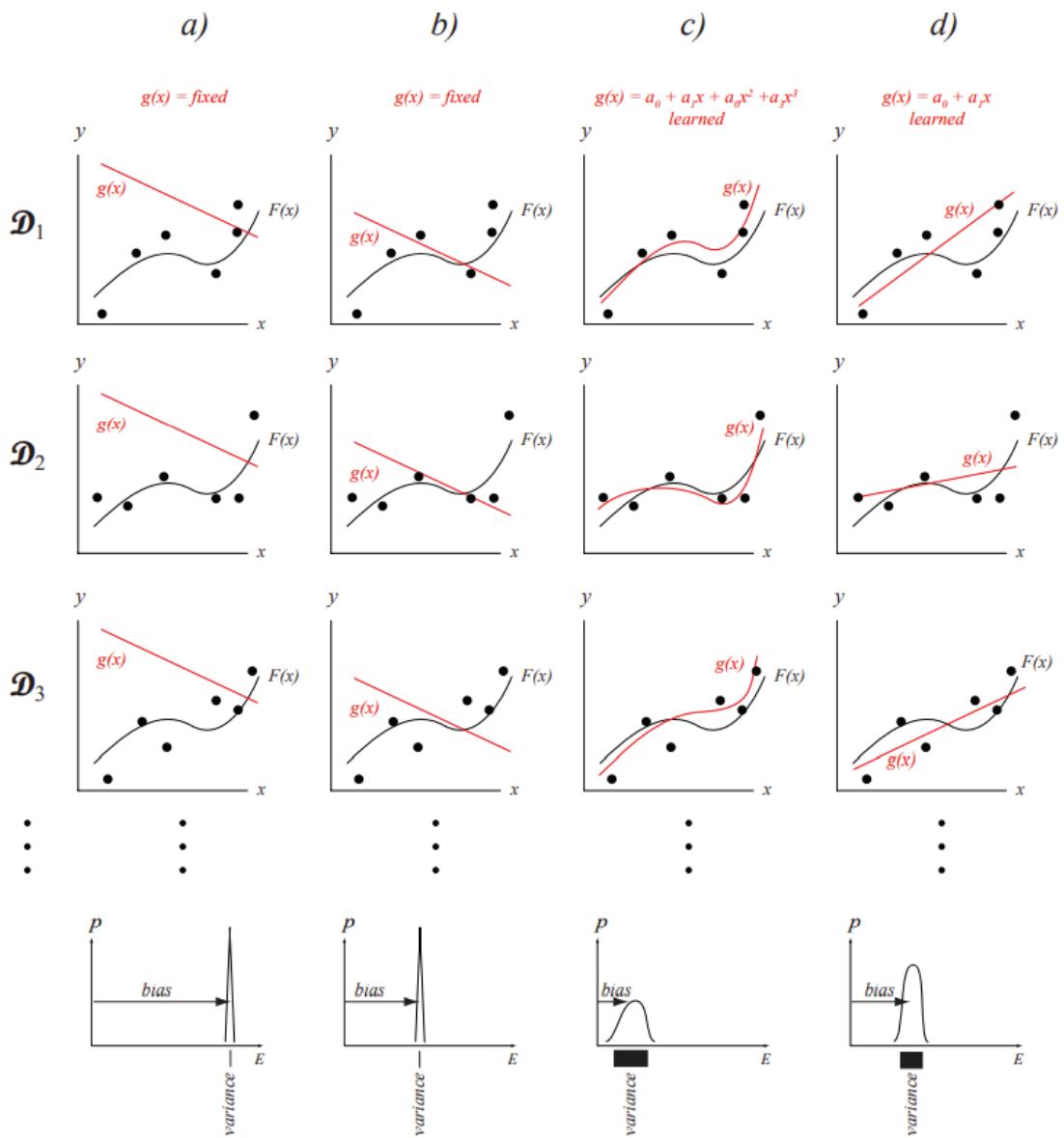


Ilustración 5 Ejemplo sesgo-varianza [7]

La ilustración 5 tiene tres conjuntos de datos distintos del mismo UT (D1, D2 y D3). En la primera columna (a), se puede ver que, para un sesgo muy alto y una varianza casi nula, la función $g(x)$ no se adapta a ninguno de los tres conjuntos de datos. En la segunda columna, con un sesgo más pequeño que el de la comuna a) y una varianza cercana a cero, $g(x)$ sigue sin adaptarse a los datos de entrenamiento, aunque se aproxima más que a). En cambio, para un sesgo muy pequeño y una varianza un poco más grande, la función $g(x)$ se entrena mejor y es capaz de adaptarse a los datos casi a la perfección, sin embargo, si el conjunto de datos fuese más grande, el sesgo disminuiría, mientras que en el ejemplo d), que es un poco más rígido que c), no lo haría y proporcionaría mejores resultados al ser un más genérico y no adaptarse solo a los datos de entrenamiento.

Por lo tanto, el clasificador que mejor resultados va a proporcionar será aquel cuya varianza y sesgo sean lo más reducidas posible.

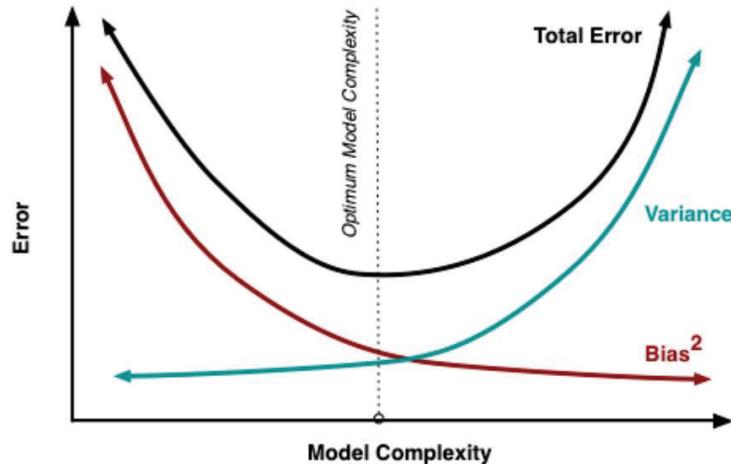


Ilustración 6 Evolución varianza - sesgo - error [12]

Dado que no se conoce el valor de la función $f(x)$, que es la que permitiría pintar la gráfica de la ilustración 6, la mejor forma de encontrar un valor para la varianza y el sesgo adecuado es estudiando el número de datos con los que se entrena el clasificador y el error total de este.

La ilustración 7 muestra la relación entre el error que se produce al clasificar las muestras de entrenamiento y evaluación con respecto a la complejidad del clasificador.

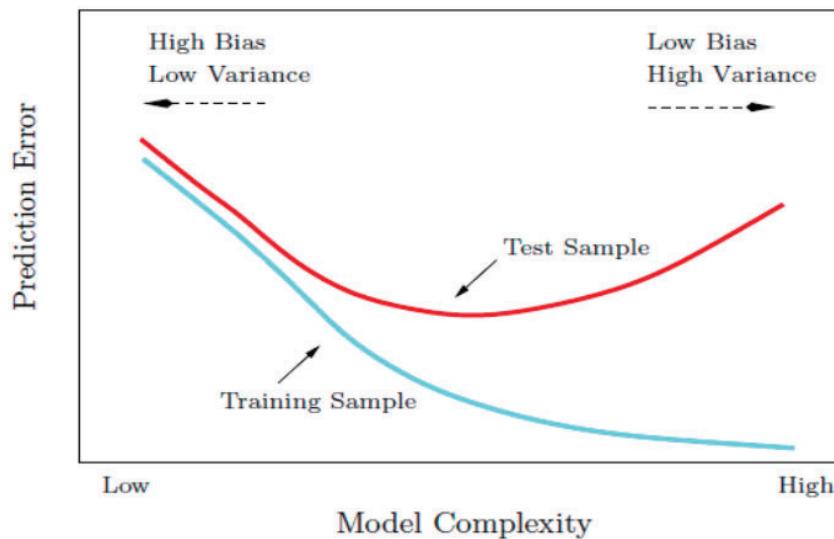


Ilustración 7 Evolución sesgo-varianza en función de los datos de entrenamiento y evaluación [12]

Entrenar y evaluar un clasificador con un número muy bajo de muestras de entrenamiento y evaluación (low model complexity), dará lugar a un rendimiento deficiente dado que el clasificador no podrá asimilar todas las características necesarias para realizar una correcta clasificación. Eso se denomina “underfitting”.

Por otro lado, si se utilizan demasiadas muestras en el entrenamiento (high model complexity), se reducirán el número de datos mal clasificados en este conjunto, pero no el de los datos de evaluación dado que es posible que se haya entrenado el clasificador con ruido. Esto se denomina “overfitting”

Para estimar la tasa de error existen distintos métodos en función de cómo se dividan los datos del UT para el entrenamiento y evaluación del clasificador, dado que no se puede utilizar el mismo conjunto de datos para las dos tareas porque se clasificarían datos ya conocidos.

Los métodos son:

- Método de partición: se selecciona un conjunto de datos aleatorios para el entrenamiento (Tr) y con el resto de los datos se evalúa el clasificador (Ts). Esto puede dar resultados poco precisos dado que puede que no se seleccionen datos de todas las categorías para el entrenamiento [13].

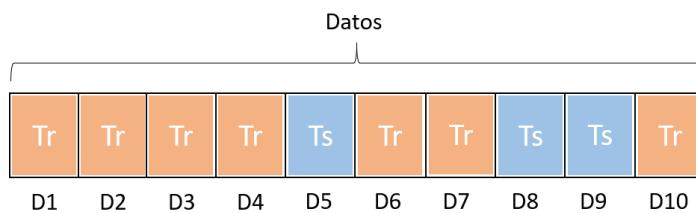


Ilustración 8 Método de partición

- Método de exclusión: se dispone de datos específicos para el entrenamiento y de otros para la evaluación. Ninguno de los datos que componen cada grupo se encuentra en el otro. Además de que en cada conjunto existen datos de todas las clases.

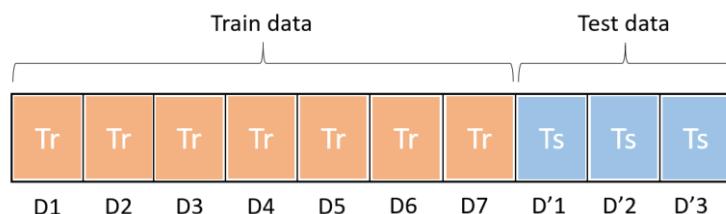


Ilustración 9 Método de exclusión

- Método de validación cruzada: se seleccionan X elementos del conjunto total de datos para el conjunto de entrenamiento y el resto para la evaluación de manera sucesiva hasta que todos los datos han formado parte de ambos grupos. El error total será la media del error obtenido en cada iteración. Tiene un alto coste computacional.



Ilustración 10 Método de validación cruzada

En función de lo comentado en este apartado, el método de validación que va a seguir el clasificador de este trabajo es el de exclusión dado que el UT elegido dispone de dos grupos de datos, una para el entrenamiento y otro para la evaluación.

2.2 Entorno de programación

En este apartado se va a comentar las herramientas que se han utilizado para el desarrollo del sistema.

Como se comentó al final del [apartado 1.2](#), el lenguaje de programación que se va a emplear es Python tanto para la programación del clasificador como para la interfaz, además de para conectar la interfaz con la base de datos y con el clasificador.

El entorno que se ha empleado es Spyder IDE porque proporciona información acerca el código de manera muy clara y su manejo es fácil. Sin embargo, antes de poder instalar el entorno, es necesario instalar el sistema de gestión de paquetes Anaconda. Este sistema proporciona múltiples librerías de Python, por lo que no es necesario instalarlas en el entorno.

Para la creación de las bases de datos es necesario instalar la aplicación MySQL Workbench, y el sistema de gestión de bases de datos denominado XAMPP. Se optó por el uso de estas herramientas debido a que son aplicaciones con las que se habían trabajado en otras ocasiones.

3 Desarrollo del Clasificador

El sistema propuesto se compone de tres partes diferenciables. La primera es el clasificador, que se encargará de dar la etiqueta que consideré que le corresponde a la imagen de la prenda que se le pasé. El desarrollo de este es lo que se va a detallar en los siguientes subapartados.

Para poder manejar sin complicaciones el código de la interfaz, se han creado cuatro ficheros cuyo nombre y descripción se detalla a continuación:

- Clasificador_v3.py: contiene la clase Classifier. En ella están todas las funciones necesarias para el correcto funcionamiento del clasificador. El esqueleto de este fichero fue proporcionado por el profesorado en la asignatura de Reconocimientos de formas de la Universidad Politécnica de Madrid [14] y se ha retocado para adaptarlo al objetivo de este trabajo.
- entrenamientoClas_v1.py: contiene la clase EntrenamientoClasificador. Esta clase está formada por una función para cargar el dataset con los datos que seleccionados y otra que lleva a cabo el entrenamiento del clasificador.
- imageRecognizer.py: contiene la clase ImageRecognizer que se encarga de seleccionar el contorno de una prenda en una imagen. El esqueleto de este fichero fue proporcionado por el profesorado en la asignatura de Reconocimientos de formas de la Universidad Politécnica de Madrid [14] y se ha retocado para adaptarlo al objetivo de este trabajo.
- etiquetarImagen_v2: contiene la clase Etiquetado. Se encarga de llamar a las funciones anteriores en el orden necesario para poder llevar a cabo la clasificación de una prenda.

3.1 Implementación del algoritmo de clasificación

El algoritmo se ha desarrollado en el fichero “Clasificador_v3.py” y en este apartado se va a comentar su implementación.

La implementación se desarrolla en cuatro funciones dentro de la clase ClassifBayesiano:

- fit(self,X, y): función que se encarga de calcular los elementos necesarios para poder hallar las funciones de pertenencia de los objetos.
- decision_function(self, X): función que estima el grado de pertenencia de una serie de datos (los de entrenamiento) calculando por cada objeto el valor de sus funciones de pertenencia.
- predict(self, X): función que, de todas las estimaciones, elige la que proporciona el valor más alto para cada objeto. Se emplea en el entrenamiento.
- predict2(self, X): función que, de todas las estimaciones, devuelve las dos con valor más alto, siempre que la diferencia de ambas no sea superior a un valor establecido. Se emplea para la clasificación de una imagen de cara a la aplicación.

Las librerías empleadas para ellos son:

- Math [15]: para poder realizar diversas funciones matemáticas.

- Numpy [16]: paquete que proporciona funciones para manejar matrices multidimensionales.
- Sklearn [17]: proporciona diferentes funciones para Machine Learning.

Se recuerda que la fórmula escogida para la obtención de las funciones de discriminación sigue la siguiente forma:

$$f_i(x) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x - \mu_i)^T * \Sigma_i^{-1} * (x - \mu_i) + \ln P(C_i)$$

Σ_i : matriz de covarianza de la clase i.

μ_i : media de la clase i

$P(C_i)$: probabilidad a priori de la clase i.

La primera función denominada “fit” admite dos parámetros: “X” que es una matriz formada por los vectores de características de los objetos de nuestro UT (cada fila es un objeto) e “y”, que es un vector que cuyos valores corresponden a la etiqueta de cada objeto. Lo primero que hace la función es comprobar que existen tantas etiquetas en el parámetro “y” como objetos (filas) en “X”. Dado que para calcular la función de discriminación se necesitan ciertos valores como la matriz de covarianza, la media y la probabilidad a priori de cada clase, esta función los calculará.

A continuación, se muestra una parte del código de esta función para ver la forma en la que se calculan esos valores:

```
# Obtenemos nuestra lista de clases y sus cardinalidades
self.clases = np.unique(y)
cardinalidades = np.array([len(X[y == c]) for c in self.clases])

# Calculamos el logaritmo neperiano de la probabilidad apriori para cada clase
self.ln_apriories=np.array([math.log((cardinalidades[c]/cardinalidades.sum())) for c in self.clases])

# Calculamos la media de cada clase
self.means= np.array([np.mean(X[y==c],axis=0) for c in self.clases])

# Calculamos la matriz de covarianzas para cada clase
cov = np.array([ShrunkCovariance(self.shrinkage).fit(X[y==c]) for c in self.clases ])
covarianzas=np.array([cov[c].covariance_ for c in self.clases])
```

```

# Calculamos para cada clase el logaritmo neperiano del determinante de
# su matriz de covarianzas
sign, self.ln_determinants = np.linalg.slogdet(covarianzas)
if np.any(np.isinf(self.ln_determinants)):
    print("Warning: There is a covariance matrix with determinant
= 0")

# Guardamos la inversa de la matriz de covarianzas de cada clase
self.inv_covs = np.array([np.linalg.inv(c) for c in covarianzas])

```

Dado que los valores que se obtengan de esta función se van a emplear en otras funciones es conveniente crear variables globales añadiendo `self.` delante del nombre de la variable.

Como se puede leer en los comentarios del código, se ha calculado el valor para todos los elementos de la función de discriminación dado que así, en la función `decision_function` solo se calculará el valor de la función por cada clase. Esta función recibe como parámetro una matriz `X` formada por los vectores de características de los objetos del UT y devuelve una matriz denominada “`funcion`” con todos los valores de las funciones de discriminación de cada clase por cada objeto.

En la ilustración 11 se adjunta una representación gráfica de esta matriz para su mejor compresión. Las celdas que tienen fondo verde se corresponderían con los valores de la matriz “`funcion`”.

	Clase 1	Clase 2	Clase 3	Clase 4		Clase m
Objeto 1	f1(Objeto 1)	f2(Objeto 1)	f3(Objeto 1)	f4(Objeto 1)	...	<u>fm(Objeto 1)</u>
Objeto 2	f1(Objeto 2)	f2(Objeto 2)	f3(Objeto 2)	f4(Objeto 2)	...	<u>fm(Objeto 2)</u>
Objeto 3	f1(Objeto 3)	f2(Objeto 3)	f3(Objeto 3)	f4(Objeto 3)	...	<u>fm(Objeto 3)</u>
Objeto 4	f1(Objeto 4)	f2(Objeto 4)	f3(Objeto 4)	f4(Objeto 4)	...	<u>fm(Objeto 4)</u>

Objeto n	f1(Objeto n)	f2(Objeto n)	f3(Objeto n)	f4(Objeto n)	...	<u>fm(Objeto n)</u>

Ilustración 11 Representación de la matriz que contiene los valores de la función de discriminación

La función predict(self, X) se encarga de construir un vector denominado “sol”, de tamaño igual al tamaño de X (objetos) en el que almacenar la clase predicha para cada objeto. Para ello, se selecciona por cada fila de la matriz devuelta por la función decision_function(self, X) la columna con el máximo valor. Dado que cada columna se corresponde con una clase, se sabe con certeza cuál es la clase más probable y se añade al vector denominado “sol”. Con esto se obtiene la predicción de clase para un objeto.

