



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

SISTEMA DE DETECCIÓN DE EMOCIONES MULTIMODAL PARA EL SEGUIMIENTO DEL APRENDIZAJE EN EL AULA INTERACTIVA E INCLUSIVA

Presenta:

Luis Fernando Gadea Sánchez

Tesis presentada para obtener el título de:

Licenciatura en Ingeniería en Ciencias de la Computación

Asesora de tesis:

Dra. Carmen Cerón Garnica

Puebla, Puebla Enero 2026

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Luis Fernando Gadea Sánchez, declaro bajo protesta de decir verdad que el presente trabajo de tesis titulado "Sistema de detección de emociones multimodal para el seguimiento del aprendizaje en el aula interactiva e inclusiva" es producto de mi autoría.

Manifiesto que no ha sido presentado con anterioridad para obtener algún grado académico, y que las fuentes consultadas han sido debidamente citadas.

Asumo la responsabilidad total por el contenido del documento.

Puebla, a 28 de enero del 2026.

Luis Fernando Gadea Sánchez

AGRADECIMIENTO

A mis padres que me dieron la oportunidad de estudiar y culminar mis estudios. A mis hermanos. A mi directora de tesis que me apoyó durante todo este proceso. A mis gatos, Veres y Luci, por su compañía y amor incondicional. Y a Ramón, mi prometido, por su compañía y apoyo incondicional durante los momentos más difíciles de mi vida, antes de la pandemia, y por devolverme las ganas de continuar con mis estudios cuando sentía que ya no podía seguir.

DEDICATORIA

A mis padres por su apoyo
incondicional. A mis hermanos.
A Ramón, por creer en mí cuando yo no
podía hacerlo.

Luis Fernando

Índice general

Contenido

Índice general	6
Índice de figuras	9
Índice de tablas	10
Capítulo 1 Introducción	12
Resumen.....	14
Capítulo 2 Marco teórico.....	16
2.1 Machine Learning.....	16
2.2 Deep Learning.....	18
2.3 Detección facial.....	19
2.4 Reconocimiento facial	20
2.5 Reconocimiento de emociones.....	21
2.6 Sistemas en tiempo real	21
2.7 Estado del arte	22
2.8 Herramientas tecnológicas	23
2.8.1 Almacenamiento de datos	23
2.8.1.1 MS SQL SERVER	24
2.8.1.2 MySQL	25
2.8.1.3 Oracle Database.....	26
2.8.1.4 Comparación de sistemas de almacenamiento	27
2.8.2 Visión por computadora	28
2.8.2.1 OpenCV	28
2.8.2.2 Pillow	29
2.8.2.3 Scikit-Image	29
2.8.2.4 Comparación entre las herramientas para visión por computadora ...	30
2.8.3 Herramientas para el desarrollo de la Inteligencia Artificial	32
2.8.3.1 Tensorflow.....	32

2.8.3.2 PyTorch	33
2.8.3.3 Scikit-learn.....	33
2.8.3.4 Keras	34
2.8.3.5 PyBrain	34
2.8.4 Herramientas para el reconocimiento de voz y transcripción	37
2.8.4.1 GTTS	37
2.8.4.2 Whisper	38
2.8.4.3 Speech_Recognition	38
2.8.4.4 Wave.....	39
2.8.4.5 Pyaudio.....	40
2.8.4.6 DeepSpeech.....	40
2.8.4.7 Pydub	41
2.8.4.8 Comparación entre las herramientas para el desarrollo del reconocimiento de voz y transcripción	41
2.8.5 Herramientas seleccionadas	43
2.8.5.1 Almacenamiento de datos.....	43
2.8.5.2 Visión por computadora	44
2.8.5.3 Desarrollo de la Inteligencia Artificial	44
2.8.5.4 Reconocimiento de voz y transcripción	44
2.8.5.5 Voz a texto	45
Capítulo 3 Propuesta de solución.....	46
3.1 Análisis y diseño.....	46
3.1.1 Requisitos funcionales del proyecto.....	46
3.1.2 Requisitos no funcionales del proyecto	48
3.1.3 Requisitos de información	48
3.1.4 Restricciones	48
3.1.5 Diagrama de casos de uso	48
3.1.6 Diagramas de secuencia	51
3.2 Implementación y desarrollo.....	59
3.2.1 Desarrollo del algoritmo para la detección facial	59
3.2.1.1 Explicación del desarrollo del algoritmo	59

3.2.2 Desarrollo de la red neuronal para el reconocimiento de emociones.....	61
3.2.2.1 Descripción del modelo	61
3.2.2.2 Entrenamiento del modelo	70
3.2.3 Desarrollo del algoritmo para el reconocimiento facial	72
3.2.3.1 Codificación del rostro de una nueva persona.....	72
3.2.3.2 Ejemplo de cómo agregar un nuevo rostro para su reconocimiento ...	72
3.2.4 Desarrollo de la base de datos	75
3.2.4.1 Estructura de la base de datos.	75
3.2.4.2 Desarrollo de la base de datos.	75
3.2.5 Funcionamiento en conjunto para el reconocimiento facial y de emociones	77
3.2.5.1 Algoritmo que combina el reconocimiento facial y de emociones.....	78
3.2.5.2 Ejemplo del funcionamiento del reconocimiento facial y de emociones	78
3.2.6 Diseño de la visualización de gráficas de barras	81
3.2.7 Desarrollo del sistema de texto a voz	82
3.2.8 Desarrollo del sistema de voz a texto	83
Capítulo 4 Pruebas del modelo y resultados	84
4.1 Pruebas del reconocimiento facial y de emociones	84
Capítulo 5 Conclusiones.....	110
5.1 Conclusiones del proyecto	110
5.1.1 Reconocimiento facial y de emociones	110
5.1.2 Texto a voz	112
5.1.3 Voz a texto	113
Capítulo 6 Trabajos futuros	114
6.1 Desarrollo de red neuronal para diferenciar entre usuarios agachados o inclinados y usuarios tristes	114
6.2 Implementación de un nuevo campo para los usuarios que NO son maestro en el apartado “Reconocimiento facial y de emociones” para poder agregar los datos faciales de otros usuarios, usando su “Nombre de usuario” para su reconocimiento facial.	114
6.3 Implementación de una mejora al método de voz a texto para que funcione en tiempo real.....	115
Bibliografía.....	116

Índice de figuras

Fig. 1: Ejemplo de una red neuronal y sus diferentes capas.	18
Fig. 2 Diagrama de casos de uso	49
Fig. 3 Diagrama de secuencia para el caso de uso Crear cuenta.	51
Fig. 4 Diagrama de secuencia para el caso de uso Iniciar sesión.	52
Fig. 5 Diagrama de secuencia para el caso de uso Crear materia.	52
Fig. 6 Diagrama de secuencia para el caso de uso Agregar alumnos a materia.	53
Fig. 7 Diagrama de secuencia para el caso de uso Cambiar rol de usuario.	54
Fig. 8 Diagrama de secuencia para el caso de uso Registrar fotografía	55
Fig. 9 Diagrama de secuencia para el caso de uso Reconocimiento facial y de emociones	56
Fig. 10 Diagrama de secuencia para el caso de uso Voz a Texto	57
Fig. 11 Diagrama de secuencia para el caso de uso Texto a voz	58
Fig. 12 Ejemplo de capa densa: Cada dato de entrada tiene una conexión a la capa densa.	67
Fig. 13 La primer imagen la foto tomada y la segunda imagen es el rostro reconocido al que se le extraerá el vector de características.	74
Fig. 14 Vector de características resultante del rostro de la Fig. 13.	74
Fig. 15 Clase con la que se creó la tabla FotoUsuario.	76
Fig. 16 Clase con la que se creó la tabla Materia.	76
Fig. 17 Clase con la que se creó la tabla Materia_Alumnos.	77
Fig. 18 Clase con la que se creó la tabla ArchivoCSV.	77
Fig. 19 Al presionar el botón iniciar detección se muestra un cuadro de confirmación.	79
Fig. 20 Ejemplo de detección Facial y de emociones, emoción: Felicidad.	79
Fig. 21 Ejemplo de reconocimiento facial y de emociones, emoción: Enojo.	80
Fig. 22 Al presionar el botón detener detección se muestra un mensaje de aviso.	80
Fig. 23 Emociones de Ian Jarquin en el transcurso de 114 segundos.	86
Fig. 24 Emociones de Rafael en el transcurso de 247.5 segundos.	86
Fig. 25 Emociones de Ángeles Ortiz en el transcurso de 21 segundos.	86
Fig. 26 Emociones de Dania García Hernández en el transcurso de 22.5 segundos.	87
Fig. 27 Emociones de Karla Aguilar en el transcurso de 54 segundos.	87
Fig. 28 Emociones de Gianeli en el transcurso de 13.5 segundos.	87
Fig. 29 Emociones de Luis Santiagi en el transcurso de 87 segundos.	88
Fig. 30 Emociones de Aliah en el transcurso de 13.5 segundos.	88
Fig. 31 Emociones de Yuridia Díaz en el transcurso de 33 segundos.	88
Fig. 32 Emociones de Rodrigo Fernández en el transcurso de 22.5 segundos.	89
Fig. 33 Emociones de Flores Márquez en el transcurso de 21 segundos.	89
Fig. 34 Emociones de Taide en el transcurso de 99 segundos.	89
Fig. 35 Emociones de Inan en el transcurso de 9 segundos.	90
Fig. 36 Emociones de Roy Jesús en el transcurso de 3 segundos.	90
Fig. 37 Emociones de Luis Alberto en el transcurso de 1.5 segundos.	90
Fig. 38 Emociones de Alejandro López en el transcurso de 3 segundos.	91
Fig. 39 Emociones de Rogelio en el transcurso de 3 segundos.	91
Fig. 40 Emociones de Diego en el transcurso de 1.5 segundos.	91
Fig. 41 Emociones de Karen Reyes en el transcurso de 4.5 segundos.	92

Fig. 42 Emociones de Sara en el transcurso de 42 segundos.	96
Fig. 43 Emociones de Juan Héctor en el transcurso de 40 segundos.	97
Fig. 44 Emociones de Linus en el transcurso de 54 segundos.	97
Fig. 45 Emociones de Ximena en el transcurso de 12 segundos.	98
Fig. 46 Emociones de Isabella en el transcurso de 34 segundos.	98
Fig. 47 Emociones de Xavier en el transcurso de 33 segundos.	99
Fig. 48 Reporte de donde se muestran los datos, el estado emocional y las gráficas de un alumno.	102
Fig. 49 Se inserta el texto en el recuadro de la izquierda y se presiona el botón "Escuchar".	103
Fig. 50 Se procesa el texto y se comienza a reproducir el audio.	104
Fig. 51 Texto más largo para probar la optimización del sistema.	104
Fig. 52 Audio creado y tiempo que se tardó en crearse.	105
Fig. 53 Resultado del Caso 1: Música de fondo y el usuario frente a la computadora.	106
Fig. 54 Resultado del Caso 2: Sin música de fondo y el usuario frente a la computadora.	107
Fig. 55 Resultado del Caso 3: Sin música de fondo y el usuario a 1 metro de la computadora.	108
Fig. 56 Resultado del Caso 4: Con música de fondo y el usuario a 1 metro de la computadora.	108

Índice de tablas

Tabla 1 Tabla comparativa. Sistemas para el almacenamiento de datos.	27
Tabla 2 Tabla resumen. Ventajas y desventajas de las herramientas para la visión por computadora.	31
Tabla 3 Tabla resumen. Ventajas y desventajas de las herramientas para el desarrollo de la inteligencia artificial.	37
Tabla 4 Tabla resumen. Ventajas y desventajas de las herramientas para el desarrollo del reconocimiento de voz.	43
Tabla 5 Tabla resumen. Modelo de la red neuronal creada	70
Tabla 6 Tabla resumen: Encuestas realizadas al terminar la detección	93
Tabla 7 Tabla resumen: Tabla de las encuestas realizadas al terminar la detección.	100

Capítulo 1 Introducción

En los últimos años el reconocimiento facial se ha vuelto una herramienta muy importante para la tecnología, sus diversas aplicaciones, principalmente en la seguridad biométrica, han llevado a que se siguiera investigando y mejorando su implementación. Gracias a estas investigaciones y avances tecnológicos, es posible implementar esta herramienta en cualquier dispositivo, desde la autenticación de acceso en un dispositivo móvil o una aplicación bancaria, hasta la búsqueda de personas desaparecidas utilizando las cámaras de la ciudad.

Otra aplicación utilizando el reconocimiento facial es el reconocimiento de emociones, el cual, utilizando el reconocimiento facial puede reconocer la emoción que presentan los rostros de cada usuario. Esta tecnología tiene una aplicación, por ejemplo, en el reconocimiento de emociones en el aula y es importante porque de acuerdo con investigaciones, por ejemplo, la realizada en 2002 por Nolfi Ibáñez S. LAS EMOCIONES EN EL AULA [1], las emociones que presentan los alumnos influyen en la capacidad para aprender, relacionarse y concentrarse. Las emociones positivas pueden aumentar la motivación y participación mientras que las negativas pueden dificultar su aprendizaje.

