



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Plataforma web para el reconocimiento de emociones  
basada en características faciales

Web platform for emotion recognition based on facial  
features

Autor

Simón Alonso Gutiérrez

Directores

Sandra Baldassarri

José Ramón Beltrán

ESCUELA DE INGENIERÍA Y ARQUITECTURA  
2024





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./D<sup>a</sup>. Simón Alonso Gutiérrez ,

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Grado en Ingeniería Informática  (Título del Trabajo)

Plataforma web para el reconocimiento de emociones basada en características faciales

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 26/06/2024

Fdo: Simón Alonso Gutiérrez



# AGRADECIMIENTOS

A mis profesores Sandra Baldassarri y José Ramón Beltrán por su guía y acompañamiento durante el proyecto que se presenta en el documento.

A mi familia por su incondicional apoyo y brindarme la motivación para superar cada desafío.

A mis compañeros por los buenos momentos compartidos durante este camino.



# RESUMEN

En la actualidad, la gestión y realización de experimentos son fundamentales para la investigación en la computación afectiva, disciplina que estudia la interacción entre los sistemas informáticos y las emociones humanas. Sin embargo, estos experimentos presentan desafíos como la recopilación de información sobre el estado emocional del usuario mientras realiza una tarea. Esta información puede obtenerse mediante el reconocimiento de expresiones faciales utilizando una cámara o a través de diversos sensores fisiológicos.

En este trabajo de fin de grado se desarrolla una plataforma que facilita a los investigadores recopilar, visualizar y analizar datos experimentales. Los investigadores tienen la capacidad de crear y configurar experimentos, personalizando el conjunto de tareas y su estructura para ofrecer una experiencia que se adapte a sus necesidades.

La plataforma permite a los usuarios acceder y participar en experimentos previamente creados. Está integrada un modelo propio de predicción de emociones basado en características faciales, el cual predice la emoción experimentada por el usuario durante la ejecución del experimento. Además de recopilar y almacenar la emoción predicha por el modelo, la plataforma registra las respuestas a las encuestas planteadas. Además, estos resultados se pueden visualizar de forma individual para cada usuario o grupal para un análisis colectivo.

El modelo propio de predicción de emociones a partir de las características faciales utilizado por la plataforma, ha sido desarrollado desde cero utilizando una base de datos pública. Este modelo tiene la capacidad de predecir las emociones de disgusto, tristeza, alegría, neutral, enfado, sorpresa y miedo exclusivamente a partir de características faciales, logrando una precisión global del 62.46 %.

La plataforma web se ha desplegado para que sea accesible a través del navegador. Adicionalmente, en este trabajo se lleva a cabo un experimento con un grupo reducido de usuarios, para mostrar la creación, realización y análisis del mismo.



# ABSTRACT

Currently, the management and execution of experiments are fundamental for research in affective computing, a discipline that studies the interaction between computer systems and human emotions. However, these experiments present challenges such as collecting information about the user's emotional state while performing a task. This information can be obtained through facial expression recognition using a camera or various physiological sensors.

In this bachelor thesis, a platform is developed that facilitates researchers in collecting, visualizing, and analyzing experimental data. Researchers have the capability to create and configure experiments, customizing the set of tasks and their structure to offer an experience that suits their needs.

The platform allows users to access and participate in pre-created experiments. It integrates its own emotion prediction model based on facial features, which predicts the emotion experienced by the user during the execution of the experiment. In addition to collecting and storing the emotion predicted by the model, the platform records the responses to the surveys provided. Furthermore, these results can be visualized individually for each user or collectively for a group analysis.

The platform's own emotion prediction model based on facial features has been developed from scratch using a public database. This model is capable of predicting emotions such as disgust, sadness, joy, neutral, anger, surprise, and fear exclusively from facial features, achieving an overall accuracy of 62.46 %.

The web platform has been deployed to be accessible through a browser. Additionally, in this work, an experiment is conducted with a small group of users to demonstrate its creation, execution, and analysis.



# Índice

<b>1. Introducción y objetivos</b>	<b>3</b>
1.1. Introducción . . . . .	3
1.2. Contexto . . . . .	4
1.3. Objetivos del trabajo . . . . .	4
1.4. Estructura del documento . . . . .	4
<b>2. Análisis del sistema</b>	<b>7</b>
2.1. Estudio de aplicaciones de reconocimiento emocional . . . . .	7
2.2. Requisitos de la aplicación . . . . .	8
<b>3. Diseño del sistema</b>	<b>11</b>
3.1. Patrón de diseño . . . . .	11
3.2. Arquitectura software del sistema . . . . .	12
3.3. Interfaz de usuario . . . . .	13
3.4. Uso de JSON Web Tokens para garantizar la seguridad . . . . .	25
3.5. Contenido de la API del backend . . . . .	26
3.6. Diseño del acceso al modelo de inteligencia artificial . . . . .	27
3.7. Diseño de la base de datos NoSQL . . . . .	28
<b>4. Implementación del sistema</b>	<b>31</b>
4.1. Componentes utilizados en el frontend de la aplicación . . . . .	31
4.2. Componentes utilizados en el backend de la aplicación . . . . .	38
4.3. Gestión de los archivos en el servidor . . . . .	40
4.4. Integración del modelo . . . . .	41

4.5. Despliegue de los servidores en Azure . . . . .	41
<b>5. Modelo de reconocimiento de emociones</b>	<b>43</b>
5.1. Descripción del proceso . . . . .	43
5.2. Selección de los datos a utilizar . . . . .	44
5.3. Preprocesamiento de los datos . . . . .	44
5.4. Extracción de características . . . . .	46
5.5. Selección del modelo . . . . .	47
5.6. Entrenamiento y validación . . . . .	48
5.7. Evaluación . . . . .	51
<b>6. Evaluación del sistema</b>	<b>53</b>
6.1. Metodología . . . . .	53
6.2. Resultados y análisis . . . . .	54
<b>7. Gestión del proyecto</b>	<b>61</b>
7.1. Desarrollo del proyecto . . . . .	61
<b>8. Conclusiones y trabajo futuro</b>	<b>63</b>
8.1. Conclusiones . . . . .	63
8.2. Conocimientos adquiridos . . . . .	64
8.3. Trabajo futuro . . . . .	65
<b>9. Bibliografía</b>	<b>67</b>
<b>Lista de Figuras</b>	<b>69</b>
<b>Lista de Tablas</b>	<b>73</b>
<b>Anexos</b>	<b>74</b>
<b>A. Análisis detallado de aplicaciones de reconocimiento emocional</b>	<b>77</b>
<b>B. Detalles de la API del backend</b>	<b>79</b>

<b>C. Tipos de gráficas disponibles en la plataforma</b>	<b>83</b>
<b>D. Modelos y técnicas descartadas para el reconocimiento emocional</b>	<b>87</b>
D.1. Modelo para el reconocimiento de estrés/no estrés . . . . .	87
D.2. Exploración de modelos y técnicas para el reconocimiento de emociones básicas . . . . .	89
<b>E. Herramientas de trabajo</b>	<b>91</b>



# Capítulo 1

## Introducción y objetivos

Este capítulo presenta la introducción del trabajo, el contexto en el que se desarrolla, los objetivos planteados y la estructura general de la memoria.

### 1.1. Introducción

Las emociones, esos estados afectivos que moldean nuestra experiencia y guían nuestras acciones, son un elemento fundamental de la condición humana. A través de ellas percibimos el mundo, interactuamos con los demás y tomamos decisiones [1]. Si bien Paul Ekman identificó las 7 emociones básicas (ira, asco, miedo, desprecio, alegría, tristeza y sorpresa), existe un rango más amplio de emociones que también incluye por ejemplo el estrés o el aburrimiento, fundamentales para comprender el comportamiento humano. Aquí entra en juego la computación afectiva, que se encarga de desarrollar sistemas capaces de reconocer e interpretar las emociones humanas a través de las señales biométricas del cuerpo humano. Si bien existen diversas señales biométricas como el ritmo cardiaco, la temperatura o los electroencefalogramas, este trabajo se enfoca principalmente en las expresiones faciales, que son una forma de comunicación no verbal que refleja nuestras emociones.

El reconocimiento de emociones a partir de expresiones faciales (FER, por sus siglas en inglés) [2] ha ganado considerable relevancia en los últimos años debido a su gran variedad de aplicaciones en diversos campos, como la salud mental, la educación o incluso el marketing. Tradicionalmente, se han empleado técnicas de *machine learning*, como *Support Vector Machine* (SVM) y *Random Forest* (RF) para el reconocimiento automático de emociones. Sin embargo, el auge del *Deep Learning* ha permitido mejorar los resultados otorgando sistemas más precisos y robustos.

## 1.2. Contexto

Este trabajo de fin de grado tiene relevancia en el ámbito de la computación afectiva, ya que proporciona una plataforma que permite el reconocimiento automático de emociones. Además, permite a los investigadores realizar experimentos configurables, evaluando a diversos usuarios y facilitando el análisis posterior de los datos obtenidos. Estas funcionalidades resultan especialmente útiles en proyectos de investigación relacionados con la salud, como por ejemplo para ayudar al estudio de los cambios emocionales en pacientes cardíacos [3].

## 1.3. Objetivos del trabajo

Este trabajo fin de grado tiene como objetivo general desarrollar una plataforma web que permita el reconocimiento de las emociones basada en características faciales. Para ello se proponen los siguientes objetivos:

- Investigar y analizar aplicaciones similares disponibles en la actualidad.
- Diseñar y desarrollar una plataforma web que permita evaluar a usuarios.
- Permitir la creación de experimentos configurables.
- Gestionar los datos para permitir su visualización.
- Estudiar las diferentes técnicas utilizadas para el reconocimiento de emociones mediante características faciales.
- Integrar un modelo de aprendizaje automático que sea capaz de realizar la predicción emocional.
- Desplegar la aplicación para que sea accesible a través del navegador.

## 1.4. Estructura del documento

Este documento está dividido en dos partes. La primera parte consiste en la memoria del proyecto, la cual resume el trabajo realizado. La segunda parte incluye los anexos, que complementan la información proporcionada en la memoria principal.

La memoria cuenta con los siguientes capítulos:

- Capítulo 1 - Introducción y objetivos: explica el contexto de la aplicación y los objetivos a abordar en el proyecto.
- Capítulo 2 - Análisis del sistema: presenta las aplicaciones similares investigadas, junto con los requisitos funcionales y no funcionales del sistema.
- Capítulo 3 - Diseño del sistema: comenta el patrón de diseño utilizado, la arquitectura software, la interfaz de la plataforma y el diseño de la base de datos utilizada.
- Capítulo 4 - Implementación del sistema: contiene los aspectos más importantes de las funcionalidades de la plataforma y la explicación detallada de los principales componentes del backend y frontend.
- Capítulo 5 - Modelo de aprendizaje automático: explica los procesos seguidos para el desarrollo del modelo de predicción emocional a partir de características faciales, que cuenta con la recolección de los datos, limpieza de los mismos, selección del modelo, entrenamiento, validación y la evaluación del modelo.
- Capítulo 6 - Evaluación del sistema: se evalúa el sistema llevando a cabo un experimento con un grupo reducido de usuarios.
- Capítulo 7 - Gestión del proyecto: comenta el desarrollo del proyecto, la metodología seguida y el tiempo dedicado al proyecto.
- Capítulo 8 - Conclusiones y trabajo futuro: explica las conclusiones obtenidas tras la realización del proyecto y se comentan posibles tareas a implementar en el futuro.

El apartado de anexos incluye los siguientes subapartados:

- Anexo A: Análisis detallado de aplicaciones de reconocimiento emocional.
- Anexo B: Detalles de la API del backend.
- Anexo C: Tipos de gráficas disponibles en la plataforma.
- Anexo D: Modelos y técnicas descartadas para el reconocimiento emocional.
- Anexo E: Herramientas de trabajo.



# Capítulo 2

## Análisis del sistema

Este capítulo muestra las aplicaciones de reconocimiento emocional investigadas, lo que permite concluir sobre las funcionalidades necesarias para la aplicación, seguido del análisis de requisitos que la aplicación debe cumplir.

### 2.1. Estudio de aplicaciones de reconocimiento emocional

En esta sección se analizan diversas aplicaciones similares a la presentada en este proyecto para identificar las características clave y sus limitaciones. Durante el proceso de búsqueda, se investiga principalmente aplicaciones web que permitan reconocer emociones, almacenar información para su posterior análisis y que sean configurables.

La aplicación web a desarrollar en este trabajo debe contar con distintos perfiles: el administrador, quien tiene permisos de superusuario para modificar los roles de los usuarios; el experimentador, quien puede crear y visualizar experimentos; y el usuario que puede participar en los experimentos. La aplicación web debe permitir la creación y configuración de experimentos con el objetivo de recopilar información acerca de las emociones del usuario, permitiendo una visualización posterior de los resultados. En base a estas características se muestran en la tabla 2.1 las aplicaciones más similares que se han encontrado: Morphcast [4], FaceReader [5], Afectiva [6], Emotimeter [7] e Imentiv [8]. Se encuentra un análisis más detallado en el Anexo A: Análisis detallado de aplicaciones de reconocimiento emocional.

Aplicación	Plataforma	¿Recopila información?	¿Permite análisis?	¿Es configurable?	Tipo de entrada
Morphcast	Escritorio Web	Sí	Sí	Sí	Cámara
FaceReader	Escritorio	Sí	Sí	Sí	Cámara
Afectiva	Escritorio Web	Sí	Sí	Sí	Cámara
Emotimeter	Android	No	No	No	Cámara
Imentiv	Escritorio Web	Sí	Sí	No	Video pregrabado

Nota: Las versiones web de Morphcast y Afectiva son demostraciones que muestran lo que son capaces de hacer, pero no integran todas las funcionalidades que tienen en su aplicación de escritorio.

Tabla 2.1: Tabla de aplicaciones significativas

En la tabla se observa que hay tres aplicaciones que cumplen con los principales requerimientos, ser configurable, recopilar información y permitir el análisis. A pesar de que en la tabla hay tres aplicaciones que son configurables, el tipo de configuración es diferente al requerido, ya que éstas se centran en la configuración de las emociones a predecir y en los métodos a utilizar para dicha predicción. El requerimiento específico es permitir la creación y configuración de un experimento, especificando su contenido.

Esto resalta la relevancia de este proyecto, el cual tiene como objetivo desarrollar una plataforma web para el reconocimiento de emociones basado en características faciales con la capacidad de configurar experimentos específicos que permitan su visualización y análisis posterior.

## 2.2. Requisitos de la aplicación

En base a los resultados del estudio previo, se determina que se debe desarrollar una aplicación web que permita la configuración de experimentos. La aplicación debe permitir a los experimentadores seleccionar los archivos multimedia sobre los cuales se recopila información de los usuarios, además de la capacidad de realizar encuestas antes y después de cada evaluación para obtener retroalimentación de los usuarios. Una vez hay datos en los experimentos, los experimentadores deben poder visualizar las emociones predichas y los resultados de las encuestas. Este resumen de funcionalidades se traduce en los siguientes requisitos mostrados en la tabla 2.2.

<b>Requisitos Funcionales</b>	
<b>Requisito</b>	<b>Descripción</b>
RF-1	El sistema debe permitir a los usuarios registrarse e iniciar sesión.
RF-2	El sistema debe gestionar diferentes roles de usuario: Administrador, Experimentador y Usuario.
RF-2.1	El Administrador debe tener la capacidad de cambiar roles y realizar experimentos.
RF-2.2	El Experimentador debe poder realizar experimentos, configurar experimentos y consultar estadísticas de sus experimentos.
RF-2.3	El Usuario debe poder participar en experimentos.
RF-4	El sistema debe permitir al Experimentador configurar los experimentos.
RF-4.1	Debe ser posible configurar el título, la descripción, un pretexto, un posttexto y la contraseña del experimento.
RF-4.2	El Experimentador debe poder añadir tareas al experimento.
RF-4.3	El Experimentador debe poder seleccionar el contenido a mostrar en cada tarea o subir archivos propios.
RF-4.4	Para cada tarea, se debe permitir la inclusión de encuestas previas, contenido y encuestas posteriores.
RF-4.5	Las encuestas deben poder configurarse como formularios tipo SAM o preguntas.
RF-4.6	Las preguntas de las encuestas pueden ser de tipo sí/no, elección múltiple (permitiendo agregar opciones) o rango numérico (especificando el intervalo).
RF-4.7	El sistema debe permitir añadir un pretexto y un posttexto, configurando el título y la descripción de cada uno.
RF-5	El sistema debe almacenar los resultados a las encuestas y las emociones que se han predicho
RF-6	El sistema debe permitir la captura y almacenamiento de imágenes.
RF-7	El sistema debe proporcionar retroalimentación en tiempo real sobre la emoción predicha.
RF-8	El sistema debe permitir mostrar una vista previa en tiempo real de la cámara web del usuario
RF-9	El sistema debe permitir al experimentador visualizar las estadísticas de las emociones durante la realización del experimento.
RF-9.1	El sistema debe permitir al experimentador mostrar estadísticas sobre las emociones durante cada tarea
RF-9.2	El sistema debe permitir al experimentador observar las estadísticas tanto de forma global, haciendo un promedio entre todos los usuarios, como de forma individual
RF-9.3	El sistema debe permitir al experimentador observar las imágenes tomadas por la webcam del usuario
RF-10	El sistema debe permitir al experimentador observar los resultados de las encuestas por usuario o por experimento.
RF-11	El sistema debe permitir al Administrador cambiar los roles de los usuarios.

RF-12	El sistema debe permitir integrar al menos un modelo de aprendizaje automático
<b>Requisitos No Funcionales</b>	
RNF-1	El sistema requiere permisos de cámara durante la realización de un experimento.
RNF-2	El sistema debe ser utilizado con conexión a Internet.
RNF-3	El sistema debe ser ejecutable en un navegador web.
RNF-4	El modelo de aprendizaje automático tiene que ser ejecutable en el servidor.

Tabla 2.2: Requisitos Funcionales y No Funcionales

# Capítulo 3

## Diseño del sistema

Este capítulo presenta el patrón de diseño adoptado, la arquitectura software utilizada, la interfaz de usuario final de la plataforma, el por qué del uso de tokens para garantizar la seguridad, el contenido de la API del backend, la interacción con el modelo de inteligencia artificial y el diseño de la base de datos utilizada para almacenar la información.

### 3.1. Patrón de diseño

En la figura 3.1 se presenta el patrón de diseño utilizado en la plataforma. Este se basa en el patrón Modelo Vista Controlador (MVC) para un diseño modular, complementado con el patrón Repositorio para abstraer el acceso a los datos y una capa *WebSocket* para permitir la interacción en tiempo real.

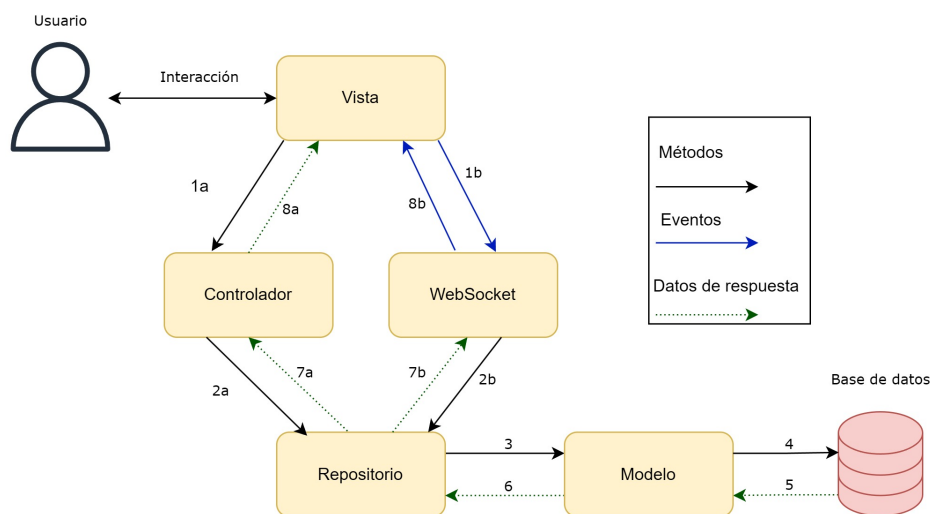


Figura 3.1: Patrón de diseño MVC con las capas adicionales de websocket y repositorio. Ver texto.