La última función que se encuentra en el fichero es predict2(self, X). Su funcionamiento es prácticamente idéntico al anterior, sin embargo, en lugar de devolver un vector, devuelve una matriz de tamaño $2 * \text{número de objetos}$, para así poder devolver por cada objeto las dos etiquetas o clases más probables. Dado que hay ciertos casos en los que no es necesario devolver dos clases porque los valores están demasiado separados se ha establecido, para este trabajo, una diferencia máxima entre los dos valores más altos de 150. Este valor se ha ajustado en función de la observación y estudio de varias clasificaciones.

3.2 Entrenamiento y evaluación del clasificador

Previamente ya se comentó las características del Dataset con el que se iba a entrenar el clasificador, así como el tipo de entrenamiento que se iba a llevar a cabo.

El fichero entrenamientoClas_v1.py contiene las siguientes funciones necesarias para el entrenamiento del clasificador: cargarDataSet y trainClasificador.

La primera función se encarga de cargar el dataset usando la librería tensorflow [18], dado que era una de las formas que se describía para la obtención del dataset y separa los datos en imágenes y etiquetas de entrenamiento (train_images y train_labels) e imágenes y etiquetas para su evaluación (test_images y test_labels).

Dado que, como se comentó previamente, no todas las clases se van a usar en la aplicación, se tienen que eliminar todos aquellos valores cuyas clases sean camisa o sandalias. Tras esto, las matrices train_images y test_images y los vectores train_labels y test_labels están listos para ser usados por la función trainClasificador(self). Dado que se usan fuera de la función, estos valores se han almacenado en variables globales.

A continuación, se muestra el código de la función trainClasificador(self), en que se puede ver como se realiza el entrenamiento y la evaluación:

```
print("-> Clasificador Bayesiano")
print("--> Training classifier")

clf = ClassifBayesiano()
print("X_train: " + str(self.X.shape))
```

```

clf.fit(self.X, self.Y)

print("--> Evaluating classifier")
y_pred = clf.predict(self.X)
print("----> Training accuracy: {:.2f}%".format(100 * np.mean(self.Y == y_pred)))

y_pred = clf.predict(self.Xtest)
print("----> Test Accuracy: {:.2f}%".format(100 * np.mean(self.Ytest == y_pred)))

```

Está compuesto por varios “print” dado que así es más fácil seguir la ejecución.

En primer lugar se crea un objeto de la clase ClassifBayesiano() llamado “clf” y se procede al entrenamiento llamando a la función fit() a la que se le pasa como parámetros los datos de entrenamiento (self.X) y sus etiquetas (self.Y). Con ello se obtienen los valores necesarios para crear las funciones de discriminación.

Posteriormente se ejecuta la función predict() con el parámetro self.X para así poder obtener el porcentaje de aciertos que se producen en un conjunto conocido.

Y para finalizar, se vuelve a ejecutar la función predict(), solo que esta vez pasando como parámetro la matriz self.Xtest, es decir, los datos de evaluación. Comprobando el vector resultante de la función con el vector que contenía las etiquetas de cada objeto, se obtiene el porcentaje de aciertos en un conjunto desconocido. Este dato es muy importante porque en caso de ser muy bajo, revelaría que los parámetros del clasificador no están bien ajustados.

En la siguiente tabla se muestra el resultado de estas evaluaciones:

Conjunto de datos	Porcentaje de aciertos
Entrenamiento	84,16%
Evaluación	81,80%

Tabla 1 Clasificador - Porcentaje de aciertos

Como se puede observar el porcentaje no llega a ser excelente, sin embargo, para este trabajo es aceptable. Además, el conjunto se ha entrenado los suficientemente bien como para que no se haya solo adaptado a los datos de entrenamiento porque la diferencia de porcentajes es pequeña.

En vista del resultado se decidió crear la función predict2() para así poder proporcionar al usuario la etiqueta correcta.

3.3 Comprobación del funcionamiento del clasificador

En este apartado se va a describir paso por paso el recorrido de la imagen de una prenda desde su selección hasta su clasificación.

Estos pasos son:

1. Obtención de la imagen de una ruta proporcionada.
2. Binarización de la imagen.
3. Detección de la silueta de la prenda en la imagen.
4. Envío de la sección de la imagen que contiene la prenda al clasificador previamente entrenado.
5. El clasificador devuelve la etiqueta o etiquetas de la prenda.

La llamada a las funciones de todos estos pasos se realiza desde el fichero etiquetarImagen_v2, así como el entrenamiento del clasificador que se usará. Este fichero contiene las siguientes funciones para llevar a cabo el etiquetado:

- `entrenarClasificador(self)`: crea una variable global de la clase EntrenamientoClasificador y ejecuta la función `cargarDataSeet()` y `trainClasificador`. De esta forma se obtiene en la variable “`self.clf`” un clasificador entrenado y listo para usar.
- `etiquetarPrenda(self,path_img)`: recibe como parámetro la ubicación de la imagen. Crear una variable de la clase `ImagineRecognizer(clasificador, etiquetas)` y le pasa como parámetro clasificador la variable global `self.clf` y las posibles etiquetas que puede detectar, que en nuestro caso es un array con la forma `[0,1,2,3,4,5,6,7]`.
Posteriormente llama a la función `detect_imagen_clothes(imagen_path)` que se encuentra dentro del fichero `imageRecognizer.py` y pertenece a la clase `ImagineRecognizer` para proceder con la clasificación. Esta función empleará otras funciones de la misma clase para conseguir el objetivo.

Los siguientes subapartados describirán los pasos que la función `detect_imagen_clothes` lleva a cabo.

Hay que destacar que la clase `ImagineRecognizer` emplea la librería cv2 [19] dado que es una librería de software de visión artificial y aprendizaje automático y contiene funciones que facilitan el desarrollo del proyecto, como por ejemplo para la lectura de la imagen y detección de contornos.

3.3.1 Obtención de la imagen

En este y en los siguientes apartados se van a mostrar unas series de imágenes de una prenda con el fin de representar de manera gráfica el tratamiento de la imagen.

La ilustración 12 muestra la imagen original cuya ubicación se le ha pasado a la función `detect_imagen_clothes(imagen_path)`.



Ilustración 12 Prenda de original

A partir de la ubicación de la imagen, se carga la imagen original en una escala de blanco y negro, debido a que todos los elementos con los que se entrenó el clasificador estaban en una escala de grises y es más fácil realizar la detección de siluetas. Posteriormente se dimensiona la imagen en un tamaño de 1600 píxeles de altura para visualizar lo mejor posible la imagen y se procura eliminar el posible ruido de ella.

Estas acciones dan lugar a la imagen que se muestra a continuación:



Ilustración 13 Prenda en escala de grises

3.3.2 Binarización

Para reducir la escala de grises y facilitar la detección de contornos en una imagen con fondo blanco o similar, se emplea la función threshold de la librería cv2.

```
bin_img = cv2.threshold(blur, 0, 255, cv2.THRESH_TOZERO +  
cv2.THRESH_OTSU)[1]
```

Esta función recibe como parámetros la imagen sin ruido (blur) el valor máximo (255) y mínimo que tendrá la imagen (0) para que este en escala de blanco y negro y el tipo de umbral, que, en este caso, se ha empleado THRESH_TOZERO

para acentuar las zonas y que no hubiese tanta variedad de colores, pero sin perder la profundidad, y THRESH_OTSU para no perder el fondo.

En la ilustración 14 se muestra un ejemplo de lo que realiza cada posible umbral de la función threshold.

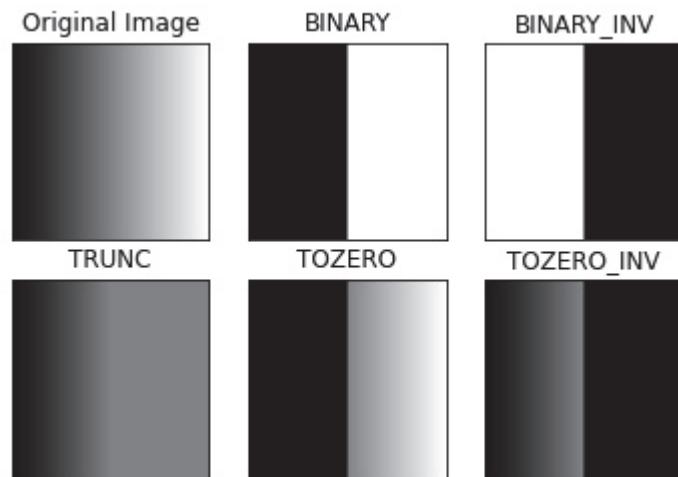


Ilustración 14 Resultado de aplicar distintos umbrales [20]

A continuación, se muestra el resultado de la ejecución de la función threshold:



Ilustración 15 Prenda binarizada

3.3.3 Detección de la silueta de la prenda en la imagen.

Una vez se obtiene la imagen binarizada, se llama a la función detect_clothes(bin_img, self.predict_category) pasándole como parámetros la imagen y la función predict_category de la clase ImageRecognizer.

La función predict_category es la que función que se encarga de llamar a la función predict2 del clasificador pasándole el vector de características del objeto a clasificar.

Pero antes de ello, es necesario establecer en qué parte de la imagen está la prenda, para así tener un vector de características lo más ajustado posible. Para

hallar los posibles contornos se emplea la función `findContours` de `cv2`, que dibuja los contornos que encuentre en la imagen que reciba como parámetro.



Ilustración 16 Contornos detectados en la prenda

En la ilustración 16 se han marcado en rojo todos los contornos que la función ha detectado.

Como se puede apreciar, hay contornos que no son válidos, por lo que para filtrarlos y obtener solo el contorno de la prenda, se realiza una restricción de que todos aquellos contornos cuya superficie sea menor a 10000 pixeles se eliminan. Este valor se ha establecido en función de la observación y estudio de varias clasificaciones.



Ilustración 17 Contornos seleccionado de la prenda

En la ilustración 17 se puede ver en verde el contorno que tras la restricción se ha quedado como válido y en rojo los que no.

Una vez se tiene la prenda ubicada, se recorta la imagen y se le pasa la sección con la prenda a la función predict_category(cropped_image).

3.3.4 Llamada al clasificador

La función predict_category() se encarga de llamar a la función predict2() del clasificador, previamente entrenado, pasándole la imagen recortada como parámetro. Con esta acción se obtiene en un array las posibles etiquetas de la prenda.

Como se puede ver, en este ejemplo, la etiqueta que el clasificador ha decidido asignar a esta prenda es “Zapatillas”, la cual se corresponde con la prenda.



Ilustración 18 Prenda con etiqueta

Hay que resaltar que todas estas imágenes el usuario no las llega a ver en ningún momento, dado que solo se pintaron para facilitar la comprensión de la clase ImageRecognizer.

4 Desarrollo de la Base de datos

En los siguientes subapartados se va a comentar como se ha llevado a cabo la implementación de la base de datos que va a usar el sistema. La base de datos es la segunda parte de la que se compone el sistema.

4.1 Desarrollo del modelo

En base de la cantidad de información que el sistema va a requerir de la base de datos, se optó por la creación de tres tablas, cuyo nombre e información se va a detallar a continuación.

- **Usuarios:** dado que al sistema podrán acceder distintos usuarios es necesario una tabla que almacene ciertos datos de estos. Se compone de tres columnas:
 - id INT (11): será la clave primaria de esta tabla, de tipo numérico y no superará los 11 dígitos. El sistema será el encargado de asignar este valor en función del último número de usuarios registrados, por lo que será único.
 - nameUser VARCHAR (45): en esta columna se almacena el nombre que el usuario ingrese al momento de darse de alta. Tiene que estar compuesto por caracteres y como mínimo, estará formado por 1 carácter y, como máximo, por 45.
 - password INT (11): almacena el valor numérico que el usuario ingreso para darse de alta como contraseña y tendrá una longitud máxima de 11 dígitos. Esta columna no puede estar vacía.
- **Prendas:** en esta tabla se almacena información acerca de las prendas que el usuario ha clasificado.
 - idPrenda INT (11): será la clave primaria de esta tabla, y tiene las mismas características que la clave primaria de la tabla “Usuarios”, es decir, es de tipo numérico y no superará los 11 dígitos y el sistema se encargará de asignar su valor.
 - nombrePrenda VARCHAR (45): almacena el nombre de la imagen que se le pasa al clasificador. Esta columna no puede no tener ningún valor y como máximo, se guardarán 45 caracteres.
 - pathImagen VARCHAR (45): almacena la ubicación de la imagen que se le pasa al clasificador. Dado que toda foto tiene una ruta, no puede estar vacía.
 - etiquetaPrenda VARCHAR (45): guarda la etiqueta que el clasificador ha asociado a la imagen de la prenda. Siempre que se guarde una imagen, se guardará su etiqueta, por lo que esta columna no puede estar vacía.
 - idUsuario INT (11): almacena el identificador del usuario que clasificó la prenda, por lo que se trata de una foreign key y enlaza las tablas usuarios y prendas. Tiene las mismas características que la columna id en la tabla Usuarios.
- **Conjuntos:** esta tabla contiene los conjuntos que un usuario crea y las prendas por las que este compuesto.
 - idConjuntos INT (11): será la clave primaria de esta tabla. Con esto se podrá identificar el conjunto que es dado que su valor es único. Su valor se asigna automáticamente.

- nombreConjunto VARCHAR (45): en esta columna se colocará el nombre del conjunto. Este valor se asigna automáticamente, y dado que el sistema solo puede almacenar seis conjuntos por usuario, este puede ser Conjunto1, Conjunto2, Conjunto3, Conjunto4, Conjunto5 y Conjunto6.
- camiseta INT (11): si el conjunto se compone de una camiseta, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- pantalon INT (11): si el conjunto se compone de un pantalón, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- sudadera INT (11): si el conjunto se compone de una sudadera, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- vestido INT (11): si el conjunto se compone de un vestido, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- abrigo INT (11): si el conjunto se compone de un abrigo, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- zapatillas INT (11): si el conjunto se compone de unas zapatillas, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- bolso INT (11): si el conjunto se compone de un bolso, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- botin INT (11): si el conjunto se compone de unas botas, se almacenará su identificador, por lo que esta columna contiene una foreign key.
- idusuario INT (11): almacena el identificador del usuario que creo el conjunto.

A continuación, se muestra el diagrama entidad-relación de las tablas que se han comentado:

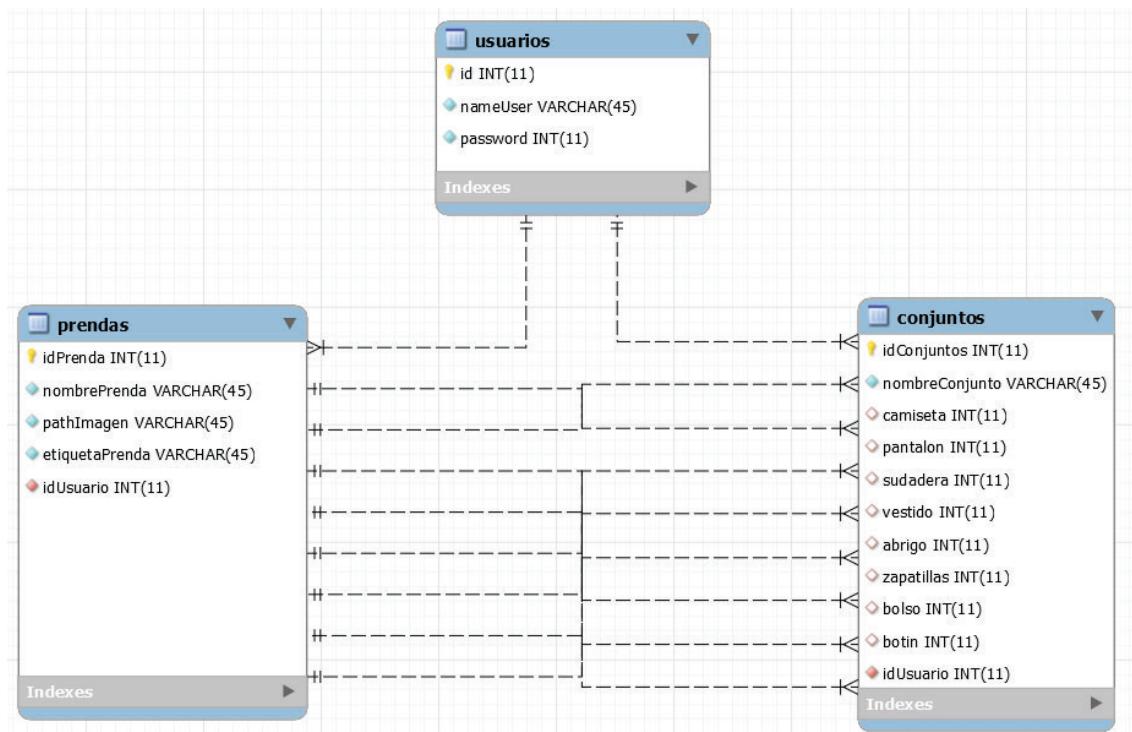


Ilustración 19 Diagrama entidad-relación de la base de datos

En la ilustración 19 se puede apreciar que las tres tablas se encuentran enlazadas unas con otras dado que tanto en la tabla “prendas” como “conjuntos” hay una columna que hace referencia a la clave primaria de la tabla usuarios, y en “conjuntos” se hace referencia a la clave primaria de la tabla prendas.

4.2 Implementación de la base de datos

Con el modelo relacional se puede obtener las sentencias SQL para la creación de la base de datos. En este apartado se comentarán estas sentencias.