En la actualidad, existen sistemas y aplicaciones que pueden reconocer las emociones de los usuarios que las usan, el problema cae en que la mayoría de estos sistemas están bloqueados detrás de suscripciones, sin embargo, debido a las limitaciones económicas de muchas escuelas se dificulta el acceso a estas herramientas. Por ello, un sistema gratuito y de código abierto es un tema importante para mejorar la calidad de la educación en México.

Dado que en el aula hay muchos alumnos y los profesores no pueden estar atentos a cada uno de ellos mientras, además, están impartiendo su clase, este sistema puede ayudar a reconocer las emociones de cada alumno y generar reportes para el profesor, con gráficas que contengan toda la información necesaria, analizando qué alumnos presentan una cantidad significativa de emociones

negativas a lo largo de la clase. Con esta información el profesor podría acercarse a dichos alumnos, platicar los problemas que presentan y poder llegar a una solución para mejorar sus emociones y con esto su capacidad de aprendizaje.

Este estudio se centra en el desarrollo de un prototipo funcional de un sistema de reconocimiento facial y de emociones, con un enfoque en su aplicación práctica en el aula, además de un pequeño sistema adicional para una herramienta que convierta el texto escrito a voz y la voz a texto. El sistema emplea tecnologías de acceso libre que garantizan su funcionalidad y escalabilidad. Entre ellas destacan:

- La librería OpenCV y imutils para el procesamiento del vídeo y las imágenes obtenidas del mismo.
- La librería Tensorflow para la creación de la red neuronal del análisis de las emociones.
- La librería GTTS para la herramienta de texto a voz.
- Las librerías Whisper y Speech_recognition para la herramienta de voz a texto.
- El sistema gestor de bases de datos MySQL para la creación de la base de datos.
- El framework Django para el desarrollo de la aplicación WEB.

El sistema dará la opción de agregar el rostro del usuario con su nombre para poder realizar el reconocimiento facial, si el usuario no está registrado aun así podrá usar el sistema, pero su nombre será 'Desconocido'. El sistema detectará a los usuarios a través del vídeo obtenido de la cámara, si hay más de una cámara en el equipo o conectada al equipo, dará la opción de elegir la cámara, esto significa que es un sistema en tiempo real. Las emociones que podrá reconocer serán felicidad, tristeza, miedo, enojo, asco, sorpresa y neutral.

El sistema no tiene un límite fijo para la cantidad de usuarios que podrá analizar al mismo tiempo, debido a que está hecho para procesar múltiples rostros en una imagen con los algoritmos de detección de rostros utilizados. Sin embargo, su capacidad de análisis depende directamente de la resolución y calidad de las

imágenes capturadas por la cámara. Por ejemplo, Una cámara con resolución alta podrá identificar y analizar más rostros a mayor distancia y con mayor precisión, mientras que, si la resolución de la cámara es baja o los rostros están borrosos, el reconocimiento facial y de emociones podrían no ser precisos o incluso llegar a fallar.

Por lo tanto, el sistema es escalable en cuanto a la cantidad de personas siempre y cuando las condiciones de captura, iluminación, posición y calidad de la cámara sean buenas para garantizar que los rostros sean claramente visibles en la imagen.

En conclusión, el sistema de reconocimiento facial y de emociones se propone como una herramienta gratuita y de código abierto, enfocada principalmente al entorno educativo, pero adaptable a otros entornos diferentes. Siendo capaz de analizar en tiempo real, además de accesible, busca mejorar el aprendizaje a través de la comprensión emocional.

Resumen

En el presente documento se presenta el Sistema de detección de emociones multimodal para el seguimiento del aprendizaje en el aula interactiva e inclusiva, un sistema en el que se hace uso de la Inteligencia artificial en un contexto de educación. EL objetivo principal de este documento es desarrollar un sistema de reconocimiento facial y de emociones que se pueda utilizar dentro de un aula analizando las emociones y creando reportes de los alumnos que faciliten a los profesores identificar qué alumnos muestran emociones negativas con más frecuencia. Así, los profesores pueden prestar mayor atención a estos alumnos, ya que las emociones negativas pueden influir en el aprendizaje dentro del aula.

El sistema podrá ser desplegado en el contexto educacional, se podrá utilizar en otros contextos, pero con algunas limitaciones. Se espera que este ayude a solucionar el problema de que estos sistemas están detrás de suscripciones, como

se expuso en la introducción del presente documento, junto a este se entrega todo el código desarrollado.

Capítulo 2 Marco teórico

En este apartado se describen los conceptos de Machine learning, Deep learning, detección facial, reconocimiento facial, reconocimiento de emociones y sistemas en tiempo real, el impacto de las emociones en el aula, así como el estado del arte y herramientas tecnológicas existentes para el desarrollo un sistema parecido.

2.1 Machine Learning

El Machine Learning es la ciencia de desarrollo de algoritmos y modelos estadísticos que utilizan los sistemas de computación con el fin de llevar a cabo tareas sin instrucciones explícitas, en vez de basarse en patrones e inferencias. Los sistemas de computación utilizan algoritmos de machine learning para procesar grandes cantidades de datos históricos e identificar patrones de datos. Esto les permite generar resultados con mayor precisión a partir de un conjunto de datos de entrada. [2]

Los algoritmos con los que trabaja esta ciencia necesitan muchos datos para entrenar el sistema que se va a usar. Estos algoritmos se pueden clasificar en cuatro estilos de aprendizaje dependiendo de la salida esperada y el tipo de entrada.

- **Supervised machine learning o aprendizaje supervisado [3]**: para usar este tipo de algoritmos se requiere tener datos previamente etiquetados, esto para, por ejemplo, entrenar un algoritmo que pueda clasificar el spam en una carpeta separada de la bandeja de entrada. Estos datos etiquetados incluyen entradas y salidas, para que el sistema aprenda con cada iteración que realiza. Su precisión se mide con una función de pérdida, la cual se ajusta hasta que el error haya disminuido a un valor aceptable.
- **Unsupervised machine learning o aprendizaje sin supervisar**: este algoritmo, a diferencia del *aprendizaje supervisado*, los datos que requiere son sin

etiquetar. A partir de estos datos, se encargan de descubrir patrones que ayuden a resolver problemas de agrupamiento.

- **Aprendizaje semisupervizado:** este algoritmo es una mezcla entre el aprendizaje supervisado y el no supervisado. En muchas ocasiones es difícil encontrar una cantidad grande de datos etiquetados, este algoritmo combina datos etiquetados y no etiquetados para construir un modelo supervisado mejorado.

- **Reinforcement machine learning o aprendizaje por refuerzo:** la diferencia de este algoritmo con los tres mencionados no viene de los datos utilizados para su entrenamiento, sino de la manera que tiene de aprender. Su aprendizaje se basa en recompensar los comportamientos deseados o correctos y castigar los no deseados o incorrectos. Un ejemplo de este tipo de algoritmos es en el que un sistema aprende a jugar ajedrez, la información que se le brinda son las reglas del juego y este sistema aprende a jugar por sí mismo.

En la actualidad existen muchas aplicaciones para el Machine Learning, algunos ejemplos son los siguientes:

- Detección de rostros.
- Algoritmos anti-spam.
- Comprensión de textos.
- Motores de búsqueda.
- Antivirus.

2.2 Deep Learning

El Deep Learning o aprendizaje profundo es un subconjunto de Machine Learning que utiliza redes neuronales multicapa, llamadas redes neuronales profundas, para simular el complejo poder de toma de decisiones del cerebro humano.[4] Es decir, se usan estructuras que asemejan el funcionamiento del cerebro humano, para detectar ciertas características existentes en los datos que se están utilizando.

Una red neuronal contiene capas de neuronas conectadas unas con otras, cada capa activándose o no, de acuerdo con un valor determinado llamado sesgo. Las neuronas de la capa de entrada se activan de acuerdo con los datos de entrada que proporciona el usuario. Las capas subsecuentes, también llamadas capas ocultas, se activan de acuerdo con la conexión con capas anteriores y los datos que estas proporcionan. Y la salida es la predicción obtenida de esta red, por ejemplo, una red neuronal hecha para reconocer imágenes de gatos obtiene como entrada una imagen de un perro, la procesa en las capas ocultas y obtiene un resultado en la salida que diría que esa imagen no es la de un gato.

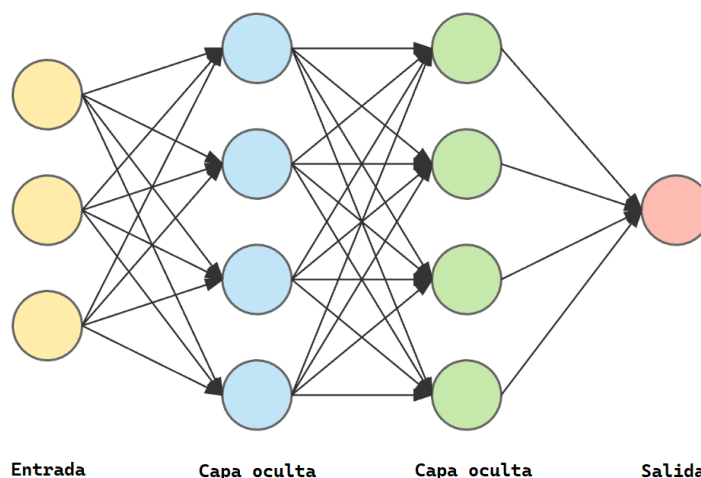


Fig. 1: Ejemplo de una red neuronal y sus diferentes capas.

El uso del Deep Learning está aumentando, esto debido a los avances en la velocidad y rendimiento general de los procesadores actuales, siendo posible usar esta disciplina sin necesidad de usar la última tecnología disponible en el mercado. Algunos ejemplos del uso del Deep Learning o redes neuronales son los siguientes:

- Reconocimiento facial.
- Reconocimiento de voz.
- Visión por computadora.
- Traducción automática.
- Asistentes virtuales.
- Reconocimiento de emociones.

2.3 Detección facial

La detección facial es una tecnología que, utilizando redes neuronales, es capaz de analizar imágenes y videos para detectar rostros. Esta tecnología utiliza algoritmos como redes neuronales convolucionales (CNN), para identificar patrones faciales en las imágenes o videos.

Se limita únicamente a localizar y delimitar la presencia de un rostro. Aunque no es muy compleja o no tiene muchas aplicaciones por si sola, es base para algoritmos de detección más complejos, como el reconocimiento facial o el reconocimiento de emociones.

2.4 Reconocimiento facial

El reconocimiento facial es una aplicación más profunda de la detección facial, ya que esta no solo es capaz de detectar rostros, sino que además puede reconocer las caras de las personas. Esto se logra al agregar un algoritmo al sistema de detección facial que permita analizar los rasgos faciales. Por ejemplo, al recibir una fotografía con el rostro de una persona y su nombre, el sistema detecta el rostro, analiza los rasgos distintivos de esta persona, y genera un registro de características con la información asociada a este rostro.

Una vez creado este registro, el sistema puede comparar los rostros detectados en imágenes o videos con los registros almacenados anteriormente, e identificar a las personas conocidas. Este proceso puede llevarse a cabo con diferentes tecnologías, por ejemplo, métodos modernos como redes neuronales convolucionales (CNN), que trabajan analizando patrones en las imágenes, o métodos más antiguos como EigenFaces y FisherFaces, que trabajan con la reducción de dimensionalidad para encontrar patrones en los rostros.

El uso de esta tecnología es tan frecuente que estamos acostumbrados a ella, estando en sistemas como el desbloqueo de un dispositivo móvil con el rostro. Otras aplicaciones de esta tecnología están más enfocadas en la seguridad, por ejemplo:

- Acceso a aplicaciones bancarias.
- Cámaras de vigilancia.
- Control de aeropuertos y fronteras.
- Acceso a edificios inteligentes.

2.5 Reconocimiento de emociones

Al igual que el reconocimiento facial, esta tecnología usa la detección facial como base para su funcionamiento. Esto porque una vez detectado el rostro, se utilizan sus parámetros de delimitación como la entrada de un algoritmo para reconocer las emociones en el rostro. Por ejemplo, al recibir una fotografía con el rostro de una o varias personas se pasa por el algoritmo de detección de rostros y se obtienen las ubicaciones de los rostros de las personas encontradas.

Una vez obtenidos estos valores se toma la ubicación de un rostro encontrado y se procesa el rostro con un algoritmo de reconocimiento de emociones que analiza la posición de las cejas, la curvatura de los labios o la apertura de los ojos, para predecir una emoción. Este proceso puede realizarse con diferentes métodos, pero uno de los más eficaces es con una Red Neuronal Convolucional, la cual trabaja analizando patrones en las imágenes.