La *vista* proporciona la interfaz que permite al usuario interactuar con la aplicación y se comunica con la capa *controlador* (1a) para realizar peticiones a través del protocolo HTTP. El controlador procesa la petición y ejecuta la lógica de negociación, solicitando la información (en alto nivel) que necesita el *repositorio* (2a). El repositorio se encarga de realizar consultas de bajo nivel al modelo de datos (3), el cual ejecuta las consultas sobre la base de datos (4). La *base de datos* responde (5) y el *modelo* envía la respuesta al *repositorio* (6). El *repositorio* transforma y procesa la información, enviándola al *controlador* (7a), que finalmente responde a la *vista* con la información solicitada (8a)

En caso de que la vista requiera acceso a datos en tiempo real, se comunica con el *websocket* (1b) mediante eventos. El *websocket* recibe y procesa el evento, y si necesita acceso a datos, envía la solicitud al repositorio (2b), siguiendo el mismo proceso que el controlador (3,4,5,6). Cuando el repositorio responde con la información (7b), el *websocket* envía un evento a la vista con los datos solicitados (8b)

### 3.2. Arquitectura software del sistema

La arquitectura software está compuesta por una base de datos y dos servidores: uno para el frontend y otro para el backend, como se puede observar en la figura 3.2.

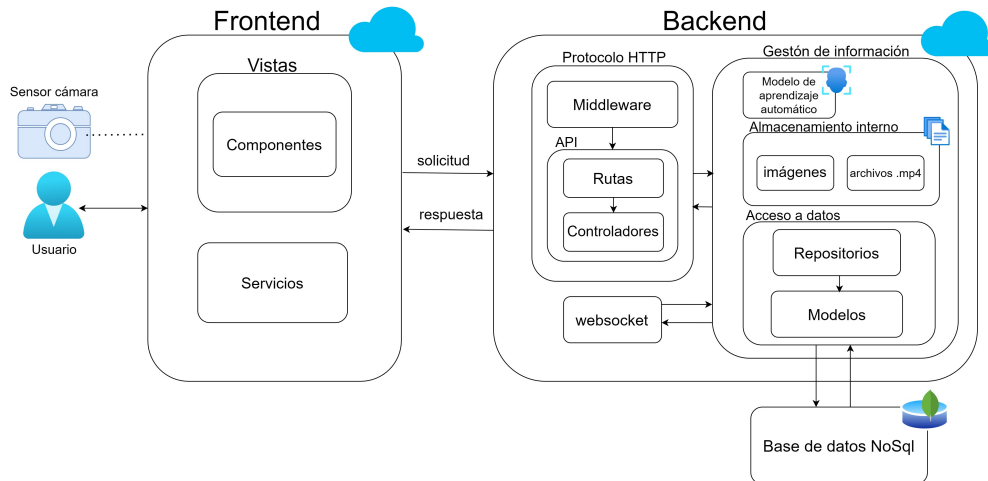


Figura 3.2: Diagrama de la arquitectura software utilizada

El servidor del frontend gestiona la web estática y proporciona la interfaz del usuario mediante las vistas compuestas por componentes modulares. Además, incluye servicios para realizar peticiones al backend mediante websockets o el protocolo HTTP.

En el backend, la recepción de información se realiza de dos formas: mediante el protocolo HTTP, con un middleware encargado de gestionar los permisos a la API y

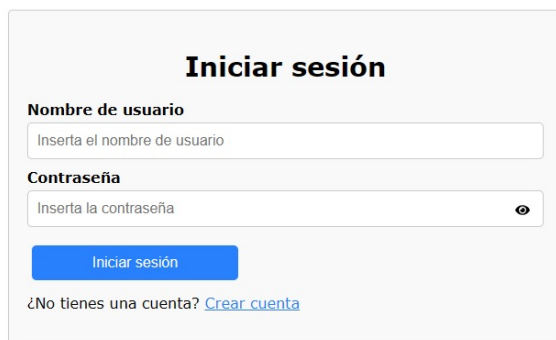
manejar errores para devolver códigos de error adecuados. La API contiene un conjunto de rutas para acceder a recursos, cada colección de la API se asocia a un controlador que maneja las peticiones, como la creación de experimentos o la obtención de la información de un experimento. O mediante websockets, utilizados para transmitir información en tiempo real, los cuales reciben la foto tomada por el usuario y el segundo en el que ha sido tomada.

El servidor del backend cuenta con un almacenamiento interno donde se guardan las fotos de los usuarios recopiladas y los archivos multimedia utilizados durante el experimento. Además, también reside el modelo de aprendizaje automático, empleado para inferir emociones basadas en las imágenes de los usuarios recibidas.

En el backend, el acceso a los datos almacenados en la base de datos se gestiona a través de repositorios. Cada repositorio está asociado a una colección específica de la base de datos y se encarga de ejecutar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) utilizando los modelos correspondientes.

### 3.3. Interfaz de usuario

A continuación se presentan las pantallas de la interfaz de la aplicación, dividiéndolas en las pantallas para configurar los experimentos, la realización de estos, las pantallas de visualización de las estadísticas y las del administrador.



**Iniciar sesión**

**Nombre de usuario**  
Inserta el nombre de usuario

**Contraseña**  
Inserta la contraseña

Iniciar sesión

¿No tienes una cuenta? [Crear cuenta](#)



**Crear una cuenta**

**Nombre de usuario**  
Nombre de usuario

**Contraseña**  
Contraseña (mínimo 8 caracteres)

**Confirmar contraseña**  
Contraseña (mínimo 8 caracteres)

Registrarse

Ya tienes una cuenta? [Iniciar sesión](#)

Figura 3.3: Pantalla de inicio de sesión.      Figura 3.4: Pantalla de registro de sesión.

La figura 3.3 contiene una captura de la pantalla que permite iniciar sesión dentro de la plataforma. Y la figura 3.4 contiene una captura de la pantalla que permite crear una cuenta de usuario. El administrador cuenta con una pantalla específica muy similar a la de inicio de sesión de un usuario, diseñada para reforzar la seguridad de la plataforma.

### 3.3.1. Pantallas para la configuración de un experimento

A continuación se presentan las pantallas que permiten a un usuario con el rol de 'experimentador' crear un experimento.



Inicio | Crear experimento | Mis experimentos | Cerrar sesión

## Configura el experimento

**Título**  
Inserta el título del experimento

**Descripción**  
Inserta la descripción del experimento (opcional)

**Contraseña**  
Inserta una contraseña para el experimento

Pretexto

Postexto

Añadir Tarea

Guardar experimento

Figura 3.5: Pantalla de configuración de un experimento.

La figura 3.5 muestra una captura de la pantalla inicial para crear un experimento. El experimentador puede seleccionar si quiere mostrar un pretexto o postexto. Si selecciona cualquiera de ellos, aparecerá un campo título y un campo descripción para que pueda rellenarlo.

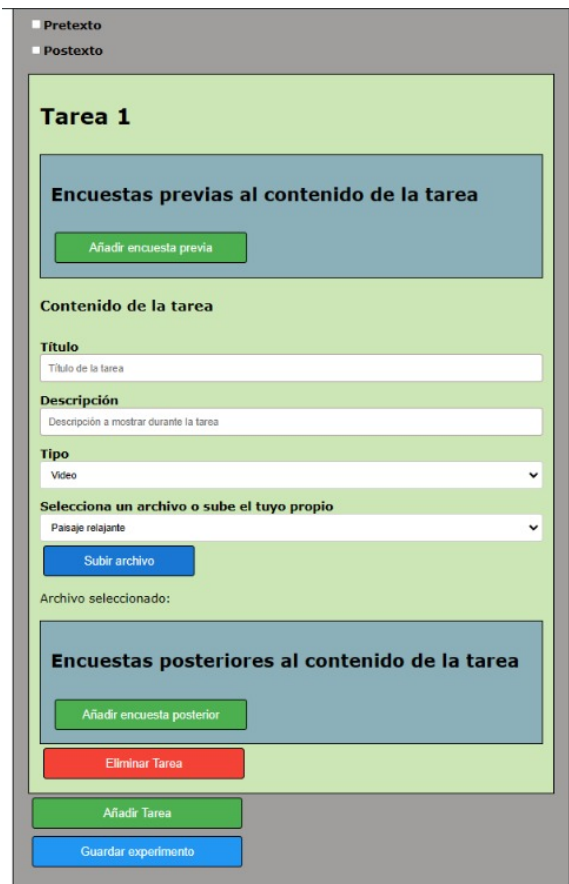


Figura 3.6: Pantalla de configuración de una tarea.



Figura 3.7: Pantalla con varias encuestas en la sección de encuestas previas.

La figura 3.6 exhibe una captura de pantalla de la configuración de una tarea. Cada tarea tiene tres bloques: encuestas previas, contenido de la tarea y encuestas posteriores. La figura 3.7 muestra la configuración de encuestas, en la sección de encuestas previas. La primera encuesta es de tipo formulario *Self-Assessment Manikin* (SAM) [9] y la segunda encuesta es de tipo pregunta, con la capacidad de añadir múltiples preguntas en la misma encuesta.

**Encuesta 1**

Pregunta

**Selecciona las preguntas a mostrar**

**Pregunta 1**

Rango numérico

Texto de la pregunta

Introduce una descripción (opcional)

**Valor mínimo**

0

Etiqueta a mostrar para el valor mínimo

**Valor máximo**

5

Etiqueta a mostrar para el valor máximo

Eliminar pregunta

Añadir pregunta

Eliminar encuesta

Figura 3.8: Pantalla de configuración de una pregunta tipo rango numérico

**Encuesta 1**

Pregunta

**Selecciona las preguntas a mostrar**

**Pregunta 1**

Elección múltiple

Texto de la pregunta

Introduce una descripción (opcional)

**Opciones**

Texto de la opción

Eliminar opción

Añadir opción

Eliminar pregunta

Añadir pregunta

Eliminar encuesta

Figura 3.9: Pantalla de configuración de una pregunta tipo selección múltiple

**Encuesta 1**

Pregunta

**Selecciona las preguntas a mostrar**

**Pregunta 1**

Si/No

Texto de la pregunta

Introduce una descripción (opcional)

Eliminar pregunta

Añadir pregunta

Eliminar encuesta

Figura 3.10: Pantalla de configuración de una pregunta tipo verdadero/falso.

En las figuras 3.8, 3.9 y 3.10 se presentan las pantallas de configuración de los

distintos tipos de preguntas: rango numérico, que permite añadir una etiqueta opcional para mostrar al lado del valor inferior y superior; selección múltiple, que permite añadir opciones; y verdadero/falso, respectivamente.

The screenshot shows a form titled "Contenido de la tarea" with a light green header. It contains several sections: "Título" with a text input field containing "Título de la tarea"; "Descripción" with a text input field containing "Descripción a mostrar durante la tarea"; "Tipo" with a dropdown menu showing "Video"; and "Selecciona un archivo o sube el tuyo propio" with a dropdown menu showing "Video de terror". Below this is a list of files: "Archivos por defecto" (Paisaje relajante) and "Mis archivos" (Video de terror). The "Video de terror" file is highlighted in blue, and the text "Archivo seleccionado: video de terror" is visible at the bottom.

The screenshot shows a modal titled "Subir archivo" with a white background and a dark border. It contains several sections: "Título" with a text input field containing "Título del archivo para luego identificarlo"; "Descripción" with a text input field containing "Descripción del archivo(opcional)"; "Tipo de archivo" with a dropdown menu showing "Video"; and "Archivo" with a button labeled "Seleccionar archivo" and the text "Ningún archivo seleccionado". At the bottom, there are two blue buttons: "Cancelar" and "Guardar".

Figura 3.11: Pantalla de selección del contenido de una tarea. Figura 3.12: Pantalla con el modal para subir un archivo.

La figura 3.11 muestra la pantalla de configuración del contenido de una tarea, permitiendo seleccionar un archivo de entre los que existen por defecto en la plataforma o aquellos que el usuario ha subido previamente. Por otro lado, la figura 3.12 muestra el modal para subir un archivo al servidor para utilizarlo como contenido.

### 3.3.2. Pantallas de realización de un experimento

A continuación se presentan las pantallas que visualiza un usuario a la hora de realizar los experimentos.



Figura 3.13: Pantalla con el listado de los experimentos disponibles.

La figura 3.13 muestra la pantalla que contiene el listado de todos los experimentos disponibles, junto con su información. Además, los usuarios tienen la opción de buscar experimentos por título o utilizar la función de ordenación según los diferentes campos.

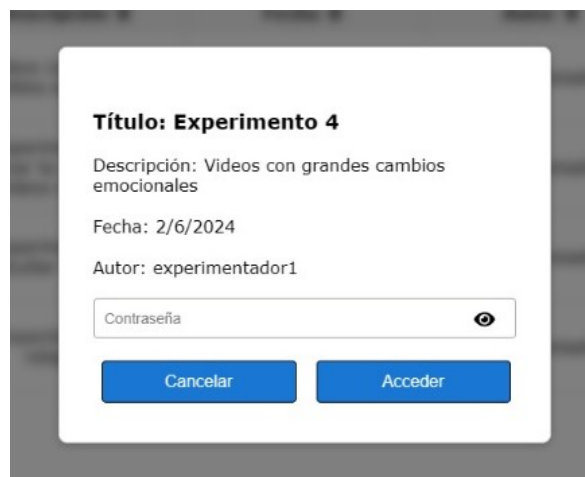


Figura 3.14: Pantalla con el modal de acceso a un experimento seleccionado.

La figura 3.14 ilustra la pantalla con el modal que muestra los detalles del experimento seleccionado y la solicitud de la contraseña para poder realizarlo.

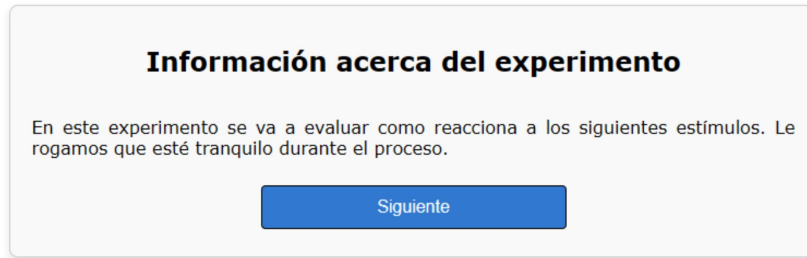


Figura 3.15: Pantalla del pretexto de un experimento.

La figura 3.15 muestra la pantalla con la información del pretexto del experimento. La pantalla del posttexto sigue el mismo formato.

### ¿Cómo se siente actualmente?

Utilice la barra deslizante para seleccionar el valor deseado. Si tiene alguna pregunta o necesita más información, consulte la esquina izquierda.

**Valencia**

Triste Contento

**Activación**

Calmado Activado

**Dominancia**

Se refiere al grado de control percibido sobre una situación. Valores bajos pueden indicar sumisión o falta de control, mientras que valores altos pueden indicar confianza y dominio.

No controlado Controlado

**Siguiente**

Figura 3.16: Pantalla del formulario SAM.

La figura 3.16 presenta la pantalla del formulario SAM. En el se permite seleccionar un valor de valencia, activación y dominancia.

The image shows a vertical sequence of three question panels, each with a light blue background and a white border. The first panel asks "¿Había visto el video previamente?" with radio buttons for "si" and "no". The second panel asks "¿Cuál es la frecuencia con la que suele ver videos graciosos?" with radio buttons for "Nunca", "Una vez a la semana", "Casi todos los días", and "Todos los días". The third panel asks "En una escala del 1 al 10, ¿que tan gracioso le ha parecido el video?" and features a horizontal slider with a blue track and a white handle. The slider is labeled "Nada" at 0, "5" in the middle, and "Mucho" at 10. Below the panels is a blue button labeled "Siguiete".

Figura 3.17: Pantalla con los tres tipos de preguntas existentes en la aplicación.

La figura 3.17 muestra la pantalla con el formato de visualización de los tres tipos de preguntas dispuestas en orden descendente: primero una pregunta de tipo verdadero/falso, seguida de una de tipo selección múltiple y, finalmente, una de rango numérico.

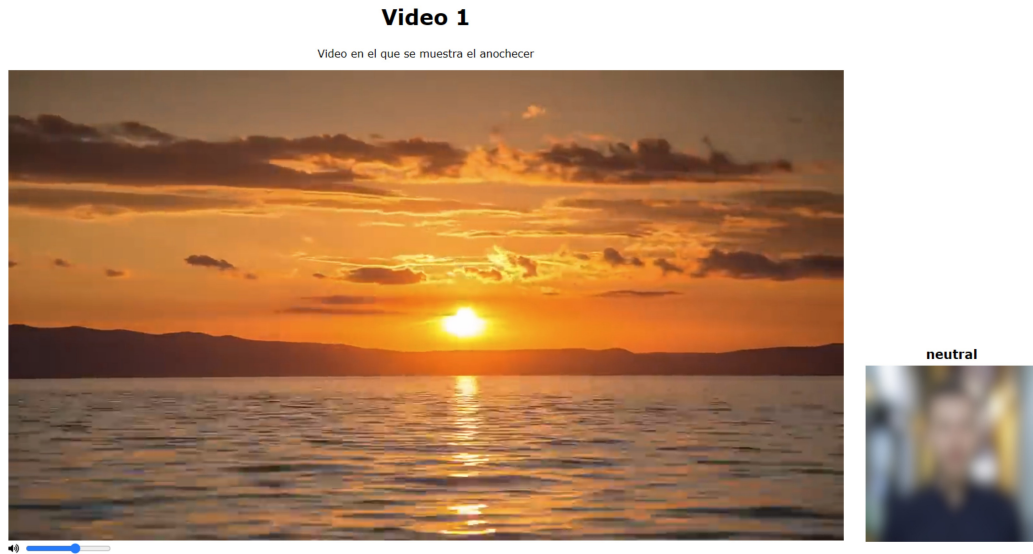


Figura 3.18: Pantalla del contenido de una tarea, la imagen del usuario y la emoción predicha.

La figura 3.18 muestra la pantalla con el contenido a mostrar, la imagen del usuario (pixelado por privacidad) y la emoción que predice el sistema.

### 3.3.3. Pantallas para la visualización de los datos recopilados

A continuación se presentan las pantallas que permiten a un experimentador visualizar la información relativa a sus experimentos.

Inicio | Crear experimento | Mis experimentos | Cerrar sesión

## Mis Experimentos

Título ↕	Descripción ↕	Fecha ↕	Número de respuestas ↕	Estadísticas
Experimento 1	Experimento de relajación	20/5/2024	5	<div style="background-color: #4a90e2; color: white; padding: 2px 10px; margin-bottom: 2px;">Globales</div> <div style="background-color: #4a90e2; color: white; padding: 2px 10px;">Individuales</div>
Experimento 4	Videos con grandes cambios emocionales	2/6/2024	0	<div style="background-color: #4a90e2; color: white; padding: 2px 10px; margin-bottom: 2px;">Globales</div> <div style="background-color: #4a90e2; color: white; padding: 2px 10px;">Individuales</div>

Figura 3.19: Pantalla con los experimentos del experimentador.

La figura 3.19 muestra al experimentador su experimentos. Se permite buscar entre sus experimentos además de ordenar por los distintos campos. En ella se observa información básica del experimento y el número de usuarios que lo han realizado. Para cada experimento, existen dos opciones: visualizar las estadísticas globales, que representa un promedio de todos los usuarios, o ver las estadísticas de un usuario específico.