Con la ejecución de este fragmento se crea el esquema general al que se añadirán las tablas. El esquema se llamará “armario_virtual”.

```
CREATE SCHEMA IF NOT EXISTS `armario_virtual` DEFAULT CHARACTER SET utf8 ;
USE `armario_virtual` ;
```

El siguiente trozo de código corresponde a la creación de la tabla “usuarios”. Como se ha comentado en el apartado anterior, la primary key es el atributo “id”.

```
CREATE TABLE IF NOT EXISTS `armario_virtual`.`usuarios` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nameUser` VARCHAR(45) NOT NULL,
```

```

`password` INT NOT NULL,
PRIMARY KEY (`id`)
ENGINE = InnoDB;

```

El siguiente fragmento se corresponde a la creación de la tabla “prendas”. Aquí se declara que el atributo “idPrenda” es la primary key de esta tabla y que “idUsuario” hace referencia a “id” en la tabla “usuarios”. Además, se ha configurado la table para que, en caso de que se borre una entrada en la tabla “usuarios”, las entradas que referenciaran a ese id en esta tabla, también se eliminen.

```

CREATE TABLE IF NOT EXISTS `armario_virtual`.`prendas` (
  `idPrenda` INT NOT NULL AUTO_INCREMENT,
  `nombrePrenda` VARCHAR(45) NOT NULL,
  `pathImagen` VARCHAR(45) NOT NULL,
  `etiquetaPrenda` VARCHAR(45) NOT NULL,
  `idUsuario` INT NOT NULL,
  PRIMARY KEY (`idPrenda`),
  INDEX `idPrenda_idx` (`idUsuario` ASC) ,
  CONSTRAINT `idPrenda`
    FOREIGN KEY (`idUsuario`)
    REFERENCES `armario_virtual`.`usuarios` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

La tabla “conjuntos” se crea con las siguientes sentencias SQL. De igual manera que en la tabla “prendas”, si una entrada de las tablas “prendas” o “usuarios” se elimina, se eliminaran las entradas que referenciaran a esa entrada.

```

CREATE TABLE IF NOT EXISTS `armario_virtual`.`conjuntos` (
  `idConjuntos` INT NOT NULL AUTO_INCREMENT,
  `nombreConjunto` VARCHAR(45) NOT NULL,
  `camiseta` INT NULL,
  `pantalon` INT NULL,
  `sudadera` INT NULL,
  `vestido` INT NULL,
  `abrigo` INT NULL,

```

```

`zapatillas` INT NULL,
`bolso` INT NULL,
`botin` INT NULL,
`idUsuario` INT NOT NULL,
PRIMARY KEY (`idConjuntos`),
INDEX `idCamiseta_idx` (`camiseta` ASC) ,
INDEX `idPantalon_idx` (`pantalon` ASC) ,
INDEX `idSudadera_idx` (`sudadera` ASC) ,
INDEX `idVestido_idx` (`vestido` ASC) ,
INDEX `idAbrigo_idx` (`abrigo` ASC) ,
INDEX `idZapatillas_idx` (`zapatillas` ASC) ,
INDEX `idBolso_idx` (`bolso` ASC) ,
INDEX `idBotin_idx` (`botin` ASC) ,
INDEX `idUser_idx` (`idUsuario` ASC) ,
CONSTRAINT `idCamiseta`
    FOREIGN KEY (`camiseta`)
    REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT `idPantalon`
    FOREIGN KEY (`pantalon`)
    REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT `idSudadera`
    FOREIGN KEY (`sudadera`)
    REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT `idVestido`
    FOREIGN KEY (`vestido`)
    REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,

```

```

CONSTRAINT `idAbrigo`
    FOREIGN KEY (`abrigos`)
        REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
CONSTRAINT `idZapatillas`
    FOREIGN KEY (`zapatillas`)
        REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
CONSTRAINT `idBolso`
    FOREIGN KEY (`bolso`)
        REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
CONSTRAINT `idBotin`
    FOREIGN KEY (`botin`)
        REFERENCES `armario_virtual`.`prendas` (`idPrenda`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
CONSTRAINT `idUser`
    FOREIGN KEY (`idUsuario`)
        REFERENCES `armario_virtual`.`usuarios` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE)
ENGINE = InnoDB;

```

4.3 Comunicación entre la base de datos y la interfaz

En este subapartado se van a detallar las funciones en lenguaje Python que la interfaz usará para realizar consultas a la base de datos.

Para ello se emplea la librería mysql.connector [21] para la comunicación con el servidor MySQL.

Para realizar las consultas, se ha creado la función denominada “run”, que se encarga de crear la conexión y ejecutar la consulta. Parte de su código se detalla a continuación:

