



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Detección de estados de ánimo usando técnicas de
machine learning

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

AUTOR/A: Carceller Llorens, Fernando

Tutor/a: Insfrán Pelozo, César Emilio

Cotutor/a: Abrahao Gonzales, Silvia Mara

CURSO ACADÉMICO: 2022/2023



Valencian Graduate School
and Research Network of Artificial Intelligence



**GENERALITAT
VALENCIANA**

**Conselleria de Innovación,
Universidades, Ciencia
y Sociedad Digital**

Resumen

El estudio de los cambios en el estado de ánimo del usuario mientras utiliza un sistema software puede dar indicios de qué aspectos de la Interfaz de Usuario (IU) mejoran o empeoran la experiencia de usuario (User eXperience – UX). El cuerpo humano produce ciertos estímulos y reacciones faciales al usar una aplicación software de forma inconsciente que pueden ser detectados a través del análisis de sus expresiones faciales (capturadas por medio de una cámara de video o webcam) en momentos determinados de su interacción.

Actualmente, existen algunas librerías para el análisis de vídeo que permiten categorizar ciertas expresiones faciales con emociones (alegría, enfado, desconcierto, etc.) pero no han sido utilizadas de forma sistemática para evaluar y mejorar IUs. En este Trabajo de Fin de Máster, se plantea el diseño e implantación de una herramienta de soporte para analizar las expresiones faciales de usuarios mientras interactúan con IUs. La imagen de vídeo capturada será analizada mediante diversas técnicas de aprendizaje automático (ej. Transfer Learning o Deep Learning). La herramienta a desarrollar, además de hacer el reconocimiento facial, deberá: i) grabar en vídeo al usuario mientras interactúa con el IU, usando la Webcam del ordenador; ii) capturar en vídeo la pantalla del ordenador; y iii) guardar un log indicando el tiempo y el resultado del análisis de la expresión facial (emoción dominante) con una cierta periodicidad. Estos tres ficheros estarán temporalmente sincronizados y podrán ser analizados con posterioridad para determinar la evolución del estado de ánimo (positiva o negativa) del usuario mientras interactúa con el IU. Para finalizar, se realizará una validación de los modelos utilizados con un conjunto de datos de prueba, una validación de estos mismos modelos en tiempo real con un experimento controlado donde los sujetos observan un vídeo mientras se analizan sus expresiones faciales y, por último, se hará un estudio preliminar de la posible relación de aspectos del IU con cambios en el estado de ánimo del usuario orientado a la mejora de la experiencia de usuario.

Palabras clave: Reconocimiento de expresiones faciales, Análisis de sentimientos, Interfaces de Usuario, Machine Learning

Resum

L'estudi dels canvis en l'estat d'ànim de l'usuari mentre utilitza un sistema programari pot donar indicis de quins aspectes de la Interfície d'Usuari (IU) milloren o empitjoren l'experiència d'usuari (User eXperience – UX). El cos humà produeix uns certs estímuls i reaccions facials en usar una aplicació programari de manera inconscient que poden ser detectats a través de l'anàlisi de les seues expressions facials (capturades per mitjà d'una càmera de vídeo o webcam) en moments determinats de la seua interacció.

Actualment, existeixen algunes llibreries per a l'anàlisi de vídeo que permeten cate-

goritzar unes certes expressions facials amb emocions (alegria, enuig, desconcert, etc.) però no han sigut utilitzades de manera sistemàtica per a avaluar i millorar IUs. En aquest Treball de Fi de Màster, es planteja el disseny i implantació d'una eina de suport per a analitzar les expressions facials d'usuaris mentre interactuen amb IUs. La imatge de vídeo capturada serà analitzada mitjançant diverses tècniques d'aprenentatge automàtic (ex. Transfer Learning o Deep Learning). L'eina a desenvolupar, a més de fer el reconeixement facial, deurà: i) gravar en vídeo a l'usuari mentre interacciona amb l'IU, usant la Webcam de l'ordinador; ii) capturar en vídeo la pantalla de l'ordinador; i iii) guardar un log indicant el temps i el resultat de l'anàlisi de l'expressió facial (emoció dominant) amb una certa periodicitat. Aquests tres fitxers estaran temporalment sincronitzats i podran ser analitzats amb posterioritat per a determinar l'evolució de l'estat d'ànim (positiva o negativa) de l'usuari mentre interacciona amb l'IU. Per a finalitzar, es realitzarà una validació dels models utilitzats amb un conjunt de dades de prova, una validació d'aquests mateixos models en temps real amb un experiment controlat on els subjectes observen un vídeo mentre s'analitzen les seues expressions facials i, finalment, es farà un estudi preliminar de la possible relació d'aspectes de l'IU amb canvis en l'estat d'ànim de l'usuari orientat a la millora de l'experiència d'usuari.

Paraules clau: Reconeixement d'expressions facials, Anàlisi de sentiments, Interfícies d'usuari, Machine Learning

Abstract

The study of changes in the user's state of mind while using a software system can give clues as to which aspects of the User Interface (UI) improve or worsen the User Experience (UX). The human body produces certain stimuli and facial reactions when using a software application unconsciously that can be detected through the analysis of their facial expressions (captured by means of a video camera or webcam) at certain moments of their interaction.

Currently, there are some libraries for video analysis that allow to categorise certain facial expressions with emotions (joy, anger, bewilderment, etc.), but they have not been used systematically to evaluate and improve IUs. In this Master's Thesis, we propose the design and implementation of a support tool for analysing the facial expressions of users while they interact with IUs. The captured video image will be analysed using different machine learning techniques (e.g. Transfer Learning or Deep Learning). The tool to be developed, in addition to performing facial recognition, shall: i) video record the user while interacting with the IU, using the computer's Webcam; ii) video capture the computer screen; and iii) save a log indicating the time and the result of the facial expression analysis (dominant emotion) with a certain periodicity. These three files will be temporally synchronised and can be analysed later to determine the evolution of the user's state of mind (positive or negative) while interacting with the UI. Finally, a validation of the models used will be carried out with a test data set, a validation of these same models in

real time with a controlled experiment where the subjects watch a video while their facial expressions are analysed and, finally, a preliminary study of the possible relationship of aspects of the UI with changes in the user's mood oriented to the improvement of the user experience will be carried out.

Keywords: Facial expression recognition, Sentiment analysis, User Interface, Machine Learning

Agradecimientos

A mis tutores del Trabajo de Fin de Máster Emilio Insfrán y Silvia Abrahão por darme esta oportunidad y todo el apoyo y confianza.

A mis compañeros del laboratorio, por hacer que cada mañana sea más amena, por su apoyo y ayuda durante la realización del trabajo y por convertirse en tan buenas amistades.

A mi familia, por su apoyo incondicional, pero sobretodo a mis padres, por fomentar en mí el espíritu de siempre querer aprender más, por darme su apoyo en todo momento, aconsejarme y guiarme. Por dejarme escoger mi camino en la vida y por ayudarme a ser la persona que soy hoy.

A mis amigos, por acompañarme en mi vida, por estar en los momentos malos apoyándome y disfrutar conmigo de los momentos buenos. Por ser ese pilar tan importante en mi vida y por todo lo bueno que tenemos que vivir juntos todavía. Un recordatorio a Alan David Aguilera, porque aunque ya no estés aquí siempre estás en nuestros recuerdos.

Por último, agradecer a ValgrAI - Valencian Graduate School and Research Network for Artificial Intelligence y a la Generalitat Valenciana por su apoyo económico para mis estudios del master y en consecuencia para la realización de este trabajo.

Índice general

1. Introducción	15
1.1. Motivación	15
1.2. Objetivos	16
1.3. Contexto del trabajo de fin de máster	16
1.4. Metodología	17
1.5. Estructura del documento	19
2. Fundamentos	21
2.1. Reconocimiento de expresiones faciales	21
2.2. Librerías existentes para el reconocimiento de expresiones faciales	22
2.3. Transfer Learning y su aplicación al reconocimiento de expresiones faciales	23
2.4. Deep Learning y su aplicación al reconocimiento de expresiones faciales	24
3. Estado del arte	25
3.1. Sentimientos en el contexto de la Ingeniería de Software	25
3.2. Análisis de expresiones faciales con técnicas de Machine Learning	26
4. Tecnologías para la implementación	29
4.1. TensorFlow	29
4.2. PyTorch	30
4.3. Flask	30
4.4. Electron	31
5. Especificación de requisitos y diseño	33
5.1. Especificación de requisitos	33
5.1.1. Requisitos funcionales	33
5.1.2. Requisitos no funcionales	35
5.2. Arquitectura	36
5.3. Estructura de ficheros	37
6. Implementación de la herramienta	41
6.1. Implementación del análisis de las expresiones faciales	41
6.2. Peticiones de la API	44
6.2.1. Inicio del análisis	44

6.2.2.	Parada del análisis	45
6.2.3.	Resultado del análisis	45
6.2.4.	Estado de la API	45
6.2.5.	Configuración de modelos	46
6.2.6.	Configuración de ventana temporal	46
6.2.7.	Configuración de la grabación de video de la Webcam	46
6.2.8.	Configuración de la grabación de vídeo de la pantalla	47
6.2.9.	Configuración de la grabación de logs	47
6.3.	<i>Frontend</i>	47
6.4.	GitHub con la implementación	51
7.	Validación y pruebas	53
7.1.	Evaluación de modelos ML: Matrices de confusión, <i>precision</i> , <i>recall</i> y F1	53
7.2.	Proceso experimental con vídeos	57
7.2.1.	Motivación	58
7.2.1.1.	Objetivo del experimento	58
7.2.1.2.	Contexto	59
7.2.1.3.	Aspectos éticos y jurídicos	59
7.2.2.	Planificación del experimento	60
7.2.2.1.	Hipótesis y variables	60
7.2.2.2.	Diseño del experimento	61
7.2.2.3.	Selección de sujetos	62
7.2.2.4.	Objetos e instrumentación	62
7.2.2.5.	Evaluación de la validez	62
7.2.3.	Ejecución del experimento	63
7.2.4.	Análisis de datos e interpretación	63
7.2.5.	Conclusiones	67
7.3.	Proceso experimental con interfaces de usuario	67
7.3.1.	Motivación	68
7.3.1.1.	Objetivo del experimento	69
7.3.1.2.	Contexto	69
7.3.1.3.	Aspectos éticos y jurídicos	70
7.3.2.	Planificación del experimento	70
7.3.2.1.	Hipótesis y variables	70
7.3.2.2.	Diseño del experimento	71
7.3.2.3.	Selección de sujetos	72
7.3.2.4.	Objetos e instrumentación	72
7.3.2.5.	Evaluación de la validez	73
7.3.3.	Ejecución del experimento	73
7.3.4.	Análisis de datos e interpretación	74
7.3.5.	Conclusiones	76
8.	Conclusiones y Trabajos Futuros	77
8.1.	Conclusiones	77

8.2. Adaptación de interfaces mediante datos biométricos	78
8.3. Trabajo futuro	80
Bibliografía	81
A. Formulario de consentimiento y cuestionario demográfico	87
B. Interfaces de usuario del proceso experimental	91
C. Objetivos de desarrollo sostenible	97

Índice de figuras

1.1.	Tablero kanban de requisitos	19
1.2.	Descripción de requisitos	19
4.1.	Función de Flask para la configuración de los modelos ML	31
5.1.	Arquitectura de la herramienta	37
5.2.	Fichero <i>logs</i> de la herramienta tras el análisis	37
5.3.	Estructura de ficheros del proyecto	39
6.1.	Arquitectura de MobileNetV2	42
6.2.	Arquitectura de la red neuronal	43
6.3.	Fichero JavaScript con la implementación de Electron	48
6.4.	Mock-up de la página principal de la herramienta	49
6.5.	Mock-up de la página configuración de la herramienta	49
6.6.	Implementación con AnyChart del gráfico de radar - Definición de estructuras de datos	50
6.7.	Implementación con AnyChart del gráfico de radar - Creación del gráfico de radar	50
6.8.	Página de análisis de la herramienta finalizada	51
6.9.	Página de configuración de la herramienta finalizada	51
7.1.	Creación de los vídeos con las emociones mezcladas aleatoriamente	61
7.2.	Interfaz de usuario UI-1 o disposición en rejilla	68
7.3.	Creación de las presentaciones de interfaces mezcladas aleatoriamente	72
8.1.	Estrategia de Adaptación Inteligente de IUs	79
A.1.	Formulario de consentimiento para participar en el proceso experimental	88
A.2.	Cuestionario demográfico previo a la ejecución del proceso experimental	89
B.1.	Interfaz de usuario UI-2 o disposición en lista	92
B.2.	Interfaz de usuario UI-3 o disposición desordenada	92
B.3.	Interfaz de usuario UI-4 o tema claro	93
B.4.	Interfaz de usuario UI-5 o tema oscuro	93
B.5.	Interfaz de usuario UI-6 o tema de alto contraste	94
B.6.	Interfaz de usuario UI-7 o letra normal	94
B.7.	Interfaz de usuario UI-8 o letra grande	95

B.8. Interfaz de usuario UI-9 o letra pequeña 95

Índice de tablas

5.1. Especificación de requisitos funcionales	33
5.2. Especificación de requisitos no funcionales	35
6.1. Plantilla de especificación de peticiones	44
6.2. Especificación de la petición del inicio del análisis	44
6.3. Especificación de la petición del final del análisis	45
6.4. Especificación de la petición de resultados del análisis	45
6.5. Especificación de la petición del estado de la API	45
6.6. Especificación de la petición de configuración de modelos en uso durante el análisis	46
6.7. Especificación de la petición de configuración del modo de ventana temporal	46
6.8. Especificación de la petición de configuración de la grabación de vídeo Webcam	46
6.9. Especificación de la petición de configuración de la grabación de pantalla	47
6.10. Especificación de la petición de configuración de la grabación de <i>logs</i>	47
7.1. Matriz de confusión de DeepFace	55
7.2. Matriz de confusión del modelo basado en Transfer Learning	55
7.3. Matriz de confusión del modelo basado en Deep Learning	55
7.4. Resultados de <i>precision</i> , <i>recall</i> y F1 para DeepFace	56
7.5. Resultados de <i>precision</i> , <i>recall</i> y F1 para el modelo de Transfer Learning	56
7.6. Resultados de <i>precision</i> , <i>recall</i> y F1 para el modelo de Deep Learning	56
7.7. <i>Precision</i> , <i>recall</i> y F1 con emociones positivas y negativas para DeepFace	57
7.8. <i>Precision</i> , <i>recall</i> y F1 con emociones positivas y negativas para Transfer Learning	57
7.9. <i>Precision</i> , <i>recall</i> y F1 con emociones positivas y negativas para Deep Learning	57
7.10. Matriz de confusión en tiempo real para DeepFace	64
7.11. Matriz de confusión en tiempo real para el modelo de Transfer Learning	64
7.12. Matriz de confusión en tiempo real para el modelo de Deep Learning	64
7.13. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para DeepFace	65
7.14. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para el modelo de Transfer Learning	65
7.15. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para el modelo de Deep Learning	66

7.16. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para DeepFace con reacciones positivas y negativas	66
7.17. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para el modelo de Transfer Learning con reacciones positivas y negativas	66
7.18. Resultados en tiempo real de <i>precision</i> , <i>recall</i> y F1 para el modelo de DeepLearning con reacciones positivas y negativas	67
7.19. Tabla de resultados CSUQ por usuario e interfaz	74
7.20. Tabla de emociones dominantes por usuario e interfaz	75
7.21. p-valores del test de normalidad de <i>Shapiro-Wilk</i> por cada interfaz	75
7.22. Valores del test de correlación de <i>Spearman</i> para cada interfaz	76

Capítulo 1

Introducción

Este capítulo presenta la motivación y objetivos del proyecto, además del contexto del trabajo de fin de máster, presentando el trabajo a realizar y el marco en el que está envuelto. Por último, se describe brevemente la estructura de esta memoria.

1.1. Motivación

A la hora de tratar de descubrir los sentimientos de las personas, las expresiones faciales juegan un papel crucial. El cuerpo humano produce ciertas expresiones faciales según las emociones que se despierten en la persona, dando pistas claras sobre como se siente el usuario en un momento determinado. Pensamos que el análisis de las expresiones faciales de las personas en tiempo real puede ser realmente útil en tareas en las que se requiere conocer el estado sentimental del usuario. Un ejemplo de ello podría ser el reconocimiento de expresiones faciales para obtener indicios de que elementos de una interfaz de usuario pueden mejorar o empeorar la experiencia de usuario (User eXperience, UX), así como si al realizar alguna adaptación esta resulta satisfactoria o no para el usuario.

Para lograr este reconocimiento de expresiones faciales se pretenden utilizar librerías y modelos ML (modelos de aprendizaje automático o Machine Learning) con el fin de crear una herramienta capaz de llevar a cabo esta tarea. Esta herramienta será accesible en forma de aplicación de escritorio y en forma de API (permitiendo la conexión con otras herramientas), de forma que tomando la imagen de la webcam como entrada permita el análisis de las expresiones faciales del usuario, proporcionando un desglose de las emociones en tanto por cien.

Además de su cometido principal de reconocimiento de expresiones faciales, nuestra herramienta permitirá la grabación de vídeo de la webcam, la grabación de vídeo de la pantalla del ordenador y la grabación de los resultados en un fichero en tiempo real durante la realización del análisis, de forma que se puedan analizar los datos manualmente al terminar el análisis, y no solo en tiempo real.

1.2. Objetivos

El objetivo general de este trabajo es el diseño e implantación de una herramienta de soporte para analizar expresiones faciales de usuarios mientras interactúan con interfaces de usuario. Para lograr este objetivo, se plantean los siguientes objetivos específicos:

- Analizar las librerías existentes para el reconocimiento de expresiones faciales, las cuales usen modelos ML previamente entrenados.
- Crear, adaptar y entrenar modelos ML para su uso en el ámbito de la detección de expresiones faciales.
- Analizar la literatura para conocer las emociones y/o estados de ánimo que se pueden encontrar dentro del área de *Sentiment Analysis* (análisis de sentimientos) para el análisis de las expresiones faciales y otros ámbitos.
- Desarrollar una herramienta que permita analizar las expresiones faciales de una persona tomando la imagen de la webcam del ordenador, utilizando uno o más modelos ML simultáneamente, que además permita la grabación de los resultados y vídeos de la webcam y la pantalla en tiempo real.
- Validar los modelos ML utilizados y compararlos sobre un conjunto de datos de prueba. Tras esto diseñar un estudio que permita validar y comparar estos modelos ML para su uso en tiempo real con la herramienta.
- Realizar un estudio de cómo el análisis de expresiones faciales nos puede permitir conocer la conformidad o disconformidad del usuario frente a distintas interfaces de usuario.

1.3. Contexto del trabajo de fin de máster

Este trabajo de fin de máster se ha desarrollado en el contexto del grupo de investigación de Ingeniería del Software y Sistemas de Información (ISSI) del Instituto Universitario Mixto Tecnológico de Informática (ITI) de la Universitat Politècnica de València (UPV). En particular, este trabajo ha contribuido a los resultados del Proyecto “*AKI-LA: User Interface Adaptation through User-Experience-based Reinforcement Learning*”, de la convocatoria de Subvenciones para grupos de investigación consolidados (CIAICO/2021/303) financiado por la Generalitat Valenciana y por el Fondo Social Europeo. IP: Silvia Abrahao. De enero del 2022 a diciembre de 2024.

Como su nombre indica el objetivo principal del proyecto es definir un marco de adaptación de interfaces de usuario soportado por técnicas de machine learning, en particular, *Reinforcement Learning (RL)* que permita la adaptación inteligente de interfaces de usuario de forma automática. Para ello, se requiere una retroalimentación en tiempo real por

parte del usuario para guiar y validar las adaptaciones que se van produciendo sobre la interfaz de usuario, como puede ser el análisis de ondas cerebrales (EEG), el seguimiento ocular, etc...

Es en esta parte donde este proyecto cobra importancia, permitiendo analizar las expresiones faciales del usuario en tiempo real, de forma que al realizar una adaptación sobre la interfaz de usuario se podrá comprobar la reacción al instante del usuario, de forma que podremos comprobar si la adaptación resulta útil para el usuario, o si por lo contrario, la adaptación no se adapta a sus gustos o dificulta la tarea que se está llevando a cabo sobre dicha interfaz de usuario.

Además cabe destacar que el autor, Fernando Carceller Llorens, está adscrito a ValgrAI – *Valencian Graduate School and Research Network for Artificial Intelligence*, y que tanto ValgrAI como el proyecto AKILA de la Generalitat Valenciana son responsables del apoyo económico para la realización de este trabajo.

1.4. Metodología

Para lograr los objetivos propuestos, consideramos imprescindible hacer uso de una metodología que vertebré todas las actividades que se necesitan llevar a cabo para conseguirlos. La metodología empleada consiste en una extensión de un modelo para investigación y transferencia de tecnología basado en las necesidades de la industria [7] que incluye actividades de observación, formulación de soluciones y evaluación de resultados. Como resultado de este proceso se obtienen nuevos conocimientos que, en un futuro, se podrían transferir a la industria.

Este modelo de investigación y transferencia de tecnología se basa en ocho actividades relacionadas donde la búsqueda de soluciones adecuadas se realiza de forma iterativa por medio de la formulación de soluciones candidatas y la correspondiente validación empírica (primero internamente con participantes reclutados para participar en estudios empíricos, una vez refinada la solución, en un contexto colaborativo con las empresas). Este proceso en dos etapas, permite dirigir los esfuerzos hacia una solución realista sin comprometer las expectativas y la inversión del tiempo de las empresas.

Las actividades a realizar se describen a continuación:

1. **Análisis del problema:** Se pretende entender un problema tecnológico que se pretende resolver. Este paso es llevado a cabo a través de reuniones y *workshops* donde participan el investigador, sus directores de tesis de máster y expertos del dominio que hacen presentaciones acerca de los retos, necesidades y problemas específicos que ellos perciben en cuanto a al análisis de expresiones faciales.
2. **Formulación del problema:** Una vez que el problema está identificado, éste es formulado de manera más precisa y los factores contextuales y las asunciones de trabajo, son especificadas claramente.

3. **Revisión del estado del arte:** Se realiza una revisión de la literatura, así como de las soluciones tecnológicas comerciales y *open source* disponibles para identificar hasta qué punto los objetivos específicos han sido abordados y cuáles son los problemas abiertos a resolver con la investigación a desarrollar.
4. **Solución candidata:** Se idea y especifica una o más soluciones potenciales y se desarrollan herramientas de soporte. Dichas soluciones serán más tarde depuradas a través de las distintas actividades de refinamiento (actividades 5 y 6).
5. **Validación inicial:** Se lleva a cabo una evaluación preliminar de las soluciones, en un entorno de laboratorio, mediante pruebas y experimentos controlados con un grupo de usuarios potenciales.
6. **Validación realista:** Se llevan a cabo casos de estudio en entornos industriales, empezando por estudios piloto para después extenderse en usos más amplios. Se definen guías prácticas.
7. **Liberación de la solución:** En este último paso se valoran los resultados obtenidos y las herramientas y se preparan material de entrenamiento para su uso en contextos industriales.