Inicio | Crear experimento | Mis experimentos | Cerrar sesión

### Usuarios que han realizado el experimento

Buscar por nombre de usuario

Nombre de usuario ↕	Acciones
usuario1	Ver estadísticas del usuario
usuario2	Ver estadísticas del usuario
usuario3	Ver estadísticas del usuario
usuario4	Ver estadísticas del usuario
usuario5	Ver estadísticas del usuario

Figura 3.20: Pantalla del listado de usuarios que han realizado un experimento concreto.

La figura 3.20 muestra la pantalla que aparece tras seleccionar la opción de ver estadísticas individuales. Contiene el listado de todos los usuarios que han realizado el experimento, permitiendo seleccionar uno concreto para ver sus estadísticas.

### Emociones promedio durante el experimento

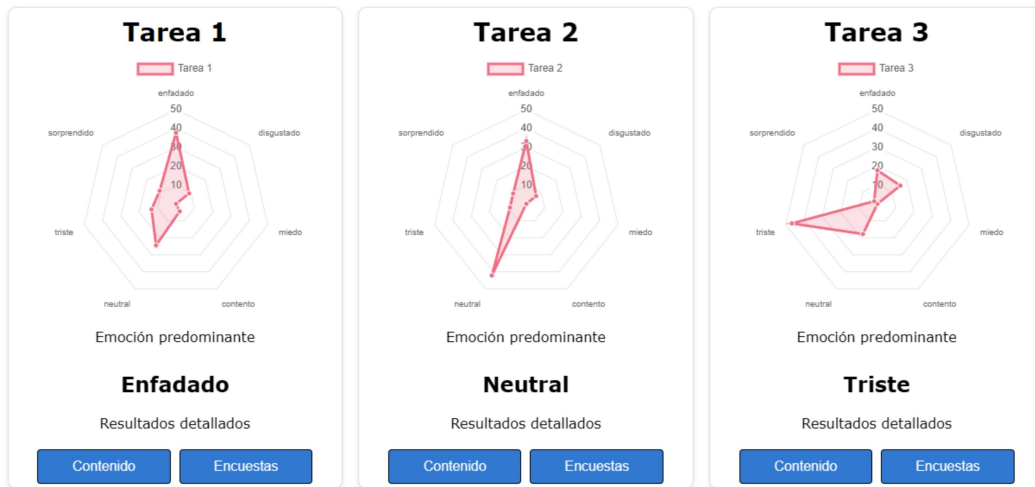


Figura 3.21: Pantalla de las emociones en las distintas tareas de un experimento.

La figura 3.21 presenta la pantalla con la información relativa a las emociones predichas en cada tarea del experimento, además de indicar la emoción predominante. Si el usuario ha seleccionado las estadísticas globales, se muestra el promedio; si es un usuario individual, se muestran solamente las del usuario seleccionado.

### Estadísticas del usuario durante la interacción con el contenido

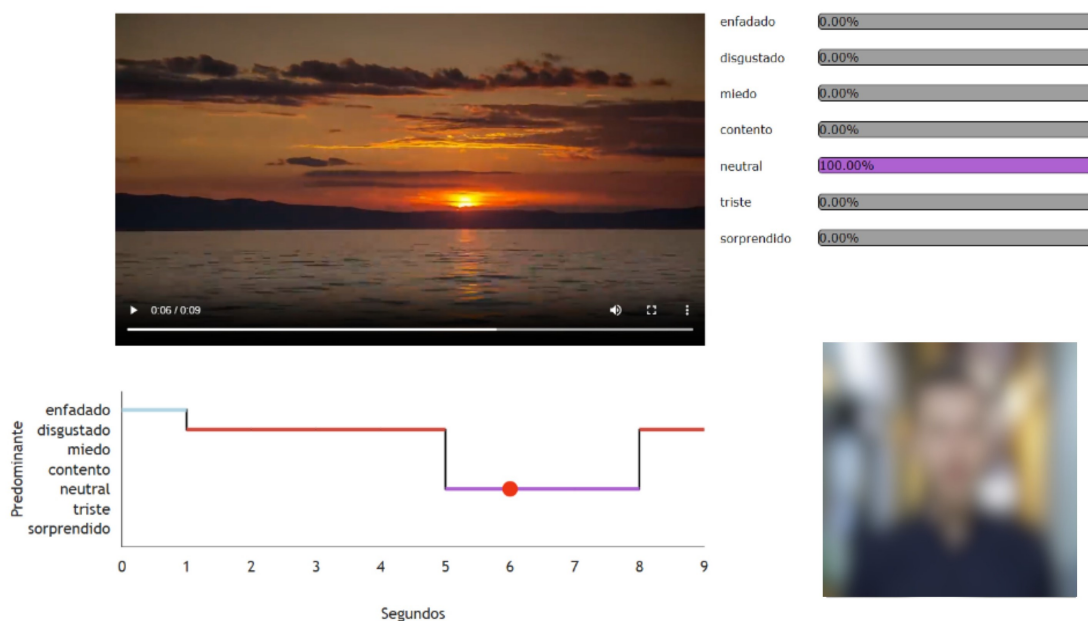


Figura 3.22: Pantalla de estadísticas de las emociones predichas para un usuario.

La figura 3.22 muestra la pantalla con las estadísticas individuales asociadas al contenido de la tarea. En la esquina superior izquierda se observa la visualización del contenido. Justo a la derecha, la gráfica con la emoción predominante en el segundo actual del vídeo. En la esquina inferior izquierda, hay una gráfica temporal con la emoción predominante en cada segundo, y justo a su derecha se encuentra la imagen del usuario capturada en el segundo actual del vídeo. Es importante destacar, que todos está sincronizado, por lo que si el experimentador avanza el vídeo se actualiza el contenido de las gráficas y la imagen a mostrar.

Si por el contrario, se han seleccionado las estadísticas globales, todos los cálculos se realizan con el promedio de los usuarios y no se muestra la imagen captura, la cual solo está disponible en el modo individual.



Figura 3.23: Pantalla de resultados de las encuestas.

La figura 3.23 muestra la pantalla con las gráficas que representan los resultados de las encuestas. El experimentador puede seleccionar la visualización de las encuestas previas o posteriores.

### 3.3.4. Pantallas del administrador del sistema

A continuación se presentan las pantallas del administrador.



Figura 3.24: Pantalla de administrador con la lista de usuarios en la aplicación.

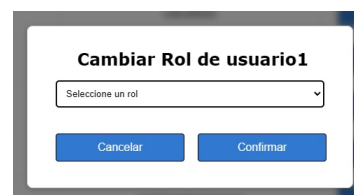


Figura 3.25: Pantalla del modal que permite cambiar el rol de un usuario.

La figura 3.24 muestra la pantalla que contiene el listado de usuarios que hay en la aplicación. Y la figura 3.25 muestra la pantalla con el modal que permite al administrador cambiar el rol del usuario seleccionado.

## 3.4. Uso de JSON Web Tokens para garantizar la seguridad

Para garantizar que un usuario acceda únicamente a los recursos que tiene que acceder, se emplean JSON Web Tokens (JWT). Estos tokens son formatos compactos y autocontenidos utilizados para la transmisión segura de información, permitiendo su verificación mediante firma digital. En la aplicación existen dos momentos en los que se generan los JWT. El primero ocurre al iniciar sesión, cuando un usuario accede al sistema, el backend responde con un JWT que incluye el identificador del usuario y la fecha de expiración del token. El segundo momento ocurre cuando un usuario decide acceder a un experimento, en caso de que ingrese la contraseña correctamente, se genera un JWT que contiene el identificador del usuario, el experimento y su fecha de caducidad. Los JWT enviados del backend al frontend se almacenan en *sessionStorage* del navegador. De esta manera, se permite realizar peticiones que requieran autorización incluyendo el token en la cabecera de la petición.

## 3.5. Contenido de la API del backend

Se ha optado por utilizar una *Application Programming Interface* (API) para abstraer el acceso a los datos. La API diseñada cuenta con distintas colecciones, dependiendo de a que se quiera obtener acceso. Para encontrar información más detallada, consultar el Anexo B: Detalles de la API del backend.

A continuación se listan las colecciones de la API junto con sus funcionalidades.

- **Administrador:** Colección diseñada para los administradores de la aplicación. Proporciona un inicio de sesión para administradores, la obtención de todos los usuarios del sistema y permitir cambiar el rol de un usuarios específico.
- **Usuario:** Colección diseñada para el acceso a la aplicación. Permite a los usuarios registrarse e iniciar sesión.
- **Experimento:** Colección diseñada para la gestión y administración de los experimentos. Permite crear nuevos experimentos, obtener todos los experimentos, obtener la información de un experimento y la generación de tokens de acceso para el experimento.
- **Archivo:** Colección diseñada para la gestión de los archivos multimedia dentro del sistema. Permite obtener las imágenes tomadas a los usuarios, los archivos multimedia y subir nuevos archivos.
- **Estadísticas:** Colección diseñada para proporcionar estadísticas. Permite obtener todos los experimentos de un experimentador concreto, usuarios que han participado en el experimento y obtener métricas detalladas relacionadas con las tareas, el contenido de las tareas y los resultados de las encuestas realizadas.
- **Resultado:** Colección diseñada para guardar los resultados asociados a un experimento. Permite almacenar los resultados de las encuestas y la completitud del experimento.

## 3.6. Diseño del acceso al modelo de inteligencia artificial

Para diseñar la interacción con el modelo de inteligencia artificial, se han valorado tres opciones:

- **Servidor separado:** Dado que el modelo está implementado en Python, se consideró la creación de un servidor dedicado que ejecute el modelo de inteligencia artificial continuamente. El servidor tendría rutas específicas para recibir imágenes y responder con la emoción predicha.
- **Integración en el cliente:** Esto implica que el backend no ejecuta el modelo, sino el ordenador del usuario que este accediendo a la plataforma. Además, es necesario convertirlo a JavaScript.
- **Integración en el backend:** Esto implica que el modelo se ejecuta en el backend del servidor, consumiendo recursos. También hay que adaptar el modelo desarrollado en Python a un formato compatible con JavaScript, el lenguaje utilizado en el backend.

La opción del servidor separado, mejora el rendimiento del backend al liberar los recursos que se utilizarían para ejecutar el modelo, el cual es costoso computacionalmente. Sin embargo, esta opción se ha descartado debido a que añade mucho tiempo de respuesta al enviar las imágenes a otro servidor y esperar la respuesta, haciendo inviable la comunicación en tiempo real.

La opción de integrarlo en el navegador del cliente también libera los recursos del backend ya que se ejecuta completamente en el ordenador del cliente. Sin embargo, esta opción se ha descartado dado que el usuario debe contar con un ordenador con buen hardware. Además, de que se rompe el esquema modular en el que el frontend pide datos al backend y el backend responde.

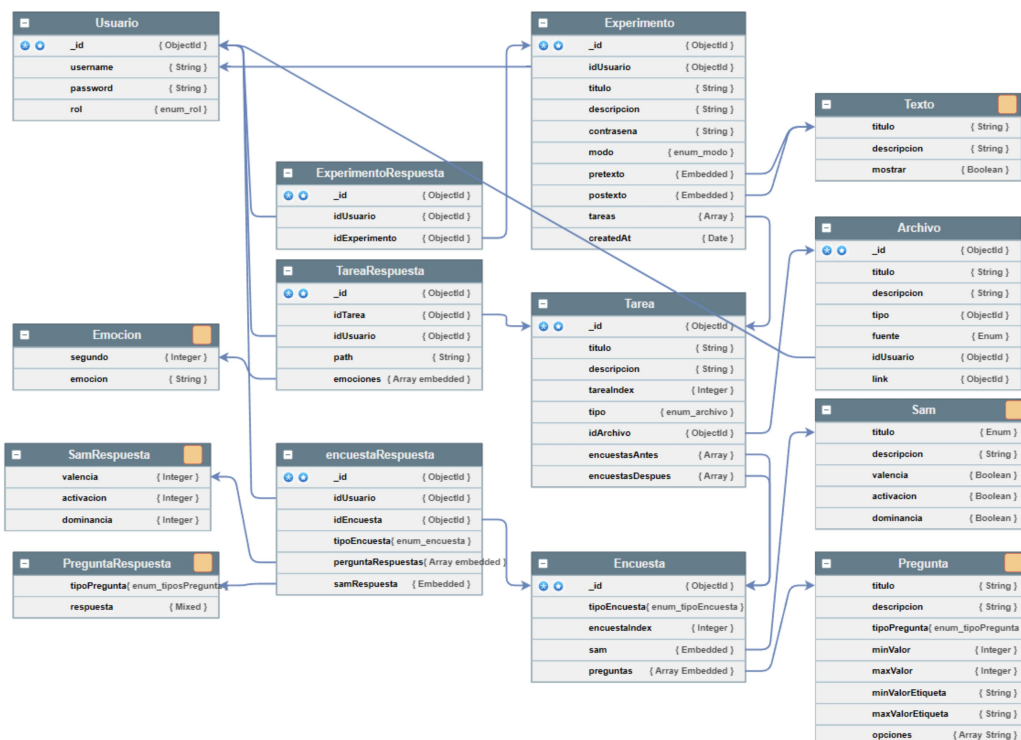
La opción elegida finalmente es la integración del modelo en el backend, se ha decidido esta opción debido a que funciona en tiempo real, evita la necesidad de que el cliente cuente con un buen ordenador y es modular. Los detalles de cómo se ha integrado se presentan en el capítulo 4.

### 3.7. Diseño de la base de datos NoSQL

La elección de una base de datos NoSQL se basa en su capacidad para manejar y almacenar información en formato JSON, el cual se utiliza tanto en el frontend como en el backend para la serialización de los datos. Dado que el formato JSON es nativo en las bases de datos NoSQL, se facilita la integración y el procesamiento de los datos. Además, las bases de datos NoSQL ofrecen una mayor escalabilidad y capacidad de procesamiento de datos en comparación con las bases de datos SQL. El gestor de bases de datos utilizado es MongoDB Atlas.

Antes de abordar el diseño de la base de datos, es importante realizar una comparación con las bases de datos SQL para aclarar algunos conceptos clave de las bases de datos NoSQL. En NoSQL, la unidad básica de datos es el documento, que es comparable a la fila en SQL. En NoSQL los datos se organizan en colecciones de documentos relacionados sin requerir un esquema rígido. A diferencia de las bases de datos SQL, donde la información se almacena en tablas interrelacionadas que siguen un esquema fijo.

La figura 3.26 muestra el diagrama que contiene todas las colecciones de la base de datos NoSQL, junto con la estructura de los documentos que almacenan.



Nota: El cuadrado naranja significa que no son colecciones, sino documentos.

Figura 3.26: Diagrama del diseño de la base de datos NoSQL.

A continuación, se explican las colecciones que contiene la base de datos.

- **Usuario:** almacena la información relacionada con el perfil del usuario, incluyendo su nombre de usuario, contraseña y rol, que se utiliza para gestionar los niveles de acceso dentro de la aplicación.
- **Experimento:** almacena la información de cada experimento. Incluye un campo “contrasena” para restringir el acceso, un campo “idUsuario” que referencia al documento correspondiente en la colección “Usuario” para saber quien lo ha creado y un campo “createdAt” para saber la fecha de creación. Además, contiene los campos “pretexto” y “posttexto”, que embeben el documento “Texto”; es decir, estos datos se almacenan directamente en el documento del experimento a diferencia de lo que sucede con “idUsuario”. Por último, para almacenar todas las tareas del experimento se tiene un vector de referencias a la colección *Tarea*.
- **Archivo:** almacena la información de los archivos del servidor.
- **Tarea:** almacena todas las tareas. Cada documento de la colección incluye una referencia al documento de la colección *Archivo* para saber el archivo asociado a la tarea. Además, las encuestas previas y posteriores se almacenan en dos vectores que referencia a los documentos de la colección *Encuesta*.
- **Encuesta:** contiene la información las encuestas. Incluye un campo para especificar su tipo. Si es de tipo “pregunta”, almacena documentos *Pregunta* embebidos en un vector. Si es de tipo “sam”, almacena el documento *Sam* embebido.
- **ExperimentoRespuesta:** almacena el identificador de usuario y experimento, que sirve para saber cuando un usuario ha realizado un experimento.
- **TareaRespuesta:** almacena las emociones predichas del usuario por cada segundo, junto con la ruta a la carpeta que contiene las imágenes del usuario.
- **encuestaRespuesta:** contiene una referencia al usuario que la ha respondido, almacenando su respuesta.



# Capítulo 4

## Implementación del sistema

En este capítulo se describen en detalle los componentes software más relevantes que se han implementado y el proceso para realizar diversas funcionalidades. Estos se dividen en componentes utilizados en el frontend de la aplicación, en el backend y detalle de algunas funcionalidades de implementación.

### 4.1. Componentes utilizados en el frontend de la aplicación

El frontend corresponde a la interfaz de usuario, abarcando desde la realización de acciones de los usuarios hasta la visualización de los datos. A continuación, se describe cómo el frontend permite configurar un experimento, mostrar un experimento completo paso a paso y visualizar los datos recopilados durante el experimento.

#### 4.1.1. Componentes de configuración de un experimento

Para permitir la creación de un experimento configurable, se ha implementado una estructura JSON que contiene toda la información necesaria. Esta estructura es generada por el frontend y enviada al backend para su almacenamiento persistente en la base de datos. Dado que el JSON incluye numerosos campos configurables, se ha optado por un diseño modular donde cada componente gestiona sus propios parámetros, facilitando así futuras modificaciones o extensiones.

La estructura general del experimento se presenta en la figura 4.1. El componente *Experimento*, se encarga de rellenar el título, descripción y contraseña del experimento. Así como la opción a mostrar un pretexto y un posttexto con sus respectivos títulos y descripciones. Además cada experimento puede tener  $n$  tareas.

```

{
  "titulo": "Experimento de prueba",
  "descripcion": "En este experimento vamos a evaluar como se siente",
  "contrasena": "password12345",
  "pretexto": {
    "titulo": "Instrucciones generales",
    "descripcion": "Mantengase relajado durante todo el experimento",
    "mostrar": true
  },
  "postexto": {
    "titulo": "",
    "descripcion": "",
    "mostrar": false
  },
  "tareas": [
    {
      "tareaIndex": 0,
      "idArchivo": "667060cf1fba433472b54fcd",
      "titulo": "Video relajante",
      "descripcion": "El anochecer",
      "tipo": "VIDEO",
      "encuestasAntes": [
        {
          "encuestaIndex": 0,
          "tipoEncuesta": "SAM",
          "sam": {
            "valence": true,
            "arousal": true,
            "dominance": true,
            "titulo": "Pre-evaluación de sus emociones",
            "descripcion": ""
          }
        }
      ],
      "encuestasDespues": []
    }
  ]
}

```

Figura 4.1: Estructura JSON para la configuración del experimento

El componente *Tarea* (figura 4.2), guarda la información asociada a cada tarea, incluyendo un título y una descripción. También se especifica el tipo de archivo, que puede ser vídeo o música, junto con el identificador del archivo seleccionado. Cada tarea puede contener múltiples encuestas previas, que se muestran antes de la reproducción del archivo, y varias encuestas posteriores, que se presentan después del archivo.

```

{
  "tareaIndex": 0,
  "idArchivo": "667060cf1fba433472b54fcd",
  "titulo": "Video relajante",
  "descripcion": "El anochecer",
  "tipo": "VIDEO",
  "encuestasAntes": [],
  "encuestasDespues": []
}

```

Figura 4.2: Estructura JSON de la tarea

El componente *Encuesta* se encarga de indicar el tipo de encuesta. Los tipos

disponibles son 'SAM' 4.3 y 'Pregunta' 4.4. Además cada encuesta puede tener múltiples preguntas o un formulario de tipo 'SAM'.

```
{
  "encuestaIndex": 0,
  "tipoEncuesta": "SAM",
  "sam": {
    "valence": true,
    "arousal": true,
    "dominance": true,
    "titulo": "fdas",
    "descripcion": ""
  }
}
```

Figura 4.3: JSON de encuesta tipo SAM

```
{
  "encuestaIndex": 0,
  "tipoEncuesta": "PREGUNTA",
  "preguntas": []
}
```

Figura 4.4: JSON de encuesta tipo pregunta

El componente *Pregunta*, permite configurar tres tipos de preguntas, mostradas en las figuras 4.5, 4.6 y 4.7. Todas tienen los campos titulo y descripción. Los tipos de preguntas disponibles son: Sí/No, elección múltiple donde se pueden definir las opciones, y rango numérico que permite especificar valores del 0 al 10, con la opción de añadir etiquetas a mostrar para los valores mínimo y máximo.

```
{
  "textoPregunta": "Enunciado 1",
  "tipoPregunta": "SI_NO",
  "minValor": 0,
  "maxValor": 10,
  "descripcion": "Descripción 1",
  "minValorEtiqueta": "",
  "maxValorEtiqueta": "",
  "opciones": []
}
```

Figura 4.5: JSON pregunta de tipo sí/no

```
{
  "textoPregunta": "Enunciado 3",
  "tipoPregunta": "RANGO_NUMERICO",
  "minValor": 2,
  "maxValor": 9,
  "descripcion": "Descripción rango",
  "minValorEtiqueta": "Poco",
  "maxValorEtiqueta": "Mucho",
  "opciones": []
}
```

Figura 4.6: JSON pregunta de tipo rango numérico

```
{
  "textoPregunta": "Enunciado 2",
  "tipoPregunta": "ELECCION_MULTIPLE",
  "minValor": 0,
  "maxValor": 5,
  "descripcion": "Otra descripción",
  "minValorEtiqueta": "",
  "maxValorEtiqueta": "",
  "opciones": [
    "Opción 1",
    "Opción 2",
    "Opción 3",
    "Opción 4",
    "Opción 5"
  ]
}
```

Figura 4.7: JSON pregunta de tipo elección múltiple

El componente que permite configurar el formulario de tipo SAM, permite configurar los campos valencia, activación y dominancia. Se tiene un campo con verdadero o falso para indicar cuales mostrar y cuáles ocultar. Y otro campo para el título y descripción a mostrar junto al formulario.