2.6 Sistemas en tiempo real

Se trata de sistemas informáticos que procesan la información y responden a eventos dentro de límites de tiempo específicos. Es decir, una vez hecha una solicitud estos sistemas procesan los datos y devuelven una respuesta en un tiempo establecido. Esto es importante en sistemas de seguridad ya que necesitan una respuesta casi inmediata. Algunos ejemplos de sistemas en tiempo real son:

- Reconocimiento facial en dispositivos móviles.
- Encendido de un dispositivo electrónico desde el móvil.
- Asistentes de voz, como Alexa o Siri.
- Sistemas de monitoreo de salud y atención médica.
- Sistemas de audio y video en directo.

2.7 Estado del arte

Actualmente hay una gran variedad de sistemas que trabajan con redes neuronales para el reconocimiento facial y de emociones, y estos tienen una gran diversidad de aplicaciones, cada uno enfocándose en un área en específico, como en el reconocimiento de emociones en el aula, al manejar, entre otras aplicaciones que se beneficiarían del poder reconocer las emociones de sus usuarios.

A través de varias búsquedas en internet, se han podido encontrar diferentes proyectos. A continuación, se describen algunos de estos:

AFFDEX SDK: Un kit de herramientas de reconocimiento de expresiones en tiempo real multiplataforma para múltiples rostros [5]: Es capaz de codificar las expresiones de varias personas simultáneamente. Este kit de herramientas ofrece la posibilidad de diseñar interfaces novedosas que respondan a los estados emocionales de los usuarios. Tiene un costo por usarlo aunque esté no está disponible para el público.

Reconocimiento facial basado en eigenfaces, lbhp y fisherfaces en la beagleboard-xM [6]: Implementó un sistema detector de rostros aplicando el procesamiento de imágenes para comparar la eficacia de los tres algoritmos, EigenFaces, LBPH y FisherFaces, sobre el dispositivo BeagleBoard. Se llegó al resultado de que EigenFaces era el mejor de los tres.

iMotions [7]: Es un software de estudio de la conducta humana. Cuenta con una interfaz para la decodificación facial y poder reconocer las respuestas emocionales utilizando algoritmos informáticos automatizados y registrando todas las emociones. Esto a través de una cámara WEB. El precio del software varía dependiendo de las características de software que se requieran.

Affective [8]: Es un software capaz de detectar estados cognitivos y emocionales humanos. Esto a través de la lectura de señales no verbales como expresiones faciales, gestos y el lenguaje corporal. El precio del software varía dependiendo de las características de software que se requieran.

Como se puede observar, hay una gran variedad de sistemas para el reconocimiento de emociones, la gran mayoría requiere algún tipo de pago, ya sea una suscripción mensual o la compra del software.

Por esto, en la siguiente sección del documento se analizarán y compararán algunas de las tecnologías actuales para el desarrollo de una herramienta de reconocimiento facial, de emociones y el reconocimiento de voz, aplicando Deep Learning.

2.8 Herramientas tecnológicas

En esta sección se analizan diferentes tecnologías que se pueden utilizar para el almacenamiento de datos, visión por computadora, desarrollo de modelos de inteligencia artificial y librerías para la herramienta de reconocimiento de voz, todas estas tecnologías juegan un papel muy importante en el desarrollo del sistema de reconocimiento facial y de emociones que se describió en la introducción de este documento.

2.8.1 Almacenamiento de datos

Para el desarrollo de este proyecto se deben almacenar varios datos, desde las cuentas de los usuarios con su información y permisos correspondientes, los datos de las materias impartidas por los maestros, las fotografías de los usuarios registrados y sus características faciales, esto para mejorar la eficacia del sistema y que solo sea necesario procesar el rostro del usuario una sola vez, logrando ahorrar tiempo y recursos de procesamiento. A continuación, se describe y comparan algunas herramientas para el almacenamiento de los datos del sistema.

2.8.1.1 MS SQL SERVER

Microsoft SQL Server es un Sistema Gestor de Bases de Datos relacionales basado en el lenguaje Transact-SQL. Su forma de funcionamiento es que las aplicaciones y herramientas se conectan a una base de datos de SQL Server y se comunican mediante el lenguaje Transact-SQL.

Este SGBD se puede instalar tanto en Microsoft, Linux o una máquina virtual de Azure u otras máquinas virtuales disponibles. Existen muchas versiones disponibles para su instalación, entre ellas las versiones MS SQL Server 2022 Express y SQL Server 2022 Developer destacan para el desarrollo de este proyecto porque son versiones gratuitas.

Ventajas

- Es fácil de integrar con otros servicios de Microsoft, como .NET, Visual Studio, Azure, entre otros, esto facilita el desarrollo y la administración de entornos Microsoft.
- Tiene funcionalidades de seguridad avanzadas.
- Existe una gran documentación.

Desventajas

- Para desbloquear funcionalidades se requiere de compra de licencias.
- Puede requerir un mayor gasto de recursos.
- No es fácil de comenzar a utilizar si no se tiene un conocimiento avanzado del software.

2.8.1.2 MySQL

Es el Sistema Gestor de Bases de Datos más popular. Este es un SGBD relacionales, esto significa que los datos se encuentran en diferentes tablas y no en una sola tabla gigante. Las relaciones entre los datos de las tablas se rigen de uno a uno, uno a muchos, únicos, obligatorios u opcionales. Esto ayuda a evitar los datos duplicados, huérfanos, desfasados o ausentes. Las consultas o Querys se pueden realizar con el lenguaje SQL, incrustar secuencias SQL en código escrito en otro lenguaje o utilizar una API.

Este SGBD cuenta con dos versiones: Community Edition que es de código abierto, lo que significa que cualquier persona puede descargar y utilizar el software de forma gratuita, y la versión de pago que es la Standard Edition.

Ventajas

- Está optimizado para consultas rápidas.
- Tiene soportes para múltiples plataformas y se integra fácil con diferentes lenguajes de programación.
- Es sencillo de instalar y configurar, contando con aplicaciones que facilitan su uso, como phpMyAdmin, MySQL Workbench, entre otras.

Desventajas

- Ofrece diferentes motores de almacenamiento lo cual puede generar inconsistencias en el manejo de transacciones.
- En comparación con otros SGBD cuenta con menos funcionalidades avanzadas.

2.8.1.3 Oracle Database

Es un Sistema Gestor de Bases de Datos de tipo relacional, desarrollado por Oracle Corporation. Este SGBD mejora la gestión de grandes bases de datos y programación, también aumenta el nivel de seguridad.

Cuenta con 7 ediciones, entre las que destaca la Express Edition dado que es la única edición gratuita.

Ventajas

- Puede manejar grandes cantidades de datos y transacciones.
- Tiene soporte para el uso de procedimientos almacenados y funciones en PL/SQL.
- Tiene una seguridad robusta, como el cifrado, control de acceso basado en roles, etc.

Desventajas

- Requiere un conocimiento profundo para muchas características y configuraciones.
- Para aprovechar sus capacidades suele requerir una gran cantidad de recursos de hardware.
- Las licencias para otras versiones son costosas.

2.8.1.4 Comparación de sistemas de almacenamiento

Sistema	Ventajas	Desventajas
MS SQL Server	Proporciona un entorno altamente escalable. Puede detener la ejecución de consultas sin que estas terminen. Mucha documentación.	Sus versiones gratuitas son para sitios ligeros, se necesita una versión de pago para desbloquear más funcionalidades. Requiere más experiencia para sacarle provecho.
MySQL	Compatible con varios lenguajes de programación y SO. Mucha documentación. Su instalación es sencilla.	No soporta transacciones. No es 100% código abierto.
Oracle Database	Cuenta con una versión gratuita. Es escalable. Seguridad basada en capaz.	Se necesita comprar para tener una versión completa. Instalación muy larga. Uso de muchos recursos de hardware.

Tabla 1 Tabla comparativa. Sistemas para el almacenamiento de datos.

2.8.2 Visión por computadora

La visión por computadora es un campo de la inteligencia artificial que utilizan las computadoras para analizar imágenes y videos, en los cuales la computadora puede extraer información para identificar objetos y personas.

En esta sección se analizan diferentes herramientas disponibles para desarrollar la visión por computadora, desde la visión en una imagen o un video.

2.8.2.1 OpenCV [9]

Fue creado en Intel en 1999 por Gary Bradsky, y la primera versión fue lanzada el año 2000. A este proyecto se unieron Vadim Pisarevsky y Willow Garage para ayudar a seguir desarrollando el proyecto. OpenCV ahora soporta una multitud de algoritmos relacionados con la Visión por computadora y Machine Learning.[9]

Esta librería está disponible para muchos lenguajes, entre ellos C++, Python, Java, etc. y también está disponible en diferentes plataformas como Windows, Linux, Android, iOS y OS X.

Ventajas:

- Tiene compatibilidad con muchos lenguajes lo cual lo hace más versátil.
- Hace uso de otras librerías como Numpy, lo cual hace más fácil integrar su uso con otras librerías que usan Numpy, como SciPy y Matplotlib.
- Su versión OpenCV-Python lo hace fácil de utilizar en Python.
- Fácil de instalar.
- Es de código abierto.

Desventajas.

- No está optimizado para manejar grandes modelos de aprendizaje profundo ni redes neuronales complejas. [10] <https://openwebinars.net/blog/opencv-introduccion-y-su-rol-en-la-vision-por-computadora/>

2.8.2.2 Pillow [11]

Su antecesor, PIL, soporta únicamente hasta la versión 2.7 de Python, y el desarrollador de esta librería dejó de hacer actualizaciones para la misma, por lo que Alex Clark en colaboración con otros programadores desarrolló Pillow. Esta librería pretende ser estable y adaptable a nuevas tecnologías, es decir, versiones más recientes de Python como lo son las 3.x. [12]

Es una librería gratuita y de código abierto, agrega la capacidad de procesar imágenes con el intérprete de Python en diferentes formatos de imagen. Está disponible en diferentes plataformas como Windows, Mac OS X y Linux.

Ventajas

- Tiene extensiones que permite leer datos de XMP, FPX e imágenes de tipo MIC.
- Fácil de instalar.

Desventajas

- Es complicado encontrar documentación para casos de uso más complejos.
- Sólo está disponible para el lenguaje Python.

2.8.2.3 Scikit-Image [13]

Es una librería para el procesamiento de imágenes con Python, esta trabaja con Numpy y SciPy que es una librería de algoritmos para el procesamiento de imágenes. Es compatible con varios sistemas operativos, como lo son, Windows MacOS y Linux.

Incluye algoritmos para la segmentación, transformaciones geométricas, manipulación del color, análisis, filtros, entre otros.

Ventajas

- Tiene mucha documentación y cuenta con foros donde investigar dudas.
- Compatible con diferentes sistemas operativos.

Desventajas

- Requiere que otras librerías estén instaladas antes de instalar Scikit-Learn.
- Solo está disponible para el lenguaje Python.

2.8.2.4 Comparación entre las herramientas para visión por computadora

	Ventajas	Desventajas
OpenCV	Es compatible con varios lenguajes de programación, como C++, Python y Java. Funciona en distintos sistemas operativos, incluyendo Windows, Linux, macOS, Android y iOS. Se integra bien con otras herramientas basadas en Numpy, facilitando su uso en análisis de datos y visualización. Su instalación es sencilla y es de código abierto.	No está diseñado para procesar modelos de inteligencia artificial avanzados ni redes neuronales de gran tamaño.

Pillow	Permite trabajar con una amplia variedad de formatos de imagen, incluyendo XMP, FPX y MIC. Su instalación es rápida y sin complicaciones. Es un proyecto de código abierto, compatible con versiones recientes de Python.	La documentación sobre usos avanzados es limitada. Solo funciona en el ecosistema de Python, no tiene soporte para otros lenguajes.
Scikit-Image	Dispone de una gran cantidad de documentación y soporte comunitario. Es multiplataforma, funcionando en Windows, macOS y Linux. Ofrece herramientas especializadas para mejorar y analizar imágenes, como filtros, segmentación y manipulación de colores.	Depende de la instalación previa de otras librerías, como NumPy y SciPy. Solo puede utilizarse con Python.

Tabla 2 Tabla resumen. Ventajas y desventajas de las herramientas para la visión por computadora.

2.8.3 Herramientas para el desarrollo de la Inteligencia Artificial

En esta sección se analizan las diferentes librerías que existen para la creación de una red neuronal para el Reconocimiento de emociones.

2.8.3.1 Tensorflow [14]

Es una librería de código abierto que facilita la creación de aprendizaje automático para escritorio, dispositivos móviles, web y la nube.[9] Fue originalmente desarrollado por Google Brain para uso interno de Google antes de ser publicado bajo la licencia de código abierto Apache 2.0 el 9 de noviembre del 2015.

Está disponible para las plataformas Windows, Linux MacOS y plataformas móviles como Android y iOS. Además, es compatible con una gran variedad de lenguajes de programación como son Python, C++, Java, Rust, entre otros.