En el marco de este proyecto de fin de máster el objetivo es abordar las actividades 1 a 5 de este método y posteriormente, dependiendo de los resultados obtenidos, contactar con empresas que estén interesadas en los resultados del proyecto para plantear la posibilidad de realizar las actividades 6 y 7.

Además, para la realización de la implementación de la herramienta se ha utilizado una metodología similar a MVP donde se buscaba un producto mínimo viable, de forma que la primera versión tuviera las funcionalidades más importantes y la herramienta fuera funcional desde el principio. A partir de esta primera versión se va ampliando poco a poco con nuevas funcionalidades más complementarias hasta conseguir un producto final.

Para ello se ha implementado un tablero *kanban* en un proyecto de GitHub con las secciones *TO DO/IN PROGRESS/DONE* (Pendiente, en progreso e implementado respectivamente) donde se encuentran todos los requisitos que se pretenden implementar en la aplicación (identificados con su id RQXXX) en la parte de *TO DO* y que conforme se vayan implementando, por orden de importancia, irán pasando de *TO DO* a la sección *IN PROGRESS* cuando se estén implementando y a *DONE* una vez las funcionalidades estén implementadas. Dentro de cada tarjeta de requisitos podemos encontrar una descripción de este (Ver Figuras 1.1 y 1.2).

Por último como método de validación se hacen reuniones periódicas donde se reorganizan y añaden requisitos para implementar en la herramienta y se observa el trabajo realizado hasta el momento para ver si hay que realizar cambios o arreglar algún error. Estas reuniones periódicas se realizan al menos una vez a la semana.

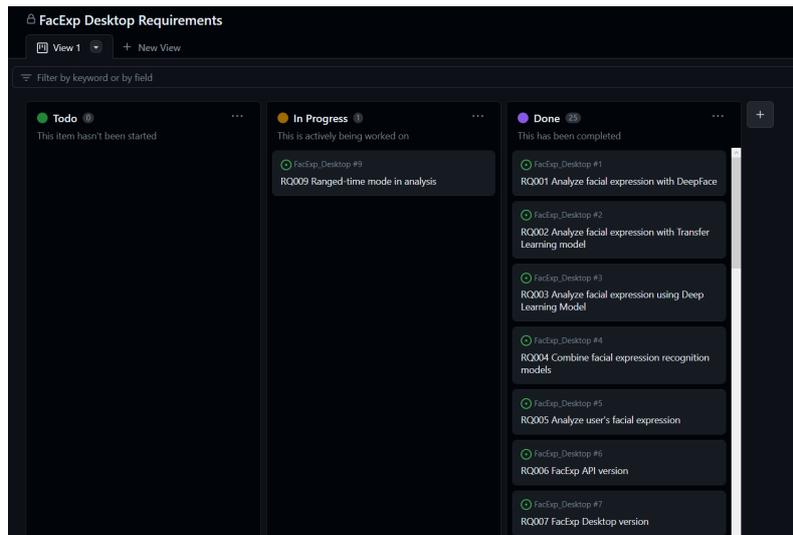


Figura 1.1: Tablero kanban de requisitos

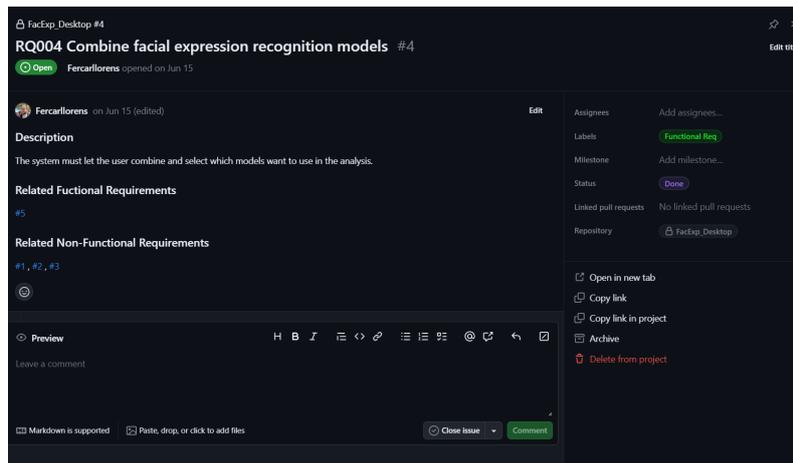


Figura 1.2: Descripción de requisitos

1.5. Estructura del documento

Los capítulos que se listan a continuación serán los que conformarán la memoria y por tanto los que documentarán todo el proceso de desarrollo del trabajo:

- **Capítulo 2. Fundamentos:** En este capítulo se entrará en detalle sobre los fundamentos del reconocimiento de expresiones faciales así como algunas de las librerías y técnicas que se pueden utilizar para este cometido.
- **Capítulo 3. Estado del arte:** En este capítulo se realiza una revisión de la literatura sobre los sentimientos dentro del contexto de la ingeniería de software y el análisis de expresiones faciales.

- **Capítulo 4. Tecnologías para la implementación:** En este capítulo se presentan las tecnologías que se utilizarán para la implementación de los modelos ML y la herramienta de análisis de expresiones faciales.
- **Capítulo 5. Especificación de requisitos y diseño:** Como el nombre del capítulo indica, en el se especificarán los requisitos funcionales y no funcionales de la herramienta y se hablará de su diseño
- **Capítulo 6. Implementación de la herramienta:** En este capítulo se hablara de la implementación tanto de los modelos ML dedicados al análisis de expresiones faciales como de la herramienta
- **Capítulo 7. Validación y pruebas:** En este capítulo se muestra la validación de los modelos ML y de la herramienta.
- **Capítulo 8. Conclusiones y trabajo futuro:** En este capítulo se presentaran las conclusiones del trabajo realizado así como algunas de las tareas que quedan pendientes tras la finalización de este trabajo.
- **Anexos:** Aquí el lector puede encontrar información complementaria no incluida en los capítulos anteriores

Capítulo 2

Fundamentos

Este capítulo tiene como objetivo presentar los principales fundamentos que envuelven al proyecto, así como poner al lector en contexto respecto a estos temas. Durante el capítulo se tratarán principalmente los conceptos de reconocimiento de expresiones faciales y como conseguirlo mediante librerías o técnicas de aprendizaje automático.

2.1. Reconocimiento de expresiones faciales

El reconocimiento o análisis de expresiones faciales consiste, como su propio nombre indica, en la detección y seguimiento de las expresiones faciales de las personas, con el fin de poder detectar sus sentimientos y emociones. Para realizar este análisis se suelen utilizar técnicas de clasificación basadas en aprendizaje automático.

El funcionamiento del análisis de expresiones faciales comienza por la detección del rostro (presencia y localización) de la persona a la que se desea analizar. Seguidamente se extraen una serie de puntos característicos del rostro para poder observar su disposición en el instante de la captura de la imagen y por último se extraen las emociones (en análisis más profundos también se pueden llegar a detectar micro-expresiones y métricas comportamentales)

El reconocimiento de expresiones faciales se puede realizar sobre imágenes o vídeos previamente tomados, lo que nos permite analizar las expresiones faciales de las personas en instantes concretos del pasado, pudiendo conocer sus sentimientos en el momento exacto de la captura de la imagen o vídeo. Pero una de las aplicaciones que puede resultar más interesante es el análisis de expresiones faciales en tiempo real, permitiendo conocer los sentimientos de la persona en el mismo instante de su grabación.

Este reconocimiento de expresiones faciales tiene gran importancia a la hora de intentar deducir las emociones de las personas en instantes concretos, cosa que puede ser de ayuda para gran cantidad de finalidades (entrevistas o interrogatorios, soporte a la psicología, validación para ciertos procesos empíricos, etc...).

En una etapa de la ingeniería de software donde se intenta acercar cada día más ésta a la inteligencia artificial, no es raro observar como el reconocimiento de expresiones faciales basado en técnicas de aprendizaje automático es un tema que cada vez se utiliza más en nuestro ámbito, ya sea para obtener datos para la validación de ciertos aspectos como la usabilidad y la experiencia de usuario, o directamente para tareas como la que se pretende utilizar esta herramienta como es la adaptación de interfaces inteligentes, donde los datos del reconocimiento de expresiones faciales se utilizarán como retroalimentación para las adaptaciones realizadas automáticamente por el sistema. Aunque nuestra aplicación no solo sera utilizada para este cometido, sino que incluirá una interfaz que permita el uso del reconocimiento de expresiones faciales para cualquier ámbito.

2.2. Librerías existentes para el reconocimiento de expresiones faciales

Si bien existen aplicaciones ya enfocadas al análisis de expresiones faciales, nosotros queremos crear nuestra propia aplicación, de forma que se pueda utilizar en forma de API por el sistema de adaptación de interfaces inteligente y de forma convencional en un formato de aplicación de escritorio.

Debido a esta necesidad de crear nuestra propia aplicación, el primer paso lógico es buscar librerías que nos ofrezcan la detección de expresiones faciales, de forma que estas se puedan incluir en nuestra propia aplicación, evitando además la complejidad de crear un modelo ML propio.

Tras explorar el catálogo de librerías de reconocimiento de expresiones faciales que nos ofrece Python pudimos destacar tres que daban un excelente rendimiento: Py-Feat, FER, y DeepFace.

Py-Feat es una librería de análisis de expresiones faciales desarrollada por CoSANLab (Computational Social Affective Neuroscience Laboratory) que pretende proveer de un conjunto de herramientas comprensibles y fáciles de utilizar para el análisis de expresiones faciales en imágenes y vídeos. Esta herramienta resultaba bastante precisa a la hora del análisis de expresiones faciales, pero su principal problema era que no permitía el análisis en tiempo real, lo que era un requisito fundamental a la hora de encontrar una librería para nuestro proyecto, por lo que tuvimos que descartar su uso.

FER es una librería desarrollada por Justin Shenk, fundador de VisioLab y desarrollador de varias librerías centradas en temas como la inteligencia artificial, el aprendizaje automático y el análisis de emociones. Pretende ofrecer una herramienta de análisis de expresiones faciales que actúe sobre vídeo e imágenes. Si bien esta herramienta si que nos permitía la posibilidad de realizar un análisis de expresiones faciales en tiempo real, la frecuencia de muestreo era muy baja, pudiendo analizar una o como mucho dos veces por segundo las expresiones faciales en tiempo real. Por tanto, a pesar de sus buenos resultados en el análisis sobre imagen y vídeo, el bajo rendimiento a la hora de un análisis en

tiempo real hizo que descartáramos el uso de esta librería.

DeepFace es la librería más famosa de Python para el reconocimiento de rostros y análisis de sus rasgos, entre los que se encuentra el análisis de expresiones faciales. Esta librería nace como una recopilación del estado del arte del reconocimiento de rostros, centrándose en DeepFace, un modelo de Deep Learning para el reconocimiento de rostros desarrollado por un pequeño equipo en Facebook [27]. Además se añadirían las funcionalidades del reconocimiento de rasgos de los rostros como pueden ser raza, edad, género o expresión facial entre otras [23]. Esta librería presenta buenos resultados en la tarea de reconocimiento de expresiones faciales además de permitirnos su uso en tiempo real con una frecuencia de muestreo adecuada, cumpliendo con los requisitos principales de nuestra aplicación.

Una vez exploradas las librerías para cumplir con la tarea del reconocimiento de expresiones faciales, como podemos observar, solo hemos obtenido una sola librería que nos ofrece buenos resultados en tiempo real. Como la intención era obtener varias librerías para ver cual de todas nos ofrecía mejores resultados, se decide crear un par de modelos ML basándonos en artículos que se han encontrado en la literatura que utilizan técnicas de aprendizaje automático como modelos basados en Transfer Learning [20] y Deep Learning [18]

2.3. Transfer Learning y su aplicación al reconocimiento de expresiones faciales

El Transfer Learning o aprendizaje por transferencia es una técnica de aprendizaje automático que trata de utilizar una red neuronal ya existente, pensada para un tipo de tarea, realizando las adaptaciones necesarias sobre esta hasta conseguir adaptarla a otro tipo de tarea [28]. De forma que una red neuronal ya existente y que de buenos resultados para un tipo de tarea puede modificarse y re-entrenarse de forma que pueda utilizarse en otro tipo de de tareas sin perder la fiabilidad de la red neuronal original. Es una aplicación del aprendizaje automático muy interesante sobre todo por el ahorro de tiempo en el diseño de una red neuronal desde cero para cada tipo de tarea, agilizando el proceso de creación de modelos basados en redes neuronales.

Realmente el uso de Transfer Learning puede resultar útil en tareas de creación de modelos de comprensión de lenguaje natural y de *computer vision* (tareas de reconocimiento de rostros, gestos, rasgos, etc...). En el ámbito del reconocimiento de expresiones faciales, el cual entraría dentro del grupo de tareas de *computer vision*, el Transfer Learning resulta realmente útil, ya que en el trasfondo la red neuronal que se pretende crear para esta tarea es similar a una red neuronal de clasificación de imágenes. Es por ello que utilizando alguna red neuronal para clasificación de imágenes que ofrezca algún *framework* de aprendizaje automático como TensorFlow o PyTorch se pueden realizar algunas modificaciones y re-entrenar la red neuronal para convertir una red neuronal de clasificación de imágenes genérico en un modelo para detección de expresiones faciales.

Explorando en la literatura se pueden encontrar ejemplos del uso del Transfer Learning en el ámbito del reconocimiento de las expresiones faciales, como el artículo de Aicha Nouisser *et al.* [20], donde utilizando una red neuronal propia de TensorFlow, *MobileNet*, consigue modificar su salida para reducirla a 7 clases y reentrenar el modelo para obtener un modelo de reconocimiento de expresiones faciales.

2.4. Deep Learning y su aplicación al reconocimiento de expresiones faciales

El Deep Learning o aprendizaje profundo es un acercamiento del aprendizaje automático que trata de crear estructuras lógicas que se asemejen lo máximo posible a nuestro sistema nervioso, formando capas de unidades de proceso (neuronas artificiales) especializadas en detectar y determinar características existentes en los objetos percibidos [13]. En otras palabras, Deep Learning envuelve a todas las técnicas de creación de redes neuronales para una gran cantidad de tareas. Incluso se podría discutir si el Transfer Learning es parte del Deep Learning al utilizar también redes neuronales, aunque su proceso no sea la creación desde cero de la red neuronal, solo la modificación y adaptación a otras tareas de redes existentes.

En la literatura se pueden encontrar diversos artículos que respaldan el uso de Deep Learning para la tarea del reconocimiento de expresiones faciales. Pero dentro de estas resulta realmente interesante el uso de redes neuronales convolucionales con módulos de atención (ACNN, *Attentional Convolutional Neural Networks*) como en el artículo de Sherwin Minaee *et al.* [18], donde se utilizan este tipo de redes neuronales para el análisis de expresiones faciales. El uso de este tipo de redes neuronales se respalda en que el módulo de atención puede ayudar a extraer las características más importantes o que más resaltan de las imágenes, pudiendo centrarse en ellas para el reconocimiento de las expresiones faciales.

Capítulo 3

Estado del arte

Durante este capítulo se repasarán brevemente algunos artículos de la literatura relacionados con el trabajo, más concretamente trabajos relacionados con el análisis de sentimientos utilizando todo tipo de técnicas, permitiendo el análisis de sentimientos sobre texto, audio e imágenes.

3.1. Sentimientos en el contexto de la Ingeniería de Software

Dentro de los campos de computación e ingeniería de software podemos encontrar diversas formas de tratar de descubrir los sentimientos y emociones de los usuarios. Una de las más comunes es el análisis de sentimientos sobre texto, aunque también podemos encontrar análisis de sentimiento para audio e imagen.

En primera instancia el análisis de sentimientos sobre texto se realizaba con un vocabulario ponderado, esto es, un vocabulario donde cada una de las palabras tiene un valor adjunto dependiendo de lo negativa o positiva que sea la palabra. Uno de estos vocabularios ponderados puede ser AFINN, una herramienta de análisis de sentimientos utilizada en muchos trabajos de este tipo [17]. Esta herramienta posee 2477 palabras ponderadas (en inglés) cuyo valor adjunto va entre -5 y +5, siendo -5 el valor más negativo que puede poseer una palabra y +5 el más positivo que puede tener.

Otras formas de análisis de sentimiento sobre texto utilizan técnicas de *Machine Learning* para este tipo de tareas. En el artículo de Saif M. Mohammad [19] se comentan diversas aproximaciones del análisis de sentimientos basadas en técnicas de *Machine Learning* para detectar las emociones y su valencia dentro de texto. Algunos ejemplos son análisis de sentimientos basados en reglas de decisión y otros como el entrenamiento de modelos de *Machine Learning* utilizando datos etiquetados.

En el campo del análisis de sentimientos para audio, una de las formas más lógicas de realizarlo es el uso de técnicas para transformar el audio a texto, de forma que se puedan

analizar estos sentimientos tratándolos como un análisis de sentimiento sobre texto. El artículo de Lakshmish Kaushik *Et al.* [11] nos muestra como se puede realizar un cambio de audio a texto y tras esto realizar un análisis de sentimiento sobre el texto obtenido.

Por otra parte, el análisis de sentimiento basado en audio no debe de estar siempre basado en la conversión de texto. Existen formas de reconocer emociones dentro del audio basándose en el tono, la energía y el volumen, entre otras cosas. Esto es exactamente lo que se realiza en el artículo de Monil Bansal *Et al.* [2], creando un reconocimiento de expresiones multimodal basado en estas características, y por tanto, creando una herramienta de análisis de sentimiento sobre audio completamente independiente del lenguaje.

3.2. Análisis de expresiones faciales con técnicas de Machine Learning

En el caso de que quisiéramos implementar una herramienta de análisis de expresiones faciales ya hemos comentado en el capítulo anterior la posibilidad de usar librerías ya existentes, como DeepFace [23] o de crear nuestros propio modelos basados en Deep Learning [18] y Transfer Learning [20]. Aunque estas no son las únicas opciones.

Revisando en la literatura podemos encontrar infinidad de trabajos relacionados con el reconocimiento de expresiones faciales. Por ejemplo, en el artículo de Yahia Said *Et al.* [21] podemos encontrar un acercamiento similar al nuestro utilizando redes neuronales convolucionales, pero de mayor complejidad, logrando una herramienta más potente, capaz de detectar las expresiones faciales de todas las personas que aparecen en imágenes de alta resolución. Algo similar se presenta en el artículo de Kim *Et al.* [12] donde utilizando CRNN (*Convolutional Recurrent Neural Networks* o Redes Neuronales Convolucionales Recurrentes) se crea un modelo ML capaz de clasificar las emociones de los usuarios mediante expresiones faciales. También en Ma *et al.* se presenta un modelo, esta vez basado en Deep Convolutional Neural Networks (Redes Neuronales Convolucionales Profundas), que permite el reconocimiento de expresiones faciales en tiempo real y con una tasa del 98,96% de acierto en la clasificación.

En cuanto a herramientas encontramos el trabajo de Huang *et al.* [10] que propone un prototipo de herramienta basado en una aplicación de chat. Este prototipo posee una gran variedad de formas de reconocer los sentimientos, siendo una de ellas el reconocimiento de expresiones faciales. Por otra parte también encontramos el trabajo de Buitelaar *et al.* [3] que presenta una *Toolbox* de código abierto para el análisis multimodal de sentimientos, presentando el análisis de sentimiento por texto audio y *linked data*, además del reconocimiento de expresiones faciales.

Aunque no siempre el reconocimiento de expresiones faciales se centra en el sentimiento general que muestra el rostro. En el trabajo de Monu Verma *Et al.* [29] podemos observar como gracias a la implementación una red neuronal mixta se consiguen captar las micro-expresiones del rostro de las personas durante el análisis de expresiones facia-

les, las cuales pueden mostrarnos emociones que puedan estar eclipsadas por la expresión facial general del rostro.

Hasta el momento encontramos herramientas que podrían ser captadas por webcam, pero en la literatura podemos encontrar trabajos sobre el análisis de expresiones faciales utilizando otros dispositivos. Es el caso de el trabajo de Wei Wei *Et al.* [30], en el que encontramos una herramienta de reconocimiento de expresiones faciales que utiliza el sensor sensible a la profundidad del Kinect para poder analizar el rostro de las personas como un objeto tridimensional.

Por último comentar que este tipo de análisis de sentimiento es realmente útil para experimentos o estudios, debido a que el reconocimiento de expresiones en tiempo real puede darnos información al momento de los sentimientos del usuario. En el caso de Shinyama *et al.* [25] se utiliza el reconocimiento de expresiones faciales para grabar a un grupo de programadores durante una sesión de programación, de forma que se puede llevar un seguimiento de las emociones de cada uno de los programadores.

Capítulo 4

Tecnologías para la implementación

En este capítulo se pretenden exponer las tecnologías que se han utilizado para el desarrollo, tanto de los modelos de reconocimiento de expresiones faciales, como de la herramienta, formada por la API y el *frontend*.

Los modelos basados en Transfer Learning y Deep Learning se han decidido implementar en Python, más concretamente, el modelo basado en Transfer Learning se ha decidido implementar con la librería TensorFlow y el modelo basado en Deep Learning se ha decidido implementar con la librería PyTorch.

Por otra parte, para la implementación de la herramienta se ha optado también por Python como lenguaje de desarrollo de la API, con la librería Flask, y Electron para el *frontend*, un *framework* que nos permite implementar aplicaciones de escritorio mediante HTML, CSS y JavaScript, como si de una aplicación web se tratase.

4.1. TensorFlow

TensorFlow es una librería de código abierto en Python para el aprendizaje automático desarrollada por Google. Esta librería es muy utilizada en el ámbito del aprendizaje automático. Realmente TensorFlow es la apuesta de Google para intentar crear un ecosistema dedicado al aprendizaje automático, y que, con sus últimas versiones (TensorFlow 2.0), se está intentando ampliar a un mayor número de usuarios intentando simplificar el uso del *framework*.

Esta ampliación del ecosistema no se queda simplemente en simplificar el *framework* para que usuarios menos expertos puedan utilizarlo, sino que también se ha ampliado a nuevos lenguajes como son JavaScript o Swift.

Además, con el fin de acelerar los procesos de entrenamiento y aprovechando el potencial de las GPUs que hay a disposición actualmente en el mercado (especialmente las GPUs NVIDIA), TensorFlow dispone de la posibilidad de utilizar nuestra GPU en caso de que tengamos una en nuestro equipo, de forma que, aprovechando los tensores de la GPU

se consigue una aceleración de hasta un 50% (según el modelo y especificaciones).

En nuestro caso nos interesa por la amplia variedad de redes neuronales existentes que posee la propia librería, pudiendo elegir entre ellas para la creación de nuestro modelo ML de Transfer Learning. Además nos ofrece muchas facilidades a la hora de implementar este modelo ML, no solo a la hora de elegir una red neuronal de base entre la gran variedad de redes neuronales que ofrece, sino que también a la hora de modificar esta red neuronal para nuestro caso.

4.2. PyTorch

PyTorch es una librería de código abierto para Python (aunque también tiene una versión para C++) enfocada en el aprendizaje automático.