## 4.1.2. Componente de gestión del ciclo de vida de un experimento

Para gestionar el ciclo de vida de un experimento se ha implementado una máquina de estados en el frontend de la aplicación utilizando el *dispatcher* de React. Este enfoque permite el envío de eventos y acciones para gestionar los cambios de estado. En la implementación, se guarda una variable que contiene el estado actual, el índice de la tarea en curso, los índices de las encuestas previas y posteriores de la tarea actual, así como el número total de tareas.

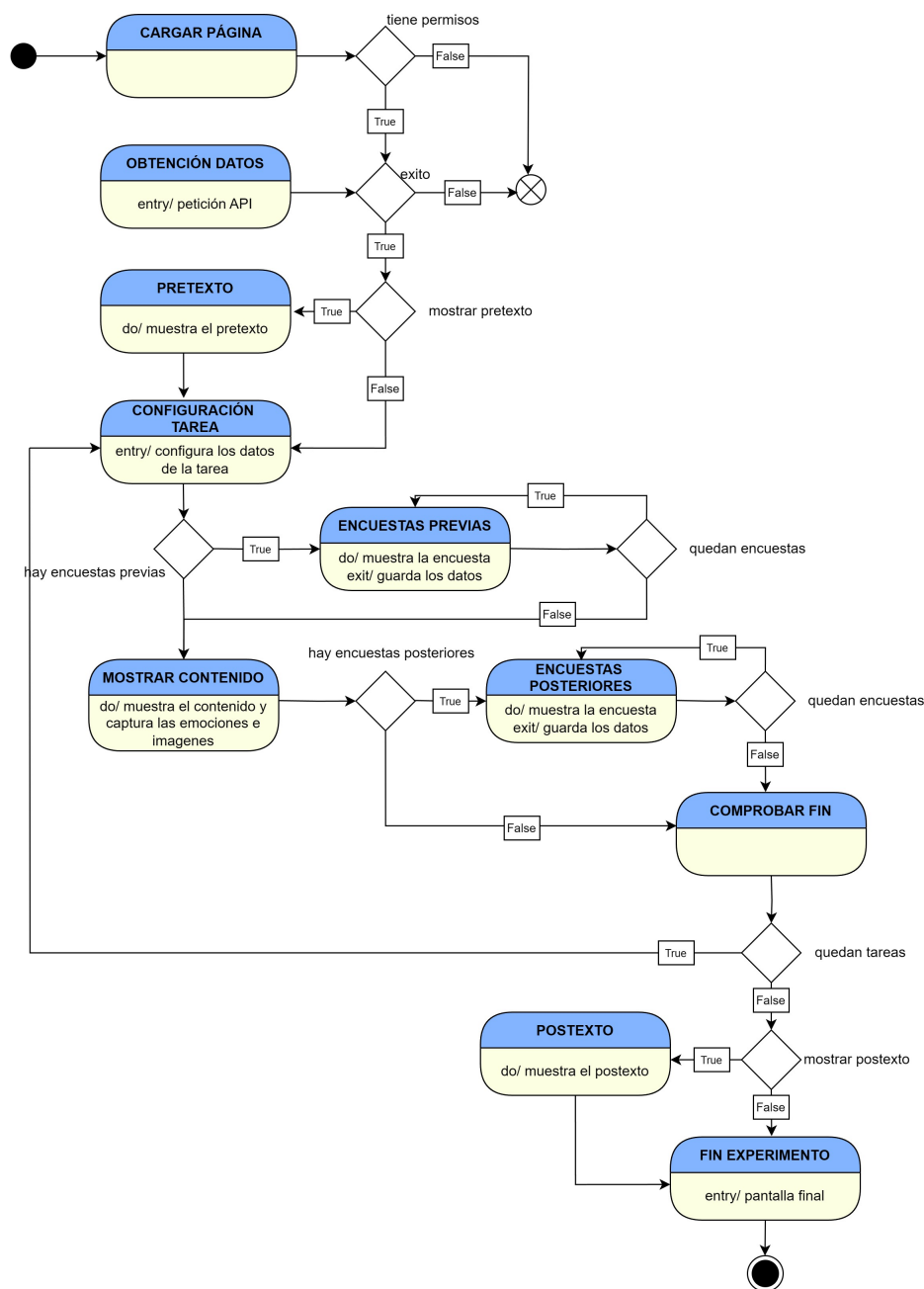


Figura 4.8: Diagrama de estados que representa el ciclo de vida de un experimento.

A continuación se explica el ciclo de vida de un experimento mostrado en la figura 4.8.

1. Se carga la página y se verifica si el usuario tiene permisos para acceder al experimento. En caso de que no los tenga, finaliza la interacción y se muestra un modal indicándole que no tiene permisos.
2. Se envía una petición al backend incluyendo el token de usuario y el identificador del experimento. En caso de que no haya éxito con la petición, se finaliza la interacción.
3. Una vez se han obtenido todos los datos del experimento, si el experimento está configurado para mostrar el pretexto, se continua al estado PRETEXTO; de lo contrario, se avanza al estado CONFIGURACIÓN TAREA.
4. En el estado PRETEXTO, se muestra el título y descripción del pretexto. Cuando el usuario presiona el botón de *siguiente*, la transición se realiza hacia el estado CONFIGURACIÓN TAREA.
5. En el estado CONFIGURACIÓN TAREA, se configuran los parámetros de estado que incluyen el índice de la tarea actual, así como el número de encuestas previas y posteriores asociadas a la tarea. Si la tarea tiene encuestas previas, el siguiente estado es ENCUESTAS PREVIAS; de lo contrario, se avanza a MOSTRAR CONTENIDO
6. En el estado ENCUESTAS PREVIAS, se muestra la información de la encuesta correspondiente según el índice almacenado. Una vez que el usuario responde a la encuesta, los resultados se envían al backend para su almacenamiento. Si esta era la última encuesta previa, se avanza hacia MOSTRAR CONTENIDO; de lo contrario, se repite el proceso incrementando el índice de encuesta previa .
7. En el estado MOSTRAR CONTENIDO, se establece una conexión mediante websockets con el backend. Está conexión permite enviar desde el frontend la imagen del usuario al backend para su procesamiento, almacenamiento y posterior devolución de la emoción predicha al frontend. Una vez finalizada la interacción con el contenido multimedia, se cierra el websocket y se avanza a ENCUESTAS POSTERIORES en caso de que haya; de lo contrario, se procede a COMPROBAR FIN.

8. En el estado ENCUESTAS POSTERIORES, se lleva a cabo el mismo proceso que en ENCUESTAS PREVIAS, pero ahora con las encuestas posteriores en lugar de las previas.
9. En el estado COMPROBAR FIN, se verifica si la tarea actual es la última del experimento. Si no lo es, se incrementa el índice de la tarea y se transiciona al estado CONFIGURACIÓN TAREA. Si es la última tarea y hay posttexto a mostrar, se continúa al estado POSTEXTO; en caso de no haber posttexto se avanza a FIN EXPERIMENTO.
10. En el estado POSTEXTO, se muestra el título y descripción del posttexto. Cuando el usuario presiona el botón de *siguiente*, la transición se realiza hacia el estado FIN EXPERIMENTO.
11. En el estado FIN EXPERIMENTO, se muestra una pantalla final y envía una solicitud a la API del backend para indicar que el usuario ha completado el experimento.

### 4.1.3. Componentes de visualización de los datos

La visualización de las estadísticas se ha realizado utilizando las librerías *Victory* y *Chart.js*. Para la visualización de todos los datos se han empleado seis gráficas distintas, que han sido creadas como componentes para permitir su reutilización a lo largo de la aplicación. A continuación, se presentan las principales gráficas: la gráfica que visualiza las emociones de forma global en una tarea se muestra en la Figura 4.9, la gráfica que visualiza la emoción predominante a lo largo del tiempo se encuentra en la Figura 4.11 y la gráfica que muestra la emoción predominante en un momento concreto se presenta en la Figura 4.10. Para visualizar todos los tipos de gráficas, consultar el Anexo C: Tipos de gráficas disponibles en la plataforma.

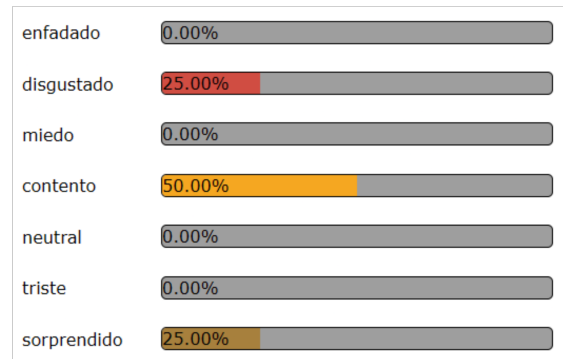


Figura 4.9: Gráfica con las emociones en una tarea

Figura 4.10: Gráfica con las emociones en un instante de tiempo concreto

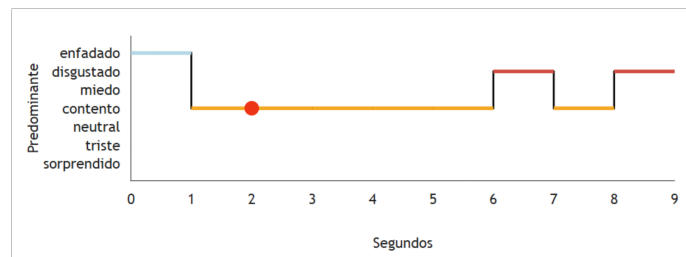


Figura 4.11: Gráfica con la emoción predominante en cada instante de tiempo

El formato en el que las gráficas reciben la información se observa en la figura 4.12. Se utilizan dos vectores: uno para las etiquetas y otro para los datos. Al acceder a ambos vectores mediante el mismo índice, se puede determinar cuál es la etiqueta correspondiente a cada valor.

```

{
  "etiquetas": [
    0,
    1,
    2,
    3,
    4,
    5,
  ],
  "emociones": [
    "contento",
    "contento",
    "contento",
    "sorprendido",
    "contento",
    "contento"
  ]
}

```

Figura 4.12: JSON con los datos a pasar a las gráficas.

## 4.2. Componentes utilizados en el backend de la aplicación

El backend de la aplicación se encarga de procesar las solicitudes enviadas desde el frontend y de generar respuestas apropiadas para cada interacción con el usuario. Aquí es donde reside la lógica de negocio del sistema.

### 4.2.1. Componentes Controladores

Debido a que cada colección de la API tiene su propio controlador, se detalla uno específico: el controlador *Experimento*. Cada controlador tiene diversas funciones las cuales cada una se corresponden a una ruta de una colección de la API. En el caso del controlador *Experimento*, las funciones son las siguientes: *crearExperimento*, *getExperimentos*, *accesoExperimento* y *getExperimentoInfo*. De entre estas funciones se va a explicar el comportamiento de la función *crearExperimento*. Esta tiene asociada una ruta invocada mediante una petición POST, por lo que recibe los datos en el cuerpo de la solicitud. Una vez se reciben los datos, se tiene que validar que sean correctos, para ello se comprueba que contiene todos los campos necesarios; de no ser así, se devuelve un error indicando que no se puede procesar. Una vez validados los campos, se procede a invocar la lógica de negocio. En este caso, se recibe en la información del experimento un vector con toda la información de las tareas a crear en formato JSON (previamente comentado en 4.1.1), por lo que se itera sobre este vector. Dado que cada tarea incluye encuestas previas y posteriores, primero se crean las encuestas correspondientes llamando a la función *crearEncuestas* del repositorio de *Encuesta*. Una vez creadas las encuestas, conocemos el identificador asociado a cada una, por lo que se pasan junto a la información de la tarea a la función *crearTarea* del repositorio de *Tarea*. Una vez creadas todas las tareas, se tiene los identificadores de cada una de ellas, por lo que junto a la información recibida del experimento se llama a la función *crearExperimento* del repositorio *Experimento*. Si todo el proceso sale con éxito, se devuelve un código de respuesta 200 para indicar que el experimento se ha creado exitosamente.

### 4.2.2. Componentes Repositorios

Como hay un repositorio por cada colección en la base de datos, se describe uno específico. En línea con el controlador previamente seleccionado, se ha optado por el repositorio *Experimento*. Este repositorio ofrece todas las funciones necesarias

para realizar cualquier operación necesaria por la aplicación con la colección *Experimento* de la base de datos, entre las cuales se encuentran: *crearExperimento*, *getExperimentos*, *getExperimentoByTarea*, *getExperimentoInfo*, *getMisExperimentos*, *getExperimentoTareas* y *comprobarAcceso*. A continuación, se explica la función *crearExperimento*. Esta función inicia validando los datos recibidos para la creación del experimento. Si la validación es exitosa, continúa con el proceso; en caso contrario responde con un error. Luego, se emplea el algoritmo *SHA256* para cifrar la contraseña y almacenarla cifrada en la base de datos. Después, se invoca al modelo *Experimento* para llevar a cabo la operación de creación del experimento. Si todo se realiza correctamente, la función devuelve el objeto del experimento recién creado, el cual contiene toda la información asociada al mismo

### 4.2.3. Componentes Middleware

La aplicación cuenta con middlewares encargados de interceptar la petición antes de llegar a la ruta específica. En la aplicación se cuenta con los siguientes middleware:

- **Middleware de control de accesos:** Se encarga de gestionar la restricción de acceso basado en el rol del usuario. Por ejemplo, en la aplicación, para restringir el acceso a una ruta específica para que sea solamente accesible por el administrador, se tiene que añadir a la ruta *verificarUsuario* (*["ADMIN"]*).
- **Middleware de validación de archivos:** Configurado con la biblioteca *multer*, este middleware se encarga de rechazar las solicitudes de usuarios que intentan subir archivos que exceden el límite máximo establecido o no cumplen con el formato requerido.
- **Middleware de gestión de errores:** Se encarga de manejar el flujo de errores en la aplicación. Está configurado como el último middleware en el flujo de la petición, asegurándose de que todas las rutas lo utilicen para capturar cualquier error que pueda surgir. Responde con un código de estado 404 si el error es generado por una acción del usuario, o 500 si el error es interno del servidor.

### 4.2.4. Componente websocket

Cuando el usuario entra en una tarea el frontend envía un evento de conexión al backend, mediante websockets. El evento contiene el identificador de la tarea y el JWT del usuario para validar el acceso. Si el acceso es autorizado, se genera una carpeta en el servidor, cuya ruta es la concatenación del identificador del experimento,

el identificador de la tarea y el identificador del usuario. Cuando el usuario decide comenzar la reproducción de la multimedia, el frontend envía al servidor un evento llamado *foto*. Este evento contiene el segundo en el que se toma la foto y la imagen de la webcam en formato base64, permitiendo así la transmisión. Cuando el backend recibe el evento deserializa la imagen y la guarda en formato JPG en la ruta previamente indicada, utilizando el segundo como nombre del archivo. Luego, la imagen se procesa y se pasa al clasificador de emociones para predecir la emoción correspondiente. El backend envía al frontend un evento llamado *info* que incluye el segundo y la emoción predicha.

### 4.3. Gestión de los archivos en el servidor

En esta sección se detalla como se almacenan y transmiten los distintos archivos almacenados en el servidor del backend. El servidor almacena dos tipos de archivos: imágenes de usuarios y archivos multimedia.

#### 4.3.1. Imágenes de usuarios

- **Recepción:** Las imágenes que se reciben a través de WebSockets están en formato base64. Por lo que primero se deserializan y convierten al formato JPG para poder guardarlas en la ruta compuesta por la concatenación del identificador del experimento, tarea y usuario. El nombre de la imagen se corresponde al segundo en el que se tomo la foto, información que se recibe junto a la imagen. En la figura 4.13 se muestra la estructura de carpetas. Al almacenarse en una carpeta cuyo nombre es el identificador del experimento, únicamente el creador del experimento puede acceder.
- **Acceso:** Para que el usuario sea capaz de solicitar la imagen de un usuario, se ha creado una URL que permite manejar peticiones GET. La URL está compuesta por la ruta exacta de la imagen.



Figura 4.13: Directorio con las imágenes almacenadas de un usuario

### 4.3.2. Archivos multimedia

- **Recepción:** Para permitir que el frontend suba archivos al backend del servidor, se ha creado una ruta que cuenta con un middleware para evitar su uso indebido. El middleware solo permite archivos con extensión MP4 y un tamaño máximo de 10MB. Esta medida se ha implementado debido a las limitaciones de espacio del servidor. Los archivos que pasan el filtro se almacenan en el directorio *uploads*, cuyo nombre es el segundo UNIX en que se guarda seguido de una cadena de números aleatorios.
- **Acceso:** Para permitir que el frontend tenga acceso a los archivos multimedia y se puedan reproducir, hay habilitada una URL que recibe como parámetro el nombre del archivo y esté responde con el archivo solicitado.

## 4.4. Integración del modelo

Dado que el modelo se ha desarrollado utilizando el lenguaje de programación Python y la aplicación web del backend está escrita en JavaScript, es necesario realizar una conversión del formato *pt* que tiene el modelo entrenado a *onnx* que es el que se puede ejecutar en JavaScript. Para ello hay que comprobar que en ambos formatos el modelo tenga las mismas entradas y salidas, además de que no todo lo que está en Python existe en JavaScript, por lo que requiere de adaptar el modelo. Una vez se ha adaptado correctamente el modelo, la librería *onnxruntime-node* permite la ejecución del modelo *onnx* en el servidor. Esta librería permite crear una sesión que simula un entorno virtual para la ejecución del modelo. Una vez la sesión iniciada, el modelo puede recibir entradas, las cuales hay que preprocesar de forma similar a la utilizada durante el entrenamiento del modelo. Una vez se tiene la imagen preprocesada, se tiene que convertir a un tensor que es lo que acepta el modelo. El modelo realiza los cálculos y devuelve una salida que también es un tensor, el cual hay que procesar para obtener la emoción.