Ventajas

- Tiene compatibilidad con una gran variedad de plataformas y lenguajes de programación.
- Cuenta con una amplia documentación y ejemplos.
- Ofrece múltiples herramientas de datos para consolidar, limpiar y procesar datos a escala. [14]

Desventajas

- Tiene una curva de aprendizaje empinada y una sintaxis compleja.
- Tiende a ser lento y necesitar mucha memoria, especialmente para modelos dinámicos y recurrentes. [15]

2.8.3.2 PyTorch [16]

Es una librería de aprendizaje automático de código abierto basada en Torch. Cuenta con una interfaz muy desarrollada y amigable para Python y una interfaz en C++ y Java, aunque estas no están tan desarrolladas. Fue desarrollada por el equipo de Meta AI y liberada a código abierto en GitHub en 2017.

PyTorch se integra bien con librerías como TensorRT para inferencia de alto rendimiento y soporta formatos como ONNX para facilitar el despliegue en diversas plataformas.[17]

Ventajas

- Es simple y fácil de usar.
- El uso de memoria es eficiente.
- Cuenta con buena documentación y ayuda de la comunidad.
- Es de código abierto.

Desventajas

- Se necesita un servidor API para producción. [18]
- Se necesitan aplicaciones adicionales para la visualización de los datos.[18]

2.8.3.3 Scikit-learn [19]

Es una librería gratuita para Python. Cuenta con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Además, presenta la compatibilidad con otras librerías de Python, como NumPy, SciPy y matplotlib. [19] Además, esta librería soporta una gran variedad de algoritmos, como lo son KNN, bosque aleatorio, SVM, entre otros.

Ventajas

- Cuenta con mucha documentación.
- Proporciona muchos parámetros de configuración.
- Compatibilidad con otras librerías de Python.

Desventaja

- Sólo está disponible en Python.

2.8.3.4 Keras [20]

Es una librería de redes neuronales de código abierto desarrollada por François Chollet escrita en Python y lanzada en 2015.[21] Tiene la posibilidad de ejecutarse sobre TensorFlow, Microsoft Cognitive Toolkit o Theano.

En 2017, su marco fue adaptado e integrado en Tensorflow, esto lo hizo disponible para los usuarios de Tensorflow a través del módulo tf.keras.

Ventajas

- Es amigable con el usuario, modular y extensible.
- Tiene soporte con la librería de Tensorflow.
- Contiene modelos preentrenados. [21]

Desventajas

- Requiere mucha depuración ya que no es fácil encontrar la causa del error.
- Se tiene mucho menos control sobre la manipulación de las funciones.

2.8.3.5 PyBrain [22]

Es una librería de código abierto para el aprendizaje automático desarrollada con Python. Permite crear modelos de aprendizaje automático de forma rápida y sencilla [22].

Tiene compatibilidad con algunas plataformas como lo son Windows, Linux, entre otras. Y solo está disponible para el lenguaje de Python.

Ventajas

- Contiene algoritmos fáciles de utilizar.
- Fácil de integrar con otras librerías de Python para visualizar los datos.

Desventajas

- No hay mucha información en el compilador cuando ocurren errores.
- No ha sido actualizado recientemente.
- No hay mucha documentación para solucionar errores que puedan aparecer.

2.8.3.6 Comparación entre las herramientas para el desarrollo de la inteligencia artificial

	Ventajas	Desventajas
TensorFlow	Compatible con múltiples plataformas y lenguajes, como Python, C++ y Java. Ofrece una amplia documentación y recursos de aprendizaje. Incluye herramientas para la gestión y procesamiento de datos a gran escala.	Tiene una sintaxis compleja y requiere un aprendizaje considerable. Puede ser lento y demandante en términos de memoria, especialmente con modelos grandes.

PyTorch	Interfaz intuitiva y fácil de usar. Optimización eficiente del uso de memoria. Cuenta con una comunidad activa y documentación extensa. Es de código abierto.	Se necesita un servidor API para implementaciones en producción. Requiere herramientas adicionales para la visualización de datos.
Scikit-learn	Dispone de una gran cantidad de documentación. Ofrece múltiples opciones de configuración. Se integra bien con otras librerías de Python, como NumPy y SciPy.	Solo es compatible con Python.
Keras	Fácil de usar gracias a su estructura modular y extensible. Compatible con TensorFlow. Incluye modelos preentrenados listos para usar.	Puede ser difícil de depurar, ya que los errores no siempre son claros. Brinda menos control sobre la personalización de funciones.
PyBrain	Implementación sencilla de algoritmos de aprendizaje automático. Se integra fácilmente con otras	Escasa información de errores en el compilador. No ha recibido actualizaciones recientes. La

	herramientas de visualización de datos.	documentación disponible es limitada.
--	--	--

Tabla 3 Tabla resumen. Ventajas y desventajas de las herramientas para el desarrollo de la inteligencia artificial.

2.8.4 Herramientas para el reconocimiento de voz y transcripción

En esta sección se analizan diferentes herramientas para el reconocimiento de voz y también la funcionalidad de texto a voz.

2.8.4.1 GTTS [23]

Google Text-to-Speech por sus siglas gTTS, es una librería de Python y una herramienta CLI para interactuar con la API de texto a voz de Google Translate. Es compatible con una gran cantidad de idiomas, como inglés, hindi, español, francés, alemán, entre otros.

Se encuentra disponible en varias plataformas como Windows, macOS, Android, iOS.

Ventajas

- Cuenta con más de 220 voces y 40 idiomas. [24]
- Fácil de instalar.
- Fácil de usar.
- Compatible con varias plataformas de escritorio y móviles.

Desventajas

- Necesita una conexión a internet para poder usarse.
- La calidad del audio generado puede no ser suficientemente buena para algunas aplicaciones que requieran una voz más natural.

2.8.4.2 *Whisper* [25]

Es una tecnología que utiliza Inteligencia Artificial para el reconocimiento y la transcripción de voz o audios. Fue creado por OpenAI y lanzado por primera vez como software de código abierto en septiembre del 2022.

Cuenta con una aplicación WEB para poder usar su transcripción. Además, cuenta con una API en Python que permite crear tu propio programa para el reconocimiento de voz y la transcripción.

Ventajas

- Tiene una tasa de error de menos del 5% al transcribir al español.[26]
- Puede transcribir del inglés y otros idiomas.
- Puede traducir al mismo tiempo que está transcribiendo.

Desventajas

- Requiere la instalación de otras librerías para funcionar.

2.8.4.3 *Speech_Recognition* [27]

Es una librería de Python diseñada para facilitar el proceso de transcripción de voz a texto. Esta librería tiene una interfaz sencilla lo cual permite su integración con otros motores de reconocimiento de voz, tanto en la nube como de forma local.

Ventajas

- Es una de las librerías más utilizadas de Python para el reconocimiento de voz.
- Permite interactuar con otros motores de reconocimiento de voz, como Google Web Speech API.
- Permite el uso de múltiples formatos de audio, como .wav, .aiff, .flac.
- Fácil de instalar.

Desventajas

- No puede traducir mientras transcribe.

2.8.4.4 Wave [28]

Es una librería que permite crear y leer archivos de audio en formato WAV. De esta forma se pueden manipular los archivos con esta extensión.

Al abrir un archivo WAV puede obtener información como la cantidad de canales, la frecuencia del muestreo, el número de frames, etc., y extraer los datos del audio.

Cuando crea un archivo WAV puede establecer los parámetros como el número de canales, la tasa del muestreo, el tamaño de la muestra, etc.

Ventajas

- Fácil de instalar.
- Interfaz amigable con el usuario.
- Gran documentación.
- Funciona en diferentes sistemas operativos.

Desventajas

- Solo es capaz de procesar archivos .WAV.
- No incluye herramientas para procesamiento avanzado de los datos.

2.8.4.5 Pyaudio [29]

Es una librería de Python que permite la captura de audio desde el micrófono para la creación de archivos de audio. Además, se puede utilizar para la reproducción de audios. Se puede hacer uso de la misma en diferentes plataformas, como Windows, Linux, y MacOS.

Ventajas

- Fácil de usar.
- Interfaz intuitiva.

Desventajas

- No cuenta con mucha documentación.

2.8.4.6 DeepSpeech [30]

Es un motor de reconocimiento de voz basado en técnicas de aprendizaje profundo. Fue desarrollado originalmente por Mozilla y se inspiró en la investigación de Baidu Deep Speech.

Ventajas

- Es de código abierto y cuenta con una comunidad activa.
- Es fácil de integrar en aplicaciones de Python.

Desventajas

- Tiene un consumo alto de recursos dado que es un modelo Deep Learning.
- No tiene una alta precisión comparado con modelos más comerciales como los de Google o Amazon.

2.8.4.7 Pydub [31]

Es una librería de Python que es capaz de manipular los archivos de audio. Permite realizar muchas operaciones como, cortar, fusionar, aplicar efectos, convertir en otros formatos, entre otros.

Ventajas

- Es fácil de usar.
- Soporta múltiples formatos de audio.
- Puede reproducir el audio desde el código.

Desventajas

- Requiere la instalación de ffmpeg en el sistema, e instalar estas herramientas es tedioso.
- Tarda mucho tiempo en procesar el audio por lo que no es lo mejor para el procesamiento en tiempo real.

2.8.4.8 Comparación entre las herramientas para el desarrollo del reconocimiento de voz y transcripción

	Ventajas	Desventajas
gTTS	Cuenta con una gran variedad de voces e idiomas. Es compatible con múltiples plataformas y fácil de instalar.	Requiere conexión a internet. La calidad del audio puede no ser óptima.

Whisper	Tiene una tasa de error muy baja. Puede transcribir en múltiples idiomas, además de traducir mientras transcribe.	Requiere librerías adicionales para funcionar.
SpeechRecognition	Es muy utilizada en Python. Es compatible con muchos motores de reconocimiento. Soporte múltiples formatos de audio.	No puede traducir mientras transcribe.
Wave	Es fácil de instalar. Tiene una interfaz amigable. Cuenta con una documentación extensa y es compatible con múltiples sistemas operativos.	Solo trabaja con archivos .WAV. No incluye herramientas avanzadas de procesamiento.
PyAudio	Es fácil de usar ya que tiene una interfaz intuitiva.	Tiene una documentación limitada.
DeepSpeech	Es de código abierto y tiene una comunidad activa. Tiene una fácil integración con Python.	Tienen un alto consumo de recursos. Tiene menor precisión que modelos comerciales.

Pydub	Es fácil de usar. Puede cambiar el formato de audio.	Requiere instalaciones adicionales. No es muy rápido.
-------	--	--

Tabla 4 Tabla resumen. Ventajas y desventajas de las herramientas para el desarrollo del reconocimiento de voz.

2.8.5 Herramientas seleccionadas

Después del análisis de las diferentes tecnologías realizado en los puntos anteriores, para la correcta realización de este sistema, se han seleccionado las siguientes:

Almacenamiento de datos	MySQL
Visión por computadora	OpenCV
Desarrollo de la inteligencia artificial	TensorFlow, Keras
Reconocimiento de voz y transcripción	Wave, Speech_recognition y Pydub
Voz a texto	gTTS

Cuadro 2.5: Tabla resumen. Tecnologías seleccionadas para el desarrollo del sistema.

2.8.5.1 Almacenamiento de datos

Se ha seleccionado esta herramienta porque, a pesar de que no es completamente de código abierto, las funcionalidades que ofrece son más que suficientes para poder almacenar los datos de forma correcta. Además, al ser el más popular, existe una gran variedad de documentación y ejemplos de su uso y corrección de errores que puedan presentarse durante su uso.

2.8.5.2 Visión por computadora

Se ha seleccionado OpenCV debido a la gran cantidad de documentación existente. Además, contiene modelos de detección facial pre entrenados que facilitan la implementación del sistema para la detección facial, además su amplia variedad de herramientas para el tratamiento de imágenes y la facilidad de uso de estas es de mucha ayuda para poder modificarlas y aplicar los filtros necesarios a las mismas. Al ser un proyecto que necesitará muchas librerías OpenCV es de mucha ayuda por su facilidad de integrarse a diferentes librerías.

2.8.5.3 Desarrollo de la Inteligencia Artificial

Para el desarrollo de la red neuronal se ha seleccionado Keras, utilizándolo a través del módulo de TensorFlow. Esto por la facilidad que da al desarrollar la red neuronal, ya que es posible implementar las redes neuronales es pocas líneas de código.

La razón por la que se ha seleccionado TensorFlow es porque, además de la facilidad que da al poder implementar Keras como un módulo de TensorFlow, es compatible con varias plataformas y lenguajes de programación. Además, cuenta con el soporte de Google y de una gran comunidad a través de foros, esto beneficia mucho ya que se tiene mucha documentación y muchas páginas donde se resuelven errores comunes que se podrían presentar durante la implementación del software.

2.8.5.4 Reconocimiento de voz y transcripción

Se ha seleccionado Wave porque trabaja con archivos de audio de extensión .WAV, y trabajará en combinación con Pydub, que es capaz de convertir los archivos de audio de cualquier extensión a .WAV.