Una subdivisión de Facebook, FAIR (*Facebook's AI Research lab*), fue la desarrolladora de esta librería que hoy en día es utilizada en variedad de proyectos de aprendizaje automático y Deep Learning por otras empresas como Tesla o Uber.

Al igual que Tensorflow nos permitía utilizar nuestra GPU para acelerar los procesos de entrenamiento de los modelos implementados, PyTorch también lo permite con su clase Tensor, que permite estructurar los datos de forma que sean operables por una GPU de NVIDIA compatible con CUDA.

En nuestro caso resulta interesante debido a la facilidad que posee esta herramienta a la hora de crear una red neuronal desde cero. Es realmente sencillo definir la estructura de la red neuronal con las herramientas que PyTorch proporciona, permitiéndonos crear redes neuronales complejas en unas pocas líneas de código.

4.3. Flask

Para la implementación de la API para el reconocimiento de expresiones faciales en tiempo real hemos optado por utilizar Flask, una librería de Python para la creación de APIs REST.

Si bien existen otras opciones, Flask resulta interesante porque esta pensada para APIs pequeñas, de pocos recursos y funciones. Es por ello que Flask resulta conveniente para la implementación de nuestra API, puesto que las únicas funciones serán la configuración de la aplicación y el análisis, arrancar y parar dicho análisis y conocer el estado y configuración de la Aplicación.

Flask nos permite crear APIs de forma muy sencilla, solo debemos de marcar la función que queramos que se ejecute con cada petición con la ruta de la petición, el tipo de petición que debe atender y los parámetros del cuerpo o URI si fuera necesario (Ver Figura 4.1). De esta manera al ejecutar la aplicación y lanzarle peticiones, estas se atenderán con las funciones marcadas con estos parámetros siempre y cuando la petición sea correcta

y se refiera a alguna de las direcciones con las que se han marcado las funciones. Estas peticiones, además de un código, devolverán a la aplicación que ha mandado la petición una respuesta en formato JSON que contendrá la información que se pretendía obtener con la petición (con las peticiones tipo GET) o información sobre los cambios en la API (con las peticiones PUT y POST) entre otras cosas.

```
@app.route('/models', methods=['POST'])
def updateModels():
    request_data = request.get_json()

    if 'DeepFace' not in request_data or 'Transfer' not in request_data or 'Deep' not in request_data:
        return jsonify({
            "Error": "Incorrect body data"
        })

    if not isinstance(status_memory['DeepFace'],bool) and not isinstance(status_memory['Transfer'],bool) and not isinstance(status_memory['Deep'],bool):
        return jsonify({
            "Error": "Incorrect value types"
        })

    deepFaceNewStatus = request_data['DeepFace']
    transferNewStatus = request_data['Transfer']
    deepLearningNewStatus = request_data['Deep']

    status_memory['DeepFace'] = deepFaceNewStatus
    status_memory['Transfer Learning'] = transferNewStatus
    status_memory['Deep Learning'] = deepLearningNewStatus

    f = open("Backend//Status//config.json", 'w+')
    f.write(json.dumps(status_memory))
    f.close()

    return status_memory
```

Figura 4.1: Función de Flask para la configuración de los modelos ML

Como podemos ver, Flask es una opción sencilla para la implementación de APIs y que nos permite no solo crearlas de forma sencilla, sino que también de forma compacta. En nuestro caso la API simplemente ocupa un fichero Python.

4.4. Electron

Para la creación de la aplicación de escritorio que permitirá el análisis de expresiones faciales en directo, mostrando la cámara, un gráfico de radar con las emociones que presenta el usuario en cada momento y los logs de los distintos modelos que se utilizan durante el análisis, se ha decidido utilizar Electron.

Electron es un *framework* para el desarrollo de aplicaciones de escritorio basada en Node.js. Su principal característica es que permite desarrollar el *frontend* como si de una aplicación web se tratase. De esta manera podemos utilizar HTML, CSS y JavaScript para la creación de nuestras aplicaciones de escritorio, pero no solo eso, sino que también tiene compatibilidad con todo tipo de *frameworks* para la implementación del *frontend* de aplicaciones web como Vue, React o Angular, por ejemplo.

Esto nos permite crear aplicaciones de escritorio de forma mas sencilla utilizando nuestro *framework* favorito para el *frontend* o simplemente HTML, CSS y JavaScript, opción que hemos decidido utilizar nosotros debido a que nuestra aplicación de escritorio es sencilla y que se puede implementar en una sola ventana, sin necesidad de realizar cambios de

página y donde toda la lógica de éste se puede realizar mediante JavaScript sin que pueda suponer un problema o suponer dificultad añadida a la implementación.

Gracias a JavaScript, no es necesario crear un *backend* específico para nuestra aplicación, puesto que nuestra API se puede interconectar con el *frontend* de nuestra aplicación de escritorio en Electron de manera sencilla, utilizando de esta manera la API como *backend* de la aplicación de escritorio. La función *fetch* de JavaScript nos permitirá lanzar peticiones a la API y manejar las respuestas posteriores, permitiendo esta conexión API-*frontend*.

Capítulo 5

Especificación de requisitos y diseño

Este capítulo se utilizará con el fin de presentar el diseño de la API y la aplicación de escritorio además de definir los requisitos funcionales y no funcionales del proyecto, aspectos que deben estar claros a la hora de la implementación. La definición de los requisitos sirve a su vez como una definición de que debe hacer y como esperamos que se comporte el sistema una vez implementado, por tanto será también un resumen de las funcionalidades y características de nuestro sistema.

5.1. Especificación de requisitos

Los requisitos, como ya he mencionado, se presentarán en dos listados diferentes. En primer lugar un listado para requisitos funcionales, estos son, requisitos que declaran los servicios que prestará el sistema, es decir, lo que debe hacer el sistema, su funcionalidad. Tras este listado, otro similar para los requisitos no funcionales, estos son, requisitos que no se refieren directamente a las funciones específicas sino a las propiedades o características del sistema, es decir, no hablan de lo que hace el sistema, sino de como lo hace.

5.1.1. Requisitos funcionales

Tabla 5.1: Especificación de requisitos funcionales

ID	Nombre	Descripción
RQ004	Combinar los modelos de reconocimiento de expresiones faciales	El sistema deberá permitir seleccionar y combinar que modelos queremos que se utilicen durante el análisis de expresiones faciales

RQ005	Analizar la expresión facial del usuario	El sistema deberá analizar la expresión facial del usuario utilizando modelos de reconocimiento de expresiones faciales
RQ009	Modo de ventana temporal en análisis	El sistema deberá permitir utilizar un modo de ventana temporal para el análisis de expresiones faciales. Podremos configurar el rango de tiempo que queramos. Este rango será dividido en dos partes, una parte antes de la petición y otra después de la petición de resultados a la API. El sistema calculará el promedio de resultados de esta ventana temporal para dar respuesta a la petición
RQ010	Análisis en tiempo real	El sistema deberá permitir utilizar un modo en tiempo real para el análisis. En este modo se devolverá el resultado exacto del análisis de expresiones faciales en el instante de la petición
RQ011	Mostrar webcam en tiempo real	El sistema deberá mostrar la imagen en vivo proveniente de la webcam en la versión de escritorio
RQ012	Grabar las imágenes webcam	El sistema deberá permitir grabar un vídeo con las imágenes captadas por webcam durante el análisis
RQ013	Cambiar carpeta y nombre del video de la webcam	El sistema deberá permitir cambiar la carpeta destino y el nombre del vídeo grabado de la webcam
RQ014	Habilitar/deshabilitar grabación webcam	El sistema deberá permitir habilitar o deshabilitar la grabación webcam
RQ015	Mostrar <i>logs</i> en tiempo real	El sistema deberá mostrar los <i>logs</i> del análisis para cada uno de los modelos utilizados en tiempo real al utilizar la aplicación de escritorio
RQ016	Grabar <i>logs</i>	El sistema debe permitir grabar los <i>logs</i> durante el análisis
RQ017	Cambiar carpeta y nombre de los <i>logs</i>	El sistema debe permitir cambiar la carpeta destino y el nombre del fichero de la grabación de los <i>logs</i>
RQ018	Habilitar/deshabilitar la grabación de <i>logs</i>	El sistema debe permitir habilitar o deshabilitar la grabación de <i>logs</i>

RQ019	Grabar pantalla	El sistema debe permitir grabar la pantalla en vídeo durante el análisis
RQ020	Cambiar la carpeta y el nombre del archivo de la grabación de pantalla	El sistema debe permitir cambiar la carpeta destino y el nombre del vídeo de la grabación de pantalla
RQ021	Habilitar/deshabilitar la grabación de pantalla	El sistema debe permitir habilitar o deshabilitar la grabación de pantalla
RQ022	Navegación en la aplicación de escritorio	La aplicación de escritorio debe permitir navegar por dos ventanas, una dedicada al análisis y otra a la configuración. La ventana de análisis debe contener el vídeo de la webcam y los <i>logs</i> . La ventana de configuración debe contener todos los parámetros sobre: modelos, grabaciones...
RQ023	Estado de la API en la aplicación de escritorio	El sistema deberá mostrar en el apartado ' <i>Settings</i> ' de la aplicación de escritorio el estado actual de la API (modelos en uso, conexión, versión de la API, carpetas destino de las grabaciones...)
RQ024	Marcas temporales en los logs	El sistema deberá permitir añadir marcas temporales en los logs desde la aplicación de escritorio
RQ025	Cambiar webcam	El sistema deberá permitir cambiar la webcam para el análisis y la grabación
RQ026	Cambiar pantalla a grabar	El sistema deberá permitir cambiar la pantalla que se quiere grabar durante el análisis

5.1.2. Requisitos no funcionales

Tabla 5.2: Especificación de requisitos no funcionales

ID	Nombre	Descripción
RQ001	Analizar expresiones faciales con DeepFace	El sistema deberá utilizar DeepFace como modelo de análisis de expresiones faciales
RQ002	Analizar expresiones faciales con el modelo de Transfer Learning	El sistema deberá utilizar nuestro modelo de Transfer Learning como modelo de análisis de expresiones faciales

RQ003	Analizar expresiones faciales con el modelo de Deep Learning	El sistema deberá utilizar nuestro modelo de Deep Learning como modelo de análisis de expresiones faciales
RQ006	Versión API del sistema	El sistema deberá poder utilizarse en formato API
RQ007	Versión aplicación de escritorio del sistema	El sistema deberá poder utilizarse en formato aplicación de escritorio
RQ008	API como backend	El sistema deberá utilizar su versión API como <i>backend</i> para la versión de aplicación de escritorio. La configuración de la API afectará a la configuración de la aplicación de escritorio y viceversa

5.2. Arquitectura

Como hemos mencionado anteriormente, la herramienta estará formada por una API implementada en Flask (librería de Python) donde se alojarán los modelos de reconocimiento de expresiones faciales, DeepFace, Transfer Learning y Deep Learning, todas basadas en un modelo de emociones en común, y el algoritmo que hace uso de ellos para el análisis del usuario en tiempo real. Esta API se lanzará en local (en la propia máquina), debido a que debe de poder acceder a la imagen de la webcam en todo momento para poder analizar las expresiones faciales del usuario. La API facilitará los datos del análisis siempre que se lance una petición, además de poder arrancar y parar el análisis y configurar los parámetros del análisis como la ventana temporal del análisis o los modelos a utilizar, entre otras cosas.

Para facilitar el uso manual de esta herramienta con el propósito de comparar modelos en tiempo real o configurar la API de una forma más sencilla y amigable con el usuario, se decide implementar un *frontend* que se conectará con la mencionada API. Este *frontend* se presentará en forma de aplicación de escritorio y estará implementada con Electron. Esta aplicación tiene como objetivo simplificar el uso y configuración de la API, así como observar en tiempo real el resultado del análisis de expresiones faciales, mostrándonos a su vez la webcam del usuario y el desglose del análisis de expresiones faciales para cada uno de los modelos que se están utilizando para el análisis en ese momento.

La principal ventaja que presenta esta arquitectura que hemos elegido es la facilidad que ofrece para conectar la herramienta a otras aplicaciones, herramientas, programas, etc... Además al estar funcionando la propia API como *backend* de la aplicación de escritorio evitamos repetir código además de mantener el mismo estado de configuración en la API y la aplicación de escritorio (Ver Figura 5.1). De esta forma no es necesario configurar la API por peticiones, si no que se puede utilizar la propia aplicación de escritorio como un panel de configuración para la API.



Figura 5.1: Arquitectura de la herramienta

La herramienta guardará al terminar el análisis un fichero con los datos del análisis o *logs*. Cada fila del fichero será un instante del análisis de expresiones realizado, donde aparecerán el tiempo en formato Horas:Minutos:Segundos, la expresión dominante por cada uno de los modelos utilizados, y el desglose de emociones para cada uno de los modelos en ese instante concreto (Ver Figura 5.2)

id	HMS	task	dominantDF	dominantTL	dominantDL	neutralDF	happyDF	surpriseDF	disgustDF	fearDF	sadDF	angryDF	neutralTL	happyTL	surpriseTL	disgustTL	fearTL	sadTL	angryTL	neut
3	0:00:03	0	angry	angry	happy	0	0	0	0	0	0	100	0,5547	0,2499	0,2371	4,4114	14,486	3,0972	76,9636	
4	0:00:03	0	sad	angry	neutral	31,3038	0,0001	0	0	0,2147	45,5665	22,9148	0,5547	0,2499	0,2371	4,4114	14,486	3,0972	76,9636	
5	0:00:03	0	angry	sad	sad	0	0	0	0	0	0	100	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
6	0:00:03	0	angry	sad	fear	0	0	0	0	0	0	100	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
7	0:00:03	0	angry	sad	happy	0,2628	0,0002	0,0094	0	0,067	0,8282	98,8324	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
8	0:00:03	0	sad	sad	surprise	2,9558	0,0002	0,0001	0	7,473	87,7049	1,866	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
9	0:00:03	0	angry	sad	neutral	0,0002	0,0006	0,0004	0	0,015	0,0127	99,9711	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
10	0:00:03	0	angry	sad	happy	0,0002	0,0006	0,0004	0	0,015	0,0127	99,9711	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
11	0:00:03	0	angry	sad	happy	0	0,0001	0	0	0,0003	0,001	99,9986	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
12	0:00:03	0	angry	sad	angry	0	0	0	0	0	0,0001	99,9999	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
13	0:00:04	0	angry	sad	fear	0	0	0	0	0,0001	0,0004	99,9994	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
14	0:00:04	0	angry	sad	neutral	0	0	0	0	0,0001	0,0004	99,9994	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
15	0:00:04	0	angry	sad	fear	0,0084	0,0001	0,0001	0	0,9027	0,1847	98,904	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
16	0:00:04	0	angry	sad	sad	0,0113	0,0004	0,0001	0	6,4411	0,7658	92,7814	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
17	0:00:04	0	angry	sad	surprise	0,0014	0,0012	0	0	6,7087	0,2306	93,058	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
18	0:00:04	0	angry	sad	neutral	0	0,0001	0	0	0,8767	0,0077	99,1156	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
19	0:00:04	0	angry	sad	fear	0	0,0001	0	0	2,4969	0,0168	97,4862	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
20	0:00:04	0	angry	sad	sad	0	0,0013	0	0	0,5711	0,0168	99,4107	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
21	0:00:04	0	angry	sad	sad	0,0031	0,0142	0,0067	0,0023	5,3645	2,2664	92,3429	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	
22	0:00:05	0	angry	sad	neutral	0	0,0033	0	0	0,1544	0,121	99,7212	6,6375	1,0376	0,0431	0,0637	1,1887	74,7984	16,231	

Figura 5.2: Fichero logs de la herramienta tras el análisis

5.3. Estructura de ficheros

En cuanto a la estructura de ficheros, como hemos mencionado anteriormente, la herramienta se divide en dos partes, una API y un *frontend*, por tanto la estructura de ficheros tendrá una distribución similar.

En el primer nivel de carpetas de nuestro proyecto tenemos el fichero *main.js* encargado de gestionar el *framework* de Electron junto con una carpeta *node-modules* y algunos archivos de configuración (tanto de Git como de Node.js) a los que no daremos mucha importancia. Además encontramos nuestras dos carpetas principales: *Backend* donde encontraremos todos los ficheros relativos a la API, y *Frontend* donde encontraremos los ficheros que componen la interfaz de usuario de la aplicación de escritorio.

Dentro de la carpeta *Backend* encontramos una carpeta *Video* donde almacenaremos por defecto los vídeos de la Webcam tomados durante el análisis, una carpeta *Screen*, donde se almacenarán por defecto los vídeos de la pantalla, una carpeta *CSV* donde se almacenarán los *logs* de los análisis por defecto, una carpeta *Image* que contendrá en todo momento la imagen de la webcam para poder pasarla al *frontend*, una carpeta *haarcascade* que contiene un fichero con un algoritmo de detección de rostros, una carpeta *Models* con los diferentes modelos de análisis de expresiones faciales y una carpeta *Status* que contiene el fichero en formato JSON con la configuración actual de la API. Además de todas estas carpetas encontramos el fichero *deep-emotion.py* que contiene la estructura de la red neuronal del modelo de Deep Learning (requerido en los modelos de PyTorch para la carga de modelos y su posterior uso) y el fichero *app.py* que contiene la implementación de la API Flask de nuestra herramienta.

Por último, dentro de la carpeta *Frontend* podemos encontrar: una carpeta *CSS* que contendrá todos los archivos CSS utilizados en la aplicación, una carpeta *Scripts* con todos los archivos de JavaScripts utilizados dentro del *frontend*, una carpeta *SRC* con los archivos como imágenes o similares utilizados en la aplicación y una carpeta *UI* con los archivos HTML utilizados en la aplicación para componer el *frontend*.

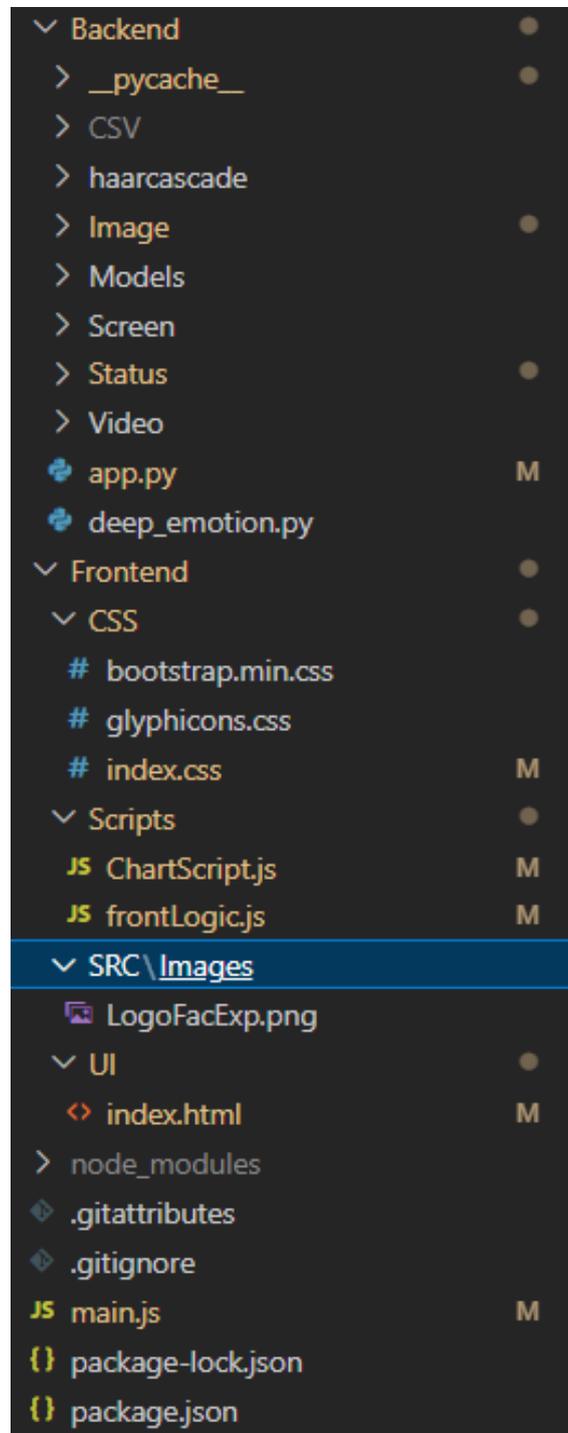


Figura 5.3: Estructura de ficheros del proyecto

Capítulo 6

Implementación de la herramienta

En este capítulo se pretende documentar la implementación de los modelos de reconocimiento de expresiones faciales, así como la herramienta de análisis de expresiones faciales, compuesta por una API que realizará el análisis y permitirá la conexión de la herramienta de manera sencilla, y el *frontend* o interfaz de usuario, que nos permitirá utilizar la herramienta con uno o varios de los modelos disponibles, además de configurar la propia API.

6.1. Implementación del análisis de las expresiones faciales

Lo primero de todo es comentar que para el análisis de expresiones faciales se ha utilizado tanto una librería ya existente, DeepFace, como dos modelos de implementación propia, uno basado en Transfer Learning y otro en Deep Learning.

Previamente a la implementación de nuestros modelos ML dedicados al análisis de expresiones faciales, debemos definir un modelo de emociones que será el que contenga las diferentes emociones que deberán ser capaces de clasificar nuestros modelos ML. En la literatura se pueden observar varios trabajos donde se define un modelo de emociones concreto para tareas de clasificación de emociones como puede ser el trabajo de Herrmann *et al.* [9] o el trabajo de Sánchez-Gordón *et al.* [22]. Nuestro modelo de emociones se inspira en el trabajo de Shaver *et al.* [24] que las agrupa en ramas por similitud. De las emociones que plantea tomamos: *happiness* como representante de la rama *joy*, *surprise* como representante de la rama *surprise*, *anger* y *disgust* como representantes de la rama *anger*, *sadness* como representante de la rama *sadness* y *fear* como representante de la rama *fear*. Para completar el modelo hemos añadido neutral para los casos de emociones neutrales. Una vez definido este modelo de emociones ya se pueden comenzar a implementar los modelos ML basados en Transfer Learning y Deep Learning.

Para la implementación del modelo basado en Transfer Learning hemos utilizado una

red neuronal proporcionada por TensorFlow, más concretamente MobileNetV2 (Ver Figura 6.1), una red neuronal de clasificación de imágenes con una salida de 1000 clases que ya ha sido utilizada para propósitos similares [20]. Obviamente este número es excesivo para nuestro caso, puesto que nuestro modelo de emociones consta de tan solo 7 emociones, y por tanto la salida que nos interesa tener es de 7 clases. Para adaptar MobileNetV2 a nuestro caso simplemente eliminemos la última capa (la que ofrece la salida en 1000 clases) y añadimos tras la penúltima capa de la red neuronal Capas de Conexión Total (*Fully Connected Layers*) que fueran bajando gradualmente el número de nodos, de forma que pudiéramos obtener una salida en 7 clases sin perder la fiabilidad de la red neuronal. A la última capa se le añade la función *softmax*, función que nos proporciona un desglose de la clasificación obtenida en el análisis entre todas las clases. Los modelos que utilizan este tipo de función en la salida se denominan clasificadores débiles.