## 4.5. Despliegue de los servidores en Azure

Para el despliegue de la aplicación, se han utilizado dos servidores en la nube: uno para el frontend y otro para el backend. Este enfoque fue posible gracias a un diseño modular que separa la interfaz con la lógica de negocio.

El frontend ha sido desplegado utilizando Azure Static Web Apps, que permite

desplegar una web estática. Este servicio obtiene el código del repositorio de GitHub y se ha automatizado el despliegue utilizando un fichero de configuración que contiene los flujos de trabajo para que utilizando GitHub Actions se consiga desplegar.

El backend ha sido desplegado utilizando Azure App Service, ya que éste permite procesamiento del lado del servidor, a diferencia del servicio utilizado en el frontend. Esto permite al backend ejecutar la lógica de negocio, acceder a datos y responder a las peticiones. Su despliegue también se ha automatizado utilizando GitHub Actions, en el que se ha configurado otro flujo de trabajo.

# Capítulo 5

## Modelo de reconocimiento de emociones

En este capítulo se detalla el desarrollo del modelo de aprendizaje automático utilizado para el reconocimiento de emociones integrado en la plataforma. Adicionalmente, para no sobrecargar la memoria principal, se encuentran los distintos modelos que se han probado en el Anexo D: Modelos y técnicas descartadas para el reconocimiento emocional.

### 5.1. Descripción del proceso

La creación del modelo de reconocimiento que se propone como solución, involucra una serie de procesos que se muestran en la figura 5.1. Esto inicia con la selección de la base de datos a utilizar. Después de elegir los datos a utilizar, se realiza un proceso de limpieza y filtrado de los datos, que continúa con la extracción de características. Luego, el conjunto de características y sus respectivas etiquetas sirven para el entrenamiento y validación del modelo. Cuando se tiene un modelo entrenado se procede a evaluar su rendimiento para la selección o descarte del modelo.

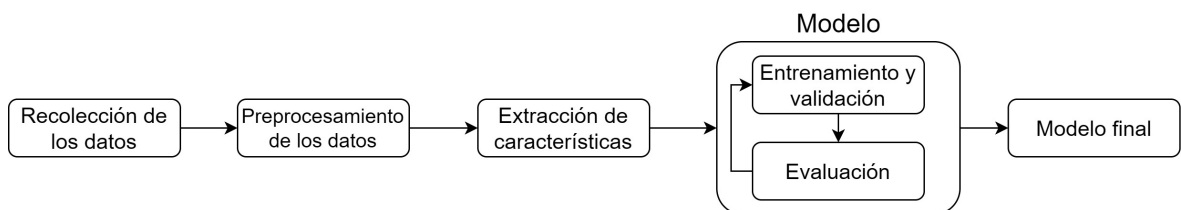


Figura 5.1: Proceso seguido para la realización del modelo de inteligencia artificial.

## 5.2. Selección de los datos a utilizar

En la figura 5.1 se observa que el primer paso es la recolección de los datos. Dado que el objetivo es desarrollar un modelo capaz de reconocer emociones como enfado, disgusto, miedo, alegría, tristeza, sorpresa y neutral exclusivamente a partir de características faciales. Inicialmente se optó por la base de datos FER2013 [10], pero tras varias pruebas no se consiguió resultados satisfactorios, debido a que los datos habían sido obtenidos de forma automáticamente mediante la API de Google imágenes. Tras esto se exploró otras soluciones y se descubrió que Microsoft había mejorado esta base de datos mediante la colaboración de personas que etiquetaban manualmente las imágenes, esta base de datos es FER+ [11]. Esta es la base de datos que se ha utilizado para el desarrollo del modelo. La base de datos contiene una emoción más que las que hay en FER2013, desprecio. Sin embargo, esta emoción se ha descartado ya que no entra dentro de los objetivos especificados anteriormente.

Debido a que las etiquetas de las imágenes han sido validadas por personas, éstas son mucho más precisas que en la base de datos original, lo que permite alcanzar mejores resultados por la calidad de los datos. La base de datos cuenta con más de 32000 imágenes de 48x48 píxeles en escala de grises. Las imágenes están separadas en dos conjuntos de datos: uno para entrenamiento y validación; así como para test. En la figura 5.2 se observa una muestra que incluye todas las emociones presentes en la base de datos.



Figura 5.2: Muestra de imágenes de la base de datos por emoción.

## 5.3. Preprocesamiento de los datos

Esta sección se corresponde con la segunda etapa del proceso especificado en la figura 5.1. Se explica la limpieza que se aplica a la base de datos, la normalizaciones aplicadas y el balanceo de los datos.

### 5.3.1. Limpieza y normalización de los datos

El primer paso consiste en filtrar las imágenes de la base de datos, ya que muchas de ellas presentan partes de la cara tapadas o están orientadas muy de perfil, lo que no permite identificar la cara. En la figura 5.3 se observan ejemplos de imágenes descartadas. Para realizar este proceso de forma automática, se ha utilizado la librería *mediapipe* utilizando el modulo de detección facial y la extracción de puntos de referencia faciales. Mediante la detección facial, se descartan las imágenes que no contienen caras o son muy difíciles de reconocer, mientras que con los puntos faciales se ha implementado una función para determinar si la cara está de frente o de perfil, descartándola en este último caso. Esto se ha logrado calculando la distancia entre el punto facial que representa el centro de los ojos izquierdo y derecho, así como la distancia entre los puntos faciales de las comisuras izquierda y derecha de la boca.



Figura 5.3: Ejemplos de imágenes que han sido descartadas

Una vez descartadas las imágenes que no cumplen con lo mencionado anteriormente, se ha mejorado la homogeneidad de la base de datos al eliminar el ruido causado por estas imágenes, debido a que ahora no hay tanta variabilidad entre las imágenes. No obstante, aún es posible mejorar la calidad de la base de datos mediante la normalización de todas las caras, asegurando que todas las imágenes tengan una orientación muy similar y estén centradas. Para lograr esto, se aplican transformaciones a los puntos faciales extraídos. Inicialmente, se calcula el ángulo entre el ojo izquierdo y derecho, el cual se utiliza para rotar la imagen mediante una matriz de rotación. Una vez rotada la imagen, los puntos faciales ya no se corresponden, por lo que también se tiene que realizar la misma rotación sobre ellos. Una vez se han ajustado los puntos faciales con la imagen rotada, se procede a definir la región de interés de la imagen seleccionando una caja delimitadora que abarca las dimensiones del contorno de los puntos faciales detectados.



Figura 5.4: Comparación entre imágenes sin normalizar y normalizadas.

### 5.3.2. Balancear los datos

El balanceo de los datos solamente se ha aplicado al conjunto de entrenamiento. Esto es debido a que si se realiza también para la validación y test, se pueden obtener resultados que aparentemente son positivos pero en realidad son falsos resultados, ya que el modelo se ha aprendido los datos en lugar de ser capaz de generalizar hacia datos que nunca ha visto.

Debido a que la base de datos no cuenta con la misma cantidad de imágenes por cada emoción, ya que la emoción 'contento' cuenta con más de 7000 imágenes mientras que 'disgusto' menos de 500. Esta desigualdad puede llevar al modelo a sesgar sus predicciones hacia la clase mayoritaria, resultando en un rendimiento deficiente al predecir la clase minoritaria. Es decir, en este caso existe una mayor probabilidad de que el modelo clasifique la emoción de disgusto erróneamente como contento.

Para abordar este problema, se han probado dos técnicas de aumento de datos:

1. La primera técnica consiste en la generación de nuevos datos a partir de los datos existentes mediante técnicas de procesamiento de imágenes. Esto incluye: invertir la imagen horizontalmente, aplicar una corrección gamma, rotar aleatoriamente imagen en un rango de  $[-15, 15]$  grados y aplicar un filtro gaussiano.
2. La segunda técnica es Synthetic Minority Over-sampling Technique (SMOTE), que permite generar imágenes sintéticas de las clases minoritarias. Esto se consigue realizando una interpolación de las características de la imagen seleccionada con la sus vecinos más cercanos de esa misma clase.

La técnica seleccionada para la realización del modelo final es SMOTE, la cual ha demostrado mejores resultados que la otra técnica. Esto se debe a que las técnicas de procesamiento de imágenes generan nuevos datos pero que son extremadamente similares a los que ya se tenían, por lo que el modelo no lo entiende como datos nuevos.

## 5.4. Extracción de características

La extracción de características es necesaria ya que es necesario identificar las características más relevantes y utilizarlas para el modelo. Ya que las imágenes por sí solas tienen mucha información redundante, la cual no es útil para el entrenamiento del modelo. Existen distintas técnicas para extraer características faciales. Algunas de las probadas han sido: los puntos faciales para capturar la geometría del rostro, las unidades de acción (AU) para describir los diferentes movimientos faciales, el Histograma

de Gradientes Orientados (HOG) para extraer características visuales y el desarrollo de un modelo de aprendizaje profundo que sea capaz de identificar automáticamente las características más relevantes dada una gran cantidad de imágenes.

La técnica que mejores resultados ha proporcionado y la integrada en el modelo final es el desarrollo de un modelo de aprendizaje profundo que sea capaz de extraer las características más relevantes.

## 5.5. Selección del modelo

Se ha desarrollado un modelo que es capaz de extraer características y realizar la clasificación emocional. Primero se ha probado la técnica de *transfer learning* utilizando el modelo preentrenado *VGG16* que está entrenado con más de un millón de imágenes para reconocer 1000 clases diferentes. Pero no se han conseguido resultados tan positivos como los conseguidos con el modelo final, parece que *VGG16* necesita más imágenes para conseguir ajustar los parámetros de las últimas capas.

La arquitectura del modelo final se observa en la figura 5.5.

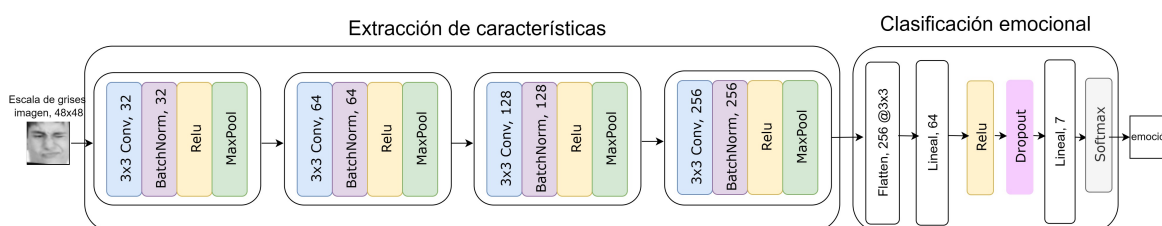


Figura 5.5: Arquitectura del modelo convolucional utilizado.

Para la extracción de características se cuenta con 4 bloques. Cada uno de estos bloques sigue la misma estructura, variando únicamente en la dimensión de la entrada y salida. Por lo que se van a explicar los elementos que conforman el primer bloque. Este primer bloque tiene una única capa convolucional bidimensional, la cual recibe como entrada una imagen 48x48 píxeles en escala de grises. Al ser una imagen en escala de grises, la capa solo necesita un canal de entrada. Está configurada de forma que tiene un tamaño de kernel de 3x3 píxeles con un desplazamiento de un píxel, para obtener el mapa de características de la imagen. A este mapa de características se aplica la técnica de *BatchNormalization*, que sirve para acelerar el entrenamiento mejorando la eficiencia de la función de activación que le sigue. En este caso la función de activación utilizada es *Relu*, que se encarga de convertir las entradas negativas en cero y dejar el mismo valor para las positivas. Para finalizar el bloque se encuentra una capa de *Pooling*, en este caso la capa **MaxPool** que se encarga de reducir la dimensionalidad de

los datos. Una vez se ha entendido un bloque los demás siguen el mismo procedimiento, lo único que cambia es que tienen distintas entradas y salidas.

Para la clasificación emocional, lo primero que se tiene que hacer es aplanar (Flatten) el mapa de características que se ha obtenido del extractor. Ya que las capas lineales necesitan los datos en un vector unidimensional. En este caso la capa de aplanado se encarga de transformar el mapa de características de dimensiones  $256 \times 3 \times 3$ , en un solo vector con 2304 características. La primera capa es una capa lineal que tiene como entrada 2304 y salida 64 neuronas. Luego se aplica la función de activación *Relu* para posteriormente utilizar un *Dropout* con una probabilidad del 50 % de que la neurona sea apagada. Esto es una técnica de regularización que se encarga de evitar el sobreajuste del modelo. La siguiente capa lineal tiene como entrada 64 y salida 7 neuronas que se introducen a la función de activación softmax para obtener la probabilidad de cada emoción. Y la emoción a devolver es la emoción con mayor probabilidad.

## 5.6. Entrenamiento y validación

El entrenamiento del modelo es el proceso mediante el cual se ajustan los parámetros utilizando los datos de entrenamiento. La validación consiste en evaluar y ajustar la capacidad de predicción del modelo sobre datos que no se han utilizado durante el entrenamiento, proporcionando una medida de cuán bien generaliza el modelo. El principal objetivo es minimizar la función de pérdida, que mide qué tan precisas son las predicciones del modelo respecto a los valores esperados.

Durante el entrenamiento del modelo, puede surgir el sobreajuste, que consiste en obtener un modelo que se comporta demasiado bien con los datos de entrenamiento pero no logra generalizar sobre datos no vistos. Para mitigar este problema, se emplea la técnica de *Early Stopping*, que implica detener el entrenamiento antes de alcanzar el número máximo de épocas especificado si no se observa una disminución en la pérdida de validación durante un cierto número de épocas, esto se llama paciencia, la cual se ha establecido en 3.

Una vez se dispone de un modelo que obtiene buenos resultados y se quiere mejorar aún más, se puede realizar el ajuste de los hiperparámetros, que consiste en reducir aún más la función de pérdida. Los principales hiperparámetros son los siguientes:

- **Época:** Indica el número de iteraciones que el conjunto completo de datos de entrenamiento se procesa por el modelo. A mayor número de épocas, mayor es la

duración del entrenamiento

- **Lote:** Indica la cantidad de muestras por grupo que se va a utilizar para calcular el gradiente y actualizar los parámetros del modelo. A mayor número de muestras mejor suele ser la generalización, ya que un caso aislado que aporte ruido no modifica significativamente el resultado.
- **Tasa de aprendizaje (LR):** Determina el tamaño del paso en cada iteración mientras avanza a un mínimo en la función de pérdida, valores muy grandes pueden hacer que no se obtenga el mínimo global y divergir. Valores muy pequeño, puede hacer que se quede atrapado en un mínimo local y nunca llegue al global.

Se ha utilizado *5-fold cross validation* para proporcionar una evaluación más precisa y robusta. Esto consiste en dividir el conjunto de datos que está dedicado a entrenamiento en validación en 5 folds, y en cada iteración selecciona 4 para entrenamiento y 1 para validación distintos. La tabla 5.1 muestra el promedio de los 5 folds para la pérdida y exactitud tanto en entrenamiento como validación durante el ajuste de los hiperparámetros.

LR	Lote	Pérdida Entrenamiento	Pérdida Validación	Exactitud Entrenamiento (%)	Exactitud Validación (%)
0.1	16	1.93	1.94	18.63	15.70
0.1	64	1.92	1.92	18.71	18.70
0.1	256	1.92	1.91	18.88	19.00
0.001	16	1.09	1.17	55.37	52.52
0.001	64	1.15	0.96	56.56	64.74
0.001	256	0.99	0.83	62.29	68.99
0.0001	16	1.16	1.19	61.84	59.62
0.0001	64	0.89	0.90	70.46	69.65
0.0001	256	0.67	0.74	77.55	74.41

Tabla 5.1: Promedio de rendimiento. En gris se muestra el que mejores resultados ha proporcionado.

Como se observa en la tabla 5.1, valores altos de la tasa de aprendizaje hacen que el modelo se quede atascado en un mínimo local, consiguiendo subóptimos resultados. A su vez, se observa que el incremento del tamaño de lote produce una mejora a la hora de la generalización del modelo incrementando el la exactitud en validación. También se observa que los mejores resultados se obtienen con una tasa de aprendizaje de 0.0001 y un tamaño de lote de 256. En la siguiente figura 5.6 se puede observar el progreso de

la media de los *folds* de la pérdida y exactitud (accuracy) a lo largo de las épocas en el mejor modelo.

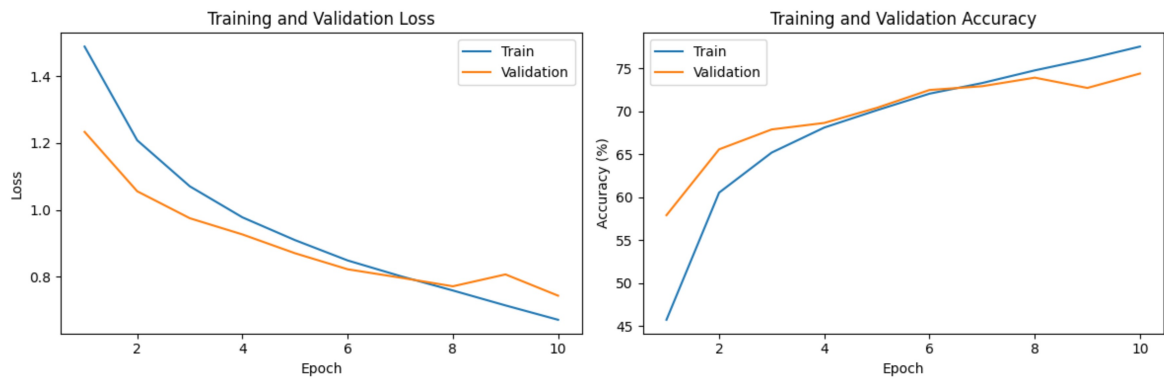


Figura 5.6: Evolución de la pérdida y exactitud, para entrenamiento y validación a lo largo de las épocas.

Se observa cómo el modelo decrementa la función de pérdida sin tener una diferencia muy grande entre la pérdida de entrenamiento y la de validación. Los resultados que se han obtenido son: 0.67 de pérdida de entrenamiento y 0.74 de pérdida de validación. Además, la precisión aumenta a medida que avanzan las épocas, sin diferir una de otra por lo que no se produce ni subajuste ni sobreajuste. La exactitud en entrenamiento ha sido de 77.55 % y la de validación de 74.41 %.

## 5.7. Evaluación

Después de seleccionar el mejor modelo, con los hiperparámetros de tasa de aprendizaje 0.0001 y tamaño de lote 256, se procede a evaluarlo con el conjunto de test, que el modelo no ha visto nunca. Esto permitirá verificar si el modelo es capaz de generalizar con datos no vistos. La evaluación con el conjunto de test ha resultado en una precisión del 62.46 %, y las métricas detalladas por emoción se pueden observar en la figura 5.7.