Se ha seleccionado Speech_Recognition debido a que permite trabajar con archivos .WAV, y dado que es una de las librerías más usadas de Python hay una gran variedad de documentación y páginas web con la resolución de muchos errores que pudieran darse durante la implementación del software. Además, cuenta con

una interfaz muy sencilla de utilizar y en muy pocas líneas de código se puede hacer la transcripción del audio.

2.8.5.5 Voz a texto

Se ha seleccionado gTTS (Google Text-To-Speech) ya que tiene una interfaz muy sencilla de utilizar, es fácil cambiar los parámetros para crear el audio del texto escrito y cuenta con una gran variedad de idiomas compatibles.

Capítulo 3 Propuesta de solución

En este capítulo de la presente documentación se describe la propuesta de solución del proyecto. Esta propuesta incluye desde el análisis y diseño del proyecto hasta las pruebas realizadas una vez implementado.

3.1 Análisis y diseño

En esta sección se exponen los requisitos del proyecto, tanto los requisitos funcionales como los no funcionales y sus restricciones. También se incluye un diagrama de casos de uso del sistema y un diagrama de secuencia para cada uno de los casos de uso incluidos en el diagrama mencionado.

3.1.1 Requisitos funcionales del proyecto

En este apartado se enlistan los requisitos funcionales que debe cumplir el proyecto:

- El sistema deberá ser capaz de reconocer a diferentes personas, tanto conocidas como desconocidas.
- El sistema deberá ser capaz de identificar las emociones de varias personas al mismo tiempo, tanto conocidas como desconocidas.
- A las personas desconocidas se les debe mostrar una etiqueta que los clasifique como *Desconocido*.
- El sistema deberá permitir agregar nuevas personas a los rostros reconocibles.

- El sistema deberá tener tres tipos de usuarios y cada uno tendrá diferentes permisos.
- El sistema deberá poder agregar nuevas materias.
- El sistema deberá crear gráficas y tablas al finalizar una sesión de reconocimiento facial y de emociones.
- El sistema deberá poder reconocer la voz y escribir lo que escuche siempre y cuando el usuario lo requiera.
- El sistema deberá poder tomar un texto escrito por el usuario y reproducirlo como audio.
- El sistema deberá ser capaz de utilizar como entrada la cámara del dispositivo o una nueva cámara conectada.
- El reconocimiento de emociones será desarrollado utilizando redes neuronales.
- Las gráficas de barras deberán presentar la información de manera legible.

3.1.2 Requisitos no funcionales del proyecto

En esta sección del documento se enlistan los diferentes requisitos no funcionales del sistema a desarrollar.

- El sistema debe ser escalable.

3.1.3 Requisitos de información

En esta sección del documento se enumeran los requisitos de información correspondientes al software necesario para el proyecto:

- Deben almacenarse las imágenes de los rostros de los usuarios, así como el vector de características de cada rostro registrado.
- Deben almacenarse las gráficas obtenidas después de cada ejecución del reconocimiento facial y de emociones para poder generar el reporte.

3.1.4 Restricciones

En esta sección del documento se enlistan las restricciones impuestas al proyecto:

- El sistema de reconocimiento facial y de emociones debe desarrollarse en Python.

3.1.5 Diagrama de casos de uso

A continuación, se incluye un diagrama con los casos de uso que se consideran necesarios y corresponden al sistema a desarrollar tras un estudio de los requisitos de este:

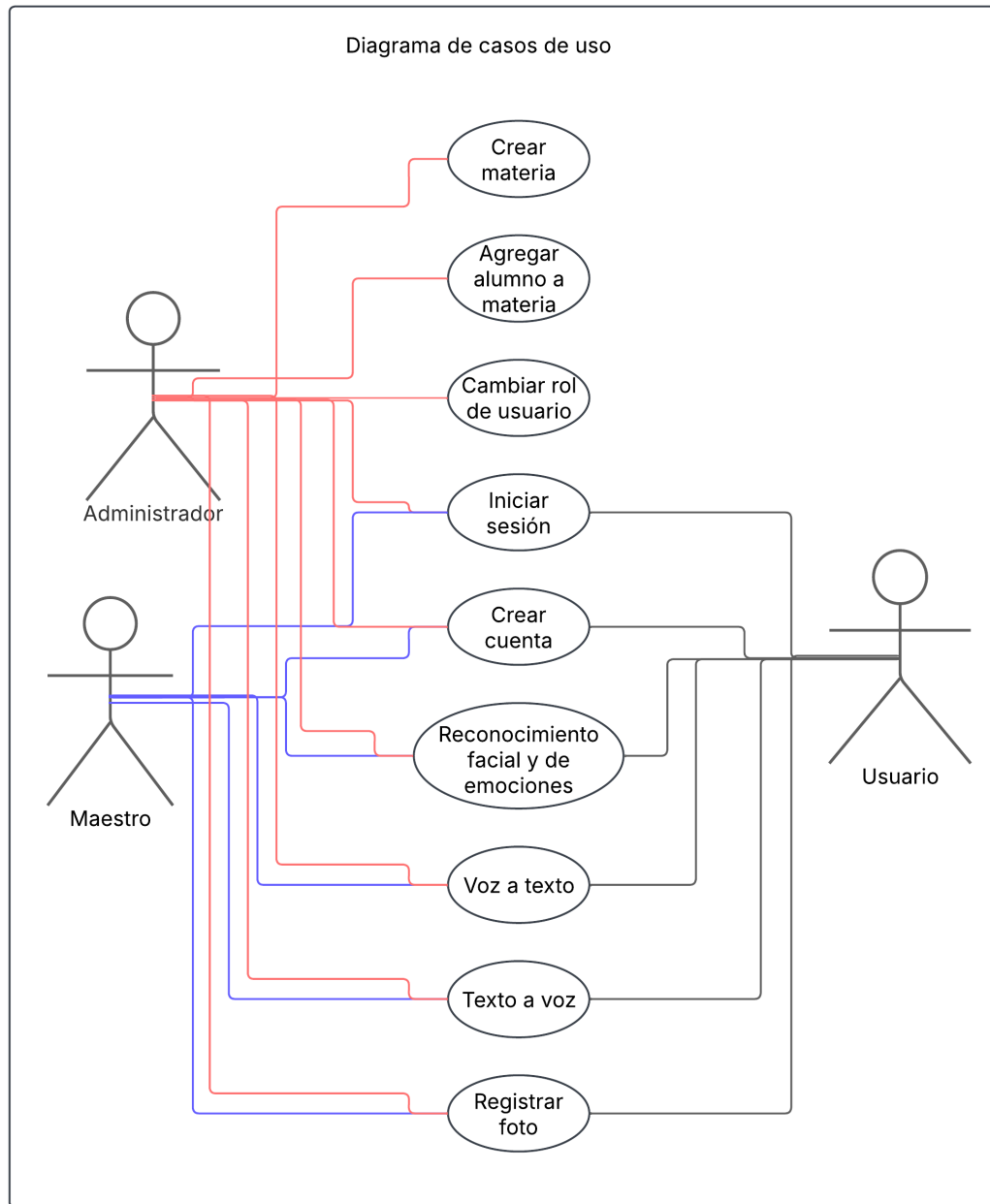


Fig. 2 Diagrama de casos de uso

Como se puede observar en el diagrama anterior el sistema cuenta con tres tipos de usuario y equis casos de uso:

Crear de cuenta: Un usuario nuevo puede crear una cuenta que por default será de usuario regular.

Iniciar sesión: Se podrá iniciar sesión con las credenciales correctas, dependiendo del tipo de usuario se mostrará una página diferente.

Crear materia: Un administrador podrá crear una nueva materia.

Agregar alumno a materia: Un administrador podrá agregar un usuario con rol de alumno a una materia.

Cambiar rol de usuario: Un administrador podrá modificar el rol de un usuario, desde convertirlo a usuario regular, maestro o administrador.

Reconocimiento facial y de emociones: Hay 2 flujos para el reconocimiento el del maestro y el de otros usuarios:

Maestro: A través de un flujo de cámara, se podrán reconocer todos los alumnos de la clase que el maestro imparta y sus emociones, si el usuario no está registrado la etiqueta del nombre será 'Desconocido' pero su emoción será reconocida. Al finalizar la detección, se guardarán los reportes de cada alumno reconocido.

Usuario: A través de un flujo de cámara, se podrá reconocer al usuario que inició sesión y sus emociones, si el usuario no tiene fotografía la etiqueta del nombre será 'Desconocido', si hay más de un usuario solo se reconocerá al usuario que inició sesión, los demás usuarios serán 'Desconocido' pero sus emociones serán reconocidas. Al finalizar la detección se redirigirá a el reporte de la sesión que se finalizó, pero los datos no se guardarán.

Registrar foto: A través de la toma de una fotografía con el sistema, se podrá agregar la foto de un nuevo usuario con su nombre para que el sistema pueda reconocer su rostro en una ejecución futura.

Maestro: Podrá elegir la materia y el alumno al que le va a asignar la fotografía.

Usuario regular: Si no es parte de una materia el usuario podrá actualizar su fotografía, si es parte de una materia el usuario no podrá actualizarla.

Voz a texto: Se podrá reconocer lo que un usuario diga y escribirlo.

Texto a voz: Se podrá tomar el texto que escriba un usuario y convertirlo en audio para su reproducción.

3.1.6 Diagramas de secuencia

En esta sección se ilustrarán los diagramas de secuencia para los casos de uso *Reconocimiento facial y de emociones*, *Registrar foto*, *Voz a texto*, *texto a voz*.

Diagrama de secuencia para Crear cuenta

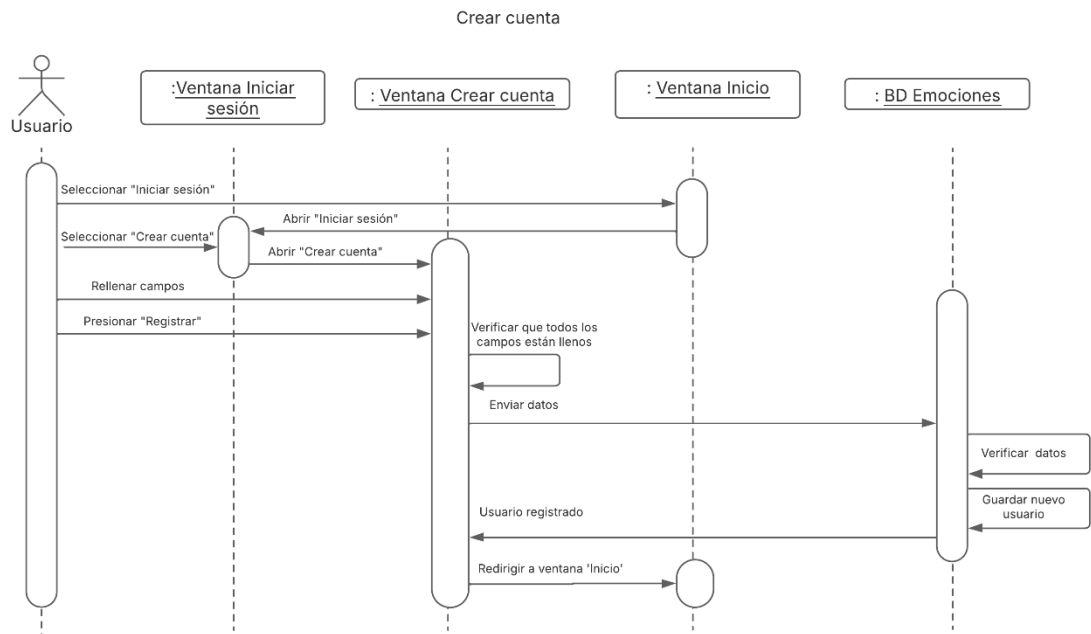


Fig. 3 Diagrama de secuencia para el caso de uso Crear cuenta.

Diagrama de secuencia para Iniciar sesión

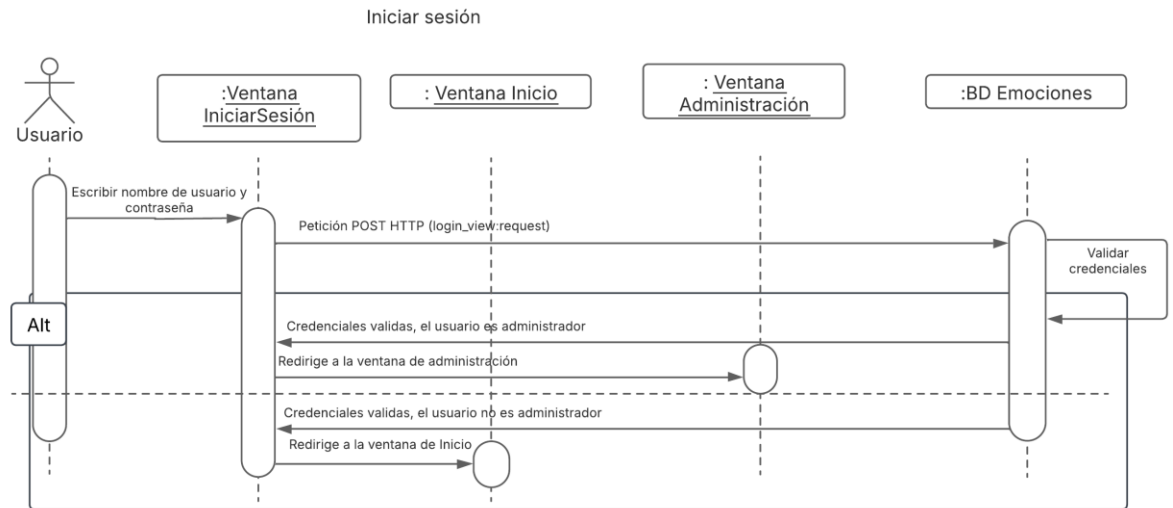


Fig. 4 Diagrama de secuencia para el caso de uso Iniciar sesión.