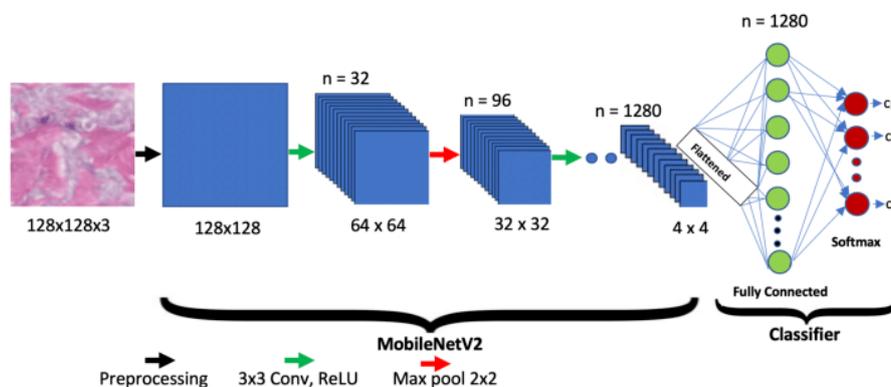


Figura 6.1: Arquitectura de MobileNetV2

En cuanto a la implementación basada en Deep Learning, se pretende realizar una red neuronal desde cero para cumplir con la tarea del análisis de expresiones faciales. Para ello se pretende utilizar una red neuronal convolucional con un módulo de atención (ACNN, *Attentional Convolutional Neural Network*), un tipo de red neuronal que ya ha sido probado para este tipo de tareas [18]. Los módulos de atención dentro de las redes neuronales tratan de imitar la atención cognitiva, haciendo que la red neuronal de más importancia a las partes más llamativas o de mayor importancia dentro de los datos de entrada. Para la implementación de esta red neuronal hemos seguido el esquema que se utiliza en el artículo de Minaee et al. [18] (Ver Figura 6.2). En resumen, se crea una red neuronal principal que consta de dos bloques, ambos formados por una capa de convolución para extraer características, una función de activación ReLU, otra capa de convolución, una capa *MaxPool* que permite tomar las características más importantes de cada sección y otra función de activación ReLU. A parte de esta red neuronal principal, se crea otra que funcionará como módulo de atención, formada por dos bloques que constan de una capa de convolución, una capa *MaxPool* y una función de activación ReLU, y un bloque de capa de conexión total (de 90 nodos a 32) con una función de activación ReLU. Estas dos redes neuronales convergen tras el segundo bloque de capas de la red neuronal principal, obteniendo como resultado las características más importantes de la entrada. A partir de

aquí se utilizan capas de conexión total hasta una salida en 7 clases (cada una de las clases será una emoción de nuestro modelo de emociones). A la última capa se le añade una función *softmax*, de esta manera obtenemos el clasificador débil.

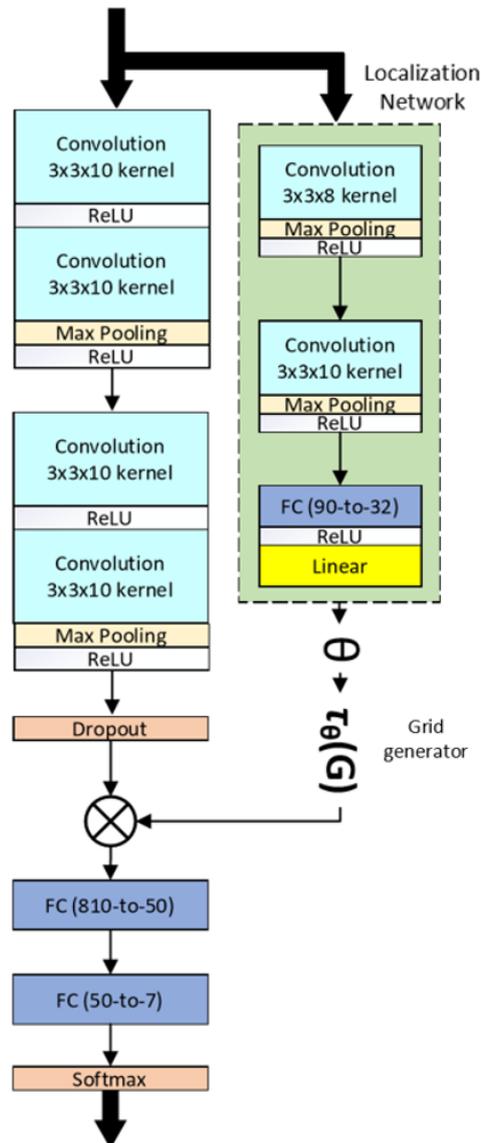


Figura 6.2: Arquitectura de la red neuronal

Durante las implementaciones de los modelos basados en Transfer Learning y Deep Learning se crearon varias versiones previas con conjuntos de datos más pequeños, pero que no nos ofrecían el desempeño esperado. Es por eso que se hizo un aumento del conjunto de datos de entrenamiento, tratando de igualar en mayor o menor medida las imágenes que había para cada una de las emociones, pudiendo así crear modelos ML más equilibrados como los actuales, cuyo desempeño en el caso del modelo de Transfer Learning supera a la librería DeepFace (Ver capítulo 7 Sección 1).

Para realizar el análisis de las expresiones faciales utilizamos tanto la librería DeepFace como nuestros modelos ML basados en Transfer Learning y Deep Learning, los cuales serán cargados para su uso dentro del fichero de la implementación de la API. Para que se produzca el análisis simultáneamente la API atiende a las peticiones que se le lanzan el análisis se realizara en un hilo que comenzará al ejecutar el inicio del análisis y morirá cuando se pare dicho análisis.

Este hilo contendrá un bucle que se ejecutara mientras no se detenga el análisis con una petición, donde se tomará la imagen del usuario en cada iteración y se analizará su expresión facial con cada uno de los modelos (DeepFace, Transfer Learning y Deep Learning) que estén activados en el momento del análisis.

Se añade también una opción donde, para obtener una mayor fiabilidad, mantendremos una ventana de tiempo (dividido entre tiempo anterior y tiempo posterior a la petición) donde se guardan todos los análisis realizados, de forma que a la hora de devolver la respuesta envíe el promedio del análisis durante esta ventana de tiempo.

Utilizando estos dos tipos de análisis (en tiempo real y con ventana temporal de datos) podemos analizar de forma óptima y fiable las expresiones faciales en tiempo real o la reacción del usuario durante un tiempo determinado.

6.2. Peticiones de la API

En esta sección se presentan todas las peticiones que se pueden lanzar a nuestra API. Para representarlas se utilizará el siguiente formato para especificar las características de cada petición:

Tabla 6.1: Plantilla de especificación de peticiones

Descripción	Breve descripción de la funcionalidad de la aplicación
Tipo de petición	[GET POST PUT DELETE]
Uri	Dirección de la petición
Parámetros del cuerpo	Parámetros que se envían en el cuerpo (si existen)
Respuesta	Respuesta proporcionada por la API

6.2.1. Inicio del análisis

Tabla 6.2: Especificación de la petición del inicio del análisis

Descripción	Esta función nos permitirá iniciar el análisis del reconocimiento de expresiones faciales y comenzará la grabación de la webcam la pantalla y los logs siempre que estén habilitadas.
Tipo de petición	GET

Uri	/start
Parámetros cuerpo	Petición sin cuerpo
Descripción	“Analysing”: “TRUE”

6.2.2. Parada del análisis

Tabla 6.3: Especificación de la petición del final del análisis

Descripción	Esta función nos permitirá parar el análisis del reconocimiento de expresiones faciales, finalizando la grabación de la webcam, la pantalla y los <i>logs</i> siempre que estuvieran habilitadas
Tipo de petición	GET
Uri	/stop
Parámetros cuerpo	Petición sin cuerpo
Descripción	“Analysing”: “FALSE”

6.2.3. Resultado del análisis

Tabla 6.4: Especificación de la petición de resultados del análisis

Descripción	Esta función nos permitirá obtener el resultado del análisis de expresiones faciales, con su desglose por emociones, para cada uno de los modelos
Tipo de petición	GET
Uri	/result
Parámetros del cuerpo	Petición sin cuerpo
Respuesta	Resultados del análisis de expresiones faciales para cada modelo mostrando el desglose de emociones formato JSON

6.2.4. Estado de la API

Tabla 6.5: Especificación de la petición del estado de la API

Descripción	Esta función nos permitirá obtener el estado de la API y los valores actuales de su configuración
Tipo de petición	GET
Uri	/status
Parámetros del cuerpo	Petición sin cuerpo
Respuesta	Estado y configuración de la API en formato JSON

6.2.5. Configuración de modelos

Tabla 6.6: Especificación de la petición de configuración de modelos en uso durante el análisis

Descripción	Esta petición nos permite indicar que modelos se deberán utilizar y que otros no durante el análisis de expresiones faciales
Tipo de petición	POST
Uri	/models
Parámetros del cuerpo	Deepface:Bool Transfer:Bool Deep:Bool
Respuesta	Configuración actual de la API en formato JSON

6.2.6. Configuración de ventana temporal

Tabla 6.7: Especificación de la petición de configuración del modo de ventana temporal

Descripción	Esta petición nos permite indicar si se debe utilizar el modo de ventana temporal durante el análisis, además de indicar en milisegundos de cuanto tiempo será dicha ventana temporal
Tipo de petición	POST
Uri	/threshold
Parámetros del cuerpo	ThresholdMode:Bool ThresholdTime:Integer
Respuesta	Configuración actual de la API en formato JSON

6.2.7. Configuración de la grabación de video de la Webcam

Tabla 6.8: Especificación de la petición de configuración de la grabación de vídeo Webcam

Descripción	Esta petición nos permite indicar si queremos habilitar la grabación de vídeo del usuario mediante la webcam, además de indicar la carpeta y nombre del vídeo
Tipo de petición	POST
Uri	/video
Parámetros del cuerpo	VideoFolder:String Video:Bool
Respuesta	Configuración actual de la API en formato JSON

6.2.8. Configuración de la grabación de vídeo de la pantalla

Tabla 6.9: Especificación de la petición de configuración de la grabación de pantalla

Descripción	Esta petición nos permite indicar si queremos habilitar la grabación de vídeo de la pantalla, además de indicar la carpeta y nombre del vídeo
Tipo de petición	POST
Uri	/screen
Parámetros del cuerpo	ScreenFolder:String Screen:Bool
Respuesta	Configuración actual de la API en formato JSON

6.2.9. Configuración de la grabación de logs

Tabla 6.10: Especificación de la petición de configuración de la grabación de *logs*

Descripción	Esta petición nos permite indicar si queremos habilitar la grabación de <i>logs</i> del análisis, además de indicar la carpeta y nombre del fichero CSV
Tipo de petición	POST
Uri	/log
Parámetros del cuerpo	LogsFolder:String Logs:Bool
Respuesta	Configuración actual de la API en formato JSON

6.3. Frontend

A la hora de implementar el *frontend*, como hemos indicado en capítulos anteriores, hemos decidido utilizar Electron, un *framework* de desarrollo de aplicaciones de escritorio que utiliza los lenguajes comúnmente utilizados para el desarrollo web: HTML, CSS, JavaScript. De esta forma, con HTML y CSS podemos lograr una interfaz de usuario con aspecto profesional y bien estructurado y utilizando JavaScript podemos dotar a dicha interfaz de funcionalidad, además de conectar el *frontend* con la API. Para utilizar este *framework* simplemente hay que instalar la librería para node.js y crear un archivo que será el que cargue nuestra estructura HTML, CSS y JavaScript en una ventana (Ver Figura 6.3).

Para empezar con la implementación decidimos utilizar Figma para realizar Mock-ups para el diseño de lo que sería la aplicación de escritorio, pensando siempre en que se busca una estética profesional y teniendo en mente los requisitos mencionados en el Capítulo 4. La intención era crear una aplicación simple, por lo tanto queríamos que constara de

```
const {app, BrowserWindow, Menu} = require('electron');
require('electron-reload')(__dirname);

function createWindow () {
  const mainWindow = new BrowserWindow({
    width:1920,
    height:1080
  });

  mainWindow.loadFile("Frontend/UI/index.html");
  mainWindow.webContents.openDevTools();
  Menu.setApplicationMenu(null);
}

app.whenReady().then(() =>
  createWindow()
);
```

Figura 6.3: Fichero JavaScript con la implementación de Electron

dos únicas páginas. Una página estaría dedicada para el análisis, donde mostraríamos: la webcam en tiempo real, un gráfico de radar que muestre el desglose de las emociones por cada modelo utilizado en el análisis y los *logs* en tiempo real. La otra página mostraría el apartado de configuración de la API, donde podríamos encontrar: estado actual de la API, configuración de la grabación de vídeo, pantalla y *logs*, configuración de la webcam y la pantalla a grabar y configuración del modo de ventana temporal. Tras varios intentos intentando diseñar las páginas que se plantean para la aplicación con Figma, nos decantamos por el siguiente diseño (Ver Figuras 6.4, 6.5)

Tras esto se realizó la implementación en HTML y CSS para dotar de un estilo y estructura similares a los mock-ups previamente creados. Para dotar de funcionalidad a la aplicación utilizaríamos JavaScript tanto para la navegación dentro de la aplicación como para la interacción con la API utilizando funciones *fetch*. Las funciones *fetch* de JavaScript nos permiten realizar peticiones a una URI y esperar una respuesta. Gracias a estas funciones podemos utilizar todas las peticiones que se han definido previamente dentro de la aplicación de escritorio, permitiéndonos configurar la API (modelos, grabación de vídeo y pantalla,etc...), conocer su estado, iniciar y parar los análisis, etc...

Para la realización del gráfico de radar que muestra el desglose de emociones del análisis para cada modelo hemos utilizado una librería de JavaScript llamada AnyChart que nos permite la creación de distintas clases de gráficos mediante funciones JavaScript. Para la implementación simplemente tomamos los datos del análisis con una petición *fetch* y los tratamos con las funciones de AnyChart, creando primero estructuras para cada modelo utilizado en el análisis con sus emociones, creando el gráfico de radar y añadiendo estas

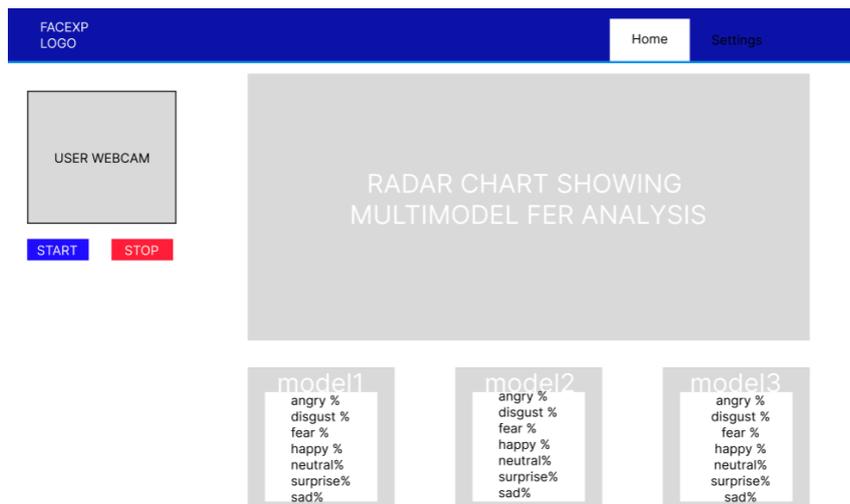


Figura 6.4: Mock-up de la página principal de la herramienta

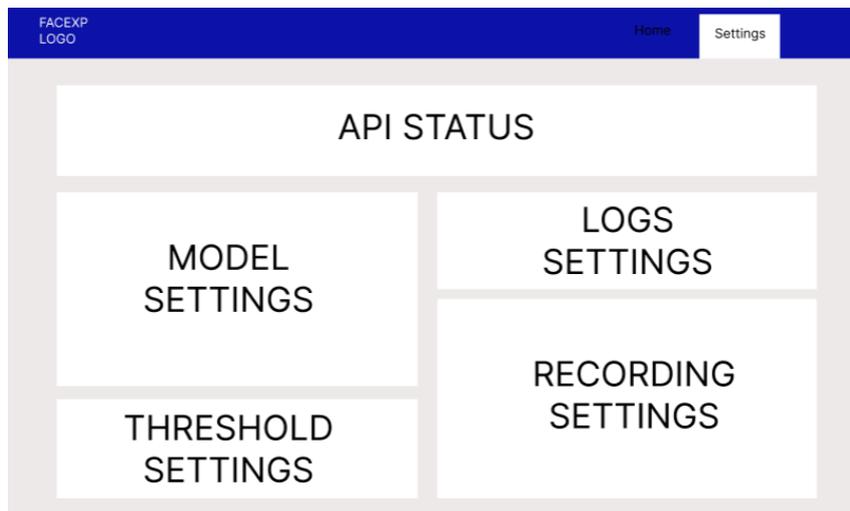


Figura 6.5: Mock-up de la página configuración de la herramienta

estructuras al gráfico. Se imprimirá un nuevo gráfico cada vez que se reciben los datos del análisis (Ver Figuras 6.6, 6.7).

Tras todo lo comentado, la aplicación de escritorio de la herramienta terminaría con el siguiente aspecto (Ver Figuras 6.8 y 6.9), siendo totalmente funcional y permitiendo tanto su configuración como el análisis en tiempo real de las expresiones faciales. Cabe destacar que se han añadido un par de botones para marcar el inicio y fin de tareas a la hora de realizar experimentos controlados en el caso de la ventana de análisis (Ver Figura 6.8) y que falta la sección de la ventana temporal en la página de configuración (Ver Figura 6.9).

```

async function plotChart (data) {
  document.getElementById('container').innerHTML = "";

  var data1 = [
    {x: "neutral", value: parseFloat(data.neutral)},
    {x: "happy", value: parseFloat(data.happy)},
    {x: "surprise", value: parseFloat(data.surprise)},
    {x: "disgust", value: parseFloat(data.disgust)},
    {x: "fear", value: parseFloat(data.fear)},
    {x: "sad", value: parseFloat(data.sad)},
    {x: "angry", value: parseFloat(data.angry)}
  ];
  var data2 = [
    {x: "neutral", value: parseFloat(data.neutral)},
    {x: "happy", value: parseFloat(data.happy)},
    {x: "surprise", value: parseFloat(data.surprise)},
    {x: "disgust", value: parseFloat(data.disgust)},
    {x: "fear", value: parseFloat(data.fear)},
    {x: "sad", value: parseFloat(data.sad)},
    {x: "angry", value: parseFloat(data.angry)}
  ];
  var data3 = [
    {x: "neutral", value: parseFloat(data.neutral)},
    {x: "happy", value: parseFloat(data.happy)},
    {x: "surprise", value: parseFloat(data.surprise)},
    {x: "disgust", value: parseFloat(data.disgust)},
    {x: "fear", value: parseFloat(data.fear)},
    {x: "sad", value: parseFloat(data.sad)},
    {x: "angry", value: parseFloat(data.angry)}
  ];
}

```

Figura 6.6: Implementación con AnyChart del gráfico de radar - Definición de estructuras de datos

```

var chart = anychart.radar();

chart.yScale()
  .minimum(0)
  .maximum(100)
  .ticks({'interval':20});

chart.yGrid().palette(["gray 0.1", "gray 0.2"]);

chart.area(data1).name('DeepFace').markers(true).fill("#E55934", 0.3).stroke("#E55934")
chart.area(data2).name('Bulbasaur').markers(true).fill("#9BC53D", 0.3).stroke("#9BC53D")
chart.area(data3).name('Squirtle').markers(true).fill("#5BC0EB", 0.3).stroke("#5BC0EB")

chart.title("")
  .legend(true);

chart.container('container');

chart.draw();

```

Figura 6.7: Implementación con AnyChart del gráfico de radar - Creación del gráfico de radar

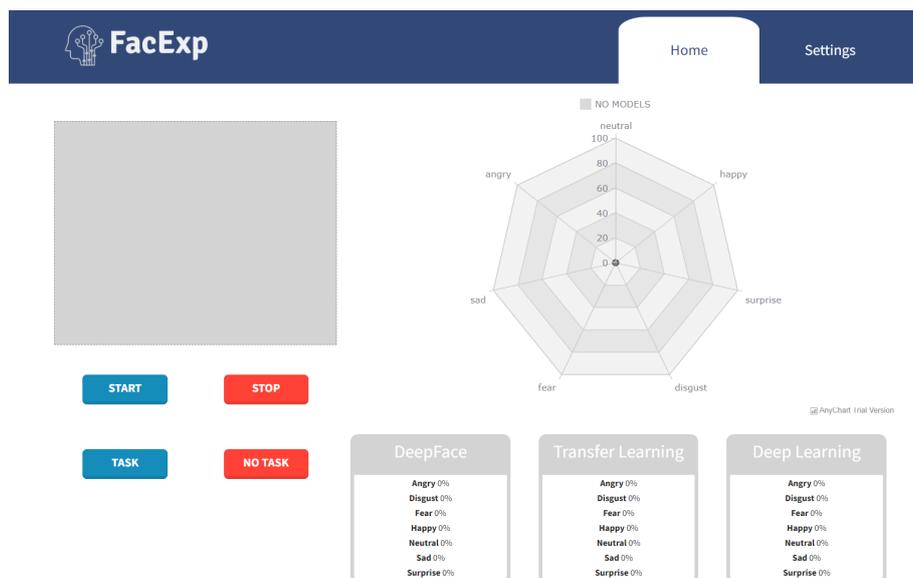


Figura 6.8: Página de análisis de la herramienta finalizada

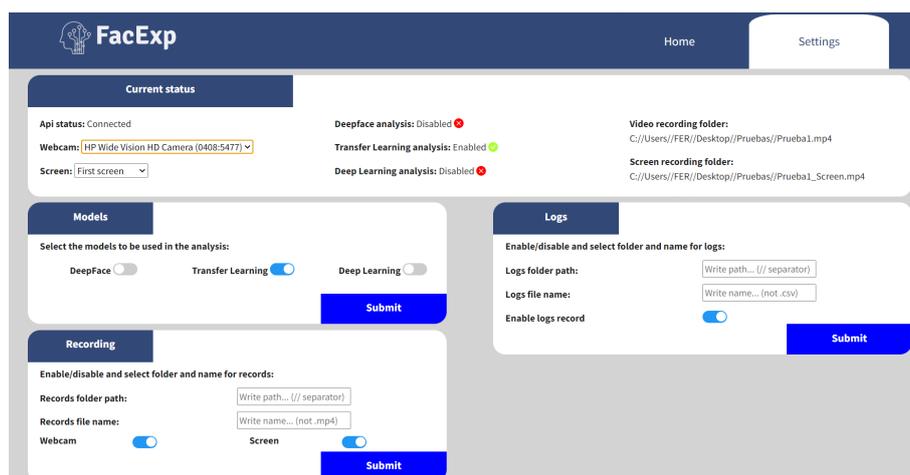


Figura 6.9: Página de configuración de la herramienta finalizada

6.4. GitHub con la implementación

La aplicación con todo lo indicado en este capítulo está disponible en GitHub. A través de el siguiente enlace se puede descargar el proyecto:

https://github.com/Fercarlllorens/FacExp_Desktop

Capítulo 7

Validación y pruebas

En este capítulo se describe la evaluación tanto de los modelos ML utilizados para el análisis de expresiones faciales como el uso de la herramienta en tiempo real. Para evaluar los modelos tomaremos las matrices de confusión de cada uno de los modelos ML, calculando *precision* y *recall*. Por otra se realizarán dos evaluaciones empíricas para proporcionar evidencias preliminares sobre el uso de la herramienta propuesta en tiempo real, una utilizando vídeos y otra utilizando interfaces de usuario. Para finalizar también realizaremos una pequeña aplicación para probar el correcto funcionamiento de la API.