```

def run(self, query, val):
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="armario_virtual"
    )
    cursor = mydb.cursor()
    ...

```

El parámetro “query” contendrá la sentencia SQL a ejecutar y “val” los valores en caso de que la sentencia SQL añada una nueva fila en alguna de las tablas.

Para crear la conexión es necesario especificar el nombre del usuario y la contraseña (password). En nuestro caso el nombre es “root” y no hay contraseña. También hay que poner el nombre de la base de datos, “armario_virtual”.

Una vez se tiene esto, se crea una variable llamada “cursor” gracias a la función cursor() de la variable “mydb”. Con esta variable se puede ejecutar la sentencia SQL que contiene la variable “query” dado que dispone de una función llamada execute(query, val) o execute(query).

El resto de las funciones que se han definido para poder acceder a la base de datos son las siguientes:

- **alta_usuario**(self, nombreUsuario, password): con esta función se puede añadir un nuevo usuario a la tabla “usuarios”. Para ello, se tiene que llamar a la función pasándole como parámetros el nombre de usuario como una cadena de caracteres y la contraseña como un valor numérico o como una cadena de caracteres cuyos caracteres solo sean números. Devuelve el número identificativo que se le ha asignado a ese usuario, si el nombre proporcionado no se encuentra en la tabla.
- **verificarUsuario**(self, nombreUsuario, contraUsuario): esta función verifica si el parámetro “nombreUsuario” y “contraUsuario” son un par nombre-contraseña correctos. Devuelve dos parámetros, “recuento” e “identificador”. El primero valdrá 1 si se ha encontrado en la tabla “usuarios” una entrada con ese nombre y contraseña, y el segundo tendrá el identificador de esa entrada. En el caso de que no exista tal par, recuento valdrá 0.
- **prendasCategoria**(self, categoria, idUser): con este método se obtiene el nombre, la ubicación de la imagen y el identificador de todas las prendas cuya categoría sea igual al parámetro “categoria” que haya guardado el usuario cuyo identificador sea igual a “idUser”. Para ello realiza una búsqueda en la tabla “Prendas”.
- **insertarPrenda**(self, nombrePrenda, dirFoto, categoria, idUser): este método permite añadir una nueva fila a la tabla “Prendas”. En el parámetro “nombrePrenda” se especificará el nombre de la imagen de la prenda, en “dirFoto” la ubicación de la imagen, en “categoria” la

etiqueta que el clasificador ha seleccionado para esa imagen y en “idUser” el identificador del usuario que ha clasificado la prenda.

- **listarConjuntos(self, idUser):** se encarga de listar el nombre de todos los conjuntos de usuario cuyo identificador sea igual a “idUser”. Para ello realiza una búsqueda en la tabla “Conjuntos”.
- **elementosConjunto(self, idConjunto, idUser):** devuelve la ubicación, el nombre de la imagen y la etiqueta de las prendas que conforman un conjunto. Para ello se realiza una búsqueda en la tabla “Conjuntos” y se retorna la fila en la que el valor de las columnas “idConjuntos” e “idUser” sea igual a los parámetros “idConjunto” e “idUser”.
- **crearConjunto(self, idConjunto, nombreConj, cam, pan, sud, vest, abr, zap, bols, bot, idUser):** añade una nueva fila a la tabla “Conjuntos”. Para ello es necesario pasarle como parámetros el identificador del conjunto a crear, el nombre de este y los identificadores de todas las prendas de las que está compuesta en el siguiente orden: camiseta, pantalón, sudadera, vestido, abrigo, zapatillas, bolso y botas. Además, en el parámetro “idUser” se especificará el identificador del usuario que crea el conjunto.
- **key_prenda(self, nombrePrenda, idUser):** obtiene de la tabla “Prendas” el identificador de la prenda del cliente con identificador igual a “idUser” y con el nombre “nombrePrenda”.
- **eliminar_prenda(self, idPrenda):** elimina la fila de la tabla “Prendas” en la que el identificador sea igual a “idPrenda”.

5 Desarrollo de la Interfaz

Dado que el sistema se ha desarrollado en tres partes, la interfaz será la que se comunique con las otras dos partes para proporcionar al cliente los servicios disponibles.

La comunicación con el clasificador y con la base de datos se hará a través de las funciones previamente comentadas.

En la siguiente ilustración se muestra una representación esquemática de estas comunicaciones:

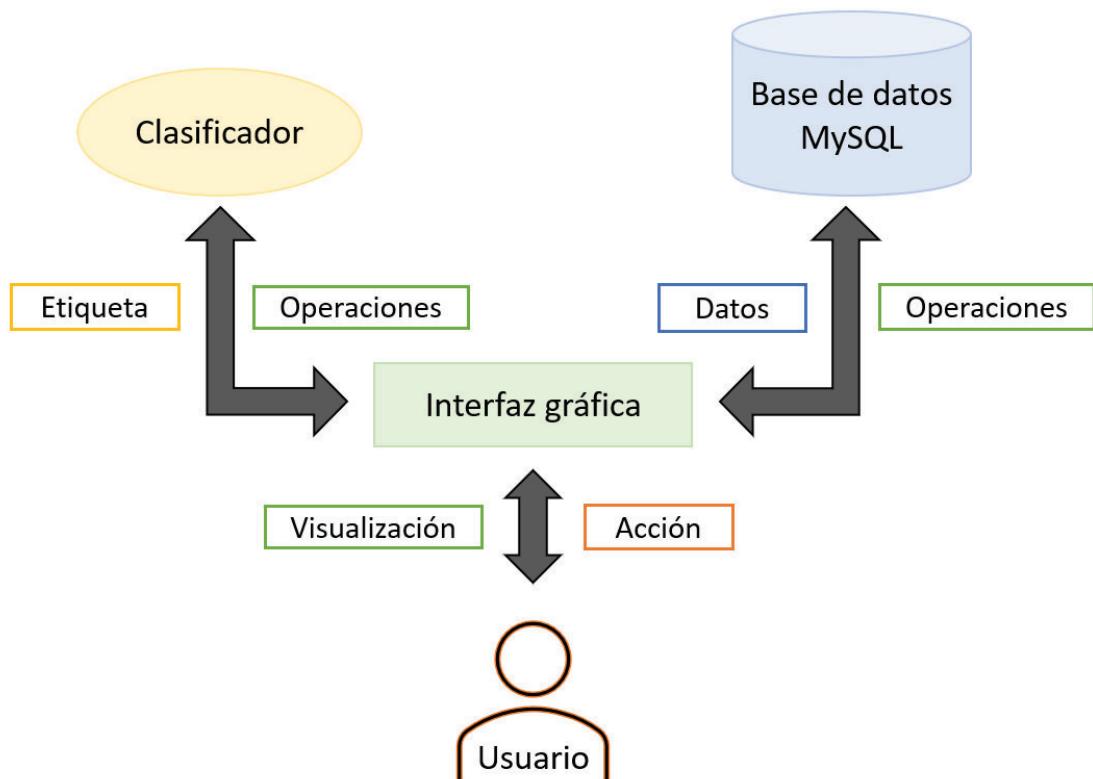


Ilustración 20 Esquema comunicación Usuario-Interfaz-BBDD-Clasificador

En naranja se representa el cliente y como interactúa con la interfaz gráfica. A su vez, la interfaz gráfica, en verde, en base de la acción realizada por el usuario, realiza operaciones en la base de datos o en el clasificador. El clasificador, en amarillo, el único dato que le envía a la interfaz gráfica es la etiqueta de una prenda; mientras que la base de datos le da los datos en función de la operación enviada por la interfaz. Para finalizar, la interfaz gráfica le mostrará al usuario los datos consultados.

Hay que destacar que en la interfaz se realiza el entrenamiento del clasificador con la función `entrenarClasificador()` nada más ejecutarse dado que, aunque cabe la posibilidad de que el usuario no quiera añadir ninguna prenda, el coste

en tiempo es más alto si se entrena cada vez que la interfaz contacta con él que si se entrena una vez al inicio.

Para el desarrollo de la interfaz se ha empleado la librería Tkinter dado que contiene múltiples funciones para realizar interfaces gráficas.

En los siguientes apartados se detallan las acciones que la interfaz le permite realizar a un usuario y lo que este visualiza.

5.1 Ventana de inicio de sesión

Es la primera ventana que se le muestra al usuario cuando se ejecuta el sistema. En ella se pueden realizar dos acciones: iniciar sesión con un usuario y contraseñas registradas o crear un nuevo perfil. En caso de querer crear un nuevo perfil se le redireccionará a la ventana descrita en el apartado 5.2.

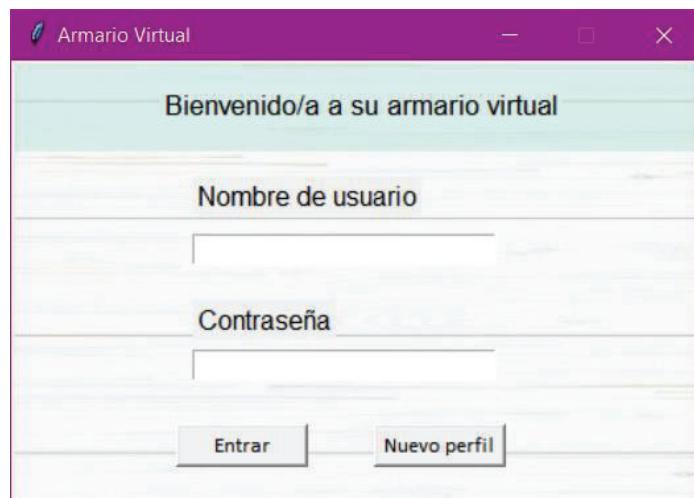


Ilustración 21 Ventana de inicio de sesión

Esta ventana, de tamaño 420x270, contiene:

- Tres etiquetas (“label” en el código): una con un mensaje de bienvenida para el usuario, otra con el texto “Nombre de usuario” y la tercera con “Contraseña”.
- Dos campos para llenar (“entry” en el código): en ellos el cliente podrá introducir un nombre de usuario y una contraseña.
- Dos botones (“button” en el código): uno de ellos muestra el texto “Entrar” y el otro “Nuevo perfil”.
- Una imagen como fondo.

Si el usuario pulsa el botón “Entrar”, la interfaz se encarga de comprobar si el texto introducido en los campos a llenar se encuentra como par usuario-contraseña en la base de datos. Para ello usará la función `verificarUsuario()`, descrita en el [apartado 4.3](#). En el caso de que el usuario no esté registrado, se mostrará como mensaje de error “El usuario o la contraseña no son correctas”, mientras que, si no se introduce algún valor en el campo del nombre o la contraseña, se mostrará “Introduzca un nombre de usuario y una contraseña”. Si el usuario ha introducido los datos correctamente, se destruirá esta ventana

y se abrirá la ventana principal, descrita en el apartado 5.3, y se guardará en una variable global el identificador del usuario para posteriores operaciones.

En caso de que el usuario pulse el botón “nuevo perfil”, la ventana se destruirá y se creará la ventana descrita en el apartado 5.2.