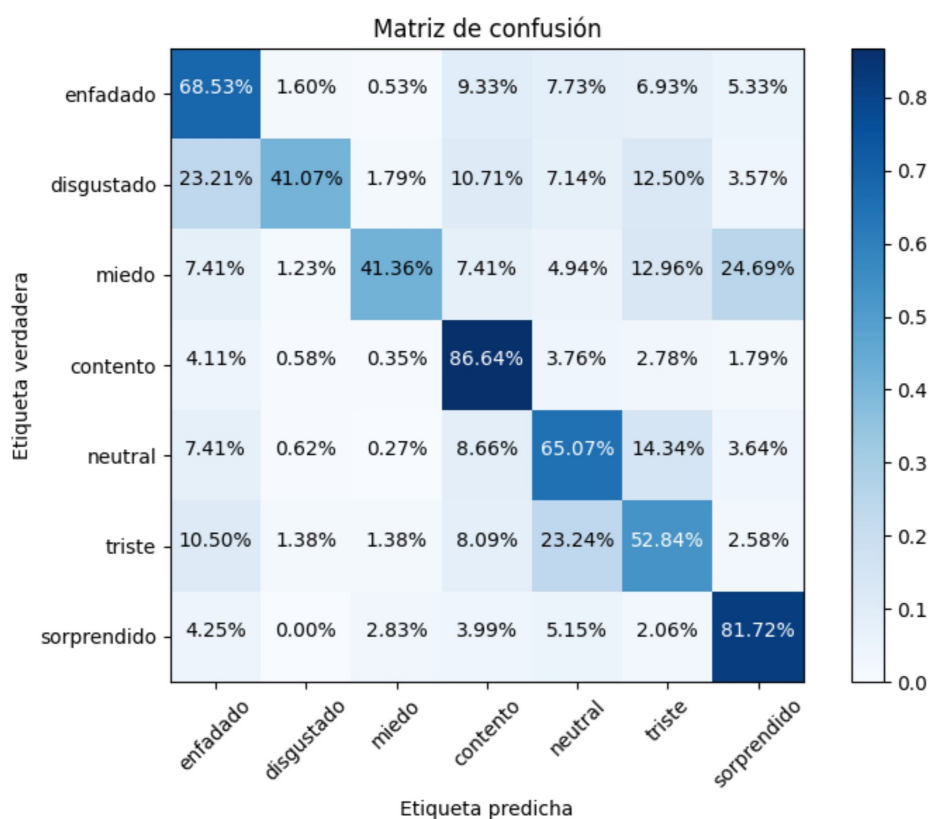


Figura 5.7: Matriz de confusión que abarca todas las emociones.

Se observa que la emoción con mayor tasa de aciertos es “contento”, con un 86.64 % de precisión en su predicción, seguida de “sorprendido” con un 81.72 %. Sin embargo, las emociones que el modelo predice con menor precisión son “disgustado” y “miedo”, con aproximadamente un 41 % de precisión en ambos casos.

Un punto destacable es que el modelo confunde “sorprendido” con “miedo” en 24.69 % de los casos. De manera, similar cuando el modelo confunde la emoción “neutral” con ‘triste’, esto ocurre un 23.24 % de las veces. Para entender por qué sucede hay que examinar la base de datos. En la figura 5.8, se observa que las emociones “sorprendido” y “miedo” comparten características comunes como tener la boca abierta y los ojos muy abiertos. De forma parecida, en la figura 5.9, se observa que las emociones

“triste” y “neutral” comparten rasgos de una expresión seria.

Sorprendido



Triste



Miedo



Neutral



Figura 5.8: Similitudes entre las emociones sorprendido y miedo.

Figura 5.9: Similitudes entre las emociones triste y neutral.

# Capítulo 6

## Evaluación del sistema

En este capítulo se presenta la evaluación del desempeño del sistema, realizando un caso real de uso. En primer lugar se presenta la metodología seguida, un análisis de los resultados obtenidos y las conclusiones finales del experimento.

### 6.1. Metodología

El objetivo de la evaluación consiste en comprobar las capacidades de la plataforma y su comportamiento frente a un caso real.

Para ello, se ha decidido llevar a cabo un experimento en el que participan 5 usuarios. Previo al experimento se les comentó a los participantes el contexto de la prueba y se solicitó a cada uno de los participantes su autorización para el uso de los datos relativos a los resultados de las encuestas y sus emociones predichas.

Cada participante se encontraba aislado de cualquier distracción, sentado frente a un ordenador equipado con una cámara para capturar imágenes. Por motivos de privacidad, no se muestran datos identificativos de los participantes, como imágenes u otra información personal.

El experimento consta de cinco tareas: tres de ellas contienen vídeos y las otras dos música. A continuación, se detallan estas tareas.

- **Tarea 1 (Vídeo - Felicidad):** tiene como objetivo observar emociones de felicidad en el usuario. Para ello, se muestra un vídeo de comedia que consiste en una recopilación de caídas graciosas.
- **Tarea 2 (Vídeo - Miedo):** tiene como objetivo ver la reacción del usuario ante un vídeo de terror que se ha seleccionado. Con el objetivo de encontrar emociones de miedo, sorpresa o indiferencia (neutral).

- **Tarea 3 (Música - Triste):** tiene como objetivo observar emociones de tristeza, para ello se ha seleccionado un fragmento de la canción "Train Wreck" de James Arthur, la cual suele producir sentimientos melancólicos.
- **Tarea 4 (Vídeo - Enfado):** tiene como objetivo observar al usuario en un ámbito de enfado o disgusto. Para ello, se muestra una noticia en la que se observa como un señor roba dinero a una persona mayor en un cajero.
- **Tarea 5 (Música - Felicidad):** tiene como objetivo observar emociones de felicidad. Para esto, se ha seleccionado un fragmento de la canción "Happy" de Pharrell Williams, conocida por ser una canción alegre.

## 6.2. Resultados y análisis

A continuación se muestran los resultados obtenidos en el experimento. Están separados por tablas, que recopilan las preguntas y respuestas que se han realizado antes y después de mostrar el contenido con el que ha sido evaluado el usuario. También se muestran las emociones que se han predicho durante la tarea. En las tablas de esta sección, la columna de respuestas se representa con la opción y el número de respuestas de esa opción, o bien, el promedio de las respuestas si son valores numéricos.

### 6.2.1. Análisis de la tarea 1

En las tablas 6.1 y 6.2 se muestran las encuestas que se han realizado antes y después de la reproducción del contenido, junto a sus resultados. En la figura 6.1 se visualizan los resultados de esta tarea: en la parte izquierda se puede observar la emoción predominante de los usuarios que ha predicho el sistema durante la reproducción del contenido y en la parte derecha se muestra en detalle la emoción predominante de los usuarios por cada segundo del contenido.

Tipo	Enunciado	Respuestas
SAM	Valencia	6.00
	Activación	5.80
	Dominancia	3.80
Pregunta	¿Es una persona que se ríe con facilidad?	Si: 4 No: 1

Tabla 6.1: Encuestas previas al contenido de la tarea 1 con sus resultados.

Tipo	Enunciado	Respuestas
SAM	Valencia	7.80
	Activación	7.00
	Dominancia	6.00
Pregunta	En una escala del 1 al 10, ¿que tan gracioso le ha parecido el vídeo?	7.2

Tabla 6.2: Encuestas posteriores al contenido de la tarea 1 con sus resultados.

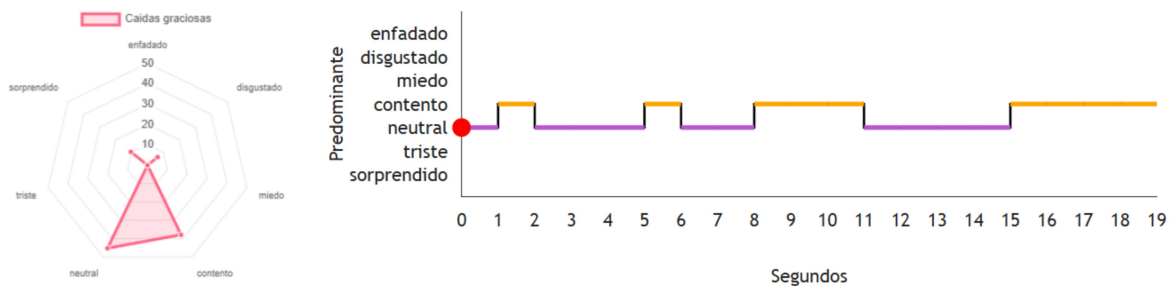


Figura 6.1: Visualización de los resultados en la Tarea 1. La gráfica de la izquierda muestra el promedio de las emociones de los usuarios, mientras que la gráfica de la derecha muestra el promedio de las emociones de los usuarios por cada segundo. Ambas gráficas son capturas de pantalla del sistema.

Al analizar los resultados, el sistema ha predicho la mayor parte del tiempo que la persona se encuentra en una emoción neutral y en ciertas partes del vídeo se reconoce la emoción 'contento'. Se puede explicar que el usuario este contento en ciertos momentos debido a que el vídeo es una recopilación de caídas, por lo que no en todo momento suceden momentos graciosos. También se observa en las encuestas respondidas, que de media a los usuarios les ha parecido un vídeo gracioso.

### 6.2.2. Análisis de la tarea 2

La tabla 6.3 muestra las encuestas realizadas justo antes del vídeo de terror, la tabla 6.4 muestra las encuestas después de reproducir el vídeo. En la figura 6.2 se observan las gráficas que muestra el sistema, en las que se observa un promedio de la emoción predominante entre todos los usuarios durante la reproducción del vídeo.

Tipo	Enunciado	Respuestas
SAM	Valencia	6.20
	Activación	5.60
	Dominancia	5.80
Pregunta	¿Es una persona que se asusta fácilmente?	Si: 3 No: 2
Pregunta	¿Que tan frecuente ve películas de terror?	Nunca: 1 De vez en cuando: 3 Frecuentemente: 1

Tabla 6.3: Encuestas previas al contenido de la tarea 2 con sus resultados.

Tipo	Enunciado	Respuestas
SAM	Valencia	4.60
	Activación	6.40
	Dominancia	4.20
Pregunta	En una escala del 1 al 10, ¿que tanto miedo ha sentido?	5.6

Tabla 6.4: Encuestas posteriores al contenido de la tarea 2 con sus resultados.

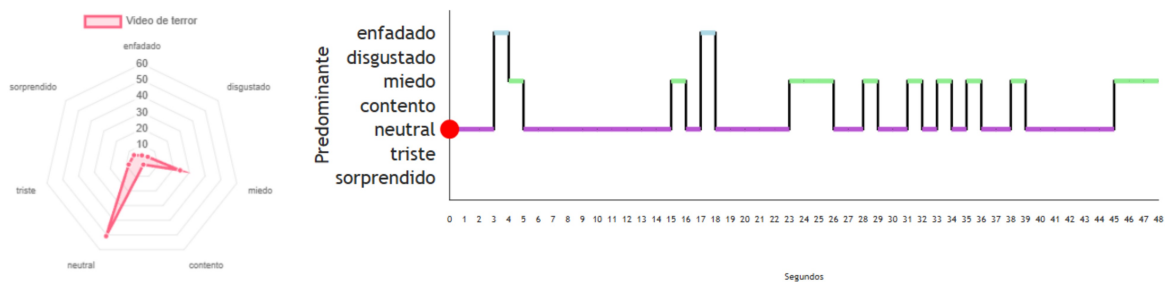


Figura 6.2: Visualización de los resultados en la Tarea 2. Ambas gráficas son capturas de pantalla del sistema.

Según los resultados, el sistema ha predicho una mayor parte del tiempo que el usuario experimenta una emoción neutral, con picos ocasionales de miedo. Observando los resultados de las encuestas, se muestra que los usuarios no han sentido gran cantidad de miedo, sin embargo, el sistema ha predicho casi la mitad de la duración del vídeo una emoción de miedo.

### 6.2.3. Análisis de la tarea 3

La tabla 6.5 muestra las encuestas realizadas justo antes de escuchar la canción triste. La tabla 6.6 muestra las encuestas justo después de escuchar la canción. Y en la figura 6.3 se observan las gráficas con las emociones predominantes durante la reproducción de la canción.

Tipo	Enunciado	Respuestas
SAM	Valencia	4.40
	Activación	5.60
	Dominancia	5.20
Pregunta	¿Es una persona que se emociona fácilmente con la música?	Si: 4 No: 1
Pregunta	¿Prefiere la música alegre o la música melancólica?	Alegre: 4 Melancólica: 1

Tabla 6.5: Encuestas previas al contenido de la tarea 3 con sus resultados.

Tipo	Enunciado	Respuestas
SAM	Valencia	4.80
	Activación	3.60
	Dominancia	5.00
Pregunta	¿Habías escuchado la canción anteriormente?	Si: 3 No: 2

Tabla 6.6: Encuestas posteriores al contenido de la tarea 3 con sus resultados.

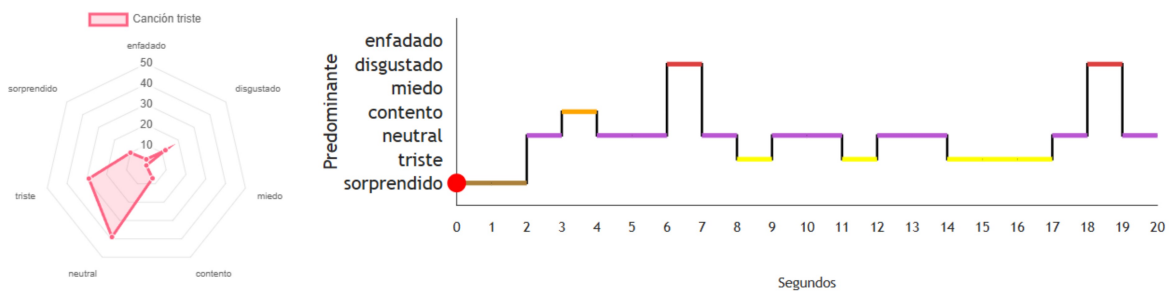


Figura 6.3: Visualización de los resultados en la Tarea 3. Ambas gráficas son capturas de pantalla del sistema.

Al analizar los resultados, el sistema ha tenido dificultades para identificar una emoción de tristeza en los usuarios, ya que la mayor parte del tiempo predice una emoción neutral. Sin embargo, en determinados momentos se predice alguna emoción un poco fuera de lugar como sorprendido, enfadado o contento. Aún así, dadas las dificultades, la segunda emoción predominante es la tristeza. Y mirando los resultados de las encuestas, en general los usuarios se encuentran más calmados después de escuchar la canción (activación).

#### 6.2.4. Análisis de la tarea 4

La tabla 6.7 muestra las encuestas realizadas justo antes de ver el vídeo donde se produce un robo. La tabla 6.8 muestra las encuestas justo después ver el vídeo.

En la figura 6.4 se observan las gráficas con las emociones predominantes durante la reproducción del vídeo.

Tipo	Enunciado	Respuestas
SAM	Valencia	4.40
	Activación	4.00
	Dominancia	4.60
Pregunta	¿Es una persona que se enfada fácilmente?	Si: 1 No: 4
Pregunta	En una escala del 1 al 10, ¿qué tan molesto le resulta ver actos de injusticia?	8.2
Pregunta	Con qué frecuencia se pone en la piel de otras personas o trata de entender sus sentimientos y perspectivas?	Nunca: 0 Raramente: 2 A veces: 2 Frecuentemente: 1

Tabla 6.7: Encuestas previas al contenido de la tarea 4 con sus resultados.

Tipo	Enunciado	Respuestas
SAM	Valencia	3.40
	Activación	6.60
	Dominancia	3.20
Pregunta	En una escala del 1 al 10, ¿cuánto enfado sintió al ver el vídeo?	6.2
Pregunta	¿Cómo se siente ahora en comparación con antes de ver el video?	Más calmado: 0 Igual: 3 Más enfadado: 2

Tabla 6.8: Encuestas posteriores al contenido de la tarea 4 con sus resultados.

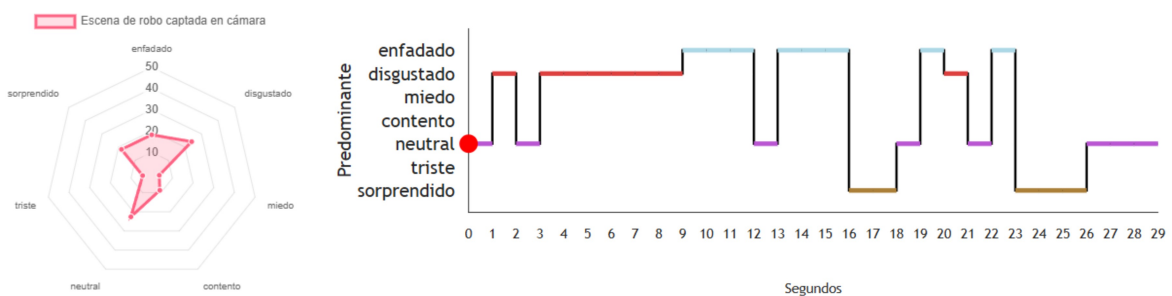


Figura 6.4: Visualización de los resultados en la Tarea 4. Ambas gráficas son capturas de pantalla del sistema.

Al revisar los resultados, un 80% de los usuarios han respondido que no se enfadan con facilidad. Sin embargo, sí que se suelen molestar bastante ante un acto de injusticia. Si se observan los resultados en las gráficas, se percibe una mezcla de emociones, viendo

como se encuentran enfadados, disgustados, neutral o sorprendidos. Esto es debido a que no todos los usuarios reaccionan igual ante la misma situación.

## 6.2.5. Análisis de la tarea 5

La tabla 6.9 muestra las encuestas realizadas justo antes de escuchar una canción alegre. La tabla 6.10 muestra las encuestas justo después escuchar la canción. Y en la figura 6.5 se observan las gráficas que muestran las emociones predominantes de los usuarios durante la reproducción de la canción.

Tipo	Enunciado	Respuestas
SAM	Valencia	4.60
	Activación	5.20
	Dominancia	4.20
Pregunta	¿Es una persona que se alegra fácilmente con la música?	Si: 4 No: 1
Pregunta	¿Qué tan frecuentemente escucha canciones alegres o positivas?	Nunca: 1 A veces: 3 Frecuentemente: 1
Pregunta	Si recuerda la última canción alegre que escuchó, ¿cómo le hizo sentir?	Más contento: 3 Igual: 1 Más triste: 2 No me acuerdo: 1

Tabla 6.9: Encuestas previas al contenido de la tarea 5 con sus resultados.

Tipo	Enunciado	Respuestas
SAM	Valencia	7.80
	Activación	6.20
	Dominancia	6.60
Pregunta	¿Habías escuchado la canción anteriormente?	Si: 4 No: 1
Pregunta	En una escala del 1 al 10, ¿qué tan feliz se sintió al escuchar la canción?	8.4
Pregunta	¿Cómo se siente ahora en comparación con antes de escuchar la canción?	Más feliz: 5 Igual: 0 Más triste: 0

Tabla 6.10: Encuestas posteriores al contenido de la tarea 5 con sus resultados.

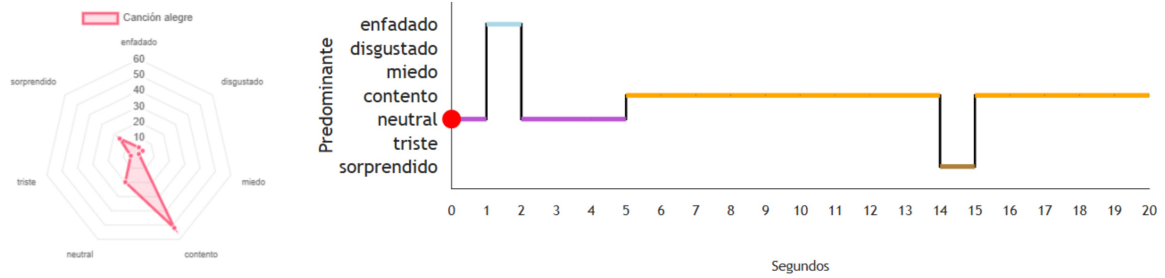


Figura 6.5: Visualización de los resultados en la Tarea 4. Ambas gráficas son capturas de pantalla del sistema.

Al observar los resultados, se observa que los usuarios han mostrado emociones de felicidad durante la mayor parte del tiempo, siendo el sistema capaz de reconocer que estaban contentos. Además, estos resultados se respaldan con los resultados de las encuestas, observando que antes de la canción los usuarios tienen una valencia de 4 de media, mientras que después de escuchar la canción su media incrementa hasta los 7.80. Esto indica que la experiencia ha sido positiva.