7.1. Evaluación de modelos ML: Matrices de confusión, *precision*, *recall* y F1

En el campo de la inteligencia artificial existen diversas formas de medir el desempeño de los modelos ML en el ámbito de los modelos con entrenamiento supervisado, entre ellas las matrices de confusión, la *precision*, el *recall* y el F1 [8].

Las matrices de confusión, como su nombre indica, son matrices donde las columnas representan las predicciones de cada clase y las filas las instancias reales de cada clase. De esta forma se realiza una representación de la clasificación de un conjunto de datos donde podemos observar las clasificaciones correctas e incorrectas de cada una de las clases, además de poder detectar sesgos de manera sencilla o observar que grupos de clases son difíciles de diferenciar entre ellas por el modelo.

A raíz de las matrices de confusión se pueden obtener dos valores que miden el desempeño del modelo de clasificación, la *precision* y el *recall*. La *precision* se calcula como la fracción de instancias bien clasificadas, los verdaderos positivos, entre el total de instancias clasificadas para la clase, la suma de verdaderos y falsos positivos (Ver 7.1). El *recall* en cambio se calcula como la fracción de instancias bien clasificadas, o verdaderos positivos, entre el total de instancias de la clase, la suma de los verdaderos positivos y los falsos negativos (Ver 7.2).

Tras el cálculo de los valores de *precision* y *recall* podremos calcular también el valor de F1. Este valor se utiliza comúnmente para combinar los valores de *precision* y *recall* en un único valor, de manera que resulte más fácil la comparación del desempeño de modelos ML. Este valor es realmente utilizado en casos donde nos resulta igual de importante los valores de *precision* y *recall*, como es nuestro caso. Para el cálculo del valor F1 se realiza la media armónica de los valores de *precision* y *recall* (Ver 7.3).

$$Precision = \frac{VerdaderosPositivos}{VerdaderosPositivos + FalsosPositivos} \quad (7.1)$$

$$Recall = \frac{VerdaderosPositivos}{VerdaderosPositivos + FalsosNegativos} \quad (7.2)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7.3)$$

Para un conjunto de datos, el cual se aparto del conjunto de entrenamiento de los modelos ML, y que a partir de ahora denominaremos el conjunto de datos de prueba, se lanza el análisis de cada una de las imágenes de este conjunto por los dos modelos de implementación propia, y también por DeepFace para poder comparar los resultados con nuestros modelos ML. La idea de utilizar el mismo conjunto de test datos de prueba para los tres modelos ML es obtener la matriz de confusión para un mismo conjunto de datos de cada modelo ML, pudiendo hacer una comparación directa de estos no solo con la matriz, sino también con la *precision*, el *recall* y el valor de F1, de manera que podremos observar cual es el modelo ML que nos proporciona un mejor desempeño sin mayores complicaciones.

El primer paso es realizar las matrices de confusión para los tres modelos ML. Para ello creamos un *dataframe* donde filas y columnas están marcados por los 7 valores que posee nuestro modelo de emociones (*angry*, *disgust*, *fear*, *happy*, *neutral*, *sad* y *surprise*) y todas las celdas tienen un valor 0. A partir de aquí se ejecuta un bucle que analiza todas las imágenes del conjunto de test con cada librería. Al estar todas las imágenes del conjunto de test etiquetadas con la emoción correspondiente, una vez obtenida la emoción del análisis de los modelos ML se incrementa en uno el valor de la celda en la fila de la emoción con la que esta etiquetada la imagen y la columna de la emoción que el modelo da como resultado del análisis. Al terminar el bucle la matriz de confusión está completa. A continuación mostramos las matrices de confusión que se obtuvieron para DeepFace (Ver Tabla 7.1), Transfer Learning (Ver Tabla 7.2), y Deep Learning (Ver Tabla 7.3).

Tabla 7.1: Matriz de confusión de DeepFace

		PREDICCIONES						
		angry	disgust	fear	happy	neutral	sad	surprise
R E A L E S	angry	337	16	181	77	137	185	25
	disgust	28	33	23	10	4	12	1
	fear	89	7	405	98	140	206	79
	happy	50	3	158	1264	150	124	25
	neutral	83	4	194	132	537	253	30
	sad	98	11	237	126	242	518	15
	surprise	23	3	136	95	83	53	438

Tabla 7.2: Matriz de confusión del modelo basado en Transfer Learning

		PREDICCIONES						
		angry	disgust	fear	happy	neutral	sad	surprise
R E A L E S	angry	481	81	131	33	78	137	17
	disgust	24	70	8	1	1	7	0
	fear	126	57	451	35	69	237	49
	happy	83	30	84	1270	184	88	35
	neutral	90	36	113	87	598	289	20
	sad	131	61	167	48	152	677	11
	surprise	35	18	225	41	41	30	441

Tabla 7.3: Matriz de confusión del modelo basado en Deep Learning

		PREDICCIONES						
		angry	disgust	fear	happy	neutral	sad	surprise
R E A L E S	angry	312	12	103	218	144	154	15
	disgust	27	15	25	16	10	16	2
	fear	150	7	200	200	181	232	54
	happy	99	4	66	1284	134	169	18
	neutral	146	8	111	204	493	243	28
	sad	164	3	135	258	290	377	20
	surprise	69	2	142	136	124	77	281

Una vez calculadas las matrices de confusión podemos observar ya a simple vista que DeepFace y el modelo de Transfer Learning ofrecen unos resultados similares, mientras que el modelo de Deep Learning nos brinda unos resultados menos prometedores. Tras esto podemos calcular la *precision* y el *recall* para cada una de las emociones con las ecuaciones previamente descritas (Ver 7.1 y 7.2). La *precision* y *recall* total de cada modelo ML serán los promedios de los valores de la *precision* y el *recall* respectivamente de cada emoción para ese modelo ML. Además, una vez calculados la *precision* y el *recall* podremos también calcular el valor de F1.

A continuación se presentan los resultados obtenidos (Ver Tablas 7.4, 7.5 y 7.6).

Tabla 7.4: Resultados de *precision*, *recall* y F1 para DeepFace

	<i>Precision</i>	<i>Recall</i>	F1
angry	47.5988 %	35.1774 %	40.4561 %
disgust	42.8571 %	29.7297 %	35.1063 %
fear	30.3598 %	39.5507 %	34.3511 %
happy	70.1442 %	71.2514 %	70.6934 %
neutral	41.5313 %	43.5523 %	42.5177 %
sad	38.3419 %	41.5396 %	39.8767 %
surprise	71.4518 %	52.7075 %	62.0614 %
TOTAL	48.8978 %	44.7869 %	46.7521 %

Tabla 7.5: Resultados de *precision*, *recall* y F1 para el modelo de Transfer Learning

	<i>Precision</i>	<i>Recall</i>	F1
angry	49.5876 %	50.2087 %	49.8962 %
disgust	19.8300 %	63.0630 %	30.1723 %
fear	38.2527 %	44.0429 %	40.9441 %
happy	83.8283 %	71.5896 %	77.2270 %
neutral	53.2502 %	48.4995 %	50.7639 %
sad	46.2116 %	54.2902 %	49.9262 %
surprise	76.9633 %	53.0685 %	62.8204 %
TOTAL	52.5605 %	54.9661 %	53.7363 %

Tabla 7.6: Resultados de *precision*, *recall* y F1 para el modelo de Deep Learning

	<i>Precision</i>	<i>Recall</i>	F1
angry	32.2647 %	32.5678 %	32.4155 %
disgust	29.4117 %	13.5135 %	18.5184 %
fear	25.5754 %	19.5312 %	22.1483 %
happy	55.4404 %	72.3788 %	62.7872 %
neutral	35.8284 %	39.9837 %	37.7921 %
sad	29.7318 %	30.2325 %	29.9800 %
surprise	67.2248 %	30.2325 %	41.7079 %
TOTAL	39.3539 %	33.8146 %	36.3745 %

Como podemos observar en los resultados se confirma lo que se podía entrever en las matrices de confusión. El modelo basado en Transfer Learning es el que mejores resultados nos ofrece para la *precision* y el *recall* totales, y por consiguiente, es el modelo ML que mejor valor F1 ofrece, seguido con no mucha diferencia por DeepFace que ofrece

también muy buenos resultados. Por desgracia el modelo implementado en Deep Learning no ofrece tan buenos resultados, quedando en último lugar en la comparativa.

En muchas ocasiones simplemente queremos conocer si el usuario muestra una reacción positiva o negativa, sin darle importancia a la emoción que presente. Para estos casos también resulta interesante calcular la *precision*, el *recall* y el valor de F1. Según el artículo de Shaver *Et al.* [24] podemos agrupar *angry*, *disgust*, *fear* y *sad* como emociones negativas y *happy* y *surprise* como emociones positivas, aunque *neutral* se puede añadir como una emoción positiva también. Con estas agrupaciones y utilizando la matriz de confusión calculada para cada modelo ML podemos obtener los valores de *precision*, *recall* y F1 de las emociones positivas y negativas de cada modelo. A continuación, se presentan los resultados obtenidos por cada modelo ML para sentimientos positivos y negativos (Ver Tablas 7.7, 7.8 y 7.9):

Tabla 7.7: *Precision*, *recall* y F1 con emociones positivas y negativas para DeepFace

	<i>Precision</i>	<i>Recall</i>	F1
Positivo	67,1052 %	74,2718 %	70,5069 %
Negativo	68,9756 %	61,8547 %	65,2214 %

Tabla 7.8: *Precision*, *recall* y F1 con emociones positivas y negativas para Transfer Learning

	<i>Precision</i>	<i>Recall</i>	F1
Positivo	70,5897 %	84,6153 %	76,9688 %
Negativo	85,2095 %	71,7418 %	77,8979 %

Tabla 7.9: *Precision*, *recall* y F1 con emociones positivas y negativas para Deep Learning

	<i>Precision</i>	<i>Recall</i>	F1
Positivo	69,3353 %	65,7420 %	67,4909 %
Negativo	57,8443 %	62,9726 %	60,2996 %

Como era de esperar, los valores que obtenidos de *precision*, *recall* y F1 de cada modelo para emociones positivas y negativas nos muestran que el mejor desempeño en este tipo de clasificación sigue siendo para el modelo basado en Transfer Learning, seguido de la librería DeepFace y no tan alejado esta vez, del modelo basado en Deep Learning.

7.2. Proceso experimental con vídeos

En este apartado se presenta el diseño de un experimento que tiene como objetivo evaluar la utilidad de la herramienta de análisis de expresiones faciales en un entorno

real. Para ello, se pretenden utilizar vídeos que susciten emociones tanto positivas como negativas para que las reacciones de los sujetos de prueba sean captadas y analizadas por dicha herramienta, pudiendo así comparar el análisis de las emociones con la reacción que produce el vídeo sobre el usuario.

El experimento está diseñado para ser ejecutado en una única sesión, con la participación de 12 personas de forma individual. Para cada participante se arrancará el análisis de las expresiones faciales, así como la grabación de pantalla y la grabación de vídeo (siempre que se permita), y se reproducirá un vídeo. Con los datos recogidos por todos los participantes obtendremos resultados reales sobre el funcionamiento de la herramienta en un entorno con usuarios en tiempo real y comprobaremos si el desempeño obtenido con el conjunto de datos de prueba se corresponde con el desempeño en tiempo real.

7.2.1. Motivación

Durante los últimos años hay una creciente tendencia por el uso de la inteligencia artificial y las técnicas de aprendizaje automático. Si esta tendencia se junta con conceptos que implican las emociones o sentimientos del usuario, como la experiencia de usuario, resulta interesante el uso de estas técnicas de aprendizaje automático para tratar de comprender las emociones del usuario en tiempo real.

FacExp es la herramienta de análisis de expresiones faciales en tiempo real que hemos desarrollado, y que utiliza la librería DeepFace y nuestros dos modelos ML de implementación propia. Esta herramienta ha sido creada con la finalidad de tratar de conocer los sentimientos del usuario en tiempo real. Previamente hemos calculado los valores de *precision* y *recall* de los modelos ML para observar su desempeño en clasificación de imágenes. La idea principal de este experimento será observar el desempeño de los modelos en tiempo real y con personas reales utilizando la herramienta propuesta.

7.2.1.1. Objetivo del experimento

El objetivo de este experimento es comprobar el desempeño del análisis de expresiones faciales utilizando la herramienta propuesta, lo cual se puede dividir en dos partes. La primera parte será comprobar la precisión de los modelos en tiempo real y con personas reales para obtener la *precision*, *recall* y el valor de F1 de los modelos ML en este contexto. La segunda parte será comparar los valores en tiempo real con los valores obtenidos con el conjunto de datos de prueba para ver si los modelos proporcionan el mismo rendimiento en tiempo real.

Según el *framework* Objetivo-Pregunta-Métrica (GQM) el objetivo del experimento es el siguiente:

Analizar tres modelos ML de análisis de expresiones faciales en tiempo real **con el propósito de** determinar si los valores de *precision*, *recall* y F1 obtenidos en tiempo real son similares respecto a los obtenidos con el conjunto de datos de prueba **desde el punto**

de vista de los investigadores en UX en el contexto de un grupo de usuarios visualizando un vídeo.

7.2.1.2. Contexto

El experimento planteado se realizará en un entorno con las siguientes características:

- **Online:** Al requerir del uso de la herramienta para el análisis en tiempo real mientras se reproduce un vídeo hemos decidido que el experimento debe realizarse de manera presencial para poder nosotros controlar tanto la herramienta como la reproducción de vídeo.
- **Estudiante:** Los participantes no tienen que ser profesionales ni conocedores de la tecnología, nosotros seremos los que llevemos el control tanto de la herramienta como del vídeo.
- **Específico:** El experimento trata de analizar las expresiones faciales con un vídeo personalizado y con unos modelos de análisis de expresiones faciales específicos.
- **Problema de juguete:** Durante el experimento los sujetos no se expondrán a un problema o situación real, simplemente observarán el vídeo mientras se analizan sus expresiones faciales.

7.2.1.3. Aspectos éticos y jurídicos

En el aspecto ético y jurídico, al ser un procedimiento experimental con sujetos deberemos tener en cuenta temas como:

- **Privacidad:** La privacidad del sujeto debe estar garantizada en todo momento, dando la opción de abandonar el proyecto o no realizar parte de el cuando crea que su privacidad peligra o está siendo violada. También se deberán asegurar el anonimato de todos los datos que se vayan a almacenar sobre los sujetos.
- **Consentimiento:** Deberemos explicar todo el proceso a los sujetos antes de comenzar y estos deberán proporcionar su consentimiento explícito.

Teniendo en cuenta estos aspectos podremos garantizar que nuestro proceso experimental es moralmente correcto y además es completamente legal, al menos en cuanto al tratamiento de datos de los sujetos se refiere. El aspecto económico no supondrá un problema debido a que todo lo utilizado durante la ejecución del experimento es de implementación propia, no se requieren pagar licencias o comprar equipo de ningún tipo.

7.2.2. Planificación del experimento

Durante esta sección se plantearán las hipótesis, definiendo tanto la variable independiente como las dependientes, además de definir el proceso que se llevará a cabo en cada sesión y los sujetos que participaron en dicho proceso experimental. También se comentarán los objetos e instrumentación requeridas y se hará una breve evaluación de la validez.

7.2.2.1. Hipótesis y variables

Como hemos mencionado en el apartado de la motivación el objetivo es obtener el desempeño de los tres modelos ML para el análisis de expresiones faciales en tiempo real utilizando la herramienta desarrollada, además de compararlos con su desempeño con el conjunto de datos de prueba.

Para ello se propone una única variable independiente: el **modelo ML utilizado** para el análisis, que puede obtener tres valores: el modelo empleado por **DeepFace**, el modelo implementado con **Transfer Learning** o el modelo implementado con **Deep Learning**. Mediante la utilización de la herramienta desarrollada se realizará simultáneamente el análisis con los tres modelos ML.

Las variables dependientes del experimento serán la *precision*, el *recall* y el valor de **F1**, ambas medibles mediante cálculo. Estas tres variables están explicadas en la primera sección del capítulo y se calculan mediante las formulas 7.1, 7.2 y 7.3 respectivamente.

Se plantean las siguientes hipótesis para el experimento:

Hipótesis nulas:

- H_{1_0} : No hay diferencias significativas del uso de los modelos ML en tiempo real, en cuanto a *precision*, con respecto a los valores obtenidos con el conjunto de datos de prueba.
- H_{2_0} : No hay diferencias significativas del uso de los modelos ML en tiempo real, en cuanto a *recall*, con respecto a los valores obtenidos con el conjunto de datos de prueba.
- H_{3_0} : No hay diferencias significativas del uso de los modelos ML en tiempo real, en cuanto al valor de F1, con respecto a los valores obtenidos con el conjunto de datos de prueba.

Hipótesis alternativas:

- H_{1_1} : El uso de los modelos ML en tiempo real nos ofrece un resultado significativamente diferente en cuanto a *precision* con respecto a los valores obtenidos con el conjunto de datos de prueba.

- H_{2_1} : El uso de los modelos en tiempo real nos ofrece un resultado significativamente diferente en cuanto a *recall* con respecto a los valores obtenidos con el conjunto de datos de prueba.
- H_{3_1} : El uso de los modelos en tiempo real nos ofrece un resultado significativamente diferente en cuanto al valor F1 con respecto a los valores obtenidos con el conjunto de datos de prueba.

Para determinar si las hipótesis son ciertas compararemos los valores de *precision* y *recall* de cada uno de los modelos ML en tiempo real con la *precision* y *recall* obtenidos en la primera sección del capítulo.

7.2.2.2. Diseño del experimento

Como hemos mencionado, los modelos estarán analizando simultáneamente las expresiones del participante en cada sesión, por lo que en una única sesión y con un único vídeo será suficiente para obtener los datos que buscamos.

Resulta interesante añadir al vídeo que vamos a reproducir un componente de aleatoriedad, de manera que el cambio entre una emoción y otra no siempre sea el mismo, ya que esto podría distorsionar en cierta manera los datos. Para ello utilizaremos 5 fragmentos de vídeo de unos segundos, cada uno de estos enfocado en una emoción en concreto, aunque debido a la dificultad de provocar una emoción en concreto al sujeto, deberemos contrastar con la grabación del usuario las expresiones que aparecen en el análisis. Debido a esta dificultad de causar emociones en concreto también hemos decidido calcular los valores conjuntos de emociones positivas y negativas, lo cual resulta una tarea más sencilla y nos permite comparar el desempeño con el calculado con el conjunto de datos de prueba. Los fragmentos de vídeo se ordenaran aleatoriamente para cada uno de los participantes (Ver Figura 7.3).

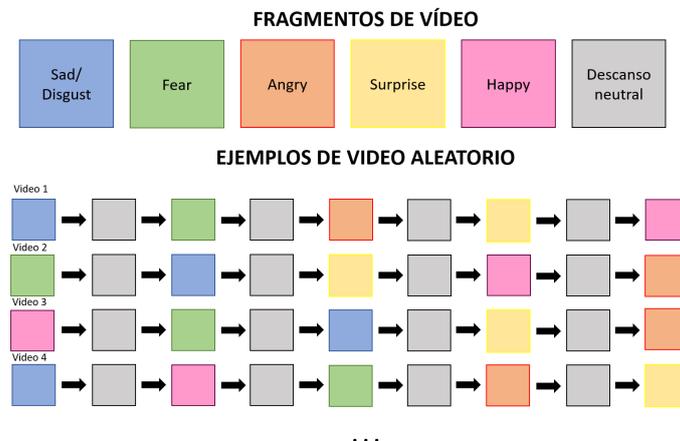


Figura 7.1: Creación de los vídeos con las emociones mezcladas aleatoriamente

Los efectos **aprendizaje** o **experiencia** no suponen un problema en este experimento, donde no se requieren usuarios expertos ni hay riesgo de que el sujeto pueda aprender nada sobre el funcionamiento del experimento que pueda alterar los resultados. En cuanto al efecto **fatiga**, el vídeo durará alrededor de los cinco minutos, tiempo insuficiente para que el sujeto pierda la atención sobre este, lo que supondría un problema para el experimento.

7.2.2.3. Selección de sujetos

Los perfiles seleccionados para llevar a acabo el experimento serán cualquier tipo de persona, sin restricción de edad ni género, y que no requieren de ningún tipo de experiencia en ningún ámbito. Esto se debe a que el experimento solo requiere la visualización del vídeo de principio a fin.

En este estudio participaron un total de **12 participantes**. Las sesiones se realizarán de forma individual y presencial y por supuesto no requerirá de una entrevista previa para agrupar por experiencia a los usuarios.

Las personas seleccionadas como sujetos fueron familiares y amigos siguiendo un modelo de bola de nieve.

7.2.2.4. Objetos e instrumentación

Al ser un experimento presencial se dispondrá de una sala de trabajo con todo lo requerido para la realización de la sesión, sin la necesidad de que los sujetos tengan que aportar nada.

En el lugar del experimento se dispondrá de un equipo con la herramienta **FacExp** para análisis de expresiones faciales, configurada para analizar con los tres modelos (**DeepFace**, **Transfer Learning** y **Deep Learning**) simultáneamente. El equipo estará conectado a dos monitores, uno para el sujeto donde se reproducirá el vídeo y otro para el experimentador donde se encontrará la aplicación con el análisis en vivo para que pueda manejarla. La separación en dos monitores permitirá que no se desvíe la atención del sujeto durante la reproducción del vídeo. Por ultimo el equipo estará conectado a una webcam que será la encargada de tomar el vídeo para el análisis.

Los participantes entrarán al lugar del experimento donde se les dará una explicación del experimento, se les mostrara brevemente el funcionamiento de la herramienta. Por ultimo antes de comenzar con el experimento, se tomará el consentimiento explícito de los participantes, mediante la firma de un documento, para permitirnos grabar el vídeo de la webcam para contrastar los datos.

7.2.2.5. Evaluación de la validez

Las amenazas a la validez interna cuestionan si realmente el tratamiento de las sesiones tendrá el efecto esperado. Ejemplos de este tipo de amenazas podrían ser la falta de

voluntarios o el efecto fatiga, aunque en nuestro caso no deberíamos preocuparnos por ello, puesto que al no requerir ningún tipo de experiencia y no tener ningún tipo de límite a la participación del experimento podremos conseguir 12 participantes de forma sencilla y el efecto fatiga no debería de aparecer en sesiones tan cortas como las que se plantean en este experimento.

Las amenazas a la validez externa tratan sobre la posible generalización de los resultados. Realmente como estamos probando tres modelos de análisis de expresiones faciales en específico para un conjunto de emociones no deberíamos de tener este problema. Existen pocas posibilidades de obtener datos incompletos o sesgados.