5.2 Ventana para crear un nuevo usuario

Esta ventana se crea si el usuario ha pulsado el botón “Nuevo perfil” en la ventana de inicio de sesión.

Tiene una dimensión de 420x270 y como título “Nuevo perfil”.

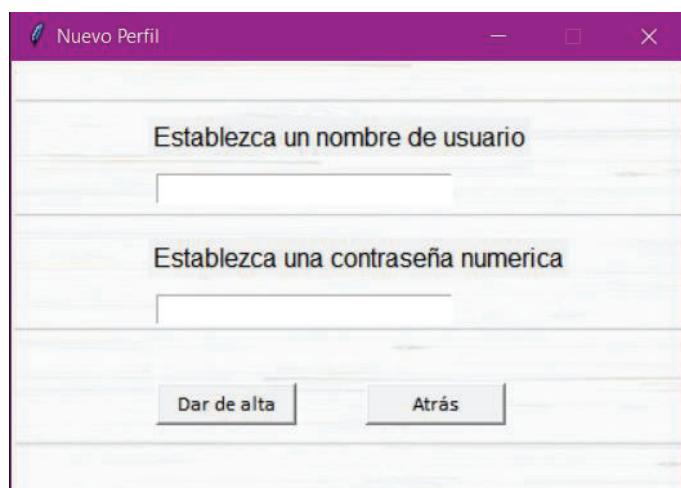


Ilustración 22 Ventana para crear un nuevo usuario

Está compuesta por:

- Dos etiquetas (“label” en el código): cada una especifica la información que el usuario tiene que escribir en los cuadros de debajo para darse de alta.
- Dos campos para llenar (“entry” en el código): en ellos el cliente podrá introducir un nombre de usuario y una contraseña numérica.
- Dos botones (“button” en el código): uno de ellos muestra el texto “Darse de alta” y el otro “Atrás”.
- Una imagen como fondo.

Si el usuario pulsa el botón “Darse de alta”, la interfaz ejecutará la función `alta_usuario ()` para añadir una nueva fila en la tabla “Usuarios”, siempre y cuando se introduzca algún nombre no existente y una contraseña. En caso de que uno de estos campos este vacío, mostrará como mensaje de advertencia “Introduzca un nombre de usuario y una contraseña”, mientras que si se ha elegido un nombre de usuario existente en la base de datos se le mostrará el mensaje “El nombre de usuario elegido ya existe”.

En caso de éxito mostrará “El perfil se ha creado correctamente”, guardará en una variable global el identificador del usuario para posibles operaciones futuras, destruirá la ventana actual y se creará la ventana principal, descrita

en el siguiente apartado, para que el usuario pueda empezar a crear su armario virtual.

Si el usuario no desea crear un nuevo perfil, puede pulsar el botón “Atrás”, que destruirá la ventana actual y creará la ventana de inicio de sesión descrita en el apartado 5.1.

5.3 Ventana principal

Esta es la ventana principal del sistema. Desde aquí se pueden realizar todas las acciones que el sistema proporciona, es decir, visualizar prendas clasificadas en función de su categoría, visualizar y crear conjuntos, añadir una nueva prenda, eliminar una prenda existente y cerrar sesión.

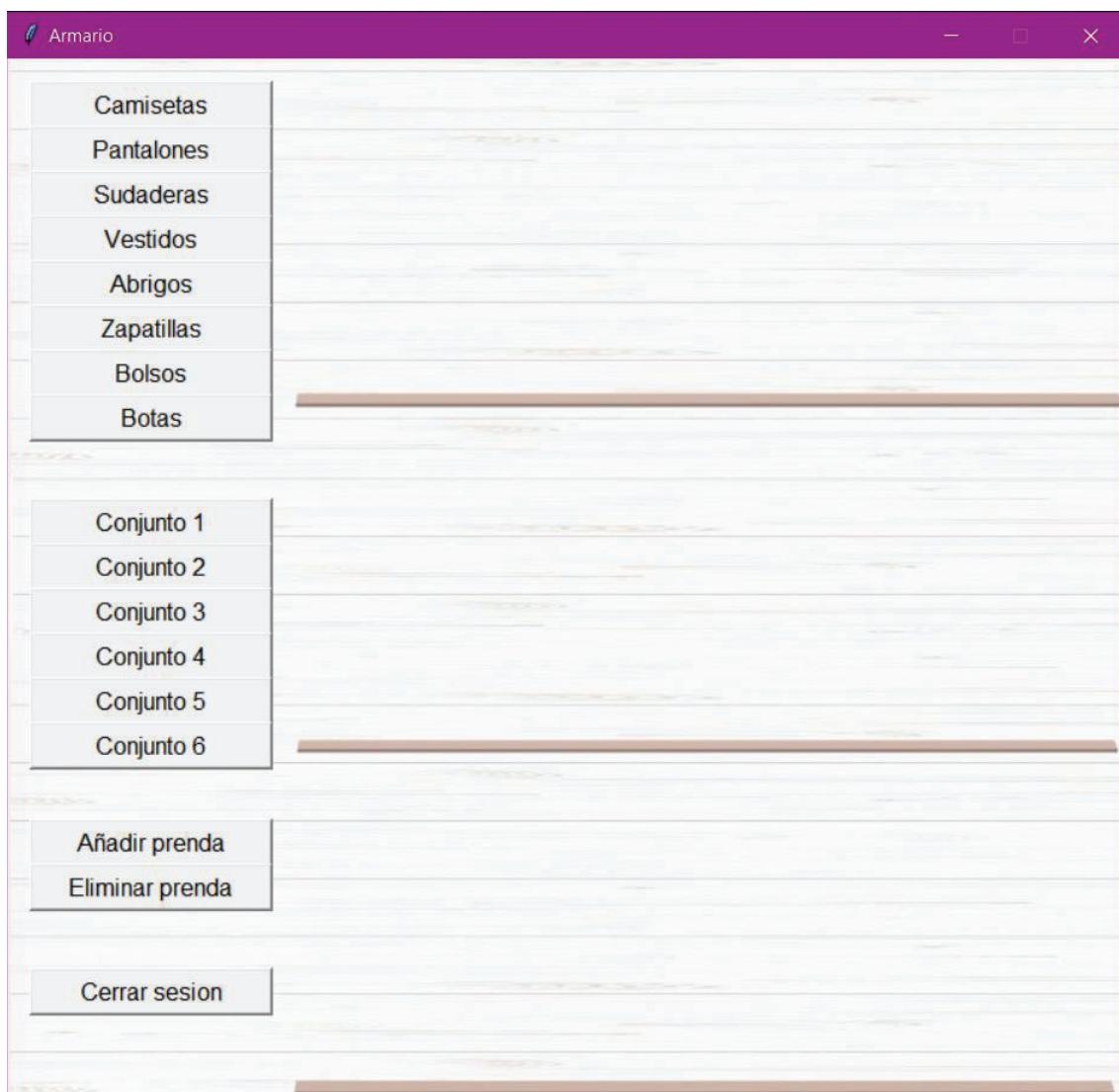


Ilustración 23 Ventana principal

Tiene una dimensión de 750x700 y está compuesta por:

- Diecisiete botones (“button” en el código): Ocho botones con el nombre de cada tipo de prenda que se puede clasificar, otros seis para identificar cada conjunto, uno para añadir una prenda al armario, otro para eliminar una prenda existente y por último, otro para cerrar sesión.
- Una imagen como fondo.

Si el usuario pulsa cualquiera de los primeros ocho botones, la interfaz ejecutará la función `prendasCategoria(categoría, idUser)`, pasándole como parámetros el nombre del botón pulsado y el identificador del usuario. De esta forma se podrán obtener de la base de datos todas las prendas de una categoría almacenadas y se mostrarán junto a su nombre. En el caso de que no se encuentre ninguna prenda, la ventana no cambiará. En el apartado 5.3.1 se muestra con más detalle.

Si el usuario tiene algún conjunto guardado, cuando pulse en el botón del conjunto en el que se almacenó, se mostrarán las imágenes de las prendas que lo conforman, mientras que, si no guardó nada, no se mostrará ninguna imagen. También se le mostrarán tantos desplegables como prendas pueden formar un conjunto. En el apartado 5.3.2 se detalla más acerca del funcionamiento de esta ventana.

Cuando se pulse el botón “Añadir prenda” el sistema permitirá que el usuario seleccione la imagen de una prenda y con la función `etiquetarPrenda(path_img)`, obtendrá una o dos posibles etiquetas para esa prenda y podrá almacenarla. En el apartado 5.3.3 se detalla más acerca de las características de esta ventana.

Por otro lado, si el usuario pulsa el botón “Eliminar prenda”, se le mostrará una serie de desplegable con todas prendas almacenadas donde podrá seleccionar la prenda a eliminar. En el apartado 5.3.4 se detalla más acerca de esta ventana.

Para finalizar, el último posible botón que el usuario puede pulsar es el que poner “Cerrar sesión”. Este botón desencadena la destrucción de la ventana principal y que se creé la ventana “Inicio de sesión” descrita en el apartado 5.1.

5.3.1 Ventana de las prendas

Esta ventana se visualiza cuando el usuario pulsa el nombre de alguna categoría. En la ilustración se muestra el resultado de pulsar “Bolsos”.

Esta ventana estará compuesta, como máximo, de:

- Nueve imágenes: dado que como máximo solo se admiten nueve prendas por categoría.
- Nueve etiquetas (“label” en el código): cada una especifica el nombre que tenía la imagen en el momento en el que se agregó.

En el caso de la ilustración, el usuario que inicio sesión solo había añadido dos prendas con la categoría “bolso” a su armario y son las dos únicas imágenes que se pueden visualizar.

Esta ventana es una ventana informativa, por lo que no se puede realizar ningún otro tipo de acción que no sea visualizar las prendas.

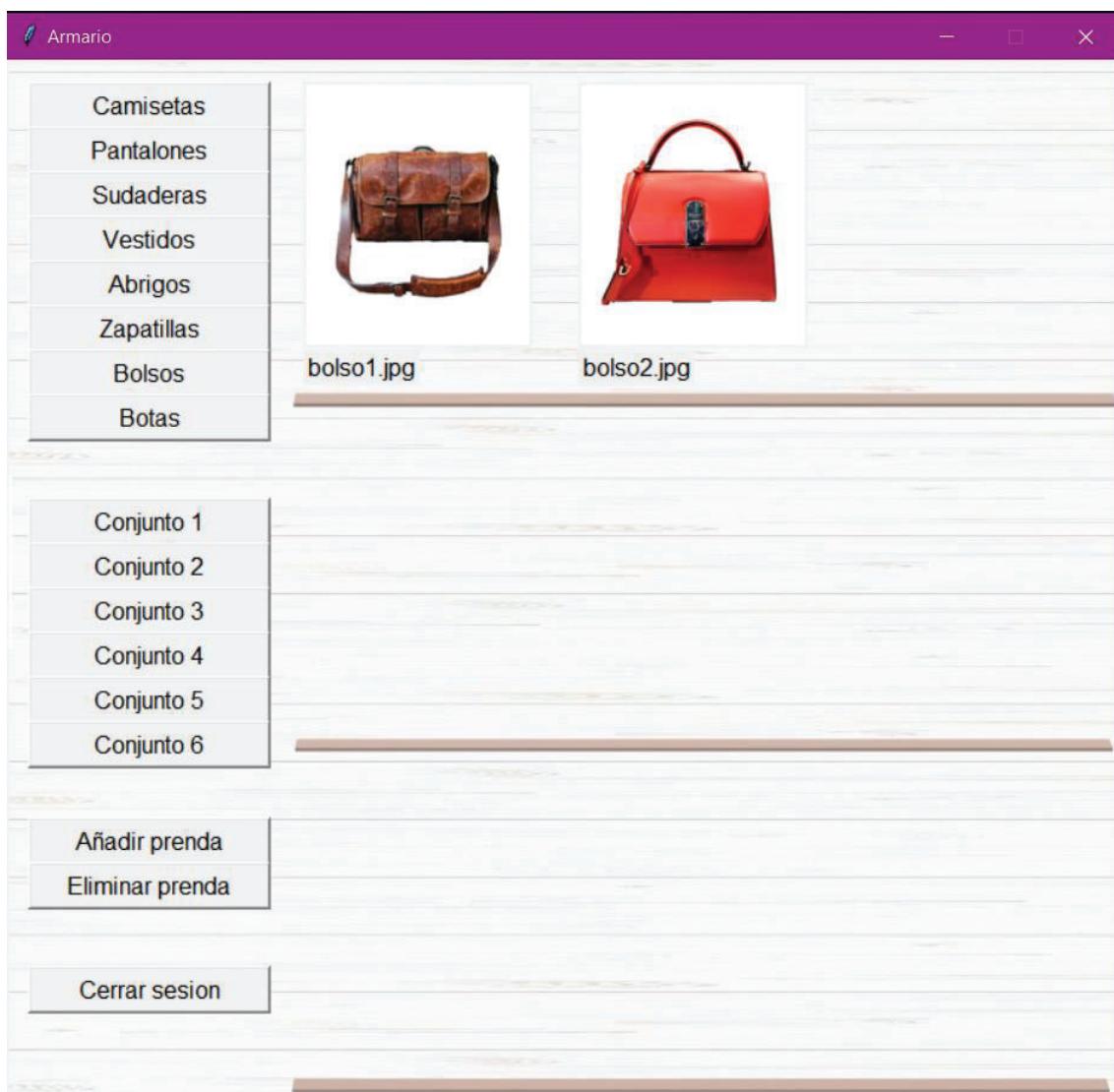


Ilustración 24 Ventana de las prendas

5.3.2 Ventana de los conjuntos

Esta ventana se visualiza si el usuario pulsa cualquier botón que empiece por “Conjuntos”.

La ventana contendrá los mismos botones que ventana “Principal”, a los que se le añadirán:

- Ocho desplegables (“OptionMenu” en el código): uno por cada tipo de prenda que se puede almacenar.
- Un botón (“button” en el código): con la etiqueta “Guardar conjunto”.

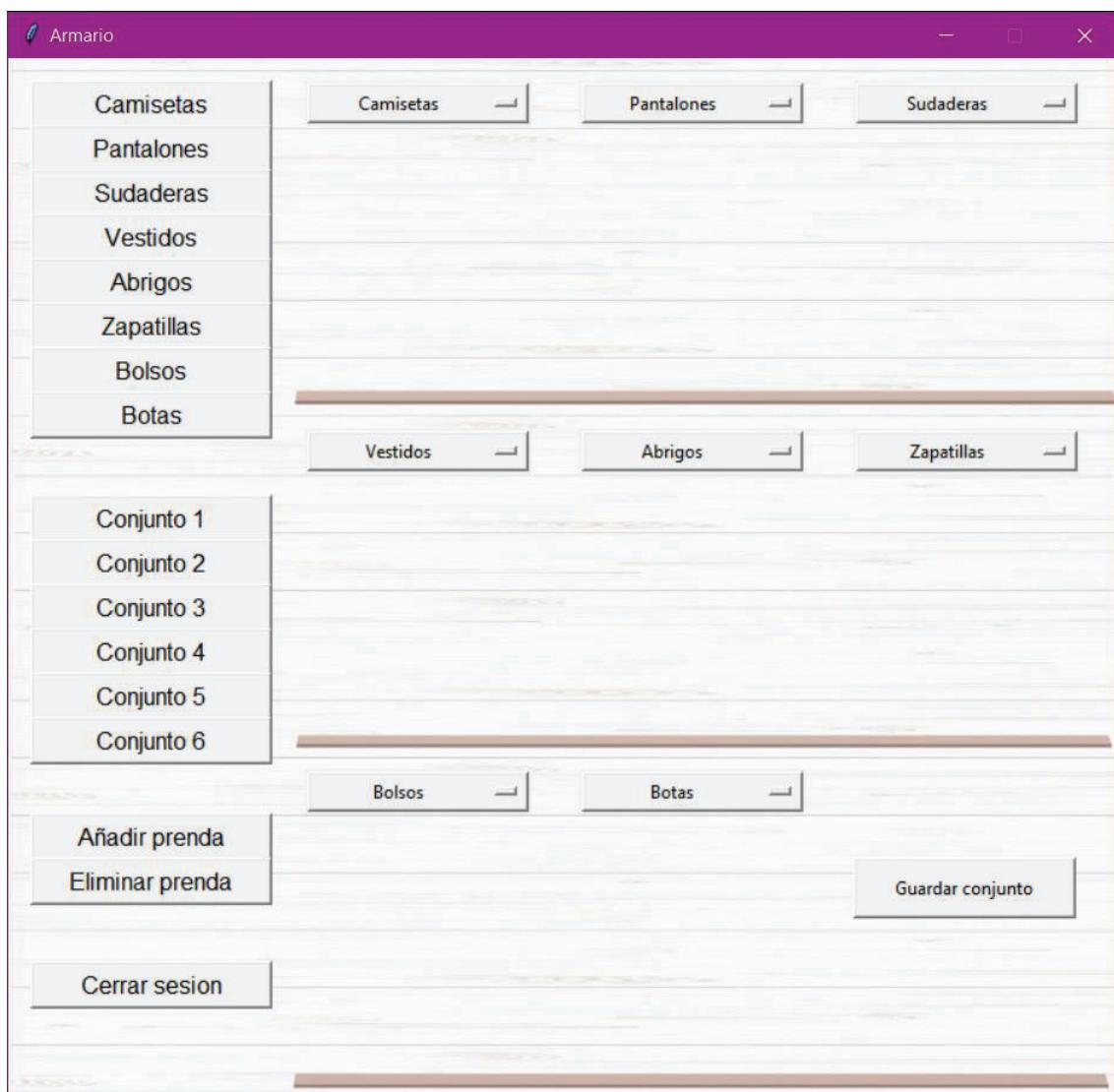


Ilustración 25 Ventana de los conjuntos

El usuario puede pinchar encima de cada desplegable y se le mostrará una lista con todas las prendas que tiene almacenadas de esa categoría, de la que podrá seleccionar una prenda si lo desea. En caso de que no tenga ninguna prenda almacenada, en la lista aparecerá el texto “No disponible”. En caso de que, aun existiendo prendas, el usuario no quiera ninguna, puede bien, no tocar ningún valor del desplegable o seleccionar el valor “Ninguno”.

Una vez que el usuario tiene todas las prendas que desea para un conjunto y pulsa el botón “Guardar conjunto”, la interfaz ejecuta la función `crearConjunto(self, idConjunto, nombreConj, cam, pan, sud, vest, abr, zap, bols, bot, idUser)`, cuyos parámetros serán: un identificador para el conjunto; un nombre, que podrá ser “Conjunto 1”, “Conjunto 2”, “Conjunto 3”, “Conjunto 4”, “Conjunto 5” o “Conjunto 6”, dependiendo del botón que se pulsase; el identificador de cada prenda seleccionada; y el identificador del usuario que lo ha creado.

En el caso de que un usuario tuviese un conjunto guardado en, por ejemplo, “Conjunto 1”, cuando pulse ese botón, se mostrarán las imágenes de las prendas que conforman el conjunto, debajo de su correspondiente desplegable.

En la siguiente ilustración se muestra un ejemplo de un conjunto creado en “Conjunto 1”:

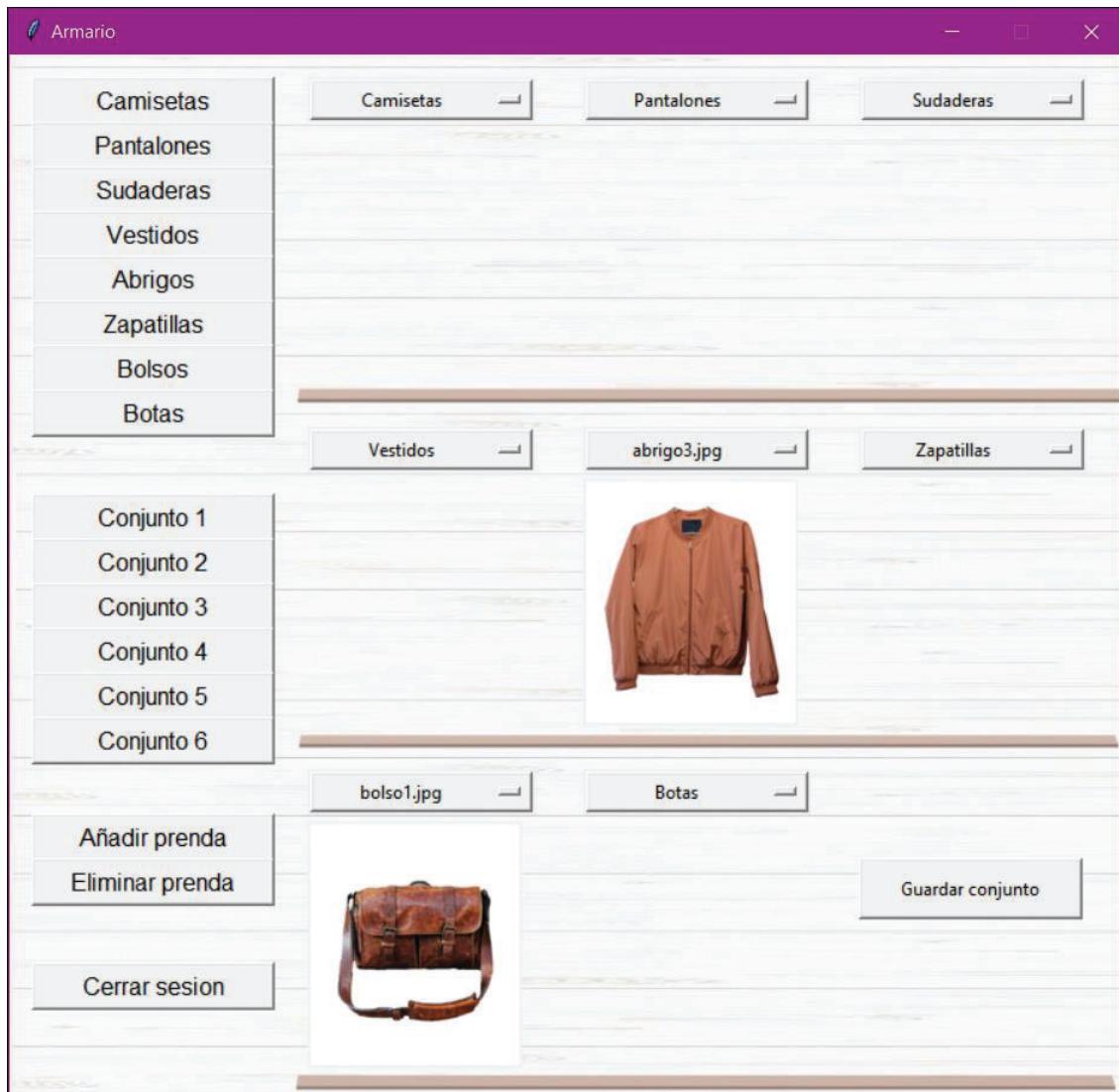


Ilustración 26 Ventana de los conjuntos

Si el usuario desea eliminar las prendas de un conjunto, solamente tendrá que abrir el desplegable de la prenda a eliminar, seleccionar el valor “Ninguno” y pulsar el botón “Guardar conjunto”.

En la ilustración 27 se muestra se muestra un ejemplo de lo que se visualizaría cuando se quiere eliminar del “Conjunto 1” el bolso:

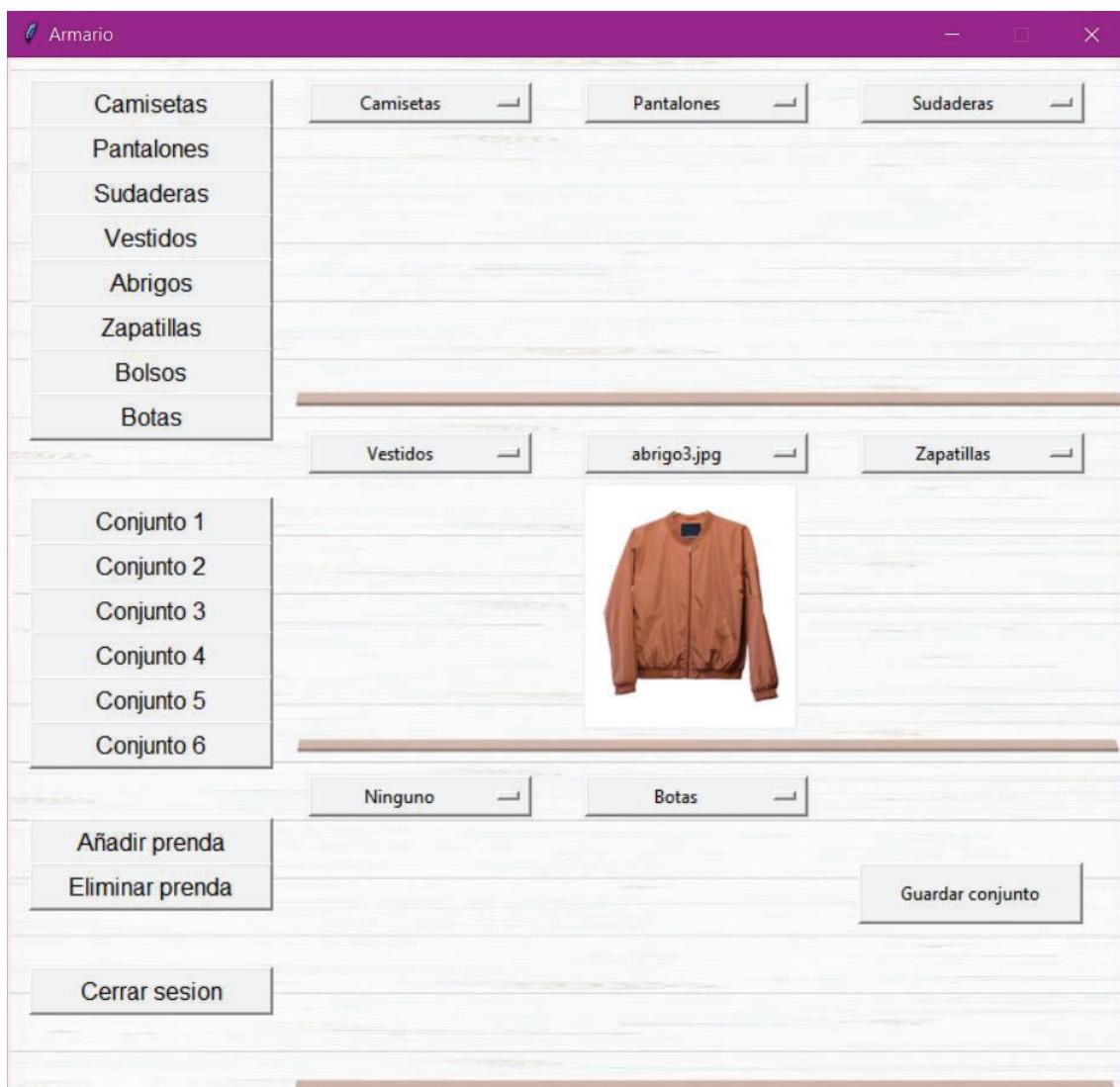


Ilustración 27 Ventana de los conjuntos

5.3.3 Ventana añadir una nueva prenda