### 6.2.6. Conclusiones finales del experimento

Tras presentar los resultados obtenidos en cada una de las diferentes tareas, se realiza un análisis detallado de los mismos. En primer lugar, se observa la viabilidad de llevar a cabo un experimento configurable, especificando el contenido de cada tarea. También se observa como recuperar los datos almacenados por la plataforma y los resultados que muestra.

En general, los resultados del sistema son similares a la retroalimentación proporcionada por el usuario, aunque como se detalla en el capítulo 5, el sistema de predicción presenta ciertas imprecisiones. Esto se evidencia en ciertas ocasiones, donde el sistema predice una emoción que el usuario no está experimentando.

# Capítulo 7

## Gestión del proyecto

En este capítulo se explica la planificación llevada a cabo para el desarrollo del trabajo, la metodología seguida y la dedicación al proyecto.

### 7.1. Desarrollo del proyecto

El trabajo se divide en diferentes tareas. La primera tarea se corresponde con la búsqueda de aplicaciones similares y una investigación de los modelos de inteligencia artificial usados en su implementación. Una vez exploradas las distintas aplicaciones, se decide realizar los requisitos del sistema. Una vez determinadas las funcionalidades de la aplicación se implementan las funcionalidades básicas, inicio de sesión y registro junto con la definición de la estructura de la información a almacenar. Paralelamente al desarrollo de la aplicación se prueban modelos de inteligencia artificial, entrenando, validando y testeando su funcionamiento. Una vez se termina el desarrollo de la aplicación y se tiene el modelo final, se llevan a cabo pruebas con usuarios. Para visualizar cómo transcurren las distintas tareas en el tiempo, se incluye un diagrama de Gantt en la figura 7.1.

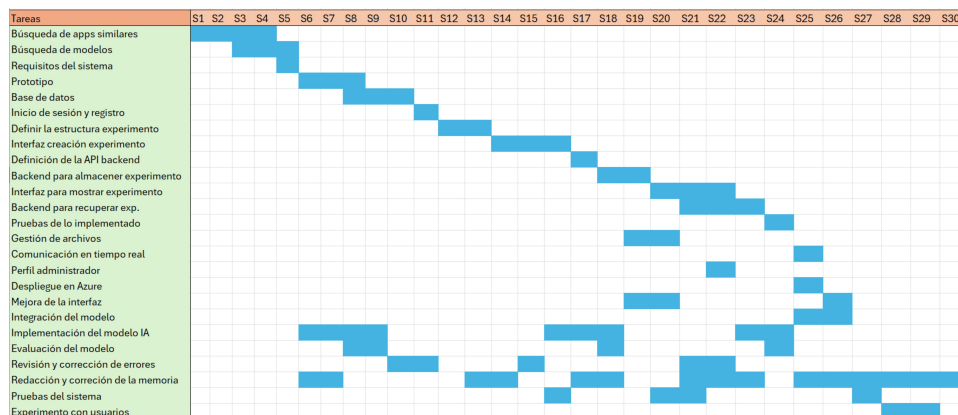


Figura 7.1: Diagrama de Gantt del desarrollo del proyecto

Para llevar a cabo el proyecto, se adoptó la metodología ágil de desarrollo con *sprints* semanales, con una reunión semanal para informar sobre el estado del proyecto. Además, cada vez que se completaba una tarea se aseguraba su funcionamiento con la realización de pruebas detalladas. El conjunto de herramientas empleadas en el proyecto se detalla en el Anexo E: Herramientas de trabajo.

### 7.1.1. Dedicación temporal

La dedicación en tiempo del proyecto se ha recogido utilizando la aplicación *Simple Time Tracker* en el cual especificas la tarea que estás realizando e inicia un cronómetro. Todos estos datos se pueden exportar posteriormente a Excel para su visualización. En la figura 7.2 se observan las horas dedicadas a cada una de las tareas. En total se han dedicado 371 horas al desarrollo de este proyecto.

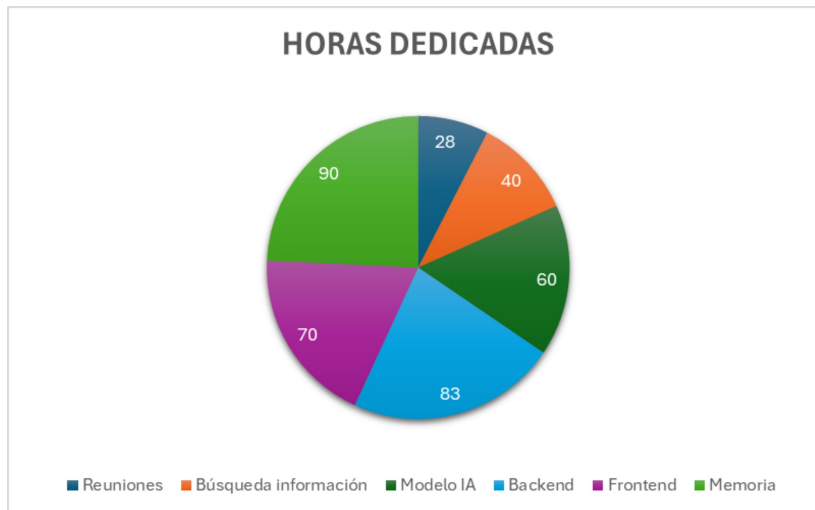


Figura 7.2: Diagrama circular con el número de horas por cada gran bloque de tareas.

# Capítulo 8

## Conclusiones y trabajo futuro

En este último capítulo se exponen las conclusiones extraídas del proyecto realizado, los conocimientos adquiridos y posibles líneas de trabajo futuro.

### 8.1. Conclusiones

Se han cumplido y desarrollado los objetivos establecidos en el primer capítulo del proyecto.

- Se han investigado y analizado las distintas aplicaciones similares a la implementada, para comprender los puntos fuertes y limitaciones que presentan.
- Se ha diseñado e implementado una plataforma web que permite evaluar usuarios que han realizado un experimento.
- Se permite la creación de experimentos configurables. Permitiendo gran variedad de configuraciones para permitir la realización de experimentos que se ajusten a las necesidades específicas solicitadas.
- Se permite la visualización de los resultados que se han obtenido tras la realización de un experimento, presentando gráficas tanto para las respuestas que proporcionan los usuarios como para los datos que otorga el sistema respecto a la predicción emocional.
- Se han estudiado las distintas técnicas que se utilizan para el reconocimiento emocional mediante características faciales, además de una implementación de un modelo propio.
- Se ha integrado el modelo de predicción desarrollado dentro de la aplicación web, para permitir obtener datos emocionales.

- Se ha conseguido desplegar la aplicación web utilizando los servidores de Azure para permitir el acceso a la plataforma mediante el navegador web.

## **8.2. Conocimientos adquiridos**

La realización de este trabajo ha posibilitado adquirir conocimientos técnicos y personales, que se presentan a continuación.

### **8.2.1. Técnicos**

Se presentan los conocimientos técnicos adquiridos:

- Prácticas recomendadas en el desarrollo del frontend de la aplicación web: diseño de las páginas, creación de los diversos componentes y estructura del proyecto.
- Prácticas recomendadas en el desarrollo del backend de la aplicación: Implementación de un diseño modular donde cada componente cumple una función específica, y gestión de proyectos de estas dimensiones.
- Creación de modelos de inteligencia artificial: Desarrollo completo del modelo desde la recolección inicial de datos hasta la implementación del modelo final.
- El diseño de una arquitectura con múltiples servidores, incluyendo la definición precisa de los datos que cada servidor recibe, transmite y el destino al cual se dirige la información.
- La importancia del diseño modular para permitir realizar pruebas específicas sobre cada componente, facilitando así el aislamiento y resolución de los posibles problemas de manera efectiva.
- La estructuración y redacción de una documentación para un proyecto con alta carga de trabajo.

### **8.2.2. Personales**

Se presentan los conocimientos personales adquiridos:

- Aprender a planificar y organizar las distintas tareas, realizando una predicción del tiempo que va a llevar, manteniendo un margen para imprevistos que puedan surgir.

- Adaptación al uso de nuevas técnicas y herramientas a través de la lectura de su documentación correspondiente.
- Organización y redacción de las tareas realizadas para futuras consultas. Sobre todo, para la organización de artículos científicos, realizando una toma de notas estructurada.

### 8.3. Trabajo futuro

Tras finalizar este trabajo se observa de cara a una expansión del proyecto diferentes tareas que sería interesante realizar.

- **Integración de otro tipo de sensores para la captura de emociones:** Ampliar la gama de sensores y características, añadiendo sensores como pulseras Empatica para obtener mediciones de la conductividad de la piel o electrodos para capturar electroencefalogramas.
- **Evaluación rigurosa:** Realizar evaluaciones más rigurosas y sistemáticas tanto con experimentadores como con usuarios finales.
- **Ampliación del reconocimiento emocional:** Incluir más emociones para reconocer, no limitándose solo a las básicas, sino también incorporando emociones como estrés, aburrimiento, confusión o envidia, entre otras.
- **Mayor configurabilidad:** Permitir añadir otros tipos de formularios como *Positive and Negative Affect Schedule* (PANAS), *Trier Social Stress Test* (TSST) y *Pick-A-Mood* (PAM). Esto no supondría una gran carga de trabajo debido a que se ha dejado la aplicación preparada para permitir nuevos tipos de formularios.
- **Nuevo modo de registro continuo:** Agregar una opción en la plataforma que permita a los usuarios registrar continuamente su estado emocional mientras están utilizando el ordenador.
- **Mejora del modelo de predicción a partir de características faciales:** Utilizar más conjuntos de datos para el entrenamiento y validación, además de utilizar diferentes datos fisiológicos para incrementar su precisión.



# Capítulo 9

## Bibliografía

- [1] Luiz Pessoa. On the relationship between emotion and cognition. *Nature reviews neuroscience*, 9(2):148–158, 2008.
- [2] Yingli Tian, Takeo Kanade, and Jeffrey F Cohn. Facial expression recognition. *Handbook of face recognition*, pages 487–519, 2011.
- [3] Proyecto de Transición Ecológica y Transición Digital. Technological ecosystem for the mood recognition in cardiac rehabilitation patients (temor). Project Report TED2021-130374B-C22, 2021.
- [4] Página oficial de morphcast - <https://www.morphcast.com/>.
- [5] Facereader - <https://www.noldus.com/facereader>.
- [6] Demo afectiva - <https://demo.mr.affected.com/>.
- [7] Emotimeter aplicación de playstore.
- [8] Imentiv - <https://imentiv.ai/product-use-cases/video-emotion-recognition/>.
- [9] Teah-Marie Bynion and Matthew T Feldner. Self-assessment manikin. *Encyclopedia of personality and individual differences*, pages 4654–4656, 2020.
- [10] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20*, pages 117–124. Springer, 2013.

- [11] Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM international conference on multimodal interaction*, pages 279–283, 2016.
- [12] Chatgpt - <https://chatgpt.com/>.
- [13] Rita Meziati, Yannick Benezeth, Pierre De Oliveira, Julien Chappé, and Fan Yang. Ubfc-phys. 2021.
- [14] Yassine Ouzar, Lynda Lagha, Frédéric Bousefsaf, and Choubeila Maaoui. Multimodal stress state detection from facial videos using physiological signals and facial features. In *International Conference on Pattern Recognition*, pages 139–150. Springer, 2022.
- [15] Clemens Kirschbaum, Karl-Martin Pirke, and Dirk H Hellhammer. The ‘trier social stress test’—a tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 28(1-2):76–81, 1993.
- [16] Página oficial de affectiva - <https://demo.mr.affectiva.com/>.
- [17] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. Affdex sdk: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*, pages 3723–3726, 2016.
- [18] Damien Dupré, Eva Krumhuber, Dennis Küster, and Gary McKeown. Emotion recognition in humans and machine using posed and spontaneous facial expression. 2019.
- [19] Yousif Khaireddin and Zhuofa Chen. Facial emotion recognition: State of the art performance on fer2013. *arXiv preprint arXiv:2105.03588*, 2021.
- [20] Amjad Rehman Khan. Facial emotion recognition using conventional machine learning and deep learning methods: Current achievements, analysis and remaining challenges. *Information*, 13(6):268, 2022.
- [21] Página de draw.io - <https://app.diagrams.net/>.
- [22] Página de overleaf - <https://es.overleaf.com/>.
- [23] Victory - <https://commerce.nearform.com/open-source/victory/>.
- [24] MongoDB compass - <https://www.mongodb.com/products/tools/compass>.

# Lista de Figuras

3.1. Patrón de diseño MVC con las capas adicionales de websocket y repositorio. Ver texto. . . . .	11
3.2. Diagrama de la arquitectura software utilizada . . . . .	12
3.3. Pantalla de inicio de sesión. . . . .	13
3.4. Pantalla de registro de sesión. . . . .	13
3.5. Pantalla de configuración de un experimento. . . . .	14
3.6. Pantalla de configuración de una tarea. . . . .	15
3.7. Pantalla con varias encuestas en la sección de encuestas previas. . . . .	15
3.8. Pantalla de configuración de una pregunta tipo rango numérico . . . . .	16
3.9. Pantalla de configuración de una pregunta tipo selección múltiple . . . . .	16
3.10. Pantalla de configuración de una pregunta tipo verdadero/falso. . . . .	16
3.11. Pantalla de selección del contenido de una tarea. . . . .	17
3.12. Pantalla con el modal para subir un archivo. . . . .	17
3.13. Pantalla con el listado de los experimentos disponibles. . . . .	18
3.14. Pantalla con el modal de acceso a un experimento seleccionado. . . . .	18
3.15. Pantalla del pretexto de un experimento. . . . .	19
3.16. Pantalla del formulario SAM. . . . .	19
3.17. Pantalla con los tres tipos de preguntas existentes en la aplicación. . . . .	20
3.18. Pantalla del contenido de una tarea, la imagen del usuario y la emoción predicha. . . . .	21
3.19. Pantalla con los experimentos del experimentador. . . . .	21
3.20. Pantalla del listado de usuarios que han realizado un experimento concreto. . . . .	22
3.21. Pantalla de las emociones en las distintas tareas de un experimento. . . . .	23

3.22. Pantalla de estadísticas de las emociones predichas para un usuario. . . . .	23
3.23. Pantalla de resultados de las encuestas. . . . .	24
3.24. Pantalla de administrador con la lista de usuarios en la aplicación. . . . .	25
3.25. Pantalla del modal que permite cambiar el rol de un usuario. . . . .	25
3.26. Diagrama del diseño de la base de datos NoSQL. . . . .	28
4.1. Estructura JSON para la configuración del experimento . . . . .	32
4.2. Estructura JSON de la tarea . . . . .	32
4.3. JSON de encuesta tipo SAM . . . . .	33
4.4. JSON de encuesta tipo pregunta . . . . .	33
4.5. JSON pregunta de tipo sí/no . . . . .	33
4.6. JSON pregunta de tipo rango numérico . . . . .	33
4.7. JSON pregunta de tipo elección múltiple . . . . .	33
4.8. Diagrama de estados que representa el ciclo de vida de un experimento.	34
4.9. Gráfica con las emociones en una tarea . . . . .	37
4.10. Gráfica con las emociones en un instante de tiempo concreto . . . . .	37
4.11. Gráfica con la emoción predominante en cada instante de tiempo . . . . .	37
4.12. JSON con los datos a pasar a las gráficas. . . . .	37
4.13. Directorio con las imágenes almacenadas de un usuario . . . . .	40
5.1. Proceso seguido para la realización del modelo de inteligencia artificial.	43
5.2. Muestra de imágenes de la base de datos por emoción. . . . .	44
5.3. Ejemplos de imágenes que han sido descartadas . . . . .	45
5.4. Comparación entre imágenes sin normalizar y normalizadas. . . . .	45
5.5. Arquitectura del modelo convolucional utilizado. . . . .	47
5.6. Evolución de la pérdida y exactitud, para entrenamiento y validación a lo largo de las épocas. . . . .	50
5.7. Matriz de confusión que abarca todas las emociones. . . . .	51
5.8. Similitudes entre las emociones sorprendido y miedo. . . . .	52
5.9. Similitudes entre las emociones triste y neutral. . . . .	52

6.1. Visualización de los resultados en la Tarea 1. La gráfica de la izquierda muestra el promedio de las emociones de los usuarios, mientras que la gráfica de la derecha muestra el promedio de las emociones de los usuarios por cada segundo. Ambas gráficas son capturas de pantalla del sistema. . . . .	55
6.2. Visualización de los resultados en la Tarea 2. Ambas gráficas son capturas de pantalla del sistema. . . . .	56
6.3. Visualización de los resultados en la Tarea 3. Ambas gráficas son capturas de pantalla del sistema. . . . .	57
6.4. Visualización de los resultados en la Tarea 4. Ambas gráficas son capturas de pantalla del sistema. . . . .	58
6.5. Visualización de los resultados en la Tarea 4. Ambas gráficas son capturas de pantalla del sistema. . . . .	60
7.1. Diagrama de Gantt del desarrollo del proyecto . . . . .	61
7.2. Diagrama circular con el número de horas por cada gran bloque de tareas. . . . .	62
C.1. Gráfica con los resultados a una pregunta de tipo rango numérico. . . . .	83
C.2. Gráfica con los resultados al formulario SAM. . . . .	83
C.3. Gráfica con el contenido emocional en un momento concreto. . . . .	84
C.4. Gráfica temporal con la emoción predominante por segundo. . . . .	84
C.5. Gráfica con los resultados a una pregunta de tipo elección múltiple. . . . .	84
C.6. Gráfica con los resultados a una pregunta de tipo elección múltiple. . . . .	85



# Lista de Tablas

2.1. Tabla de aplicaciones significativas . . . . .	8
2.2. Requisitos Funcionales y No Funcionales . . . . .	10
5.1. Promedio de rendimiento. En gris se muestra el que mejores resultados ha proporcionado. . . . .	49
6.1. Encuestas previas al contenido de la tarea 1 con sus resultados. . . . .	54
6.2. Encuestas posteriores al contenido de la tarea 1 con sus resultados. . . . .	55
6.3. Encuestas previas al contenido de la tarea 2 con sus resultados. . . . .	56
6.4. Encuestas posteriores al contenido de la tarea 2 con sus resultados. . . . .	56
6.5. Encuestas previas al contenido de la tarea 3 con sus resultados. . . . .	57
6.6. Encuestas posteriores al contenido de la tarea 3 con sus resultados. . . . .	57
6.7. Encuestas previas al contenido de la tarea 4 con sus resultados. . . . .	58
6.8. Encuestas posteriores al contenido de la tarea 4 con sus resultados. . . . .	58
6.9. Encuestas previas al contenido de la tarea 5 con sus resultados. . . . .	59
6.10. Encuestas posteriores al contenido de la tarea 5 con sus resultados. . . . .	59
B.1. Detalles de la colección Administrador. . . . .	79
B.2. Detalles de la colección Usuario. . . . .	80
B.3. Detalles de la colección Experimento. . . . .	80
B.4. Detalles de la colección Archivo. . . . .	81
B.5. Detalles de la colección Estadísticas. . . . .	81
B.6. Detalles de la colección Resultados. . . . .	82
E.1. Herramientas de trabajo utilizadas a lo largo del proyecto. . . . .	92



# Anexos



## Anexo A

# Análisis detallado de aplicaciones de reconocimiento emocional

En este anexo se detallan las aplicaciones de reconocimiento emocional investigadas que son similares a la que se muestra en este proyecto.