Las amenazas a la validez de constructo son los tipos de amenazas que surgen cuando las métricas que se utilizan para medir las variables dependientes no son las correctas. Como hemos mencionado anteriormente, la *precision*, el *recall* y el valor de F1 son las medidas comunes para determinar el desempeño de los modelos de clasificación, por lo que este tipo de amenazas no nos pueden afectar.

Por último las amenazas a la validez de conclusiones son las que pueden causar que las conclusiones no resulten estadísticamente válidas. En nuestro caso esto podría suceder si las emociones de un fragmento contrastaran mucho con las del siguiente o si las emociones del fragmento anterior persisten todavía en el fragmento siguiente. Para lidiar con ello hemos introducido el factor del orden aleatorio para los fragmentos y los fragmentos de descanso entre fragmentos que nos permiten relajar las emociones hasta el estado *neutral*. Gracias a esto no deberíamos de sufrir ninguna amenaza a la validez de este tipo en el experimento.

7.2.3. Ejecución del experimento

El experimento fue llevado a cabo según lo planeado con 12 personas cuya edad va desde los 22 años hasta los 59 años. Entre las personas seleccionadas para el experimento encontramos 3 mujeres, 9 hombres y 0 personas no binarias.

Al no requerir ningún tipo de entrenamiento para realizar el experimento la sesión comienza directamente para cada uno de los sujetos después de que rellenen un cuestionario demográfico que nos permite tener información del usuario y de firmar el consentimiento para participar en el experimento. Tras una breve explicación de la actividad se comienza a analizar las expresiones faciales del usuario con la herramienta y se reproduce el vídeo. Una vez finalizado el vídeo se finaliza el análisis de las expresiones faciales y la herramienta exporta las grabaciones de webcam y pantalla además del análisis en formato *.CSV* para su posterior análisis de datos.

7.2.4. Análisis de datos e interpretación

Como hemos indicado anteriormente para comprobar si las hipótesis se demuestran debemos comparar por cada uno de los usuarios si las emociones que se muestran en el

rostro de los usuarios es el que muestra el análisis de cada modelo ML en el mismo momento, calculando la *precision*, el *recall* y el valor de F1 de los modelos ML en tiempo real. Tras esto compararemos los resultados en tiempo real con los resultados con el conjunto de datos de pruebas.

Antes de calcular la *precision*, el *recall* y el valor de F1 deberemos crear las matrices de confusión de cada modelo. Analizando cada uno de los vídeos del sujeto en los momentos claves y comparándolo con los resultados del análisis obtenemos las siguientes matrices de confusión para cada modelo ML:

Tabla 7.10: Matriz de confusión en tiempo real para DeepFace

		PREDICCIONES						
		angry	disgust	fear	sad	happy	neutral	surprise
R	angry	25	0	0	4	1	5	0
E	disgust	0	24	0	10	0	1	0
A	fear	0	0	14	0	0	0	0
L	sad	5	10	0	39	1	29	0
E	happy	31	0	14	27	89	22	0
S	neutral	8	0	0	13	0	79	0
	surprise	-	-	-	-	-	-	-

Tabla 7.11: Matriz de confusión en tiempo real para el modelo de Transfer Learning

		PREDICCIONES						
		angry	disgust	fear	sad	happy	neutral	surprise
R	angry	30	4	0	0	0	1	0
E	disgust	0	25	0	0	0	10	0
A	fear	2	2	2	8	0	0	0
L	sad	2	0	0	66	1	7	0
E	happy	0	0	0	6	131	43	0
S	neutral	3	2	1	5	1	87	1
	surprise	-	-	-	-	-	-	-

Tabla 7.12: Matriz de confusión en tiempo real para el modelo de Deep Learning

		PREDICCIONES						
		angry	disgust	fear	sad	happy	neutral	surprise
R	angry	4	0	15	7	7	2	0
E	disgust	12	16	10	0	4	6	1
A	fear	0	0	1	7	0	5	0
L	sad	6	0	4	18	11	27	9
E	happy	19	0	0	14	119	25	7

S	neutral	13	1	5	12	8	55	5
	surprise	-	-	-	-	-	-	-

Como podemos observar los números que obtenemos en estas matrices de confusión son mucho menores que en los calculados con el conjunto de datos de prueba, cosa que es normal debido a que la muestra que tenemos es mucho menor. También cabe destacar que, como se puede observar, ninguno de los sujetos ha presentado una emoción de sorpresa en el experimento, por lo que debemos desestimar estos valores a la hora de calcular la *precision*, el *recall* y el valor de F1. Teniendo todo esto en cuenta y utilizando las formulas 7.1, 7.2 y 7.3 calcularemos la *precision*, el *recall* y el valor de F1 para cada uno de los modelos ML (Ver Tablas 7.13, 7.14 y 7.15)

Tabla 7.13: Resultados en tiempo real de *precision*, *recall* y F1 para DeepFace

	<i>precision</i>	<i>recall</i>	F1
angry	36.2319 %	71.4286 %	48.0769 %
disgust	70.5882 %	68.5714 %	69.5652 %
fear	100 %	50 %	66.6667 %
sad	46.4286 %	46.4286 %	46.4286 %
happy	97.8022 %	48.6339 %	64.9635 %
neutral	58.0882 %	85.8696 %	69.2982 %
surprise	-	-	-
TOTAL	68.1899 %	61.8220 %	60.8332 %

Tabla 7.14: Resultados en tiempo real de *precision*, *recall* y F1 para el modelo de Transfer Learning

	<i>precision</i>	<i>recall</i>	F1
angry	81.0811 %	85.7143 %	83.3333 %
disgust	75.7576 %	71.4286 %	73.5294 %
fear	66.6667 %	14.2857 %	23.5294 %
sad	77.6471 %	86.8421 %	81.9876 %
happy	98.4962 %	72.7778 %	83.7061 %
neutral	58.7838 %	87 %	70.1613 %
surprise	-	-	-
TOTAL	76.4054 %	69.6747 %	69.3745 %

Tabla 7.15: Resultados en tiempo real de *precision*, *recall* y F1 para el modelo de Deep Learning

	<i>precision</i>	<i>recall</i>	F1
angry	7.4074 %	11.4286 %	8.9888 %
disgust	94.1176 %	32.6531 %	48.4848 %
fear	2.8571 %	7.6923 %	4.1667 %
sad	31.0345 %	24 %	27.0677 %
happy	79.8658 %	64.6739 %	71.4715 %
neutral	45.8333 %	55.5556 %	50.2283 %
surprise	-	-	-
TOTAL	43.5193 %	32.6672 %	35.0680 %

Como podemos observar excepto Deep Learning que nos ofrece un resultado más similar, los otros dos modelos ML nos ofrecen unos mejores resultados. Esto se puede deber a que al ser una muestra de datos mucho más pequeña y menos equilibrada se producen resultados muy abultados para algunas de las emociones, desequilibrando un poco los resultados de algunos valores, además de que no hay muestras de la emoción *surprise* para el caso. Si tuviéramos un conjunto de muestras suficientemente grande como el conjunto de datos de prueba podrían cambiar estos valores, aunque la realidad es que en el análisis que se ha hecho observando las emociones de los sujetos y comprobando el análisis se puede observar como estos modelos funcionan realmente bien en tiempo real, por lo que también puede ser que el problema es que los resultados de *precision*, *recall* y F1 fueran bajos debido a que las imágenes del conjunto de prueba pueden dar lugar a confusión. En cuanto al orden de los modelos ML por desempeño, podemos observar como el modelo de Transfer Learning sigue siendo el que mejor desempeño ofrece, por delante de la librería DeepFace y, por último, el modelo basado en Deep Learning.

Como habíamos comentado también se pretende conocer el desempeño que dan cada uno de los modelos para emociones positivas y negativas (Ver Tablas 7.16, 7.17, 7.18)

Tabla 7.16: Resultados en tiempo real de *precision*, *recall* y F1 para DeepFace con reacciones positivas y negativas

	<i>precision</i>	<i>recall</i>	F1
positive	83.7004 %	67.1378 %	74.5098 %
negative	58.4821 %	77.9762 %	66.8367 %

Tabla 7.17: Resultados en tiempo real de *precision*, *recall* y F1 para el modelo de Transfer Learning con reacciones positivas y negativas

	<i>precision</i>	<i>recall</i>	F1
positive	93.2624 %	93.9286 %	93.5943 %
negative	92.7632 %	88.125 %	90.3846 %

Tabla 7.18: Resultados en tiempo real de *precision*, *recall* y F1 para el modelo de DeepLearning con reacciones positivas y negativas

	<i>precision</i>	<i>recall</i>	F1
positive	75.2577 %	77.3852 %	76.3066 %
negative	60.9756 %	60.2410 %	60.6061 %

Como podemos comprobar con las agrupaciones de emociones positivas y negativas también notamos una gran mejoría en el Modelo de Transfer Learning, el cual es el que mejor resultados nos ofrece, y la librería DeepFace, la cual está en segundo puesto. Es cierto que si que hay una mejoría en este ámbito en el modelo de Deep Learning respecto al calculado con el conjunto de datos de prueba, que si bien no es un cambio tan notable como los otros dos modelos ML, consideramos que el cambio respecto a los valores con el conjunto de datos de prueba es suficientemente significativo.

7.2.5. Conclusiones

Como hemos podido observar en la sección anterior, el modelo basado en Transfer Learning y la librería DeepFace nos han dado un resultado incluso mejor a la hora de realizar el análisis en tiempo real. A pesar de que hay cambios en los valores, el orden de los modelos ML según su desempeño sigue siendo el mismo, siendo el primero el modelo basado en Transfer Learning, el segundo la librería DeepFace y el último el modelo basado en Deep Learning.

Observando la diferencia de valores con detenimiento se puede decir que el modelo de Transfer Learning y la librería DeepFace ofrecen un cambio significativo, mejorando bastante en todos sus ámbitos los valores de *precision*, *recall* y F1. Para el modelo de Deep Learning, a pesar de que el cambio no es tan notable, se considera que también sufre una mejoría significativa.

Es por esto que una vez finalizado este proceso experimental podemos concluir que las hipótesis H_{1_1} , H_{2_1} y H_{3_1} se confirman.

7.3. Proceso experimental con interfaces de usuario

En este apartado se presenta el diseño de un experimento que tiene como objetivo validar el uso de la herramienta de análisis de expresiones faciales para dar retroalimentación en tiempo real sobre interfaces de usuario adaptativas, de manera que se podrá comprobar si el análisis de expresiones faciales resulta realmente útil para esta tarea.

Nuestro experimento está diseñado para ser ejecutado en una única sesión, con la participación de 12 personas de forma individual. Para cada participante se arrancará el análisis de las expresiones faciales, así como la grabación de pantalla y la grabación de vídeo (siempre que se permita), y se presentarán varias interfaces de usuario las cuales serán adaptaciones de tres interfaces de usuario principales, una tienda *online*, una aplicación de búsqueda de viviendas y la web de una cartelera de cine (Ejemplo de interfaz figura 7.2, para ver el resto ver Anexo B). Con los datos recogidos por todos los participantes obtendremos resultados reales sobre el funcionamiento de la herramienta y su utilidad en el contexto del análisis de expresiones faciales como retroalimentación para la adaptación de interfaces de usuario.

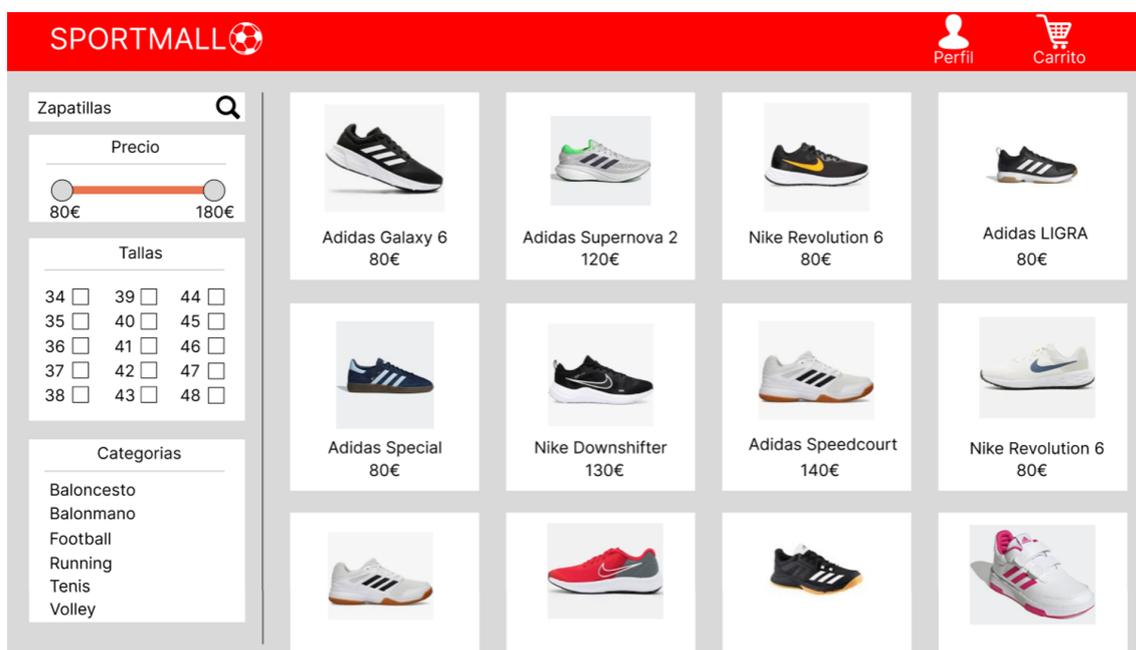


Figura 7.2: Interfaz de usuario UI-1 o disposición en rejilla

7.3.1. Motivación

Durante los últimos años hay una creciente tendencia por el uso de la inteligencia artificial y las técnicas de aprendizaje automático. Si esta tendencia se junta con conceptos que implican las emociones o sentimientos del usuario, como la experiencia de usuario, resulta interesante el uso de estas técnicas de aprendizaje automático para tratar de comprender las emociones del usuario en tiempo real. Esto puede resultar realmente útil a la hora de valorar cuando una interfaz está satisfaciendo las necesidades de un usuario o no. Dentro de este contexto, se podría utilizar el análisis de expresiones faciales en tiempo real para valorar cuando una interfaz de usuario necesita una adaptación o cuando una adaptación previamente hecha convence al usuario.

FacExp es la herramienta de análisis de expresiones faciales en tiempo real que hemos

desarrollado, y que utiliza la librería DeepFace y nuestros dos modelos de implementación propia. Esta herramienta ha sido creada con la finalidad de tratar de conocer los sentimientos del usuario en tiempo real y concretamente para ser implementada como un componente de retroalimentación para un *framework* de adaptación de interfaces inteligente. A pesar de que FacExp puede utilizar los tres modelos mencionados, este experimento lo realizaremos con el modelo de Transfer Learning únicamente, el cual dio los mejores resultados en *precision* y *recall* sobre el conjunto de test y también dio los mejores resultados en tiempo real.

7.3.1.1. Objetivo del experimento

El objetivo de este experimento es determinar si realmente el análisis de expresiones faciales en tiempo real pueda ser de utilidad a la hora de evaluar las posibles adaptaciones que se puedan realizar sobre una interfaz de usuario.

Según el *framework* Objetivo-Pregunta-Métrica (GQM) el objetivo del experimento es el siguiente:

Analizar las expresiones faciales de los usuarios durante el uso en tiempo real de varias interfaces **con el propósito de** determinar si la satisfacción del usuario es similar respecto a la satisfacción medida con un cuestionario **desde el punto de vista de** los investigadores en UX **en el contexto de** un grupo de usuarios realizando tareas en diversas interfaces de usuario.

7.3.1.2. Contexto

El experimento planteado se realizará en un entorno con las siguientes características:

- **Online:** Al requerir del uso de la herramienta para el análisis en tiempo real mientras se realiza el experimento hemos decidido que el experimento debe realizarse de manera presencial para poder nosotros controlar la herramienta.
- **Estudiante:** Los participantes no tienen que ser profesionales ni conocedores de la tecnología, nosotros seremos los que llevemos el control de la herramienta y las interfaces que se presenten serán adaptaciones de una interfaz común como puede ser la de una tienda *online*, un tipo de interfaz a la que todos estamos acostumbrados.
- **Específico:** El experimento trata de analizar las expresiones faciales con un conjunto de interfaces personalizado y con unos modelos de análisis de expresiones faciales específicos.
- **Problema de juguete:** Durante el experimento los sujetos no se expondrán a un problema o situación real, simplemente se les pedirá realizar acciones sencillas sobre las interfaces que se van mostrando mientras se analizan sus expresiones faciales.

7.3.1.3. Aspectos éticos y jurídicos

En el aspecto ético y jurídico, al ser un procedimiento experimental con sujetos deberemos tener en cuenta temas como:

- **Privacidad:** La privacidad del sujeto debe estar garantizada en todo momento, dando la opción de abandonar el proyecto o no realizar parte de el cuando crea que su privacidad peligra o está siendo violada. También se deberán asegurar el anonimato de todos los datos que se vayan a almacenar sobre los sujetos.
- **Consentimiento:** Deberemos explicar todo el proceso a los sujetos antes de comenzar y estos deberán proporcionar su consentimiento explícito.

Teniendo en cuenta estos aspectos podremos garantizar que nuestro proceso experimental es moralmente correcto y además es completamente legal, al menos en cuanto al tratamiento de datos de los sujetos se refiere.

El aspecto económico no supondrá un problema debido a que todo lo utilizado durante la ejecución del experimento es de implementación propia, no se requieren pagar licencias o comprar equipo de ningún tipo.

7.3.2. Planificación del experimento

Durante esta sección se plantearán las hipótesis, definiendo tanto la variable independiente como las dependientes, además de definir el proceso que se llevará a cabo en cada sesión y los sujetos que participaron en dicho proceso experimental. También se comentarán los objetos e instrumentación requeridas y se hará una breve evaluación de la validez.

7.3.2.1. Hipótesis y variables

Como hemos mencionado en el apartado de la motivación el objetivo es comprobar si el análisis de expresiones faciales puede ser de ayuda a la hora de evaluar la satisfacción de un usuario con una interfaz o adaptación de interfaz en concreto.

Para ello se propone una única variable independiente que será la **interfaz de usuario** que podrá tomar valores diferentes para distintos tipos de tareas que se plantearán durante el experimento.

La variable dependiente será la satisfacción del usuario, medida por las emociones detectadas en tiempo real por el modelo ML basado en *Transfer Learning* y además por un cuestionario final donde los usuarios contestaran a preguntas sobre la satisfacción frente a las interfaces presentadas. De esta forma se pretende medir la correlación para la satisfacción mediante medidas objetivas (detección de emociones en tiempo real) y subjetivas (cuestionario sobre satisfacción).

Hipótesis nulas:

- H_{10} : No existe correlación entre la detección de expresiones en tiempo real y el cuestionario de satisfacción para las interfaces.

Hipótesis alternativas:

- H_{11} : Existe correlación entre la detección de expresiones en tiempo real y el cuestionario de satisfacción para las interfaces.

Para determinar si las hipótesis son ciertas compararemos las grabaciones de los resultados del análisis de expresiones en tiempo real con las respuestas en los cuestionarios.

El cuestionario seleccionado para medir la satisfacción de los sujetos frente a las interfaces presentadas será el CSUQ (*Computer Satisfaction Usability Questionnaire*) de IBM: este cuestionario de satisfacción sobre la usabilidad de ordenadores [14] permite a los participantes expresar su grado de satisfacción con la usabilidad de sistemas e interfaces de usuario. Este cuestionario de 19 preguntas se ha validado empíricamente con un gran número de participantes con un conjunto significativo de estímulos [16]. Es aplicable a cualquier tipo de sistema interactivo [15] y proporciona un $\alpha=0.89$ de coeficiente de fiabilidad entre sus resultados y la usabilidad percibida del sistema [14].

7.3.2.2. Diseño del experimento

Como hemos mencionado anteriormente, la herramienta deberá analizar las expresiones faciales del usuario mientras este realiza tareas en un grupo de interfaces, por tanto este experimento se puede realizar en una única sesión.

Lo que si resulta interesante es añadir al grupo de tareas e interfaces que vamos a utilizar un componente de aleatoriedad, de manera que la secuencia de interfaces a mostrar no siempre sea la misma, ya que esto podría distorsionar en cierta manera los resultados. Para ello plantearemos tres tipos de tarea (compra en tienda online, búsqueda de inmuebles y compra de entradas de cine) cuyo orden de aparición será aleatorio, mostrando para cada una de estas tareas tres tipos de interfaces distintas, las cuales también aparecerán en orden aleatorio. Entre cada interfaz se dejará un tiempo de descanso para que el usuario deje de pensar en la interfaz anterior, ya que esto también podría alterar los resultados. (Ver Figura 7.3).

Los efectos **aprendizaje** o **experiencia** no suponen un problema en este experimento, donde no se requieren usuarios expertos ni hay riesgo de que el sujeto pueda aprender nada sobre el funcionamiento del experimento que pueda alterar los resultados. En cuanto al efecto **fatiga**, al utilizar 9 tipos de interfaces con tareas sencillas para cada uno de ellos no se espera que el experimento dure mucho más de 10 minutos, por lo que no debería resultar un problema.

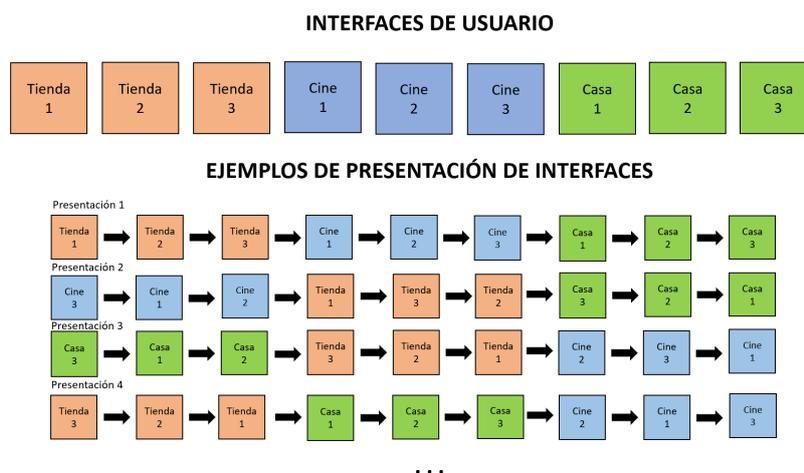


Figura 7.3: Creación de las presentaciones de interfaces mezcladas aleatoriamente

7.3.2.3. Selección de sujetos

Los perfiles seleccionados para llevar a cabo el experimento serán cualquier tipo de persona, sin restricción de edad ni género, y que no requieren de ningún tipo de experiencia en ningún ámbito. Esto se debe a que el experimento se centrará en interfaces que resulten familiares y tareas que no requerirán de conocimiento o experiencia previa.

Requerimos de un total de **12 participantes**. Las sesiones se realizarán de forma individual y presencial y por supuesto no requerirá de una entrevista previa para agrupar por experiencia a los usuarios.