Esta ventana se visualiza si el usuario pulsa el botón “Añadir prenda”.

La ventana contendrá los mismos botones que ventana “Principal”, con unas pequeñas adiciones, que variarán en función de las acciones que realice el usuario.

Inicialmente tendrá:

- Una etiqueta (“label” en el código): con un texto que pedirá al usuario que seleccione una imagen.
- Un botón (“button” en el código): con la etiqueta “Buscar imagen”.

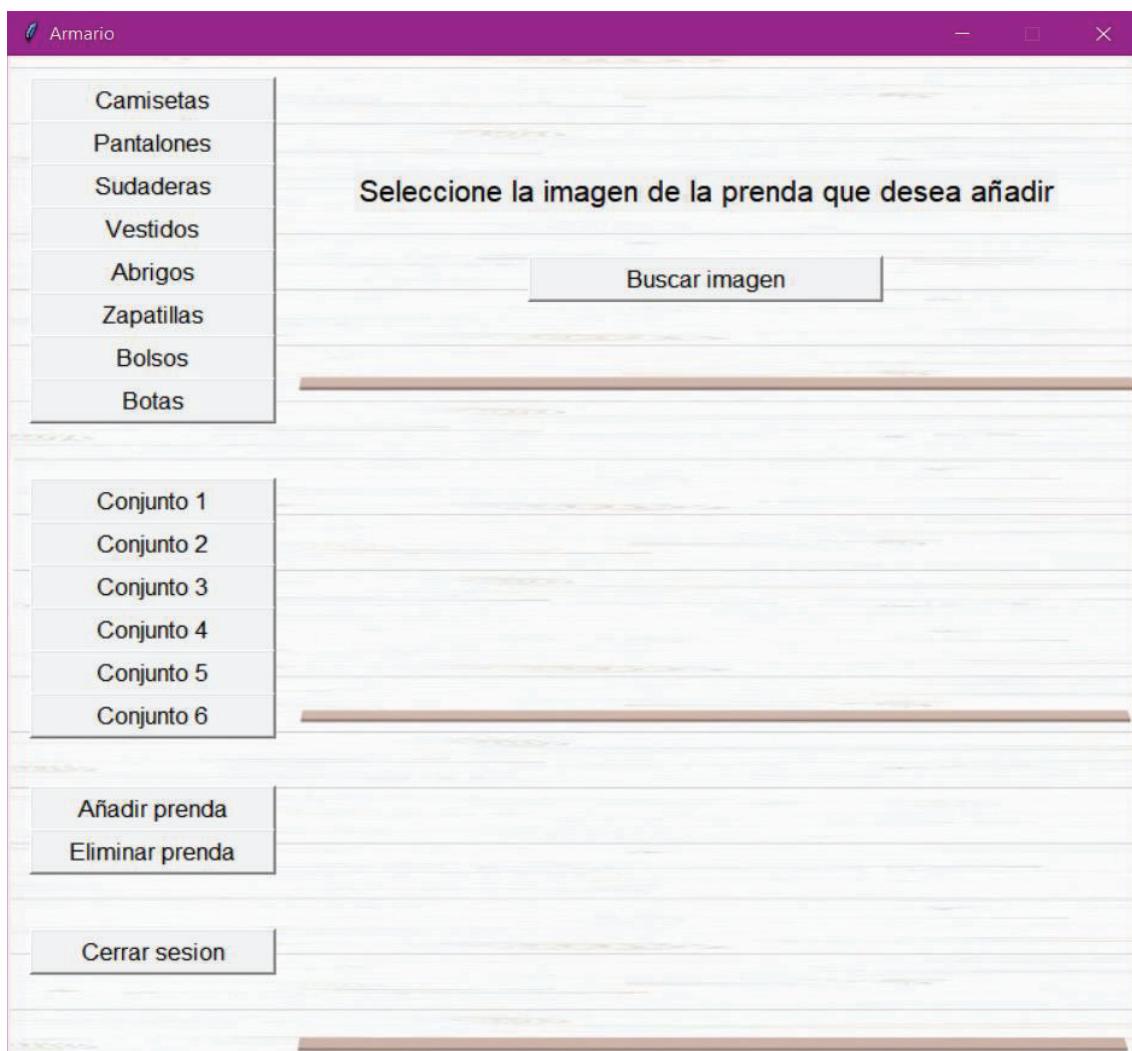


Ilustración 28 Ventana añadir nueva prenda

Cuando el usuario pulsa en “Buscar imagen”, se ejecuta una función de la interfaz denominada `selecImgPrenda`. Esta función muestra la carpeta `resources`, permitiendo así que el usuario visualice todas las imágenes de esa carpeta y seleccione la prenda que quiere añadir. Hay que destacar aquí que para que una imagen pueda añadirse, tiene que ubicarse en esta carpeta.

En el caso de que el usuario opte por no seleccionar ninguna imagen, se le mostrará un mensaje de advertencia diciendo “Debe seleccionar una imagen para poder añadirla al armario”.

En cuanto se selecciona una prenda, se ejecuta la función `etiquetarPrenda(path_img)`, propia del clasificador, para así obtener las posibles etiquetas de la prenda. Dado el rendimiento del clasificador y para proporcionar una experiencia óptima al usuario, se ha decidido mostrar las dos etiquetas más probables, en ciertos casos.

Con estos datos, la interfaz mostrará unos nuevos valores:

- Una etiqueta (“label” en el código): con un texto que pedirá al usuario que seleccione la etiqueta o categoría con la que desea guardar la prenda.
- Uno o dos botones (“button” en el código): dependerán de las etiquetas disponibles para la prenda. Cada botón mostrará en texto una categoría.
- Una imagen con la prenda seleccionada.

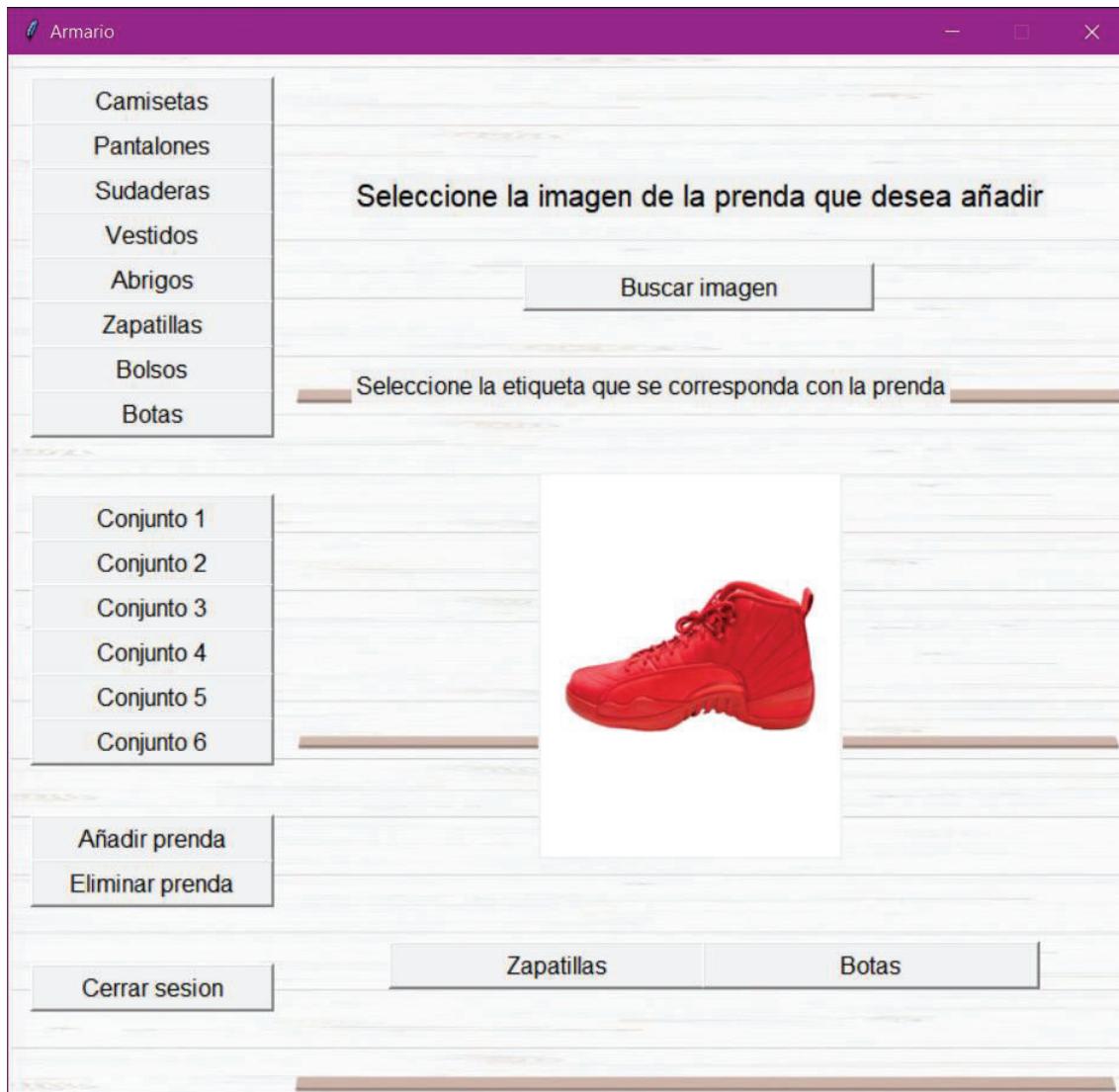


Ilustración 29 Ventana añadir nueva prenda

En este momento el usuario podrá guardar la prenda que se muestra en la imagen pulsando la categoría que considere que más se ajusta a su prenda. Una vez seleccionada, la interfaz ejecutará la función `insertarPrenda(self, nombrePrenda, dirFoto, categoria, idUser)` con los siguientes parámetros: el nombre de la imagen, la ubicación de esta, la etiqueta seleccionada por el usuario y el identificador del usuario. Tras la ejecución de esta función aparecerá un mensaje indicando que se ha añadido, y se volverá a visualizar la ventana como la ilustración anterior.

En caso de que el usuario no hubiese seleccionado la prenda que deseaba, puede volver a pulsar el botón “Buscar imagen”.

5.3.4 Ventana eliminar una prenda

Esta ventana se visualiza si el usuario pulsa el botón “Eliminar prenda”.

Tiene los mismos objetos que la ventana de los conjuntos, es decir:

- Ocho desplegables (“OptionMenu” en el código): uno por cada tipo de prenda que se puede eliminar.
- Un botón (“button” en el código): con la etiqueta “Eliminar prendas”.

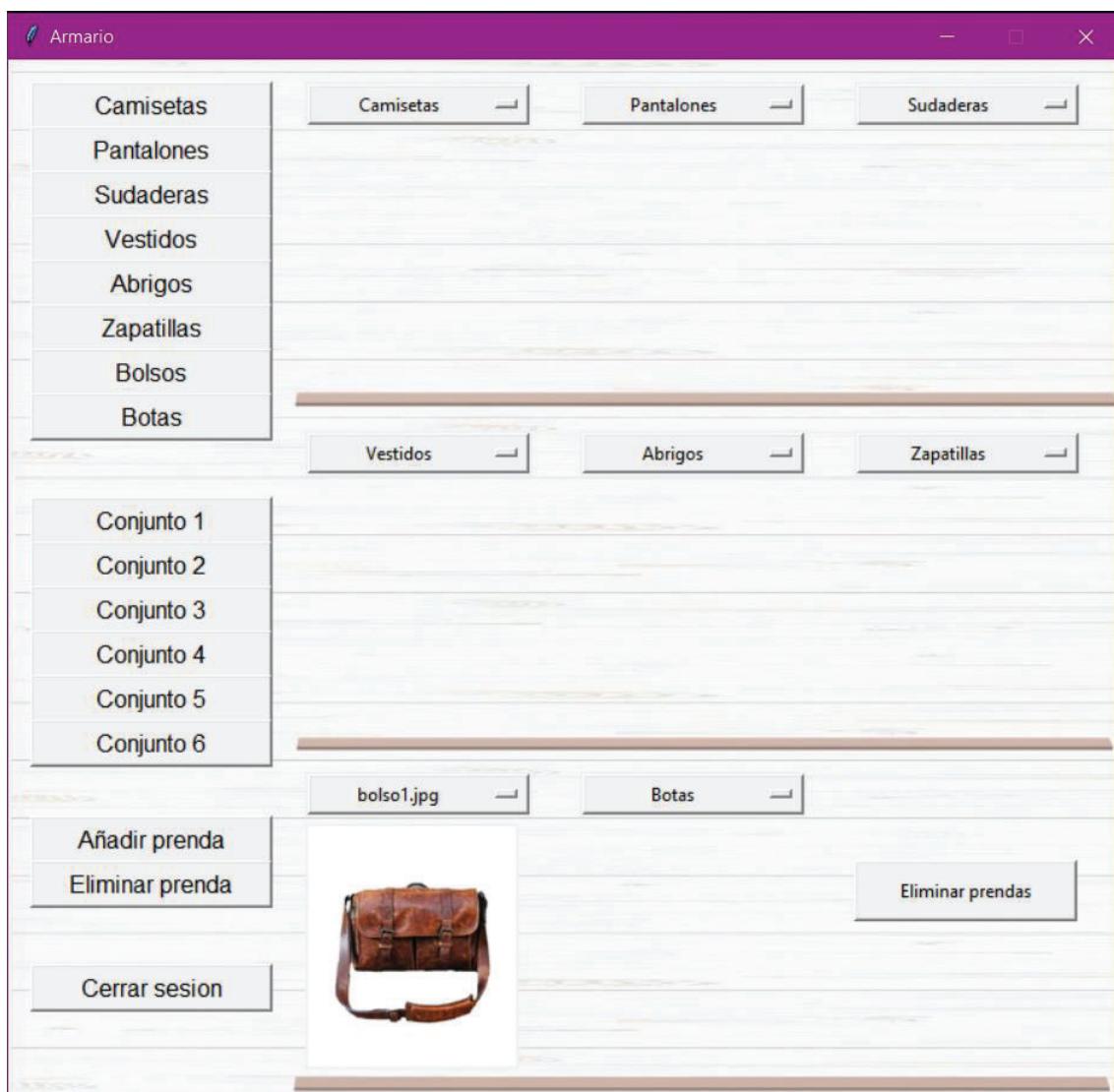


Ilustración 30 Ventana eliminar prenda

El usuario puede eliminar una o más prendas de distintas categorías. Para ello lo que tiene que hacer elegir el nombre de la prenda en el desplegable de cada

categoría de la prenda a eliminar. Se eliminarán todas las prendas seleccionadas cuando el usuario pulsé el botón “Eliminar prendas”.

La interfaz ejecuta la función `eliminar_prenda(self,idPrenda)` en un bucle que se repetirá hasta que se le hayan pasado como parámetro `idPrenda` el identificador de todas las prendas que se quiera eliminar.

Hay que destacar que si alguna de las prendas que se elimina forma parte de un conjunto, el conjunto entero desaparecerá dado que, sin una prenda, el conjunto puede no tener sentido.

6 Desarrollo de las pruebas y ejecuciones

En este apartado se va a comprobar el correcto funcionamiento del sistema analizando el comportamiento de este cuando el usuario realiza ciertas acciones.

Para la realización de las pruebas se parte de un escenario en el que todas las tablas que componen la base de datos están vacías, es decir, no existe ningún usuario registrado y, por ende, ningún tipo de prenda o conjunto.

6.1 Crear un nuevo perfil

El sistema debe permitir al usuario crear un nuevo perfil. En los siguientes subapartados se contemplan distintas posibles acciones por parte del usuario.

6.1.1 Nombre y contraseñas correctas

El usuario introduce un nombre y una contraseña numérica válidas y el sistema comunica que el usuario se ha creado correctamente.

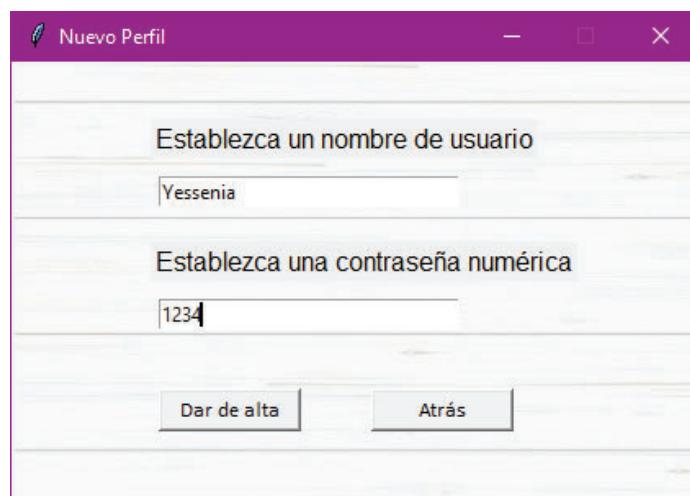


Ilustración 31 Nuevo perfil - Nombre y contraseña correctas

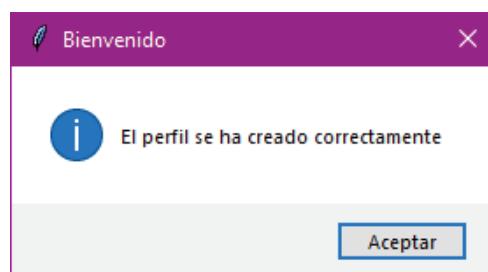


Ilustración 32 Nuevo perfil – Mensaje del sistema

En la ilustración 33 se muestra una captura de la tabla “usuarios” en la que se puede ver la creación de una nueva fila y como se le ha asociado el identificador 1.

	id	nameUser	password
▶	1	Yessenia	1234
*	HULL	HULL	HULL

Ilustración 33 Nuevo perfil - Tabla usuarios

6.1.2 Se dejan campos vacíos

El usuario puede dejar vacío dos campos distintos a la hora de crear un nuevo perfil:

- No se introduce ningún valor en el apartado de Nombre

The screenshot shows a window titled "Nuevo Perfil". It contains two input fields: one labeled "Establezca un nombre de usuario" which is empty, and another labeled "Establezca una contraseña numérica" which contains the value "5678". At the bottom are two buttons: "Dar de alta" and "Atrás".

Ilustración 34 Nuevo perfil - Nombre vacío

- No se introduce ningún valor en el apartado de Contraseña

The screenshot shows a window titled "Nuevo Perfil". The "Nombre de usuario" field contains the value "Maria123". The "Contraseña numérica" field is empty. At the bottom are two buttons: "Dar de alta" and "Atrás".

Ilustración 35 Nuevo perfil - Contraseña vacía

En ambos casos el sistema devuelve el mismo mensaje de error y no permite la creación del usuario.

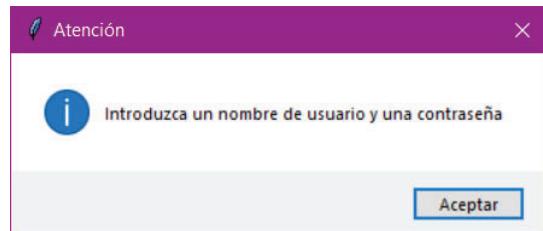


Ilustración 36 Nuevo perfil - Mensaje del sistema

6.1.3 Se introduce un nombre de usuario existente

El usuario introduce en el campo nombre un nombre de usuario ya registrado.

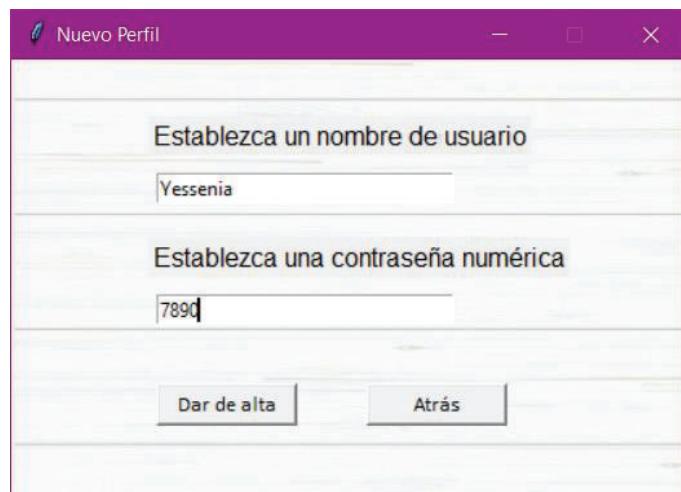


Ilustración 37 Nuevo perfil - Usuario existente

En este caso el sistema no permite que se cree el usuario y muestra un mensaje de error notificando que el nombre de usuario ya existe.

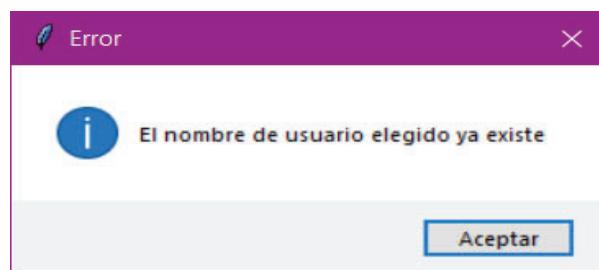


Ilustración 38 Nuevo perfil - Mensaje del sistema

6.2 Acceder con un nombre de usuario y contraseña

El sistema debe permitir al usuario acceder con un nombre de usuario y contraseña, siempre y cuando estas se encuentren en la base de datos.

6.2.1 Se accede correctamente

Si el usuario introduce un nombre de usuario y una contraseña registradas el sistema no muestra ningún tipo de mensaje de error.

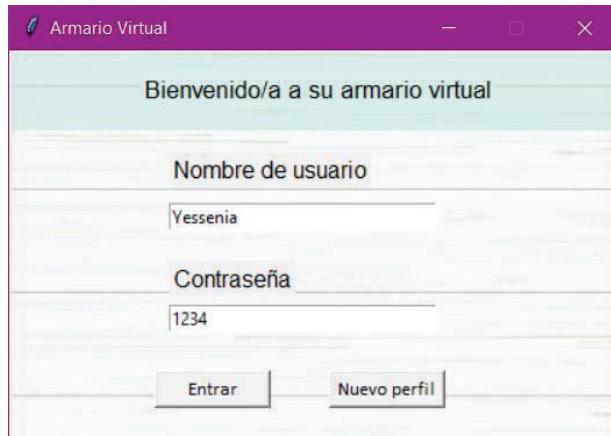


Ilustración 39 Acceder - Acceso correcto

Posteriormente se cambia la ventana de acceso por la pantalla principal del sistema, desde la que podrá realizar diversas acciones

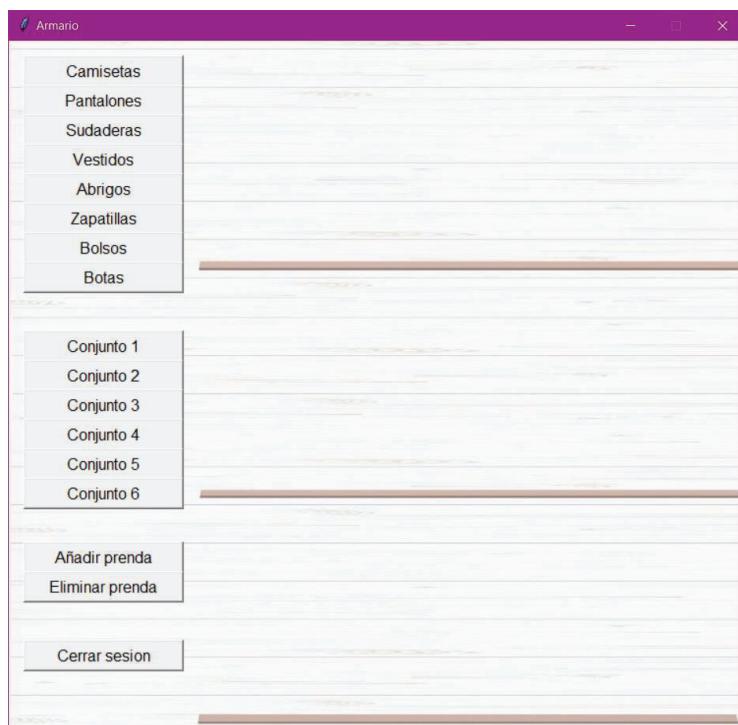


Ilustración 40 Acceder - Ventana principal

6.2.2 No se introducen correctamente los parámetros

El usuario puede no introducir correctamente:

- El nombre de usuario

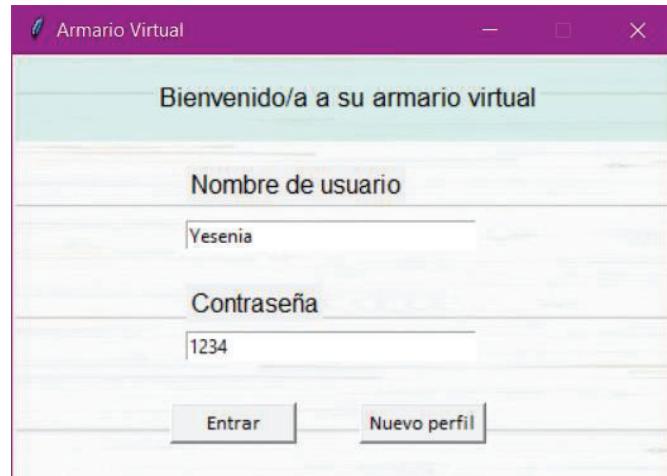


Ilustración 41 Acceso - Nombre erróneo

- La contraseña

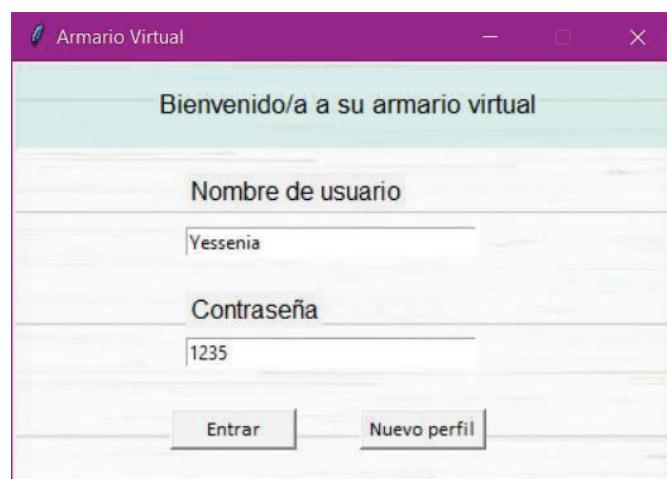


Ilustración 42 Acceso - Contraseña errónea