- **Afectiva:** es una compañía fundada por Dr. Rana el Kalioub y Dr. Rosalind Picard, los cuales han desarrollado varias aplicaciones en el campo de la computación afectiva. Cuentan con una aplicación web de detección de emociones en tiempo real que utiliza la cámara web del ordenador para identificar las emociones del usuario. Tienen una demo [6] que permite elegir entre 6 vídeos para evaluar el funcionamiento del sistema. El modelo de machine learning [17] utilizado se basa en una base de datos de más de 14,7 millones de imágenes de caras. La técnica que utilizan es el algoritmo Support Vector Machine (SVM) para obtener las Action Units (AU) y mediante el sistema de codificación de acciones faciales emocionales (EFACS) obtienen una puntuación sobre la emoción predicha.
- **Morphcast:** Es una compañía fundada por Stefano Bargagni, los cuales cuentan con varias aplicaciones en el ámbito del reconocimiento de emociones mediante características faciales. Cuentan con una demo en la web [4] la cual muestra métricas más detalladas que la anterior, como pueden ser la pose de la cabeza del usuario la valencia y activación experimentada. También contienen una opción de escritorio con algunas características adicionales. A diferencia del anterior, donde ellos han creado su propio modelo, esta aplicación evalúa distintos modelos de predicción, integrando el que mejor resultados les proporciona [18].
- **FaceReader:** Es una aplicación de escritorio [5] de la compañía Noldus que se encarga de la computación afectiva la cual cuenta con una aplicación llamada

FaceReader que proporciona a los usuarios información relativa a las expresiones faciales. Es una aplicación de pago que tiene diferentes planes, la cual permite visualizar y obtener información. Pero no cuenta con una forma de crear tu propio experimento de forma independiente, ya que se necesita contactar con uno de la aplicación e intentan adaptarlo a tus necesidades. El modelo que utilizan es un modelo que permite la clasificación emocional mediante las Action Units(AU).

- **Emotimeter:** Es una aplicación Android disponible en la PlayStore [7], que ha integrado un modelo de clasificación emocional en el dispositivo Android. No obstante, sus funcionalidades son limitadas, ya que solo permite la predicción de emociones sin la capacidad de guardarlas o proporcionar información más detallada.
- **Imentiv:** Disponible tanto en versión de escritorio como web [8]. Imentiv permite el análisis detallado de vídeos especificados por el usuario. A diferencia de las anteriores, no se enfoca en imágenes que han sido capturadas por la cámara del usuario, sino en analizar vídeos pregrabados. En ella se puede obtener las emociones experimentadas a lo largo del tiempo.

# Anexo B

## Detalles de la API del backend

En este anexo se detallan todas las rutas que componen la API del backend. Las tablas [B.1](#), [B.2](#), [B.3](#), [B.4](#), [B.5](#) y [B.6](#) presentan información detallada sobre cada conjunto de rutas. El término URL se refiere a la ruta que se debe llamar para llevar a cabo la acción descrita en la columna de descripción. El método especificado indica el tipo de solicitud requerida, mientras que la columna de rol representa el nivel de acceso del usuario necesario para interactuar con el recurso.

### Colección Administrador

En la tabla [B.1](#) se detallan las rutas disponibles en la colección Administrador.

URL	Descripción	Método	Rol
<a href="#">/admin/inicio-sesion</a>	Ruta para iniciar sesión por el administrador. Ya que es la ruta para iniciar sesión, se debe permitir el acceso a todos, debido a que aún no está iniciada ninguna sesión, por lo que no tienen rol.	POST	Todos
<a href="#">/admin/listar-usuarios</a>	Ruta para listar todos los usuarios de la aplicación.	GET	Administrador
<a href="#">/admin/cambiar-rol-usuario</a>	Ruta para permitir cambiar el rol de un usuario.	PUT	Administrador

Tabla B.1: Detalles de la colección Administrador.

## Colección Usuario

En la tabla B.2 se detallan las rutas disponibles en la colección Usuario.

URL	Descripción	Método	Permisos
/registro	Ruta para registrarse y crear una nueva cuenta de usuario.	POST	Todos
/inicio-sesion	Ruta para iniciar sesión.	POST	Todos

Tabla B.2: Detalles de la colección Usuario.

## Colección Experimento

En la tabla B.3 se detallan las rutas disponibles en la colección Experimento.

URL	Descripción	Método	Permisos
/experimentos	Ruta para obtener todos los experimentos que existen.	GET	Todos
/experimentos	Ruta para crear un experimento, especificando todos sus datos.	POST	Experimentador
/experimentos/info	Ruta para obtener los detalles de un experimento.	GET	Todos
/experimentos/acceso	Ruta para verificar el acceso a un experimento, comprobando si la contraseña es correcta. Esta devuelve un token de autorización. Para transmitir la contraseña de forma segura, se utiliza el método POST.	POST	Usuario

Tabla B.3: Detalles de la colección Experimento.

## Colección Archivo

En la tabla B.4 se detallan las rutas disponibles en la colección Archivo.

URL	Descripción	Método	Rol
/archivos	Ruta para obtener los archivos almacenados en el servidor.	GET	Experimentador
/archivo/:link	Ruta para obtener un archivo multimedia específico para permitir su visualización.	GET	Todos
/archivo/subir	Ruta para subir un archivo al servidor.	POST	Experimentador
/imagenes/tarea	Ruta para obtener el directorio de las imágenes de un usuario en una tarea.	GET	Experimentador
imagenes/: idExperimento/: idTarea/:idUsuario/: filename	Ruta para obtener la imagen de un usuario.	GET	Experimentador

Tabla B.4: Detalles de la colección Archivo.

## Colección Estadísticas

En la tabla B.5 se detallan las rutas disponibles en la colección Estadísticas.

URL	Descripción	Método	Permisos
/estadisticas/ experimentos	Ruta para obtener mis experimentos.	GET	Experimentador
/estadisticas/ experimentos/usuarios	Ruta para obtener todos los usuarios que han realizado mi experimento.	GET	Experimentador
/estadisticas/tareas	Ruta para obtener las estadísticas relacionadas con las tareas.	GET	Experimentador
/estadisticas/ contenido	Ruta para obtener las estadísticas relacionadas con el contenido de la tarea.	GET	Experimentador
/estadisticas/ encuestas	Ruta para obtener las estadísticas relacionadas con los resultados de las encuestas.	GET	Experimentador

Tabla B.5: Detalles de la colección Estadísticas.

## Colección Resultado

En la tabla B.6 se detallan las rutas disponibles en la colección Resultado.

URL	Descripción	Método	Permisos
/resultados/encuestas	Ruta para guardar el resultado de una encuesta.	POST	Usuario
/resultados/experimentos	Ruta para guardar el resultado de un experimento.	POST	Usuario

Nota: Los resultados de las tareas se guardan mediante los WebSockets, por lo que no es parte de la API.

Tabla B.6: Detalles de la colección Resultados.

## Anexo C

# Tipos de gráficas disponibles en la plataforma

En este anexo se presentan los tipos de gráficas disponibles en la plataforma, con una explicación de su función.

La figura C.1 muestra la gráfica que contiene la información de una respuesta de tipo numérico, en la que se indica el valor y el número de respuestas en ese valor. En la figura C.2 se presenta la gráfica con la respuesta a un formulario SAM, en ella se presenta el promedio de las respuesta para la valencia, activación y dominancia.



Figura C.1: Gráfica con los resultados a una pregunta de tipo rango numérico.



Figura C.2: Gráfica con los resultados al formulario SAM.

La figura C.3 muestra la gráfica que contiene el porcentaje de usuarios que han experimentado la emoción que se muestra en un instante de tiempo concreto. En la gráfica, se observa que el 50 % de los usuarios están experimentando que se encuentran contentos, un 25 % disgustados y el restante 25 % sorprendidos.

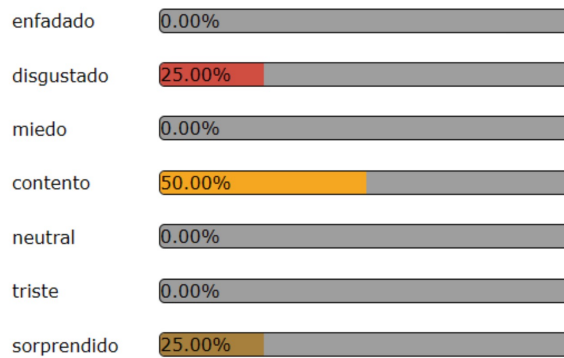


Figura C.3: Gráfica con el contenido emocional en un momento concreto.

La figura C.4 muestra la gráfica que contiene una visualización de la emoción predominante en cada instante de tiempo. El punto rojo está sincronizado con la reproducción del contenido multimedia, por lo que indica el segundo actual del contenido.

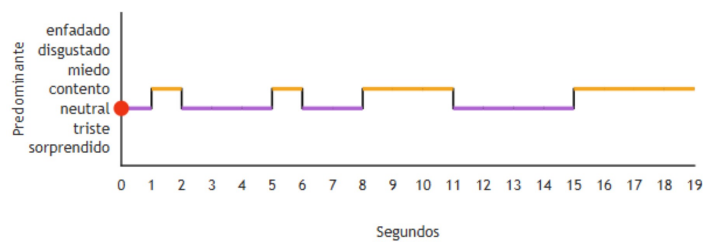


Figura C.4: Gráfica temporal con la emoción predominante por segundo.

La figura C.5 presenta la gráfica con los resultados a una pregunta de tipo elección múltiple. En ella se observa el porcentaje de respuestas por opción y una leyenda en la parte derecha. La pregunta de tipo verdadero/falso utiliza la misma gráfica pero con dos opciones.

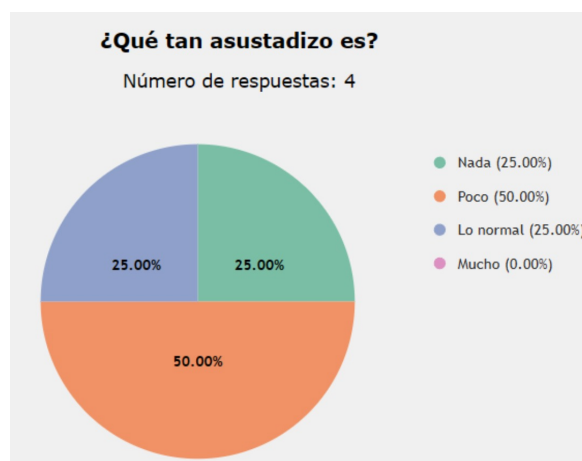


Figura C.5: Gráfica con los resultados a una pregunta de tipo elección múltiple.

La figura C.6 representa la gráfica con las emociones que han experimentado los usuarios en una determinada tarea. En ella se muestran las emociones del sistema junto el porcentaje de cada emoción experimentada.

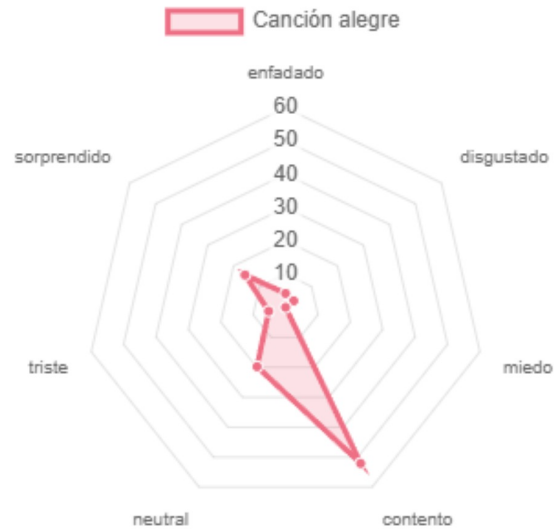


Figura C.6: Gráfica con los resultados a una pregunta de tipo elección múltiple.



# Anexo D

## Modelos y técnicas descartadas para el reconocimiento emocional

En este anexo se describe el proceso seguido para desarrollar un modelo destinado a reconocer una emoción que no pertenece a las básicas, la emoción de estrés. Además, se exploran diversas técnicas y modelos para el reconocimiento de emociones básicas.

### D.1. Modelo para el reconocimiento de estrés/no estrés

La base de datos UBFC-Phys [13] se ha utilizado para entrenar el modelo de clasificación de estrés/no estrés. Incluye datos de 56 sujetos sometidos a tres tareas: descanso, entrevista y aritmética. Para evaluar el estado emocional de los usuarios, se realiza una autoevaluación utilizando el Trier Social Stress Test (TSST) [15] antes y después de cada sesión.

Cada usuario en la base de datos tiene tres vídeos de tres minutos de duración, correspondientes a cada una de las tareas, así como la puntuación obtenida en el TSST. Además de los vídeos, la base de datos incluye otras señales fisiológicas que no se han utilizado.

Dado que la información de los usuarios consiste en vídeos, el primer paso es convertirlos en fotogramas, seleccionando un fotograma por segundo. Una vez se tienen todas las imágenes de los usuarios, se procede con su normalización, centrando la cara del usuario y ajustando su rotación con los puntos de referencia faciales que se han obtenido con la librería *mediapipe*.

Una vez recopilados todos los datos, es necesario realizar un balanceo de la base de datos, dado que presenta un alto desbalanceo; aproximadamente el 80 % de las imágenes

corresponden a no estrés. Por lo tanto se aplican técnicas de balanceo de datos, como SMOTE.

El modelo que se ha utilizado ha sido el VGG16 con la técnica de *transfer learning*, siguiendo el artículo [14]. Esta técnica implica adaptar un modelo pre-entrenado diseñado originalmente para clasificar mil clases de objetos (VGG16), para que ahora sea capaz de reconocer el estrés. Para lograr esto, solamente se modifican las capas finales del modelo que se encargan de la clasificación de objetos, pero se mantienen las capas de extracción de características. Estas capas finales se sustituyen por otras para que la capa final solamente tenga una salida que sea estrés o no estrés. Además, antes de la primera capa lineal se añade una capa Long Short-Term Memory (LSTM), debido a que se tienen secuencias de fotogramas de los usuarios, se utiliza para capturar las dependencias temporales entre los fotogramas.

Durante el entrenamiento del modelo, se han explorado distintas técnicas de partición de imágenes:

1. Cada usuario solo puede pertenecer o al conjunto de entrenamiento o al de validación con la secuencia de fotogramas completa.
2. Cada usuario puede pertenecer tanto al conjunto de entrenamiento como al de validación, estando mezclados.

La primera forma es la única que se considera correcta, ya que no se produce *Data leakage*, que consiste en que los datos utilizados para entrenar el modelo se filtran con los datos de validación, resultando en una evaluación incorrecta.

Se han evaluado ambas metodologías para comparar sus resultados. La primera metodología obtiene unos resultados deficientes, con una precisión del 36 % en el mejor modelo. En contraste, la segunda técnica ha mostrado unos resultados muy similares a los del artículo de referencia, logrando un 80 % de precisión en comparación con el 82 % mencionado en dicho artículo. Esto sugiere que posiblemente en el artículo se ha utilizado esta técnica para la separación de los datos, aunque no se especifica en ninguna parte del mismo.

A pesar de obtener un 80 % utilizando la segunda metodología, esta se ha descartado debido a que no refleja una aproximación realista. Se muestra un ejemplo de lo que sucede. Por ejemplo, el modelo es entrenado con los fotogramas del 1 al 10 de un usuario y luego evaluado con los fotogramas del 11 al 20 del mismo usuario. Esto permite que el modelo se aprenda ese usuario generando predicciones precisas en este contexto. Sin embargo, no es capaz de generalizar, debido a que si se prueban usuarios que no ha

visto nunca, este no va a ser capaz de predecir correctamente, ya que solamente sabe predecir los usuarios con los que ha sido entrenado.

## D.2. Exploración de modelos y técnicas para el reconocimiento de emociones básicas

Todas las pruebas realizadas a continuación se han llevado a cabo utilizando la misma base de datos que se ha empleado para el modelo final, FER+.

Se ha probado con Support Vector Machine (SVM) y utilizando como entrada los descriptores que se obtienen mediante el Histograma orientado de gradientes (HOG), sin embargo esta opción se descartó debido a que no se ha conseguido hacer que la extracción de las características del HOG funcione en tiempo real.

Se ha probado a extraer las *Facial Action Units* (AU) utilizando la librería *Py-Feat* y emplearlas como entradas para entrenar un modelo simple con tres capas densas. Sin embargo, esta opción fue descartada debido a la dificultad para implementar la extracción de AU en *JavaScript*, el lenguaje utilizado en el backend de la plataforma. Por lo que implementar esta opción requiere del uso de un servidor externo que utilice *Python* como lenguaje. En el capítulo 3 se muestra porque se descarta el uso de un servidor externo.

Utilizando la librería *mediapipe* se ha probado a extraer los puntos de reconocimiento facial y utilizarlos como entrada para un modelo básico con pocas capas densas. Sin embargo, al extraer 468 puntos faciales con coordenadas x,y,z se tiene un problema de dimensionalidad, lo que produce que el modelo no consiga buenos resultados. Por esta razón, se ha probado la técnica de *Principal Component Analysis* (PCA), diseñada para reducir la dimensionalidad de los datos mientras se conserva la mayor cantidad de información posible. Con PCA, se ha logrado reducir las características de 468x3x3 a solo 23 características, que explican el 96 % de la variabilidad de los datos. A pesar de esta reducción, los resultados no fueron satisfactorios debido a problemas de sobre ajuste del modelo.



# Anexo E

## Herramientas de trabajo

En este anexo presenta la tabla E.1 que contiene las diferentes herramientas y tecnologías utilizadas para el diseño, implementación, desarrollo de modelos, gestión y documentación del proyecto.

Herramientas y Grupos	Descripción
Herramientas de desarrollo	<b>Visual Studio Code:</b> IDE empleado para el backend, frontend y desarrollo del modelo.
	<b>MongoDB Compass [24]:</b> Gestor de la base de datos NoSQL utilizada.
	<b>Postman:</b> Utilizado para comprobar el correcto funcionamiento de la API del backend.
Lenguajes de programación	<b>Python:</b> Lenguaje utilizado para el desarrollo del modelo de inteligencia artificial.
	<b>JavaScript:</b> Lenguaje utilizado para el backend y frontend de la plataforma.
Bibliotecas o frameworks utilizados	<b>Pytorch:</b> Empleada para el desarrollo de los modelos de inteligencia artificial.
	<b>Mediapipe:</b> Biblioteca en Python encargada de extraer puntos faciales y detección de caras.
	<b>React:</b> Biblioteca en JavaScript utilizada para realizar el frontend de la aplicación.
	<b>Chart.js:</b> Biblioteca en JavaScript utilizada para permitir mostrar gráficas en la plataforma.
	<b>Victory [23]:</b> Biblioteca en JavaScript utilizada para realizar gráficas más complejas en la plataforma.

	<p><b>Node.js:</b> Entorno que permite ejecutar JavaScript en el servidor del backend.</p> <p><b>Express.js:</b> Framework en JavaScript utilizado para realizar el backend de la aplicación.</p>
Herramientas de gestión	<p><b>Notion:</b> Empleada para gestionar las tareas a realizar.</p> <p><b>Obsidian:</b> Empleada para tomar notas más detalladas.</p> <p><b>Zotero:</b> Empleada para gestionar los artículos científicos y sus bibliografías.</p> <p><b>Simple Time Tracker:</b> Aplicación android utilizada para llevar registro de las horas dedicadas al proyecto.</p>
Herramientas de documentación	<p><b>Overleaf [22]:</b> Empleada para la redacción de la memoria del proyecto.</p> <p><b>Excel:</b> Empleada para obtener las gráficas de la dedicación temporal.</p> <p><b>Draw.io [21]:</b> Empleada para la generación de diagramas.</p>
Otros	<p><b>Azure:</b> Servicio en la nube utilizado para desplegar los servidores.</p> <p><b>HTML:</b> Lenguaje de etiquetas utilizado para realizar las pantallas de la plataforma.</p> <p><b>CSS:</b> Lenguaje de estilo utilizado para el diseño de los documentos HTML.</p> <p><b>JSX:</b> Extensión de JavaScript utilizado para el desarrollo de la aplicación en React.</p> <p><b>JSON:</b> Formato en el que se transmiten la información entre el backend y el frontend de la aplicación.</p>

Tabla E.1: Herramientas de trabajo utilizadas a lo largo del proyecto.