Las personas seleccionadas como sujetos fueron familiares y amigos siguiendo un modelo de bola de nieve.

7.3.2.4. Objetos e instrumentación

Al ser un experimento presencial se dispondrá de una sala de trabajo con todo lo requerido para la realización de la sesión, sin la necesidad de que los sujetos tengan que aportar nada.

En el lugar del experimento se dispondrá de un equipo con la herramienta **FacExp** para análisis de expresiones faciales, configurada para analizar con el modelo de Transfer Learning. El equipo estará conectado a dos monitores, uno para el sujeto donde se mostrarán las interfaces y otro para el experimentador donde se encontrará la aplicación con el análisis en vivo para que pueda manejarla. La separación en dos monitores permitirá que no se desvíe la atención del sujeto durante la sesión. Por último el equipo estará conectado a una webcam que será la encargada de tomar el vídeo para el análisis.

Los participantes entrarán al lugar del experimento donde se les dará una explicación del experimento, se les mostrará brevemente el funcionamiento de la herramienta. Por

ultimo antes de comenzar con el experimento, se tomará el consentimiento explícito de los participantes, mediante la firma de un documento, para permitirnos grabar el vídeo de la webcam para contrastar los datos.

7.3.2.5. Evaluación de la validez

Las amenazas a la validez interna cuestionan si realmente el tratamiento de las sesiones tendrá el efecto esperado. Ejemplos de este tipo de amenazas podrían ser la falta de voluntarios o el efecto fatiga, aunque en nuestro caso no deberíamos preocuparnos por ello, puesto que al no requerir ningún tipo de experiencia y no tener ningún tipo de límite a la participación del experimento podremos conseguir 12 participantes de forma sencilla y el efecto fatiga no debería de aparecer en sesiones tan cortas como las que se plantean en este experimento.

Las amenazas a la validez externa tratan sobre la posible generalización de los resultados. Realmente como estamos probando una herramienta de análisis de expresiones faciales en específico sobre un conjunto de interfaces personalizado no deberíamos de tener este problema. Existen pocas posibilidades de obtener datos incompletos o sesgados.

Las amenazas a la validez de constructo son los tipos de amenazas que surgen cuando las métricas que se utilizan para medir las variables dependientes no son las correctas. Una posible amenaza de este tipo sería que el cuestionario no recoja correctamente la satisfacción de usuario del sujeto, y por tanto no se pueda realizar una comparación correcta con sus emociones. Realmente esto no nos puede suponer un problema al estar utilizando el cuestionario CSUQ de IBM, debido a su validez y porcentaje de fiabilidad.

Por último las amenazas a la validez de conclusiones son las que pueden causar que las conclusiones no resulten estadísticamente válidas. En este caso podría suceder que las emociones suscitadas por una interfaz persistieran en la siguiente interfaz o el contraste de emociones entre dos interfaces fuera tan fuerte que distorsionara el análisis. Para lidiar con esto introducimos los descansos entre interfaces y el orden aleatorio de la secuencia de interfaces, de manera que esto no debería suponer un problema.

7.3.3. Ejecución del experimento

El experimento fue llevado a cabo según lo planeado con 12 personas cuya edad va desde los 22 años hasta los 59 años. Entre las personas seleccionadas para el experimento encontramos 3 mujeres, 9 hombres y 0 personas no binarias.

Al no requerir ningún tipo de entrenamiento para realizar el experimento la sesión comienza directamente para cada uno de los sujetos después de que rellenen un cuestionario demográfico que nos permite tener información del usuario y de firmar el consentimiento para participar en el experimento. Tras una breve explicación de la actividad se comienza a analizar las expresiones faciales del usuario con la herramienta y se comienza a presentar las interfaces una a una. Para cada una de las interfaces presentadas se pide al usuario

que realice una tarea en concreto, de forma que este tenga que fijarse bien en la interfaz. Una vez finalizadas todas las tareas se finaliza el análisis de las expresiones faciales y la herramienta exporta las grabaciones de webcam y pantalla además del análisis en formato *.CSV* para su posterior análisis de datos.

Por último se le entrega al usuario el cuestionario CSUQ para cada una de las interfaces con las que ha trabajado durante la sesión, de forma que al rellenarlo podremos comparar los resultados de satisfacción con la usabilidad con las emociones percibidas por la herramienta de análisis de expresiones faciales.

7.3.4. Análisis de datos e interpretación

Para poder comparar la satisfacción de los usuarios con la usabilidad de las interfaces de usuario con sus emociones durante el uso de éstas se va a utilizar una plantilla de Excel que contiene las formulas para medir la satisfacción con las preguntas del cuestionario CSUQ. Una vez calculada la satisfacción de la usabilidad con cada interfaz se comparará con el análisis de las expresiones faciales durante el uso de dicha interfaz. Para ello vamos a crear una tabla con el promedio de la nota del CSUQ por cada usuario e interfaz de usuario (Ver Tabla 7.19) y otra tabla con la emoción predominante de cada usuario por cada interfaz de usuario (Ver Tabla 7.20). Cabe destacar que en este experimento tomaremos neutral como una reacción positiva puesto que la mayoría de los usuarios adoptan una expresión neutral mientras trabajan. Es decir, en este experimento solo se tomarán las reacciones negativas como indicio de que el usuario no está satisfecho y de lo contrario, se asumirá que está satisfecho.

Tabla 7.19: Tabla de resultados CSUQ por usuario e interfaz

	UI-1	UI-2	UI-3	UI-4	UI-5	UI-6	UI-7	UI-8	UI-9
S01	6.43	6.62	3.06	5.5	5.93	2.87	5.56	5.81	4.75
S02	6.83	6.83	2.5	6.94	6.94	5.22	7	5.33	4.77
S03	6.43	5.72	3	5.5	5.93	2.93	5.56	5.81	4.81
S04	6.11	6.11	6.22	5.27	5.38	2	5.33	5.44	5.44
S05	6.27	7	2	5.88	6.94	5.25	6.94	5.72	2.77
S06	6.62	5.75	2.62	6.43	6.37	5.94	6.93	5.75	4.93
S07	6.17	6.7	2.76	6.82	6.7	4.64	6.7	5.35	4.05
S08	5.16	5.83	5.94	6.22	6	6.27	6.22	6.16	6.16
S09	6.72	6.61	3.61	6.22	6.27	5.55	6.88	6.5	3.94
S10	6.16	6.61	2.44	6.33	6.27	5.55	6.88	6.83	4.22
S11	6.82	5.75	2.53	5.88	5.93	2	5.72	5.33	2
S12	5.78	6.11	3.06	5.5	6	2.87	6.62	5.16	4.05

Tabla 7.20: Tabla de emociones dominantes por usuario e interfaz

	UI-1	UI-2	UI-3	UI-4	UI-5	UI-6	UI-7	UI-8	UI-9
S01	happy	happy	sad	neutral	happy	angry	neutral	neutral	neutral
S02	happy	happy	sad	happy	neutral	sad	happy	neutral	neutral
S03	happy	neutral	sad	neutral	happy	sad	disgust	neutral	neutral
S04	neutral	neutral	neutral	neutral	neutral	disgust	neutral	neutral	neutral
S05	happy	happy	angry	neutral	happy	happy	neutral	neutral	angry
S06	happy	neutral							
S07	neutral	happy	sad	neutral	neutral	sad	happy	neutral	sad
S08	neutral								
S09	happy	happy	sad	neutral	neutral	happy	neutral	happy	sad
S10	neutral	happy	happy	happy	happy	neutral	neutral	happy	neutral
S11	happy	neutral	disgust	neutral	neutral	sad	neutral	neutral	angry
S12	neutral	happy	sad	neutral	happy	angry	neutral	neutral	sad

Tras exponer los datos en esta forma tabular, ahora debemos realizar una prueba de normalidad para cada una de las interfaces de usuario en ambas tablas, de esta forma sabremos que tipo de correlación deberemos aplicar (*Pearson, Spearman, Kendall, etc...*). Como queremos calcular la correlación entre la nota de CSUQ obtenida y la emoción del usuario, tendremos que ver si ambas variables de cada una de las interfaces de usuario siguen una distribución normal o no para aplicar un tipo de correlación u otra. En el caso de las emociones, para poder calcular su normalidad y correlación se han ordenado en un rango del 0 al 6 siendo 0 la más negativa y 6 la más positiva (0-*angry*, 1-*disgust*, 2-*sad*, 3-*fear*, 4-*neutral*, 5-*surprise*, 6-*happy*).

A continuación presentamos una tabla (Ver Tabla 7.21) donde aparece por cada interfaz de usuario el p-valor obtenido para cada una de las interfaces de usuario en cuanto a nota del CSUQ y emoción presentada por el usuario al aplicar la prueba de normalidad de *Shapiro-Wilk*. En *Shapiro-Wilk* cuando el p-valor es mayor de 0.05 podemos concluir que la variable sigue una distribución normal.

Tabla 7.21: p-valores del test de normalidad de *Shapiro-Wilk* por cada interfaz

	P-valor CSUQ	P-valor emociones
UI-1	0.19	0.00
UI-2	0.08	0.00
UI-3	0.00	0.21
UI-4	0.50	0.00
UI-5	0.32	0.00
UI-6	0.05	0.21
UI-7	0.01	0.00
UI-8	0.16	0.00
UI-9	0.56	0.00

Como podemos observar no hay ningún caso en el que una interfaz siga una distribución normal en los dos casos, por lo que no podremos aplicar la correlación de *Pearson*, debido a que es para variables que presenten una distribución normal. Utilizaremos la correlación de *Spearman* para cada una de las variables. Los valores obtenidos para la correlación utilizando la correlación de *Spearman* se presentan a continuación (Ver Tabla 7.22)

Tabla 7.22: Valores del test de correlación de *Spearman* para cada interfaz

	<i>Spearman</i>
UI-1	0.85
UI-2	0.83
UI-3	0.43
UI-4	0.52
UI-5	0.51
UI-6	0.78
UI-7	0.48
UI-8	0.65
UI-9	0.88

El rango de valores de la correlación de *Spearman* va de -1 a 1, indicando los valores próximos a -1 una correlación fuerte negativa y los valores próximos a 1 una correlación fuerte positiva. En nuestro caso podemos observar que para cada interfaz de usuario se obtiene una correlación positiva, que varía entre moderada en los peores casos (por ejemplo 0.43) hasta fuerte e incluso muy fuerte en los mejores casos.

7.3.5. Conclusiones

Como hemos podido observar en el apartado anterior, tras realizar la correlación de *Spearman* observamos que hay una correlación positiva entre la nota obtenida por cada interfaz de usuario en el cuestionario CSUQ y las emociones que presenta el usuario al momento de utilizar dichas interfaces de usuario, que varía entre moderada y fuerte. Esto nos indica que el análisis de expresiones faciales en tiempo real nos puede ayudar a evaluar la experiencia que esta teniendo el usuario al momento de interactuar con una interfaz de usuario.

Es por esto que una vez finalizado este proceso experimental podemos concluir que la hipótesis $H1_1$ se confirma.

Capítulo 8

Conclusiones y Trabajos Futuros

Para finalizar con el trabajo, en este capítulo presentaremos las conclusiones, además de comentar alguno de los trabajos futuros, recalcando la utilidad de esta herramienta dentro del contexto en el que se encuentra.

8.1. Conclusiones

Como hemos ido comentando a lo largo del trabajo, se ha desarrollado una herramienta cuya finalidad es el análisis de las expresiones faciales del usuario en tiempo real. Para ello se realiza una búsqueda entre las librerías de reconocimiento de expresiones faciales para Python, tomando finalmente de entre varias opciones la librería DeepFace, que resulta conveniente para este análisis de expresiones en tiempo real.

Tras esta búsqueda de librerías se opta por la implementación de nuestros propios modelos ML dedicados al análisis de expresiones faciales. Se decide realizar un modelo utilizando Transfer Learning y otro utilizando Deep Learning, más concretamente creando una red neuronal con módulo de atención (ACNN). Tras varios entrenamientos no conseguimos que el modelo basado en Deep Learning nos ofrezca los resultados esperados, pero en cambio conseguimos que el modelo basado en Transfer Learning supere el desempeño de la librería DeepFace.

A la hora de implementar la herramienta de análisis de expresiones faciales, se decide crear un aplicación de escritorio que nos permita utilizar los modelos en tiempo real para analizar las expresiones faciales del usuario, además de grabar los resultados en un archivo *.CSV* y grabar tanto un vídeo del usuario con la webcam como un vídeo de lo que ocurre en la pantalla mientras se realiza el análisis de expresiones faciales. Además de todo esto, implementamos el *backend* de esta aplicación en forma de API, de forma que esta puede ser accedida por otras herramientas, programas y aplicaciones para aprovechar el análisis de las expresiones faciales del usuario en tiempo real.

Posteriormente se realiza un proceso experimental cuya finalidad es comprobar si los

modelos ML que utiliza nuestra herramienta tienen un desempeño similar a el calculado con un conjunto de datos de prueba. Tras la realización de este experimento se concluye que los resultados para el desempeño no son similares a los previamente calculados, sino que en tiempo real los modelos nos ofrecen mejores valores para la *precision*, el *recall* y el F1.

Por último se realiza otro proceso experimental que pretende comprobar si nuestra herramienta puede ser útil a la hora de evaluar la experiencia de usuario en tiempo real a través de las expresiones faciales del usuario. Al concluir el experimento podemos contemplar como los resultados de los cuestionarios realizados por los usuarios para cada interfaz están correlacionados con la emoción dominante que presenta el usuario durante el uso de la misma.

Además de todo este trabajo, cabe destacar que la herramienta ya ha sido utilizada en un proceso experimental para recoger la mayor cantidad de datos biométricos posibles durante la interacción de un usuario con una interfaz de usuario. En este experimento se utilizó tanto la herramienta FacExp para el reconocimiento de las expresiones faciales en tiempo real como un casco neuronal (EEG) para recoger datos de las ondas cerebrales, un *smartwatch* para medir el pulso y un dispositivo de seguimiento ocular, entre otras cosas. También se ha utilizado en un trabajo para la validación de interfaces de usuario, el cual se puede encontrar en el TFG de Isabel Santonja Soto [26].

8.2. Adaptación de interfaces mediante datos biométricos

Como hemos podido observar en el proceso experimental con interfaces de usuario, muchas veces las emociones del usuario van ligadas a su experiencia de usuario a la hora de utilizar interfaces, pudiendo obtener fácilmente una retroalimentación de como esta siendo la experiencia de usuario con una aplicación o interfaz de usuario. Esta retroalimentación no es la única que se puede obtener, existen muchas formas de obtener una retroalimentación de la experiencia de usuario en tiempo real basado en sus datos biométricos, por ejemplo utilizando seguimiento ocular, un casco neuronal (EEG) para medir las ondas cerebrales, un *smartwear* que nos permita conocer la tensión y pulsaciones del usuario, etc... Uno de los trabajos que quedan para el futuro es la interacción de la herramienta con un *framework* de adaptación de interfaces de usuario.

En trabajos previos se ha propuesto un *framework* de Adaptación Inteligente de IUs [1, 5] que utiliza aprendizaje por refuerzo basado en modelos para adaptarse de manera inteligente y en tiempo real. Una de las principales características de esta propuesta se encuentra en la capacidad de adaptarse a cambios en el contexto, es decir cambios en la plataforma, el entorno y el usuario. Además, el *framework* permite aprender basándose en la historia de interacción y en información biométrica (EEG, seguimiento ocular, etc.) y preferencias del usuario, lo que le permite ofrecer una interfaz de usuario personalizado y mejorar su experiencia. Uno de los desafíos actuales para esta solución es la falta de información continua sobre el *estado emocional* del usuario durante la interacción con la

interfaz. La herramienta que presentamos ofrece *retroalimentación* del estado emocional del usuario al *framework* durante la interacción, mejorando así su capacidad de adaptación personalizada.

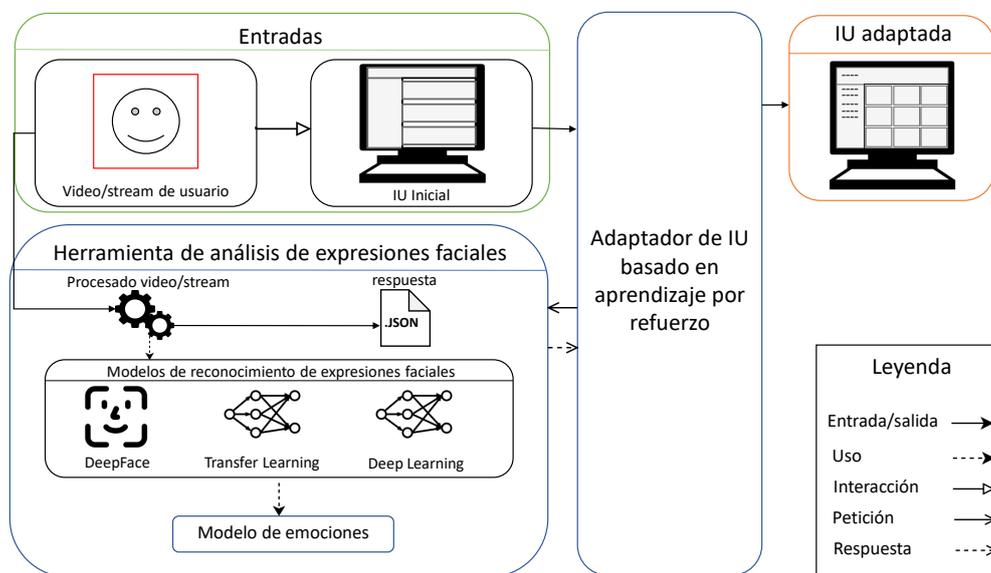


Figura 8.1: Estrategia de Adaptación Inteligente de IUs

En la Figura 8.1 se muestra la arquitectura de esta solución. Se parte de la interacción del usuario con una *IU inicial*, se captura la *webcam* en vídeo. A continuación, se procesa el vídeo usando la herramienta, que utiliza uno o varios modelos de detección de expresiones faciales, y que basado en un modelo de emociones clasifica el estado de ánimo del usuario. La herramienta funciona como una API que recibe peticiones desde el *framework* de adaptación y devuelve un *JSON* con la información de las emociones detectadas. Esto permite al *framework* conocer el estado emocional del usuario y determinar si una adaptación podría ser necesaria o si una adaptación (ej. *IU adaptada*) ha sido exitosa o no.

Por último comentar que tanto el trabajo aquí mostrado [4] como otro, relacionado con el *framework* de adaptación inteligente de interfaces de usuario con el que se pretende conectar la herramienta [6], serán presentados en las JISBD (Jornadas de Ingeniería del Software y Bases de Datos) del día 12 al día 14 de septiembre, en Ciudad Real, España.

Estas JISBD constituyen un foro de encuentro de referencia donde investigadores y profesionales de España, Portugal e Iberoamérica en los campos de la Ingeniería del Software y de las Bases de Datos pueden debatir e intercambiar ideas, crear sinergias y, sobre todo, conocer la investigación que se está llevando a cabo en nuestra comunidad. Es por eso que se ha decidido presentar el trabajo de la herramienta de análisis de expresiones faciales en el contexto de la adaptación inteligente de interfaces de usuario [4] con el fin de exponer el trabajo a la comunidad y poder recibir retroalimentación y nuevas ideas de sus integrantes.

8.3. Trabajo futuro

Como trabajo futuro se pretende mejorar la experiencia de usuario de la herramienta, añadiendo mensajes de error y mejorando la forma de seleccionar las carpetas donde guardar los archivos. También se plantea una mejora de la estética de la aplicación de escritorio de la herramienta para mejorar la experiencia de usuario.

Además se añadirá una función para dar soporte a procesos experimentales, en los cuales habitualmente se definen ciertas tareas a realizar por los sujetos. La funcionalidad que se pretende implementar nos permitirá registrar el inicio y final de estas tareas durante el análisis, generando un archivo al finalizar que presente la emoción dominante de cada tarea realizada durante el análisis.

Queda pendiente también conectar la herramienta propuesta a un *framework* de adaptación inteligente de interfaces de usuario propuesto en trabajos anteriores [1, 5], cosa que se explica más ampliamente en la anterior sección.

Por último, aunque los resultados de la herramienta son buenos, seguiremos estudiando como mejorar el desempeño de los modelos ML de análisis de expresiones faciales, ya sea con re-entrenamientos de dichos modelos ML o con la creación de otros nuevos que se puedan añadir a la herramienta.

Bibliografía

- [1] Silvia Abrahão et al. “Model-based intelligent user interface adaptation: challenges and future directions”. En: *Software and Systems Modeling* 20.5 (2021), págs. 1335-1349.
- [2] Monil Bansal, Sampriti Yadav y Dinesh K. Vishwakarma. “A Language-Independent Speech Sentiment Analysis Using Prosodic Features”. En: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. 2021, págs. 1210-1216. DOI: 10.1109/ICCMC51019.2021.9418357.
- [3] Paul Buitelaar et al. “MixedEmotions: An Open-Source Toolbox for Multimodal Emotion Analysis”. En: *IEEE Transactions on Multimedia* 20.9 (2018), págs. 2454-2465. DOI: 10.1109/TMM.2018.2798287.
- [4] Fernando Carceller et al. “Análisis de expresiones faciales para la adaptación inteligente de interfaces de usuario”. En: *JISBD2023*. SISTEDES, 2023 (enviado en Abril).
- [5] Daniel Gaspar-Figueiredo, Silvia Abrahao y Emilio Insfran. “Hacia un Framework de Adaptación Inteligente de Interfaces de Usuario”. En: *JISBD2022*. SISTEDES, 2022.
- [6] Daniel Gaspar-Figueiredo et al. “Entorno para agentes de aprendizaje por refuerzo para adaptar interfaces de usuario”. En: *JISBD2023*. SISTEDES, 2023 (enviado en Abril).
- [7] Tony Gorschek et al. “A Model for Technology Transfer in Practice”. En: *IEEE Software* 23.6 (2006), págs. 88-95. DOI: 10.1109/MS.2006.147.
- [8] Margherita Grandini, Enrico Bagli y Giorgio Visani. *Metrics for Multi-Class Classification: an Overview*. 2020. arXiv: 2008.05756 [stat.ML].
- [9] Marc Herrmann et al. “On the Subjectivity of Emotions in Software Projects: How Reliable are Pre-Labeled Data Sets for Sentiment Analysis? (Summary)”. En: *Software Engineering 2023*. Ed. por Gregor Engels, Regina Hebig y Matthias Tichy. Bonn: Gesellschaft für Informatik e.V., 2023, págs. 67-68.
- [10] Weidong Huang et al. “Sens-e-Motion: Capturing and Visualising Emotional Status of Computer Users in Real Time”. En: *2019 23rd International Conference in Information Visualization – Part II*. 2019, págs. 96-99. DOI: 10.1109/IV-2.2019.00028.