En ambos casos, tras pulsar el botón entrar, el sistema muestra el siguiente mensaje, no permitiendo el acceso a la pantalla principal.

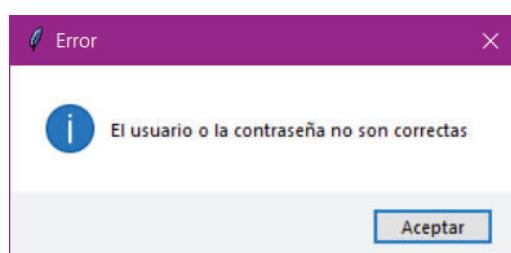


Ilustración 43 Acceder - Mensaje del sistema

6.3 Añadir una nueva prenda al armario

El sistema tiene que permitir al usuario añadir la imagen de una prenda al armario.

6.3.1 Se añade una prenda correctamente

El usuario añade correctamente una prenda a su armario. En este caso, el usuario quiere añadir una sudadera.

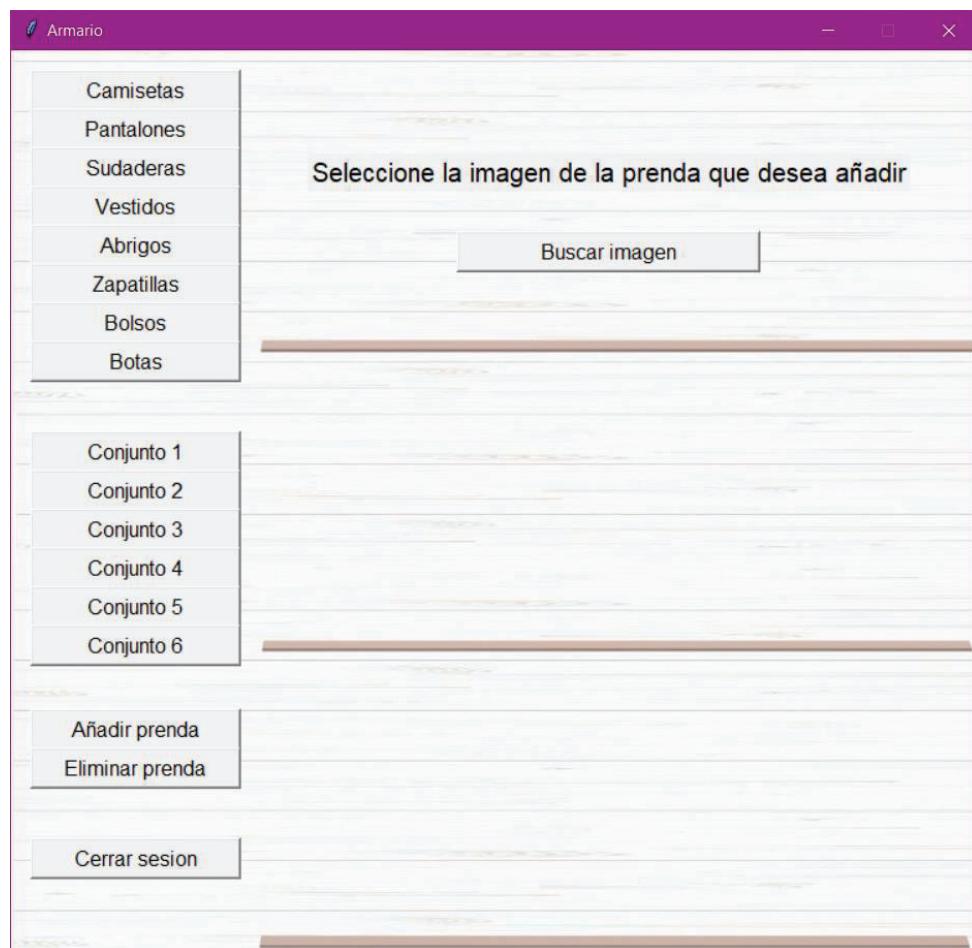


Ilustración 44 Nueva prenda - Incio

Cuando el usuario pulsar “Añadir una prenda” en la pantalla principal, se da paso a la ilustración 44. Si pulsa el botón “Buscar imagen” puede seleccionar la imagen de la sudadera dado que se le muestra el contenido de la carpeta resources, como se puede ver en la ilustración 45.

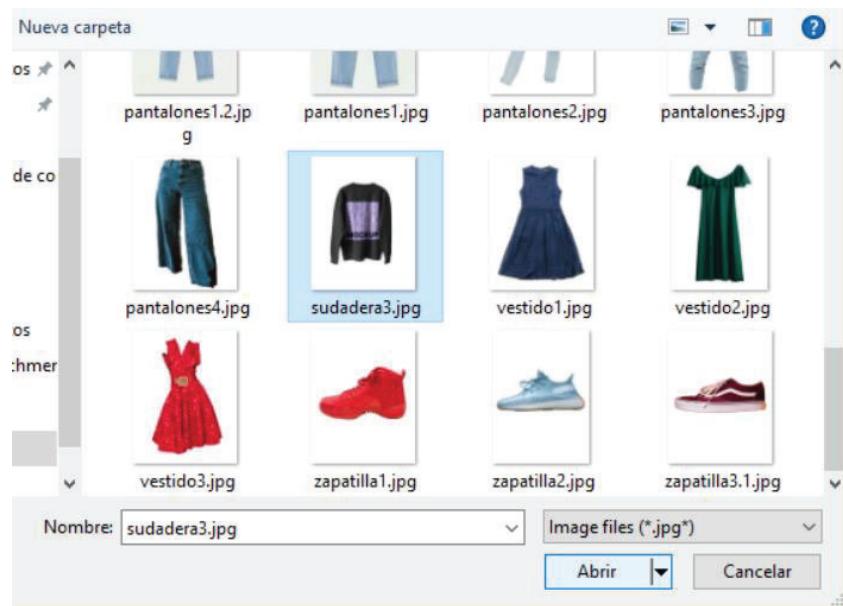


Ilustración 45 Nueva prenda - Contenido de la carpeta Resources

Cuando el usuario elige la prenda que desea añadir, pulsa Abrir y pasa a visualizar la ilustración 46.

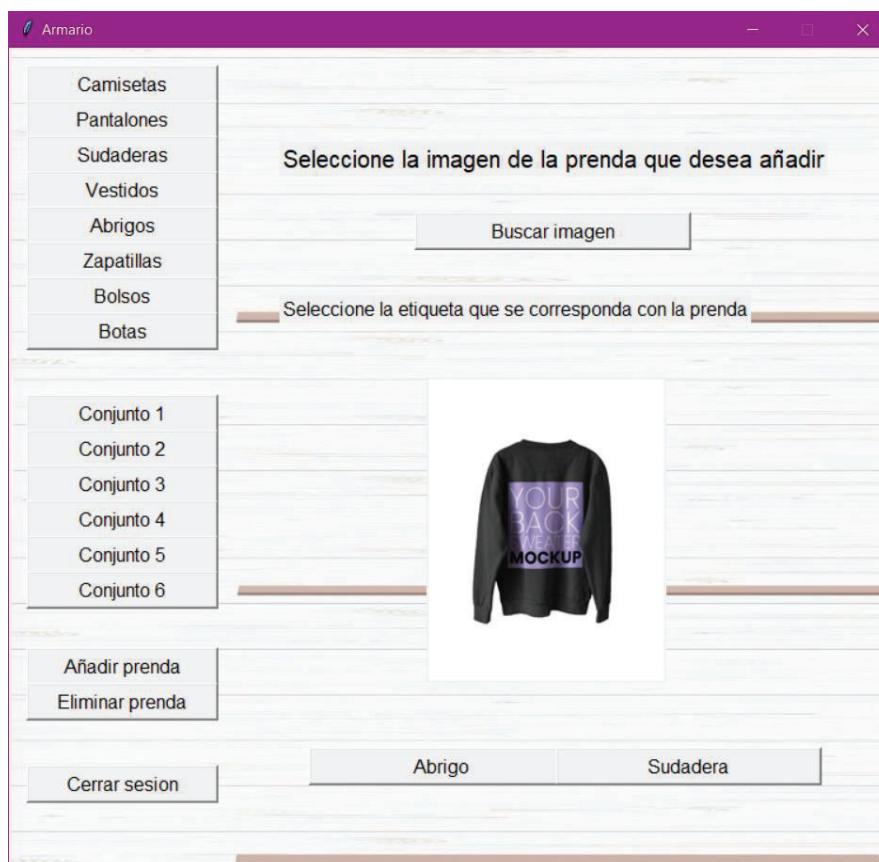


Ilustración 46 Nueva prenda - elección de etiqueta

Cuando se pulsa una etiqueta, se añade la prenda al armario y el sistema muestra el siguiente mensaje de aviso:

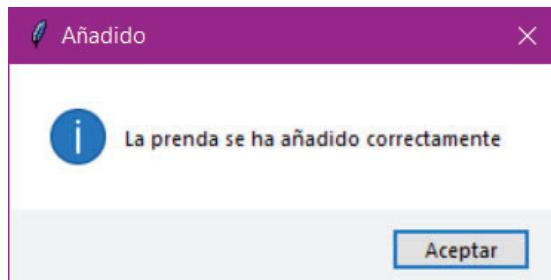


Ilustración 47 Nueva prenda - Mensaje del sistema

Para comprobar que la imagen se encuentra en el armario, se comprueban las entradas de la tabla “prendas” de la base de datos. Se pude ver que efectivamente se ha guardado la imagen y se ha establecido como identificador del usuario el valor 1.

	idPrenda	nombrePrenda	pathImagen	etiquetaPrenda	idUser
▶	1	sudadera3.jpg	resources/sudadera3.jpg	Sudadera	1
*	NULL	NULL	NULL	NULL	NULL

Ilustración 48 Nuevo prenda - Tabla prendas

6.3.2 No se selecciona ninguna imagen

En el caso de que el usuario no seleccione ninguna imagen, el sistema mostrará el siguiente mensaje de advertencia al usuario.

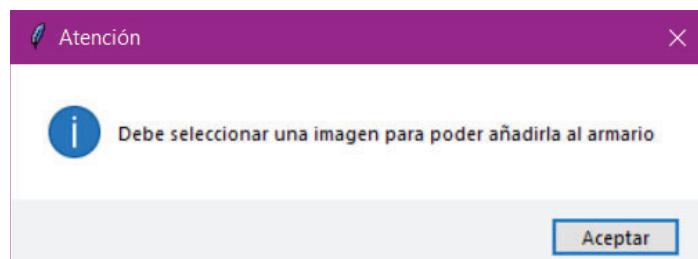


Ilustración 49 Nueva prenda - Mensaje del sistema

6.4 Visualizar las prendas de una categoría

El sistema debe permitir a un usuario visualizar todas las prendas almacenadas de una categoría.

En este caso, al solo haberse registrado una sudadera, esta será la única imagen que se muestre cuando el usuario pulse el botón “Sudaderas”.

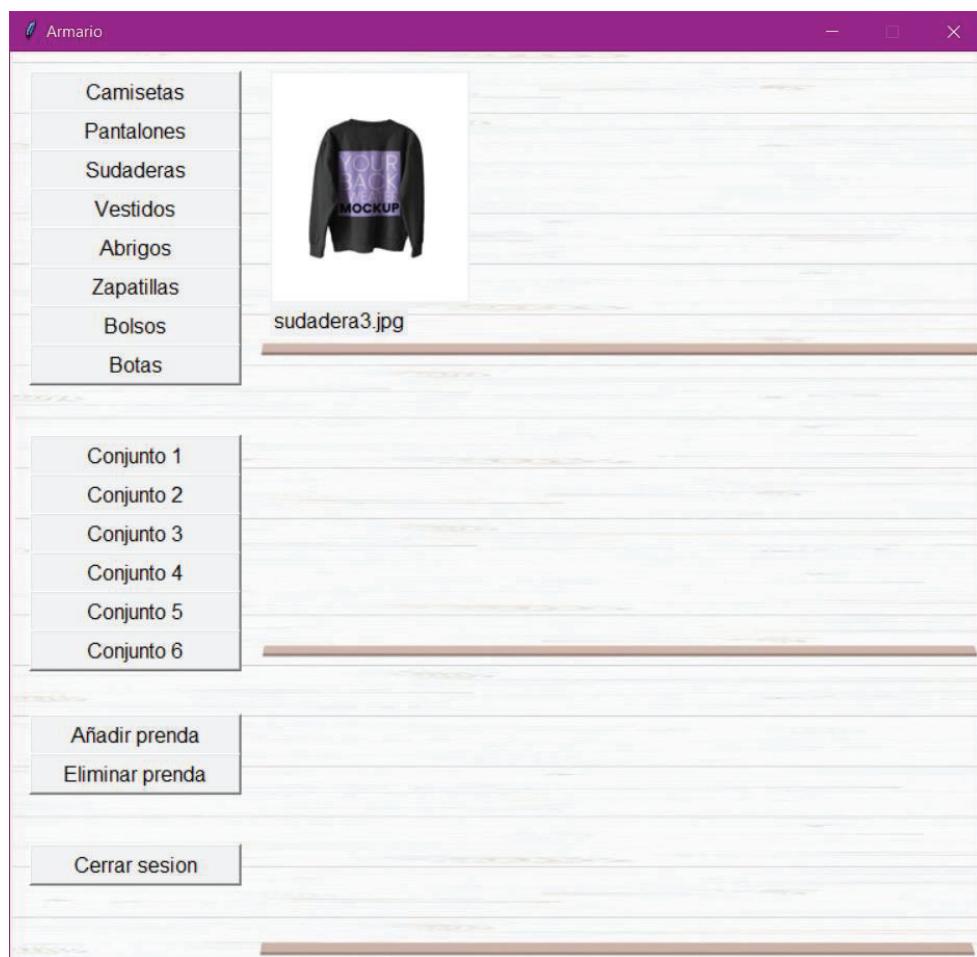


Ilustración 50 Visualización de las sudaderas

Como se muestra en la ilustración, el nombre que se muestra debajo de la foto es el mismo que tenía el archivo cuando se seleccionó.

6.5 Crear de un nuevo conjunto

Para la evaluación de este escenario se han añadido prendas a las categorías de zapatillas, pantalones, camisetas, vestidos y bolsos, dado que, aunque se puede crear el conjunto con solo una prenda, es más visual que lo compongan dos o más.

En la ilustración 51 se puede ver el contenido de la tabla “prendas” en este momento.

	idPrenda	nombrePrenda	pathImagen	etiquetaPrenda	idUser
▶	1	sudadera3.jpg	resources/sudadera3.jpg	Sudadera	1
	2	zapatilla3.1.jpg	resources/zapatilla3.1.jpg	Zapatillas	1
	3	pantalones1.2.jpg	resources/pantalones1.2.jpg	Pantalon	1
	4	camiseta3.jpg	resources/camiseta3.jpg	Camiseta	1
	5	camiseta4.jpg	resources/camiseta4.jpg	Camiseta	1
	6	vestido1.jpg	resources/vestido1.jpg	Vestido	1
	7	bolso3.jpg	resources/bolso3.jpg	Bolso	1
	8	bolso6.jpg	resources/bolso6.jpg	Bolso	1
	NULL	NULL	NULL	NULL	NULL

Ilustración 51 Nuevo conjunto - Tabla Prendas

El usuario crea un conjunto compuesto por una camiseta, un pantalón, una sudadera, unas zapatillas y un bolso:

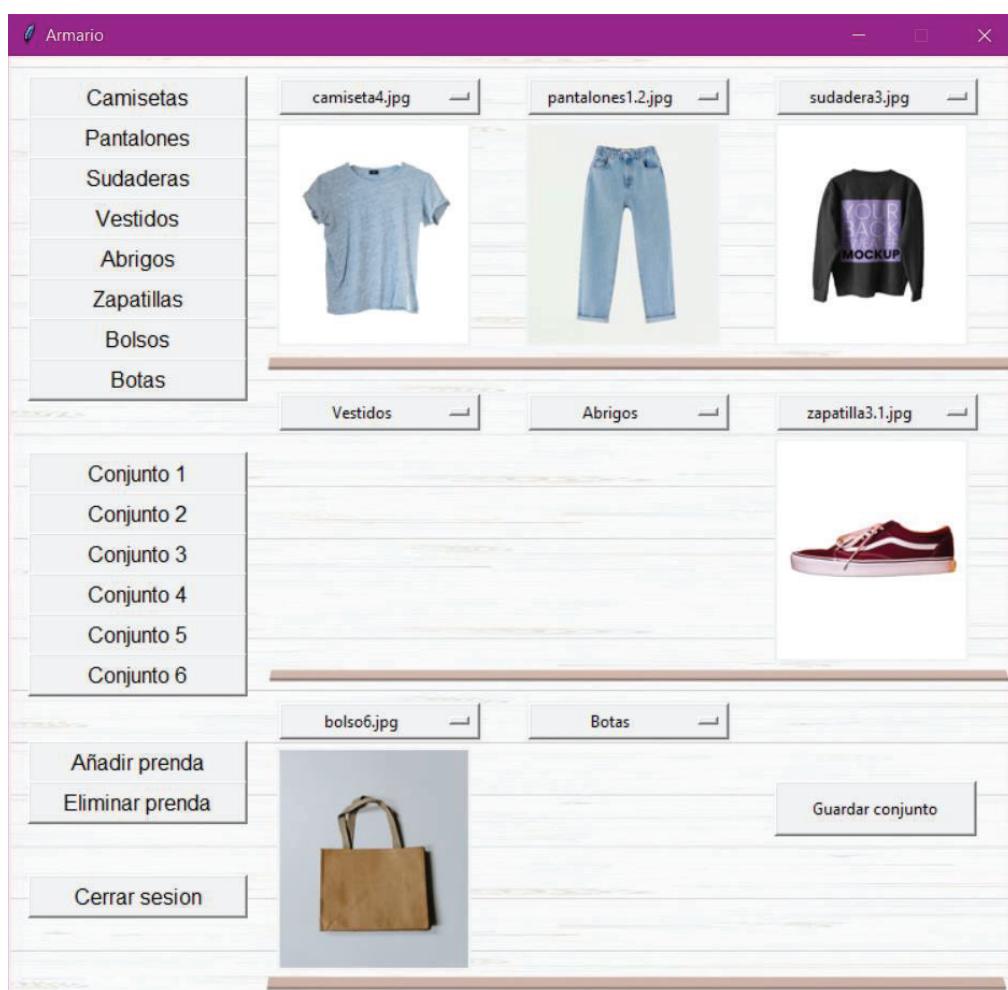


Ilustración 52 Nuevo conjunto - Conjunto 1

Cuando se pulsa “Guardar conjunto”, el sistema muestra el siguiente mensaje:

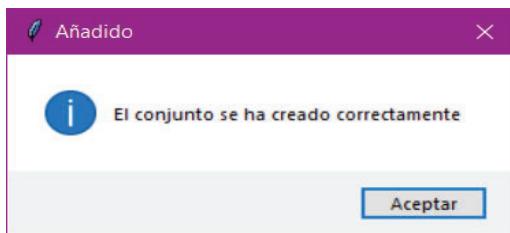


Ilustración 53 Nuevo conjunto - Mensaje del sistema

En la tabla de Conjuntos de la base de datos se puede comprobar que el conjunto se ha creado y, por lo tanto, cada vez que el usuario con identificador 1 pulse el botón “Conjunto 1” visualizará la ventana de la ilustración 52.

	idConjuntos	nombreConjunto	camiseta	pantalon	sudadera	vestido	abrigos	zapatillas	bolso	botin	idUsuario
▶	1	Conjunto1	5	3	1	NULL	NULL	2	8	NULL	1
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Ilustración 54 Nuevo conjunto - Tabla Conjuntos

6.6 Modificar de un conjunto existente

En este escenario se modifican algunas prendas que componen el conjunto creado en el apartado anterior.

Se ha decidido eliminar la sudadera y cambiar el bolso.

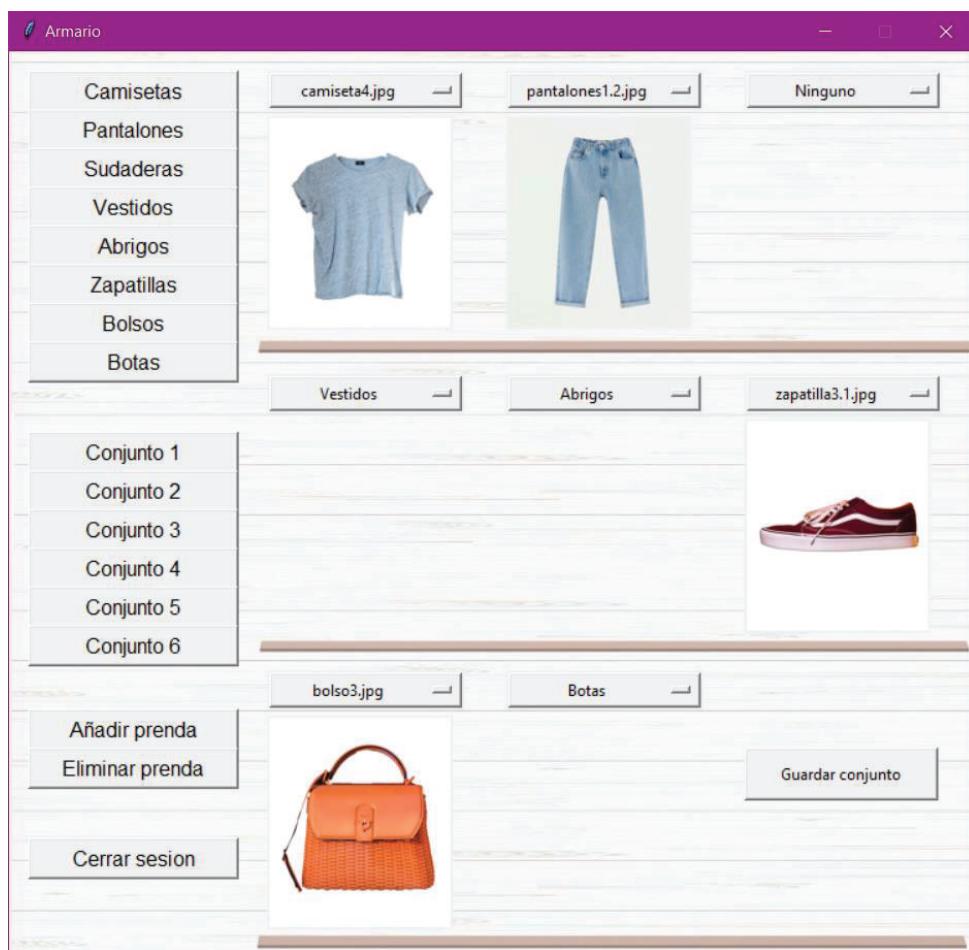


Ilustración 55 Modificar conjunto – Conjunto 1

Una vez se pulsa el botón “Guarda conjunto” se sobrescribe el conjunto y se muestra un mensaje informativo y se actualizarán los valores de la tabla “Conjuntos”.

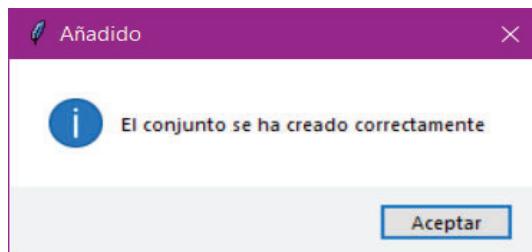


Ilustración 56 Nuevo conjunto - Mensaje del sistema

	idConjuntos	nombreConjunto	camiseta	pantalon	sudadera	vestido	abrigos	zapatillas	bolso	botin	idUser
▶	1	Conjunto1	5	3	NULL	NULL	NULL	2	7	NULL	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ilustración 57 Modificar conjunto - Tabla Conjuntos

6.7 Eliminar prendas del armario

El sistema debe permitir al usuario eliminar una prenda almacenada.

6.7.1 Se elimina una prenda que no conforma un conjunto

En ese caso, el sistema informará al usuario que la prenda o prendas se han eliminado correctamente.

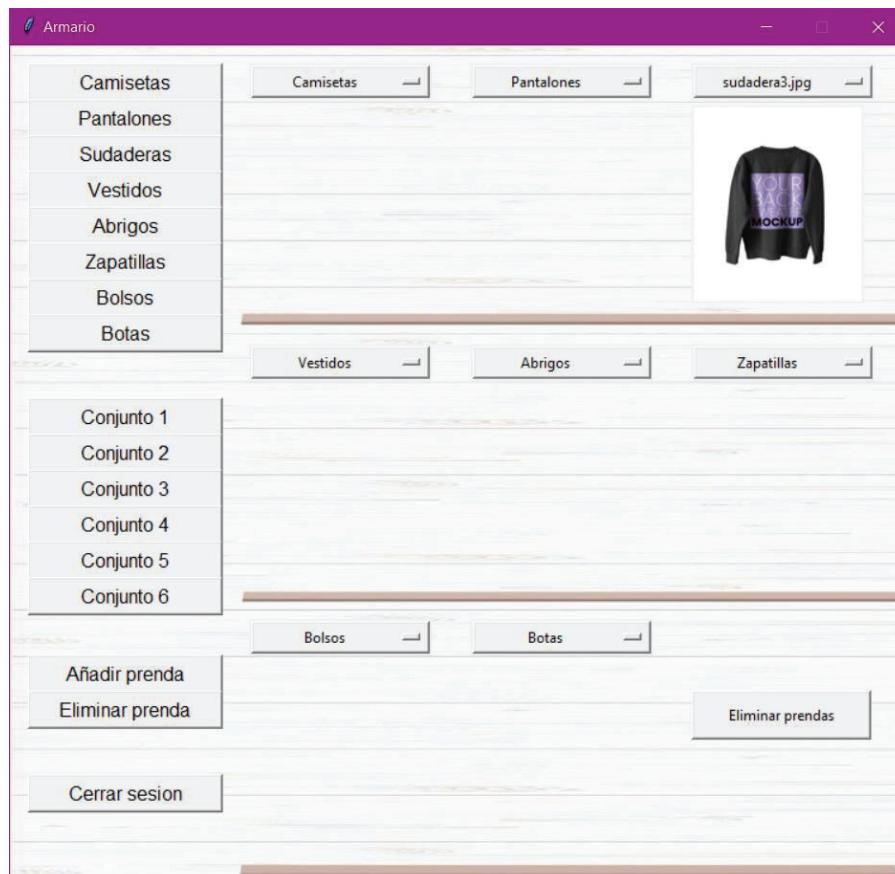


Ilustración 58 Eliminar prenda – Prenda no perteneciente a un conjunto

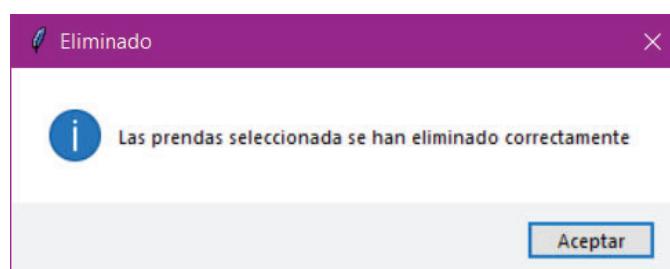


Ilustración 59 Eliminar prenda - Mensaje del sistema

En la base de datos se puede comprobar que ya no existe la prenda, pero no se modifica la tabla conjuntos. El contenido de la tabla prendas se muestra en la ilustración 60.

	idPrenda	nombrePrenda	pathImagen	etiquetaPrenda	idUser
▶	2	zapatilla3.1.jpg	resources/zapatilla3.1.jpg	Zapatillas	1
	3	pantalones1.2.jpg	resources/pantalones1.2.jpg	Pantalon	1
	4	camiseta3.jpg	resources/camiseta3.jpg	Camiseta	1
	5	camiseta4.jpg	resources/camiseta4.jpg	Camiseta	1
	6	vestido1.jpg	resources/vestido1.jpg	Vestido	1
	7	bolso3.jpg	resources/bolso3.jpg	Bolso	1
*	8	bolso6.jpg	resources/bolso6.jpg	Bolso	1
	HULL	HULL	HULL	HULL	HULL

Ilustración 60 Eliminar prenda - Tabla Prendas

6.7.2 Se elimina una prenda que conforma un conjunto

Se elimina una prenda que forma parte del conjunto 1.

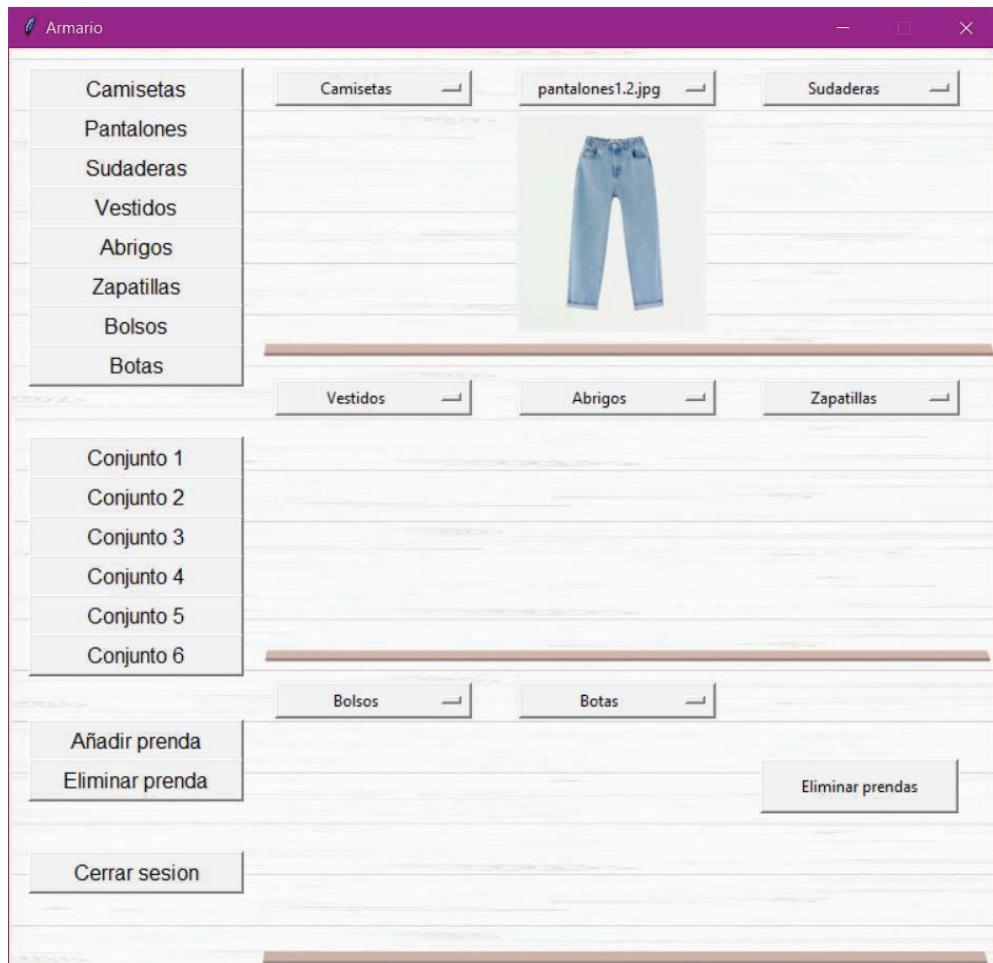


Ilustración 61 Eliminar prenda – Prenda perteneciente a un conjunto

Cuando se pulsa “Eliminar prendas” el mensaje que el sistema proporciona es el siguiente dado que existía un conjunto con esa prenda:

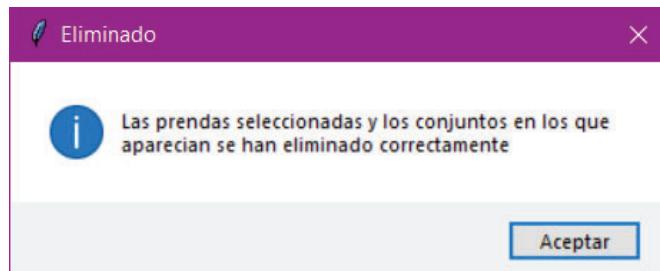


Ilustración 62 Eliminar prenda - Mensaje del sistema

6.7.3 No se selecciona ninguna prenda

En el caso en el que no se seleccione ninguna prenda, pero si se pulse el botón “Eliminar prendas”, el sistema le mostrará el siguiente mensaje de advertencia:

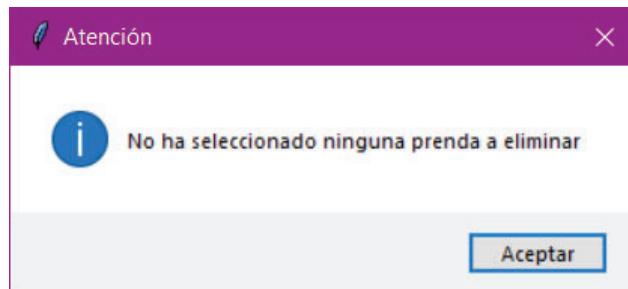


Ilustración 63 Eliminar prenda - Mensaje del sistema

7 Resultados y conclusiones

El objetivo de este Trabajo de Fin de Grado residía en el desarrollo de un sistema que emulase a un armario que tuviese un clasificador para la asignación de las categorías de las prendas. Adicionalmente, se buscaba que el sistema permitiese la creación de distintos conjuntos de ropa con las prendas que se hubiesen almacenado en el sistema.