Diagrama de secuencia para Crear materia

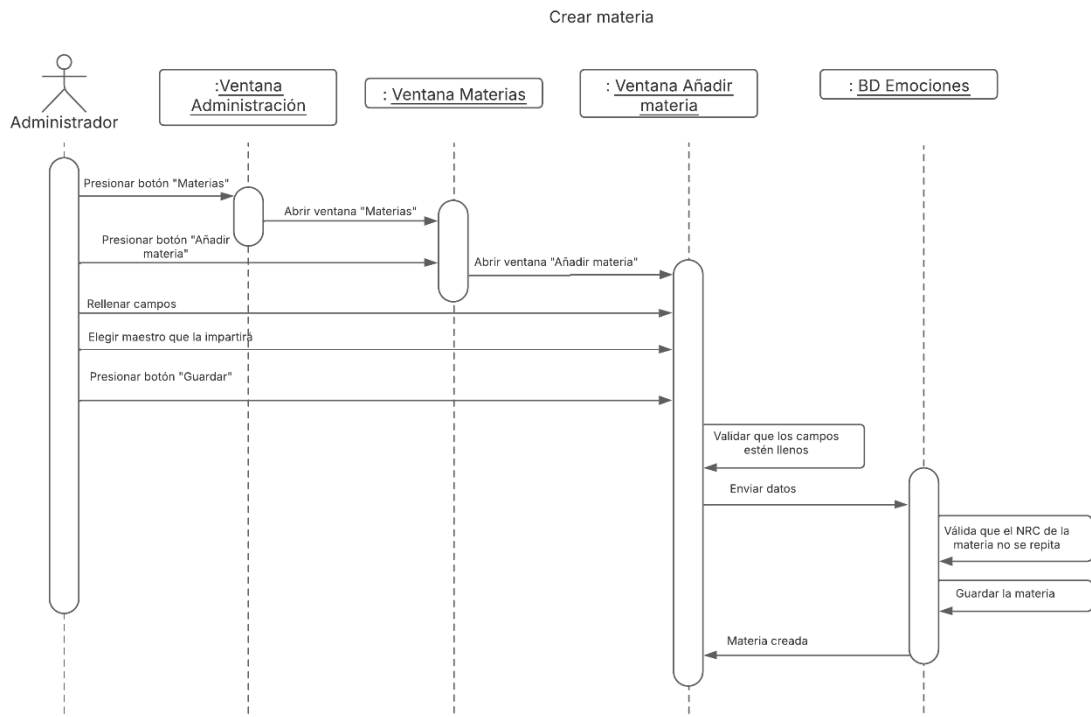


Fig. 5 Diagrama de secuencia para el caso de uso Crear materia.

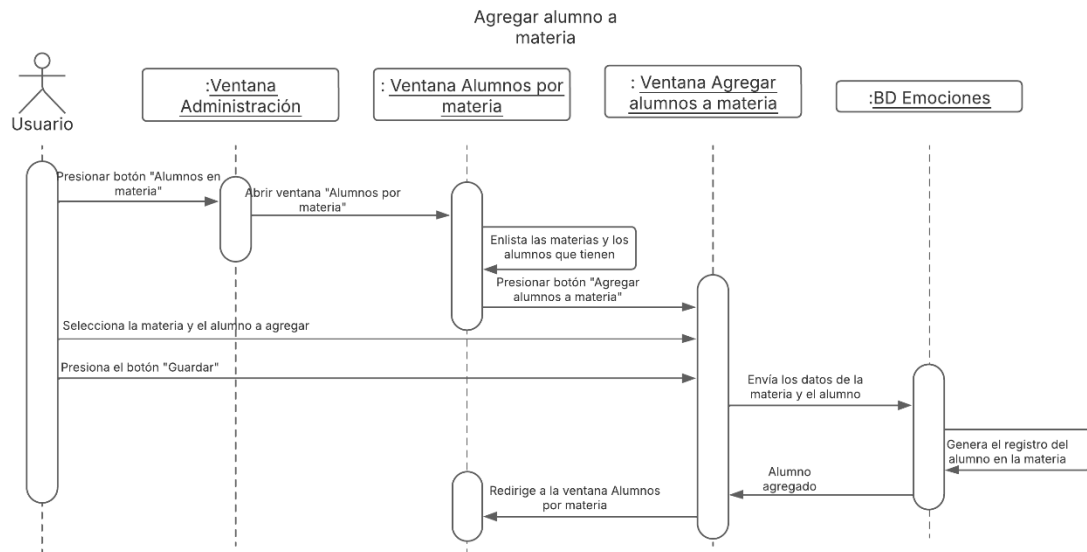
Diagrama de secuencia para Agregar alumno a materia

Fig. 6 Diagrama de secuencia para el caso de uso Agregar alumnos a materia.

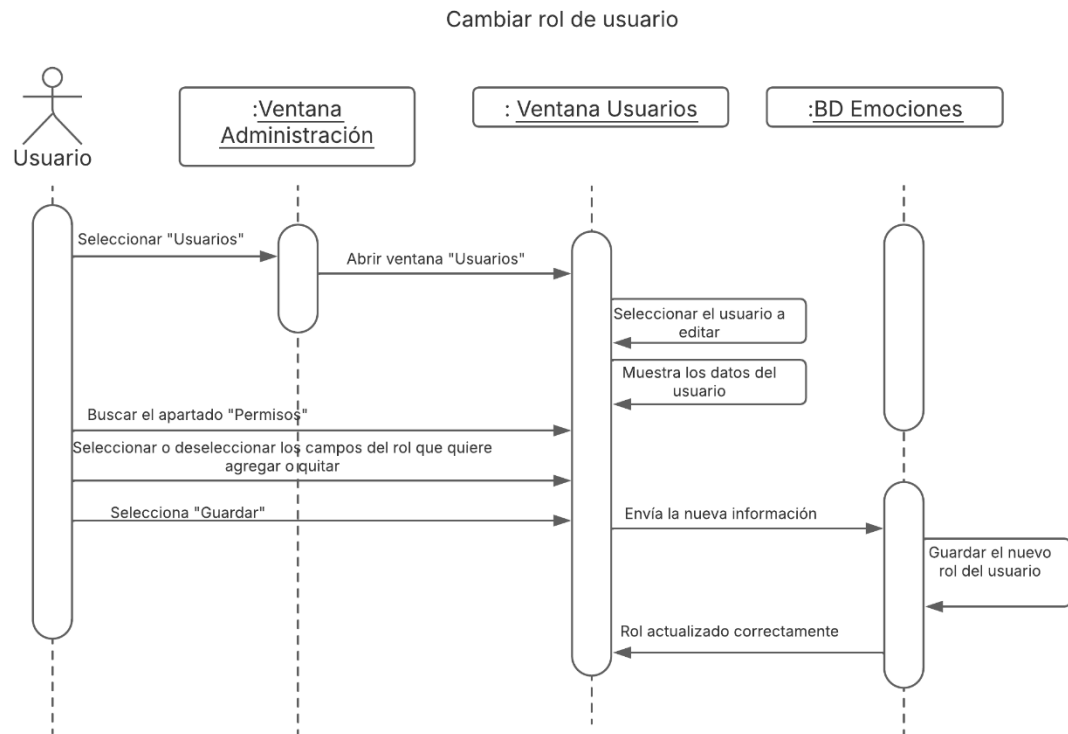
Diagrama de secuencia para Cambiar rol de usuario

Fig. 7 Diagrama de secuencia para el caso de uso Cambiar rol de usuario.

Diagrama de secuencia para Registrar fotografía

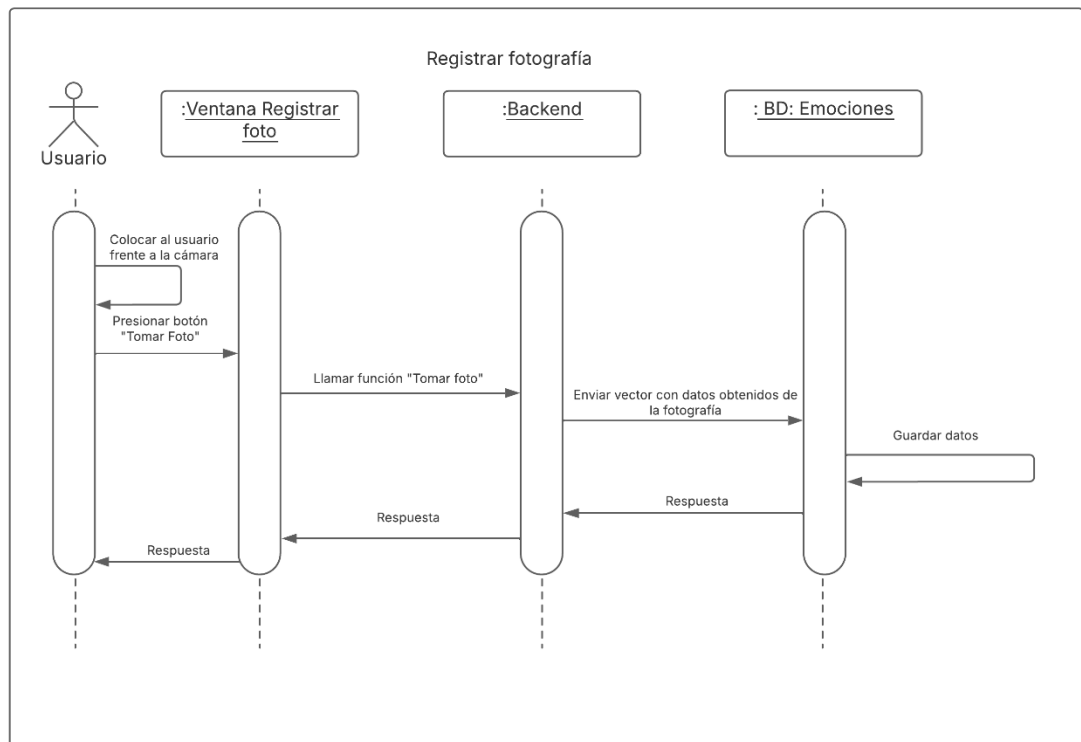


Fig. 8 Diagrama de secuencia para el caso de uso Registrar fotografía

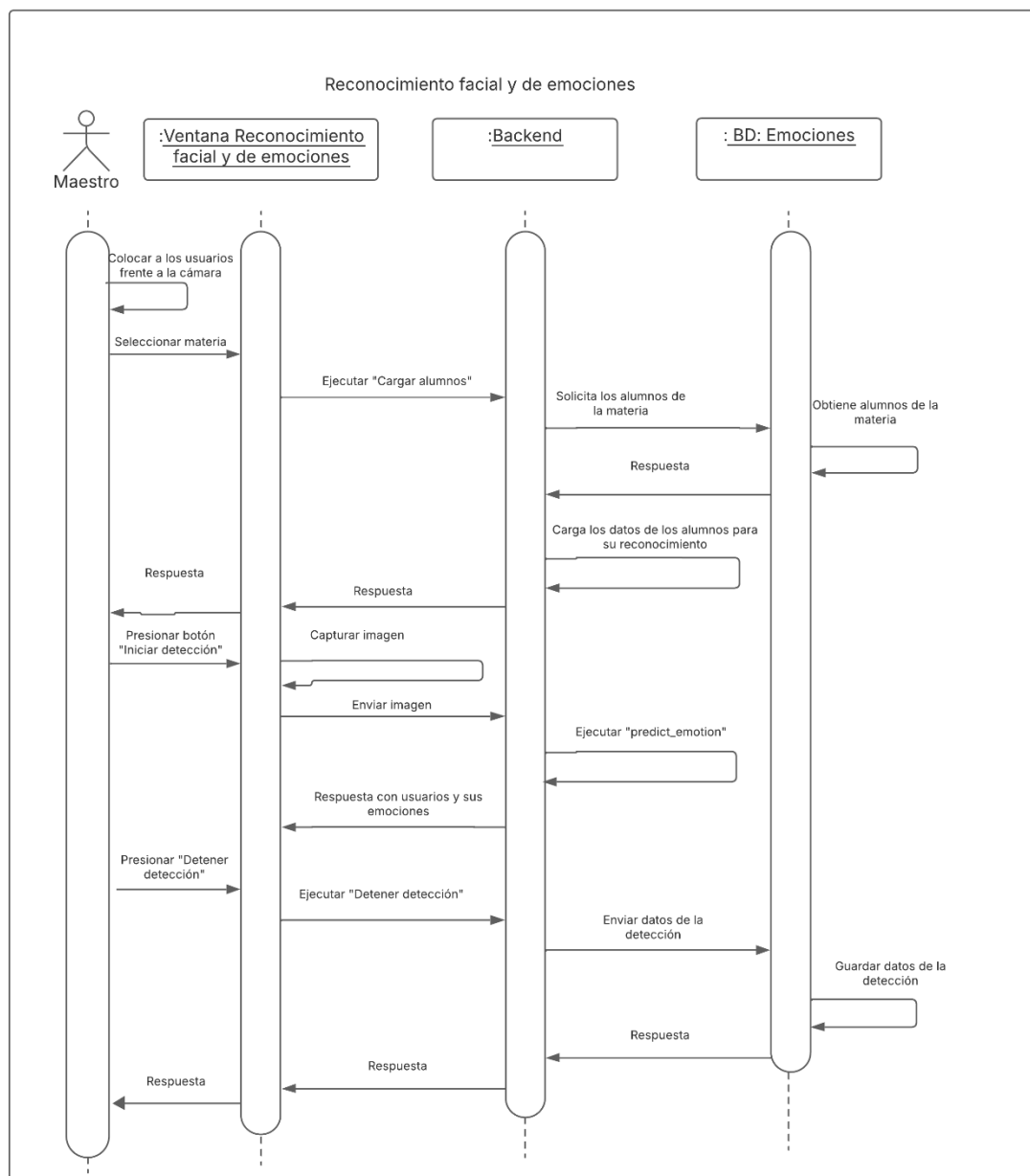
Diagrama de secuencia para Reconocimiento facial y de emociones