- [11] Lakshmish Kaushik, Abhijeet Sangwan y John H. L. Hansen. “Sentiment extraction from natural audio streams”. En: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, págs. 8485-8489. DOI: 10.1109/ICASSP.2013.6639321.
- [12] Chang-Min Kim et al. “Real-Time Streaming Image Based PP2LFA-CRNN Model for Facial Sentiment Analysis”. En: *IEEE Access* 8 (2020), págs. 199586-199602. DOI: 10.1109/ACCESS.2020.3034319.
- [13] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. “Deep learning”. En: *nature* 521.7553 (2015), págs. 436-444.
- [14] James R Lewis. “IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use”. En: *International Journal of Human-Computer Interaction* 7.1 (1995), págs. 57-78.
- [15] James R Lewis. “Psychometric evaluation of the PSSUQ using data from five years of usability studies”. En: *International Journal of Human-Computer Interaction* 14.3-4 (2002), págs. 463-488.
- [16] James R Lewis. “Sample Sizes for Usability Tests: Mostly Math, Not Magic. interactions 13, 6 (Nov. 2006), 29–33”. En: *Google Scholar Google Scholar Digital Library Digital Library* (2006).
- [17] Robert Marcec y Robert Likic. “Using Twitter for sentiment analysis towards AstraZeneca/Oxford, Pfizer/BioNTech and Moderna COVID-19 vaccines”. En: *Postgraduate Medical Journal* 98.1161 (ago. de 2021), págs. 544-550. ISSN: 0032-5473. DOI: 10.1136/postgradmedj-2021-140685.
- [18] Shervin Minaee, Mehdi Minaei y Amirali Abdolrashidi. “Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network”. En: *Sensors* 21.9 (2021). ISSN: 1424-8220.
- [19] Saif M. Mohammad. “9 - Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text”. En: *Emotion Measurement*. Ed. por Herbert L. Meiselman. Woodhead Publishing, 2016, págs. 201-237. ISBN: 978-0-08-100508-8. DOI: <https://doi.org/10.1016/B978-0-08-100508-8.00009-6>.
- [20] Aicha Nouisser, Ramzi Zouari y Monji Kherallah. “Enhanced MobileNet and transfer learning for facial emotion recognition”. En: *ACIT*. 2022, págs. 1-5.
- [21] Yahia Said y Mohammad Barr. “Human emotion recognition based on facial expressions via deep learning on high-resolution images”. En: *Multimedia Tools and Applications* 80.16 (2021), págs. 25241-25253.
- [22] Mary Sánchez-Gordón y Ricardo Colomo-Palacios. “Taking the emotional pulse of software engineering — A systematic literature review of empirical studies”. En: *Information and Software Technology* 115 (2019), págs. 23-43. ISSN: 0950-5849.
- [23] Sefik Ilkin Serengil y Alper Ozpinar. “HyperExtended LightFace: A Facial Attribute Analysis Framework”. En: *ICEET*. 2021, págs. 1-4. DOI: 10.1109/ICEET53442.2021.9659697.

- [24] Phillip Shaver et al. “Emotion knowledge: further exploration of a prototype approach.” En: *Journal of personality and social psychology* 52.6 (1987), pág. 1061.
- [25] Yusuke Shinyama, Yoshitaka Arahori y Katsuhiko Gondow. “CodeCAM: capturing programmer’s reaction during coding session”. En: *2018 IEEE Workshop on Mining and Analyzing Interaction Histories (MAINT)*. 2018, págs. 1-5. DOI: 10.1109/MAINT.2018.8323087.
- [26] Isabel Santonja Soto. “Evaluación de Usabilidad de Interfaces de Usuario: Un Enfoque Combinado de Evaluación Heurística y Testing con Usuarios”. En: *Trabajo de fin de grado, Grado en Ingeniería Informática, Escuela Técnica Superior de Ingeniería Informática*. Universitat Politècnica de València, Septiembre 2023 (a ser presentado).
- [27] Yaniv Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. En: *2014 IEEE CVPR*. 2014, págs. 1701-1708.
- [28] Lisa Torrey y Jude Shavlik. “Transfer learning”. En: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, págs. 242-264.
- [29] Monu Verma et al. “LEARNet: Dynamic Imaging Network for Micro Expression Recognition”. En: *IEEE Transactions on Image Processing* 29 (2020), págs. 1618-1627. DOI: 10.1109/TIP.2019.2912358.
- [30] Wei Wei, Qingxuan Jia y Gang Chen. “Real-time facial expression recognition for affective computing based on Kinect”. En: *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. 2016, págs. 161-165. DOI: 10.1109/ICIEA.2016.7603570.

Anexos

Apéndice A

Formulario de consentimiento y cuestionario demográfico

UPV-DSIC

Universitat Politècnica de València
Departamento de Sistemas
Informáticos y Computación

CONSENTIMIENTO A PARTICIPAR EN UN ESTUDIO EXPERIMENTAL

Título del Estudio: Estudio de reconocimiento de expresiones faciales
Investigador: Fernando Carceller Llorens
Conductores del experimento:
Correo de contacto: fercarllorens@gmail.com

Gracias por considerar participar en esta investigación. La persona encargada del experimento debe explicarle el proyecto antes de que acepte colaborar. Si tiene cualquier pregunta relativa a la hoja informativa o a la explicación proporcionada, pregunte a la persona que está llevando a cabo el experimento antes de decidir si desea participar. Si lo necesita, se le proporcionará una copia de este formulario de consentimiento para que la guarde y pueda utilizarla en cualquier momento.

Usted entiende que, si decide en cualquier momento a lo largo del experimento que ya no desea seguir participando en él, puede avisar a los investigadores involucrados y abandonarlo inmediatamente.

Consiente en el procesamiento de su información personal para las necesidades de este estudio de investigación. Esta información será considerada como estrictamente confidencial y tratada de acuerdo con lo recogido en las políticas de protección de datos de la España y Reglamento General de Protección de Datos. (GDPR).

- Por favor, marque esta casilla si acepta que los investigadores se pongan en contacto con usted en lo relativo a participar en experimentos futuros.
Si así lo desea, por favor, indique su dirección de e-mail:

Declaración del participante:

Yo,, acepto que el proyecto de investigación nombrado anteriormente me ha sido explicado satisfactoriamente y acepto formar parte del estudio. He leído tanto los datos escritos anteriormente como la hoja informativa del proyecto, y entiendo lo que conlleva el estudio en el que acepto participar.

Firmado:

Fecha:

Declaración del conductor del experimento:

Yo, Fernando Carceller Llorens, confirmo que he explicado cuidadosamente la naturaleza, requerimientos y cualquier posible riesgo (cuando sea aplicable) de la investigación propuesta a la persona voluntaria.

Firmado:

Fecha:

Figura A.1: Formulario de consentimiento para participar en el proceso experimental

UPV-DSIC

Universitat Politècnica de València
Departamento de Sistemas
Informáticos y Computación

CUESTIONARIO PARA PARTICIPAR EN UN ESTUDIO DE EXPERIMENTOS

Título del Estudio: Estudio de reconocimiento de expresiones faciales
Investigador: Fernando Carceller Llorens
Experimentadores:
Correo de contacto: fercarllorens@gmail.com

Datos personales

Código anónimo: (rellenado por el experimentador)
Nombre: (será anonimizado posteriormente)
Apellido:
Género:
Edad: (años)
Ocupación: Estudiante Ejecutivo Empleado
 Autónomo Jubilado Desempleado
 Otro: (especifique por favor)
Sector de ocupación: (e.g., administración, dirección, finanza, técnico)
Antecedentes: (sector de tu último diploma, e.g., Master, Grado, ...)

Uso un ordenador frecuentemente: En desacuerdo 1 2 3 4 5 6 7 De acuerdo
Uso un Smartphone frecuentemente: En desacuerdo 1 2 3 4 5 6 7 De acuerdo
Uso una Tablet frecuentemente: En desacuerdo 1 2 3 4 5 6 7 De acuerdo
Uso una videoconsola frecuentemente: En desacuerdo 1 2 3 4 5 6 7 De acuerdo

Figura A.2: Cuestionario demográfico previo a la ejecución del proceso experimental

Apéndice B

Interfaces de usuario del proceso experimental

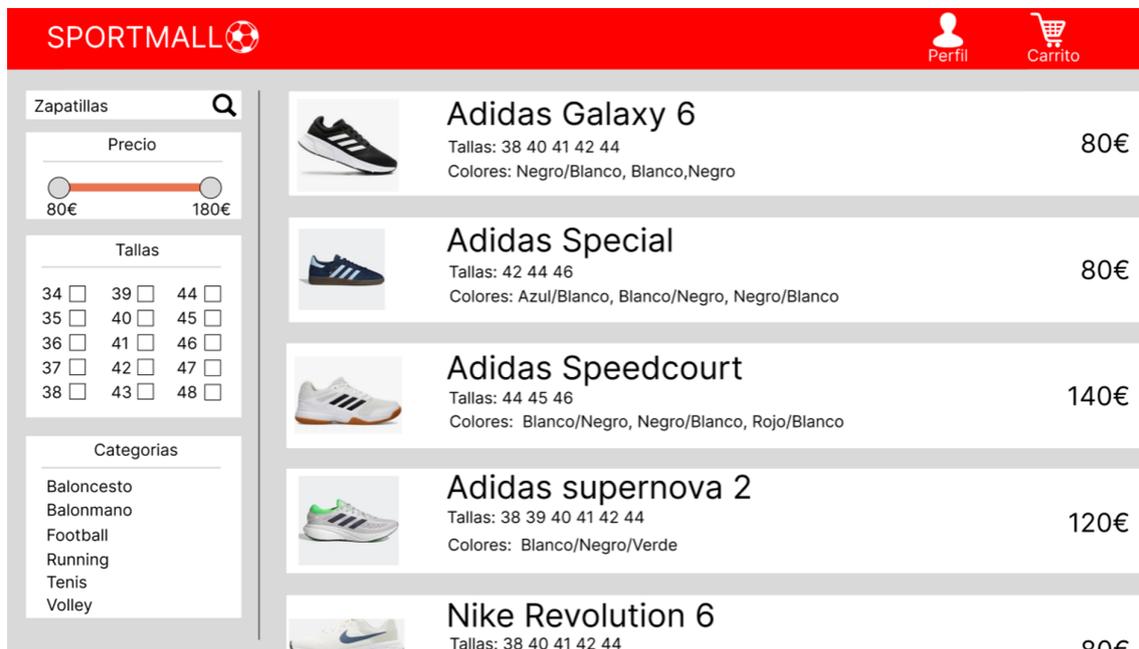


Figura B.1: Interfaz de usuario UI-2 o disposición en lista

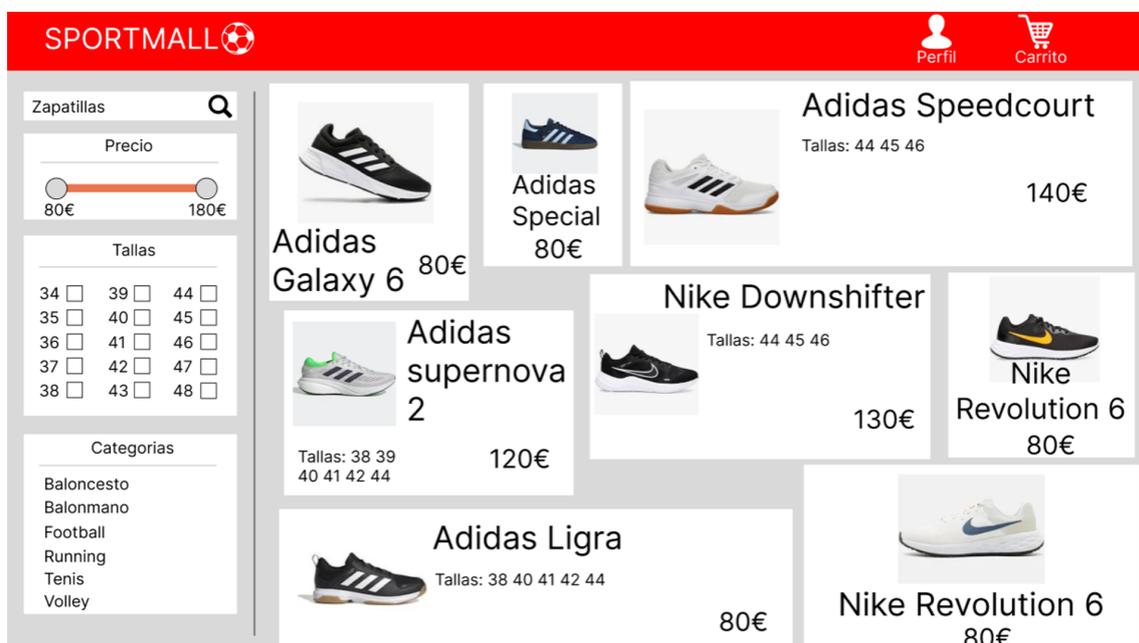


Figura B.2: Interfaz de usuario UI-3 o disposición desordenada

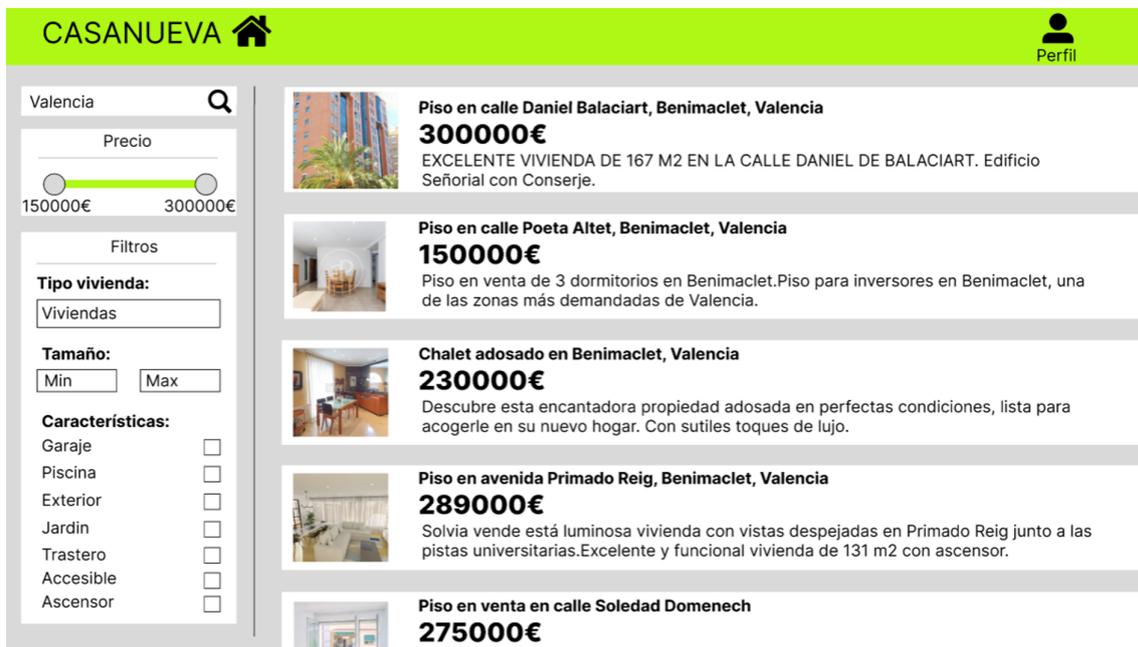


Figura B.3: Interfaz de usuario UI-4 o tema claro

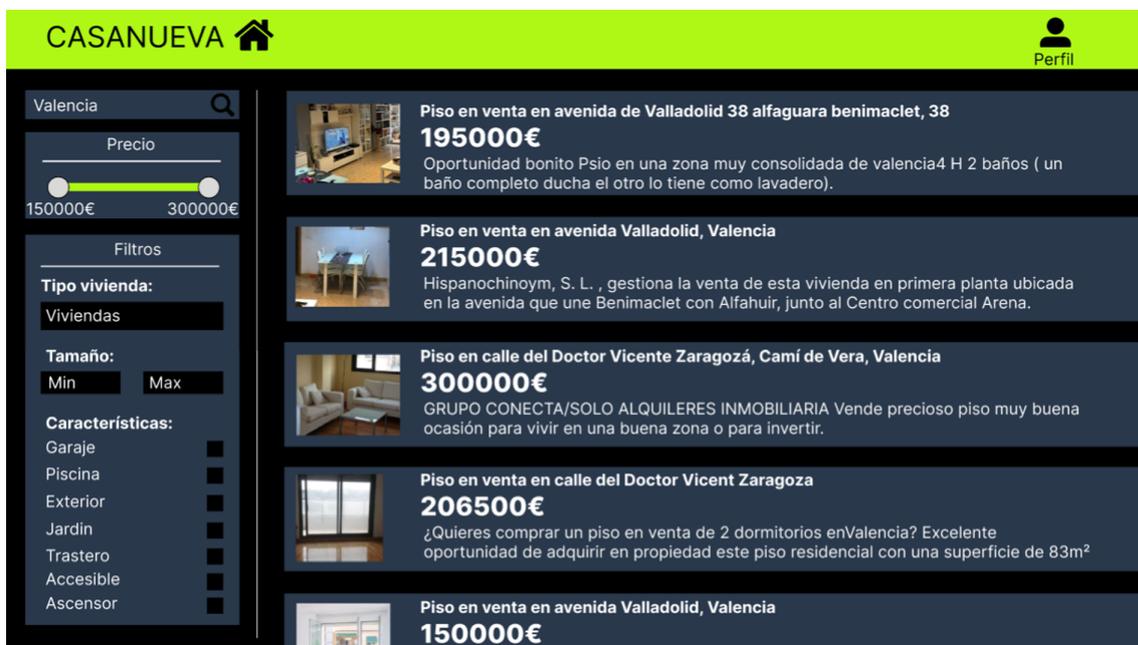


Figura B.4: Interfaz de usuario UI-5 o tema oscuro



Figura B.5: Interfaz de usuario UI-6 o tema de alto contraste



Figura B.6: Interfaz de usuario UI-7 o letra normal

CINEMAX  Perfil



Gran Turismo

La película narra la historia de un joven jugador de Gran Turismo

Tickets:

sala 4	15:00	17:00	21:00	
sala 6	16:00	18:00	20:00	22:00
sala 21	15:30	18:00	22:00	
sala 25	15:00	17:00	20:00	23:00



Elemental

El largometraje número 27 de Pixar nos traslada a una imaginaria ciudad, Ciudad Elemento

Tickets:

sala 1	15:00	17:00	21:00	
sala 2	16:00	18:00	20:00	22:00
sala 10	15:30	18:00	22:00	
sala 12	15:00	17:00	20:00	23:00



Oppenheimer

Película sobre Julius Robert Oppenheimer, físico teórico padre de la bomba atómica.

Tickets:

sala 3	15:00	17:00	21:00	
sala 7	16:00	18:00	20:00	22:00
sala 13	15:30	18:00	22:00	
sala 19	15:00	17:00	20:00	23:00



Quatre raons

Embarazada de veintisiete semanas, María Cuixart, es víctima de una agresión y entra en coma

Tickets:

sala 5	15:00	17:00	21:00	
sala 15	16:00	18:00	20:00	22:00
sala 18	15:30	18:00	22:00	
sala 23	15:00	17:00	20:00	23:00

Figura B.7: Interfaz de usuario UI-8 o letra grande

CINEMAX  Perfil



Insidious la puerta roja

Para enterrar sus demonios de una vez por todas, Josh y un ya universitario Dalton deberán profundizar en El Mas Allá más que nunca, enfrentándose al oscuro pasado familiar y a un huésped de nuevos horrores terroríficos que acechan tras la puerta roja.

Tickets:

sala 4	15:00	17:00	21:00	
sala 6	16:00	18:00	20:00	22:00
sala 21	15:30	18:00	22:00	
sala 25	15:00	17:00	20:00	23:00



Misión imposible: sentencia mortal

En "Misión: Imposible- Sentencia Mortal Parte I" Ethan Hunt (Tom Cruise) y su equipo del FMI se embarcan en su misión más peligrosa hasta la fecha: Localizar, antes de que caiga en las manos equivocadas, una nueva y terrorífica arma que amenaza a toda la humanidad. En esta tesitura, y con unas fuerzas oscuras del pasado de Ethan acechando, comienza una carrera mortal alrededor del mundo en la que está en juego el control del futuro y el destino del planeta. Enfrentado a un enemigo misterioso y todopoderoso, Ethan se ve obligado a considerar que nada puede anteposeerse a su misión, ni siquiera las vidas de aquellos que más le importan.

Tickets:

sala 1	15:00	17:00	21:00	
sala 2	16:00	18:00	20:00	22:00
sala 10	15:30	18:00	22:00	
sala 12	15:00	17:00	20:00	23:00



Indiana Jones and the dial del destino

El arqueólogo Indiana Jones deberá emprender otra aventura contra el tiempo para intentar recuperar un dial legendario que puede cambiar el curso de la historia. Acompañado por su ahijada, Jones pronto se encuentra enfrentándose a Jürgen Voller, un ex nazi que trabaja para la NASA.

Tickets:

sala 3	15:00	17:00	21:00	
sala 7	16:00	18:00	20:00	22:00
sala 13	15:30	18:00	22:00	
sala 19	15:00	17:00	20:00	23:00



Megalodón 2: La Fosa

Prepárate para el subidón de adrenalina definitivo este verano en "Megalodón 2: La fosa", un emocionante viaje al fondo marino que lleva la acción a alturas más altas y profundidades aún mayores con las criaturas más terribles del mundo. Sumérgete en aguas inexploradas con Jason Statham y Wu Jing mientras lideran un audaz equipo de investigación, en una inmersión exploratoria en las profundidades del océano. Su viaje se convierte en una espiral de caos cuando una malévol operación minera amenaza su misión y les obliga a librar una arriesgada batalla por la supervivencia. Enfrentados a megalodones colosales y a implacables saqueadores del medio ambiente, nuestros héroes deben completar su objetivo, ser más astutos y nadar mejor que sus despiadados depredadores en una trepidante carrera contrarreloj.

Tickets:

sala 5	15:00	17:00	21:00	
sala 15	16:00	18:00	20:00	22:00
sala 18	15:30	18:00	22:00	
sala 23	15:00	17:00	20:00	23:00

Figura B.8: Interfaz de usuario UI-9 o letra pequeña

Apéndice C

Objetivos de desarrollo sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.			X	
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Este trabajo presenta una innovación por varios factores como puede ser la posibilidad de dar información a aplicaciones o herramientas externas del estado anímico del usuario (por ejemplo el adaptador de interfaces inteligente mencionado en el Capítulo 8) o directamente dar soporte en procesos experimentales, puesto que muchas veces puede ser útil conocer los sentimientos de la persona durante la realización de un experimento controlado. *Realmente es difícil de encuadrar este trabajo dentro de los Objetivos de Desarrollo Sostenible, debido a que no tiene mucha relación con la mayoría. Aun así es innegable que este trabajo se puede relacionar con el ODS 9. **Industria, innovación e infraestructuras**, siendo éste un trabajo muy relacionado con las técnicas de Machine Learning y la Inteligencia Artificial, que están a la orden del día.*

Este trabajo presenta una innovación por varios factores como puede ser la posibilidad de dar información a aplicaciones o herramientas externas del estado anímico del usuario (por ejemplo el adaptador de interfaces inteligente mencionado en el Capítulo 8) o directamente dar soporte en procesos experimentales, puesto que muchas veces puede ser útil conocer los sentimientos de la persona durante la realización de un experimento controlado.