A lo largo del desarrollo del sistema, se añadieron otras funcionalidades como la creación de perfiles, la eliminación de prendas de un usuario y la opción de cerrar una sesión, con el objetivo de que el sistema proporcionase al usuario una experiencia más agradable.

En vista de los resultados obtenidos durante las pruebas del sistema descritas en el [apartado 6](#), se puede concluir que la solución propuesta cumple con todos los objetivos que se habían establecido inicialmente y los que se adoptaron a lo largo del desarrollo.

Se ha buscado que el sistema fuese lo más visual posible dado que las imágenes de las prendas son las piezas fundamentales de este sistema, lo que ha dado lugar a una aplicación muy intuitiva. A pesar de esto, se ha tenido en cuenta posibles escenarios en los que el usuario no seguía la línea establecida para el uso del sistema. Esto ha hecho que el sistema sea muy interactivo dado que apenas hay restricciones para el usuario y, además, como se ha buscado que en lo máximo posible el usuario no tuviese que tocar ningún teclado para interactuar con el sistema, se ha obtenido un sistema accesible.

Gracias a este proyecto se ha podido profundizar en el desarrollo de un clasificador y comprender mejor su funcionamiento hasta el punto de poder adaptarlo a un problema concreto. Además, se ha podido aprender más acerca del lenguaje de programación Python, de las librerías que se pueden emplear con este lenguaje relacionadas con Machine Learning y del manejo de bases de datos.

En términos generales el proyecto ha sido un éxito, sin embargo, tiene margen de mejora dado que se le pueden añadir otro tipo de acciones como la creación automática de conjuntos o ampliar el número de prendas por categoría y conjuntos que se pueden almacenar.

8 Análisis de Impacto

En base a lo que se comentó en el [apartado 1.1](#) y a los resultados obtenido, se puede decir que el impacto en el ámbito económico de los usuarios puede ser el más destacable debido a que, como se mencionó anteriormente, uno de los motivos por el que una persona puede llegar a tener una gran cantidad de ropa es por el desconocimiento de todas las prendas que posee. Si esto no fuese así, una persona podría reducir sus gastos en prendas de vestir. Como consecuencia de esto, nuestra cultura de usar y tirar disminuiría dado que, al poder planificar los conjuntos, se realizarían compras más planificadas y menos impulsivas de prendas pocos ponibles. Esto afectaría directamente al medio ambiente al reducir los desechos y los contaminantes de las fábricas textiles al disminuir la demanda.

Dado que se pueden aumentar las características del sistema a nivel social puede implicar una mejora en la calidad de vida de las personas dado que el sistema permitirá planificar una parte de la rutina diaria con mucha antelación. En casos de personas a cargo de otras a las que tienen que preparar el atuendo diariamente esto puede reducir notablemente su tiempo, por ejemplo, en familias con varios niños pequeños. Otro caso sería el de personas a las que les resulta complicado ver prenda por prenda toda la ropa para elegir un conjunto, bien por problemas de movilidad o por no ser autónomos. En este caso, el sistema facilitaría la visualización al mostrar todo en una sola pantalla sin necesidad de revolver las prendas. Es por ello por lo que se buscó que en la medida de lo posible el sistema fuese táctil.

Además, podría implicar la modernización de los armarios dado paso a armario inteligentes como en las películas.

Con respecto al ámbito empresarial se abren dos vías: la reducción de ingresos en las empresas textiles y la comercialización de sistemas parecidos tanto en formato aplicación para dispositivos móviles como en formato de un sistema complementario para los armarios. Eso dependerá del comportamiento de los usuarios.

A partir de lo analizado, los Objetivos de Desarrollo Sostenible de la Agenda 2030 en los que se podría influir son: salud y bienestar (3) y producción y consumos responsables (12) principalmente.

9 Bibliografía

- [1] I. P. Gestal, «modaes.es,» 31 Enero 2019. [En línea]. Available: <https://www.modaes.es/entorno/la-modas-a-la-reconquista-del-bolsillo-espanol-el-gasto-per-capita-crecerá-un-16-hasta-casi-500-euros-en-2023.html>.
- [2] INE, «ine.es,» 2019. [En línea]. Available: <https://www.ine.es/jaxiT3/Tabla.htm?t=25177>.
- [3] «ec.europa.eu,» Eurostat, 25 Enero 2021. [En línea]. Available: [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Online_purchases_of_goods,_EU-27\(1\),_2020_\(%25_of_individuals_who_bought_or_ordered_goods_or_services_over_the_internet_for_private_use_in_the_previous_3_months\).png#filehistory](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Online_purchases_of_goods,_EU-27(1),_2020_(%25_of_individuals_who_bought_or_ordered_goods_or_services_over_the_internet_for_private_use_in_the_previous_3_months).png#filehistory).
- [4] B. A. a. M. Anderson, «Pew Research Center,» [pewresearch.org, 7 Abril 2021.](https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/) [En línea]. Available: <https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/>.
- [5] F. K. Zlatinova, *La eficacia de la publicidad en las redes sociales*, 2020.
- [6] Universidad Politécnica de Madrid, «Introducción al Reconocimiento de Formas,» Reconocimiento de Formas, 2020.
- [7] P. E. H. D. G. S. Richard O.Duda, Pattern Classification, Jonh Wiley & Sons Inc, 2000.
- [8] K. R. R. V. Han Xiao, «Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,» 28 08 2017. [En línea]. Available: <https://github.com/zalandoresearch/fashion-mnist>.
- [9] K. D. Andrew R. Webb, Statistical Pattern Recognition, John Wiley & Sons, 2011.
- [10] Universiad Politécnica de Madrid, «Clasificador estadístico,» Reconocimiento de Formas, 2020.
- [11] M. Binieli, «freeCodeCamp,» 16 10 2018. [En línea]. Available: <https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/>.
- [12] Universidad Politécnita de Madrid, «Evaluación de rendimiento,» Reconocimiento de Formas, 2020.
- [13] G. L. d. Torre, «Validación de clasificadores,» 16 1 2020. [En línea]. Available: <https://www.gorka.io/2020/01/16/validacion-de-clasificadores/>.

- [14] Universidad Politécnica de Madrid, «Reconocimiento de formas,» Grado en Ingeniería Informática, Madrid, 2020.
- [15] Python Software Foundation, «math — Mathematical functions,» 27 6 2021. [En línea]. Available: <https://docs.python.org/3/library/math.html>.
- [16] The NumPy community., «NumPy v1.21 Manual,» 22 6 2021. [En línea]. Available: <https://numpy.org/doc/stable/index.html>.
- [17] P. e. al, «Scikit-learn: Machine Learning in Python,» *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [18] A. A. P. B. E. B. Z. C. C. G. S. C. A. D. J. D. M. D. S. G. I. G. A. H. G. I. M. I. R. J. Y. J. Martín Abadi, «TensorFlow: Large-scale machine learning on heterogeneous systems,» 2015. [En línea]. Available: tensorflow.org.
- [19] Intel Corporation, «OpenCV,» 2021. [En línea]. Available: <https://opencv.org>.
- [20] doxygen, «OpenCV,» 2 4 2021. [En línea]. Available: https://docs.opencv.org/4.5.2/d7/d4d/tutorial_py_thresholding.html.
- [21] Oracle, «MySQL Connector/Python Developer Guide,» [En línea]. Available: <https://dev.mysql.com/doc/connector-python/en/>.

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Thu Jul 01 15:09:20 CEST 2021
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)