Fig. 9 Diagrama de secuencia para el caso de uso Reconocimiento facial y de emociones

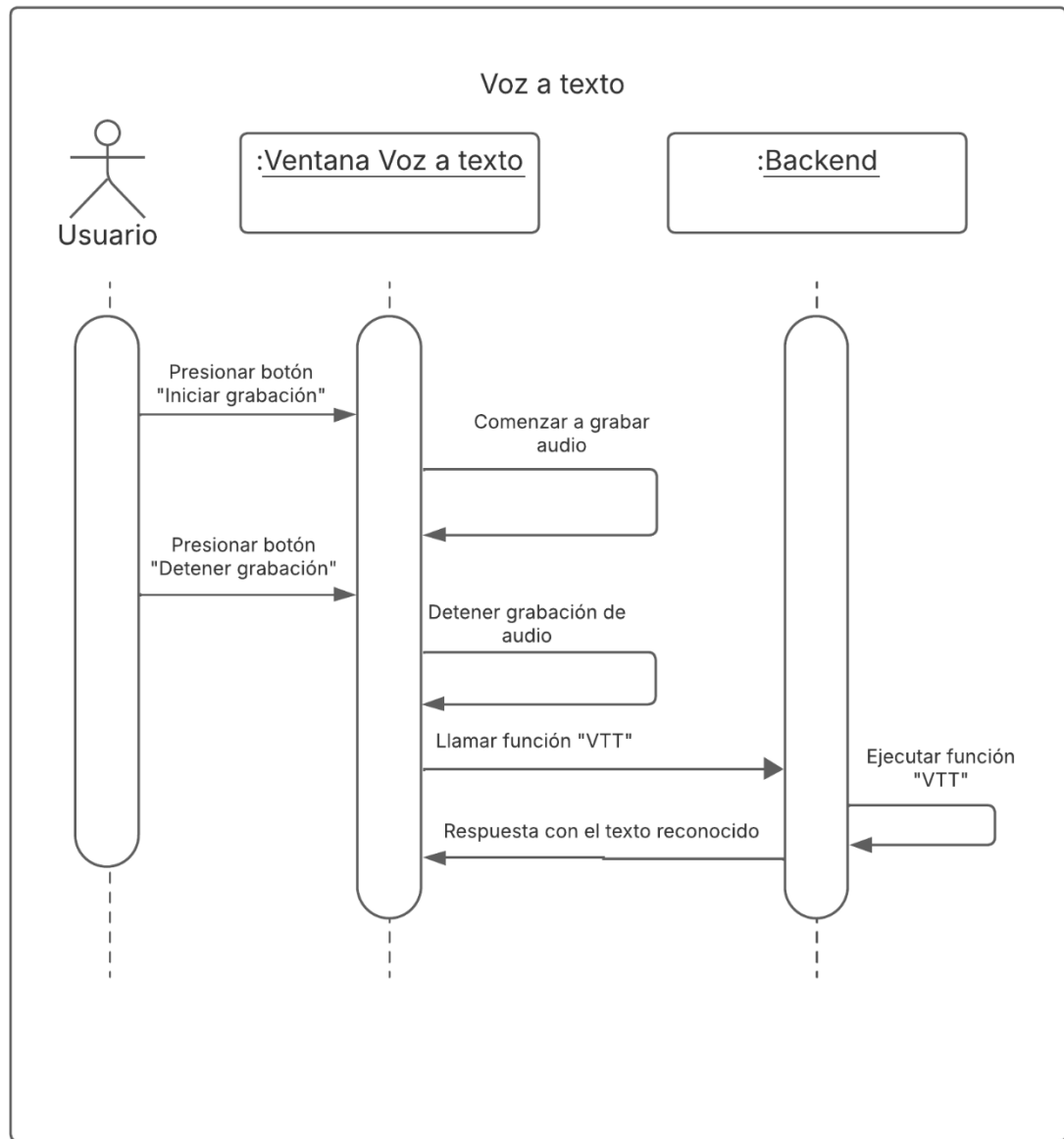
Diagrama de secuencia para Voz a texto

Fig. 10 Diagrama de secuencia para el caso de uso Voz a Texto

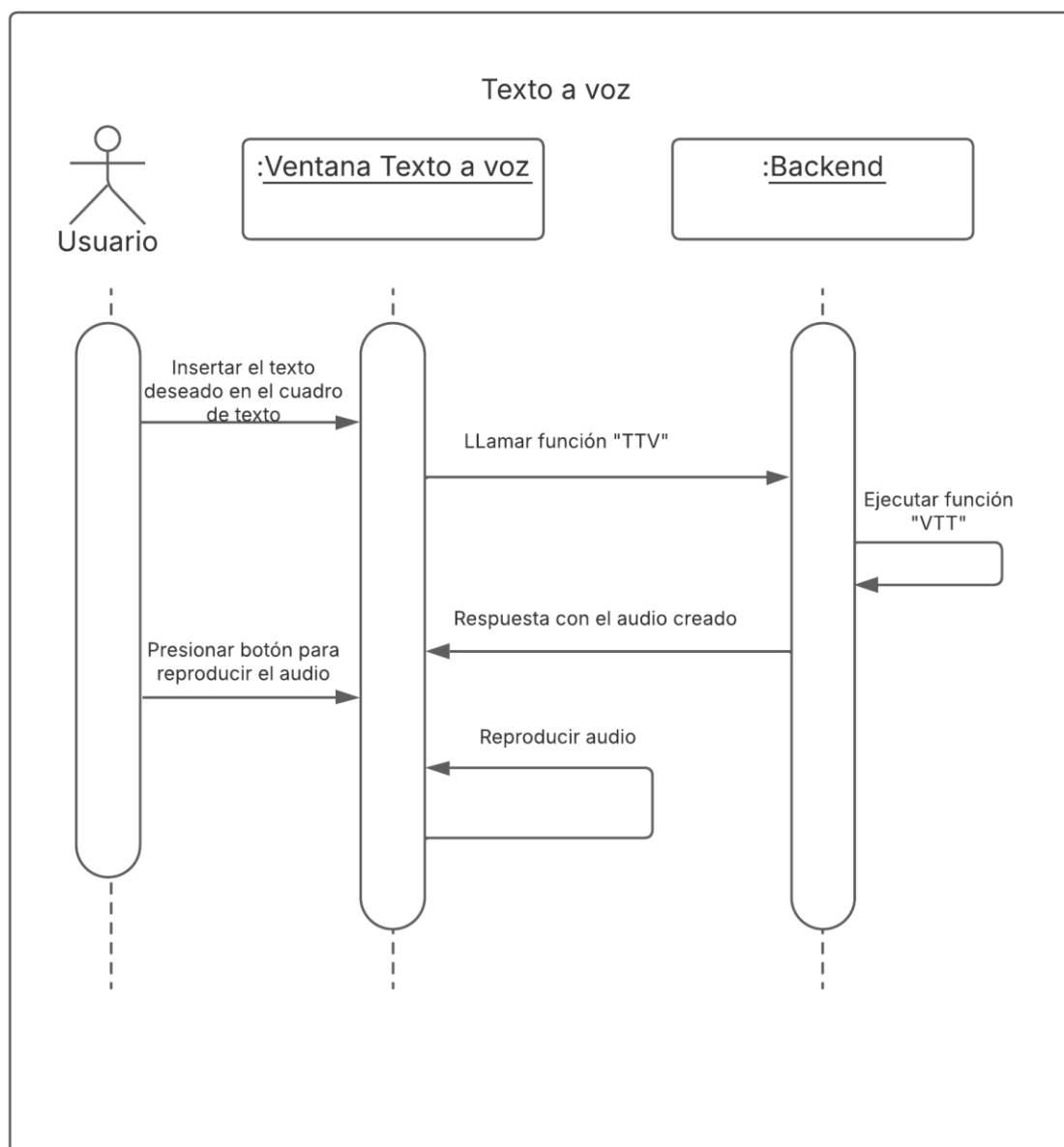
Diagrama de secuencia para Texto a voz

Fig. 11 Diagrama de secuencia para el caso de uso Texto a voz

3.2 Implementación y desarrollo

En este apartado se detallará toda la información referente al desarrollo de la red neuronal utilizada para el reconocimiento de emociones, incluyendo su entrenamiento; el desarrollo del algoritmo para el reconocimiento facial; el funcionamiento en conjunto para el reconocimiento facial y de emociones; el desarrollo de la base de datos utilizada; el desarrollo del sistema de texto a voz y de voz a texto; y las interfaces para los tres tipos de usuarios que puede haber en el sistema. Esto para dejar correctamente documentada la información del sistema a desarrollar.

3.2.1 Desarrollo del algoritmo para la detección facial

En esta sección se explica la creación del algoritmo para la detección facial que es fundamental para las pruebas de la red neuronal de reconocimiento de emociones y el funcionamiento del reconocimiento facial.

3.2.1.1 Explicación del desarrollo del algoritmo

El primer paso para realizar la detección facial es importar el módulo **cv2** de la librería OpenCV, con la que utilizaremos **CascadeClassifier** para cargar un modelo pre entrenado para la detección facial. En este caso utilizaremos el modelo para la detección facial frontal llamado **haarscade_frontalface_default.xml** ya que necesitamos ver el rostro de frente para poder reconocer las emociones de forma exitosa.

Después, desde la aplicación WEB se envía una imagen capturada con una cámara al BackEnd para ser procesada, esta imagen se obtiene en formato Base64 y se convierte en datos binarios (bytes) representativos de la imagen en su formato original. A partir de este conjunto de bytes, con la librería NumPy y su función **frombuffer**, se crea un arreglo de tipo **uint8**. El arreglo se decodifica y se crea una matriz que representa la imagen.

Se procede a redimensionar la imagen a un ancho de 640 píxeles, manteniendo la proporción del aspecto. Haciendo uso nuevamente de cv2 se convierte la imagen a escala de grises, esto para reducir la complejidad de la imagen y mejorar la eficiencia de la detección de rostros.

Por último y utilizando el modelo cargado anteriormente para la detección facial se utiliza el método **detectMultiScale** para detectar los rostros en la imagen en escala de grises y guardar sus coordenadas en una variable. Con estas coordenadas se obtiene la ubicación de cada rostro detectado dentro de la imagen, se recorta la ubicación del rostro, se redimensiona a un ancho y largo de 150x150 píxeles, es importante redimensionar las imágenes de estos rostros ya que la red neuronal para el reconocimiento de emociones se ha entrenado con imágenes de esta medida y si las imágenes son de una medida diferente no es posible procesarlas con la misma.

A continuación, se explica cada una de las funciones y archivos utilizados para el desarrollo de este algoritmo:

- **CascadeClassifier[32]**: Es una función utilizada para leer modelos de redes neuronales para la detección de objetos, esta obtiene como parámetro el archivo con el modelo para la detección, en este caso, de rostros.
- **Haarscade_frontalface_default.xml [33]**: Es un modelo pre entrenado para la clasificación de rostros frontales.
- **Frombuffer[34]**: Convierte datos de buffer a una matriz de NumPy.
- **Uint8**: Es un tipo de dato que representa un entero sin signo de 8 bits, es decir, un número entre 0 y 255.
- **detectMultiScale[35]**: Detecta los objetos de diferentes tamaños en la imagen de entrada. Los objetos detectados se regresan como una lista de rectángulos.

Los rostros obtenidos del procesamiento de la imagen de entrada se utilizan para el reconocimiento facial y el reconocimiento de emociones. En los siguientes apartados se utiliza este algoritmo de detección facial.

3.2.2 Desarrollo de la red neuronal para el reconocimiento de emociones

En esta sección se explica lo relacionado con el desarrollo de la red neuronal definida para este proyecto y también el entrenamiento de esta.

3.2.2.1 Descripción del modelo

Para el desarrollo de la red neuronal se requiere diseñar un modelo de Deep Learning. Se realizó una investigación en diferentes fuentes de internet, entre ellas un tutorial disponible en la plataforma YouTube, en donde se encontró una red neuronal que permitió comprender la implementación de redes neuronales convolucionales utilizando TensorFlow y Keras, y sirvió como base para el desarrollo del modelo propuesto [36]. A continuación, se describe la implementación de la red neuronal que se realizó para el sistema.

Para la implementación de la red neuronal se utilizará, como se ha mencionado anteriormente, el módulo Keras perteneciente a la librería de TensorFlow, por lo tanto, el lenguaje que se utilizará para el desarrollo es Python.

A continuación, se detalla la estructura de capas de la red neuronal desarrollada.

- **Modelo:** El modelo con el que se desarrollará la red neuronal es Secuencial, para esto se utiliza el método **Sequential**.
- **Entrada:** Las imágenes se redimensionan a 150x150 píxeles, y se convierten a escala de grises, por lo tanto, tienen 1 canal. Esto hace que la imagen sea más fácil de procesar sin perder características importantes.

- **Bloque 1:** Extracción de características
 - Se utiliza la clase **conv2d** para agregar una capa que convoluciona con la entrada para producir un **tensor** de salidas. Esta capa contiene 32 filtros de tamaño 3x3 lo que significa que esta capa aprenderá 32 características diferentes a través de filtros de 3x3, y cada filtro se optimiza durante el entrenamiento para obtener la información más relevante de la entrada.
 - Se utiliza **batchNormalization**, lo cual crea una capa que normaliza la salida de la convolución anterior para estabilizar el entrenamiento.
 - Se utiliza **Activation**, la cual aplica una función de activación, en este caso la función **ReLU**, a la salida.
 - Se utiliza **MaxPool2D**, la cual reduce las dimensiones espaciales de la entrada (150x150), manteniendo las características más relevantes y obteniendo una nueva dimensión de 75x75.
 - Se utiliza **Dropout**, la cual aleatoriamente ajusta los valores de entrada de la capa anterior a 0 para evitar el sobreajuste, en este caso el 20%.
- **Bloque 2:**
 - Se utiliza la clase **conv2d** para agregar otra capa que convoluciona con la salida del **Dropout** anterior. Esta capa contiene 64 filtros de tamaño 5x5.
 - Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.
 - Se agrega una nueva capa de **Activation** con la función de activación **ReLU**.
 - Se utiliza **MaxPool2D** para reducir nuevamente las dimensiones de la nueva entrada (75x75) conservando las características más relevantes y obteniendo una nueva dimensión de 37x37.
 - Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta el 20% de los valores.
- **Bloque 3:**

- Se utiliza la clase **conv2d** para agregar una nueva capa que convolucione la salida del **Dropout** anterior. Esta capa contiene 128 filtros de tamaño 3x3.

- Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.

- Se agrega una nueva capa de **Activation** con la función de activación **ReLu**.

- Se utiliza **MaxPool2D** para reducir nuevamente las dimensiones de la nueva entrada (37x37) conservando las características más relevantes y obteniendo una nueva dimensión de 18x18.

- Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta el 20% de los valores.

- **Bloque 4:**

- Se utiliza la clase **conv2d** para agregar una nueva capa que convolucione la salida del **Dropout** anterior. Esta capa contiene 256 filtros de tamaño 3x3.

- Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.

- Se agrega una nueva capa de **Activation** con la función de activación **ReLu**.

- Se utiliza **MaxPool2D** para reducir nuevamente las dimensiones de la nueva entrada (18x18) conservando las características más relevantes y obteniendo una nueva dimensión de 9x9.

- Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta el 20% de los valores.

- **Bloque 5:**

- Se utiliza la clase **conv2d** para agregar una nueva capa que convolucionada la salida del **Dropout** anterior. Esta capa contiene 528 filtros de tamaño 3x3.

- Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.

- Se agrega una nueva capa de **Activation** con la función de activación **ReLU**.

- Se utiliza **MaxPool2D** para reducir nuevamente las dimensiones de la nueva entrada (9x9) conservando las características más relevantes y obteniendo una nueva dimensión de 4x4.

- Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta el 20% de los valores.

- **Aplanamiento:** La salida del bloque 5 es un mapa de características 3D, esto porque se tiene una dimensión de 4x4x528, 4x4 son las características más relevantes y 528 son los filtros agregados en la capa **conv2d**. Esta capa utiliza **Flatten**, que es una capa de aplanamiento que convierte un mapa de características 3D en un vector 1D, en este caso, obteniendo 8448 neuronas para la clasificación en la siguiente capa.

- **Capa densa 1:**

- Se utiliza **Dense** para agregar una capa densa, esto significa que crea 128 neuronas completamente conectadas, esto quiere decir que cada neurona de esta capa tendrá 8192 conexiones con la capa de aplanamiento anterior.

- Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.

- Se agrega una nueva capa de **Activation** con la función de activación **ReLU**.

- Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta

el 20% de los valores.

- **Capa densa 2:**

- Se utiliza **Dense** para agregar una capa densa que crea 256 neuronas completamente conectadas, esto quiere decir que cada neurona de esta capa tendrá 128 conexiones con la capa de aplanamiento anterior.

- Se agrega una nueva capa de **batchNormalization** para estabilizar el entrenamiento.

- Se agrega una nueva capa de **Activation** con la función de activación **ReLU**.

- Agrega un nuevo **Dropout**, el cual vuelve a ajustar valores aleatorios de entrada de la capa anterior a 0 para evitar el sobreajuste, nuevamente se ajusta el 20% de los valores.

- **Capa de salida:** Para finalizar se utiliza **Dense** para agregar una capa de salida que tendrá 7 neuronas completamente conectadas, correspondientes a las 7 emociones, y contará con la función de activación **softmax**, esta función devuelve la probabilidad de pertenencia a cada clase o emoción.

A continuación, se explica cada una de las funciones y conceptos que se utilizaron para la creación del modelo de la red neuronal:

Sequential: Esta función es la definición de la red neuronal, es decir, que estructura tendrá esta red. Este modelo hace que las capas estén una tras otras, de esta forma la salida de una capa es la entrada de la siguiente.

conv2d: Esta función convoluciona con la capa de entrada para producir un tensor de salidas.

tensor: Son matrices multidimensionales con un tipo uniforme. Sirven para almacenar datos.

batchNormalization: Esta función aplica una transformación que mantiene la salida media cerca de cero y la desviación estándar de la salida cerca de 1.

Activation: Esta función aplica una función de activación a una salida.

ReLu: Es una función de activación de la unidad lineal rectificada, esta transforma el resultado de entrada con el siguiente algoritmo:

- Si el valor de entrada es menor que 0, muestra 0.
- Si el valor de entrada es mayor o igual que 0, muestra el valor de entrada.

MaxPool2D: Esta función reduce la entrada (alto y ancho), eliminando las características menos importantes y conservando las más importantes.

Dropout: Aunque esta no se considera una capa de las redes neuronales, es una parte importante ya que se encarga de establecer valores de entrada aleatorios a 0, lo que ayuda a evitar el sobre entrenamiento de esta.

Flatten: Función que agrega una capa de aplanamiento que convierte un mapa de características 3D en un vector 1D, es decir, si se tiene un mapa de características 3x3x3 esta función creará un vector de 9 neuronas para la siguiente capa.

Dense: Es un tipo de capa de redes neuronales, esta capa recibe una conexión (neurona) de cada valor de la capa anterior a el nuevo valor de esta capa.

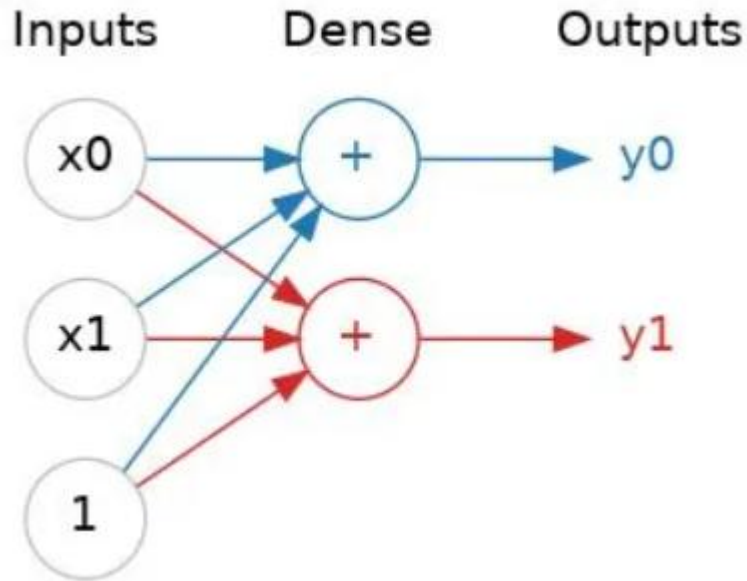


Fig. 12 Ejemplo de capa densa: Cada dato de entrada tiene una conexión a la capa densa.

Softmax: Es una función de activación que convierte las salidas de la capa anterior en probabilidades que suman uno. Estas probabilidades se usan para medir la confiabilidad de pertenencia a cada clase.

Los sesgos y pesos no se mencionan en la creación de las capas porque son aleatorios, pero se describen a continuación:

Sesgo o bias: Es un parámetro que se suma a la combinación ponderada de las entradas de una neurona antes de aplicar la función de activación. Este valor permite que la neurona ajuste su umbral de activación, para que el modelo pueda aprender patrones más complejos.

Peso: Es un valor que determina la influencia de una neurona sobre otra. Es decir, determina la importancia de la señal que transmite la neurona. Durante el entrenamiento estos valores se ajustan para que la red pueda aprender patrones más complejos.

La estructura del modelo se compila y está listo para entrenar. A continuación, se muestra una tabla que resume el proceso que seguirá la red neuronal.

Bloque / Capa	Tipo de Capa	Detalles / Parámetros	Dimensiones de Salida
Bloque 1	Conv2D	32 filtros, tamaño 3x3	150x150x32
	BatchNormalization	Normaliza la salida	150x150x32
	Activation (ReLU)	Función de activación ReLU	150x150x32
	MaxPooling2D	Reduce dimensiones espaciales a la mitad	75x75x32
	Dropout	20% de unidades desactivadas aleatoriamente	75x75x32
Bloque 2	Conv2D	64 filtros, tamaño 5x5	75x75x64
	BatchNormalization		75x75x64
	Activation (ReLU)		75x75x64
	MaxPooling2D	Reduce dimensiones espaciales a la mitad	37x37x64
	Dropout	20%	37x37x64
Bloque 3	Conv2D	128 filtros, tamaño 3x3	37x37x128
	BatchNormalization		37x37x128
	Activation (ReLU)		37x37x128

	MaxPooling2D	Reduce dimensiones espaciales a la mitad	18x18x128
	Dropout	20%	18x18x128
Bloque 4	Conv2D	256 filtros, tamaño 3x3	18x18x256
	BatchNormalization		18x18x256
	Activation (ReLU)		18x18x256
	MaxPooling2D	Reduce dimensiones espaciales a la mitad	9x9x256
	Dropout	20%	9x9x256
Bloque 5	Conv2D	528 filtros, tamaño 3x3	9x9x528
	BatchNormalization		9x9x528
	Activation (ReLU)		9x9x528
	MaxPooling2D	Reduce dimensiones espaciales a la mitad	4x4x528
	Dropout	20%	4x4x528
Aplanamiento	Flatten	Convierte el mapa 3D (4x4x528) a un vector 1D de 8448 elementos	8448
Capa densa 1	Dense	128 neuronas totalmente conectadas	128
	BatchNormalization		128
	Activation (ReLU)		128
	Dropout	20%	128

Capa densa 2	Dense	256 neuronas totalmente conectadas	256
	BatchNormalization		256
	Activation (ReLU)		256
	Dropout	20%	256
Capa de salida	Dense (Softmax)	7 neuronas (una por emoción), activación softmax (clasificación multiclase)	7

Tabla 5 Tabla resumen. Modelo de la red neuronal creada

En la siguiente sección se describe el proceso que se siguió para el entrenamiento del modelo.

3.2.2.2 Entrenamiento del modelo

Para el entrenamiento del modelo se realizó una búsqueda en distintas páginas de internet de diferentes bases de datos de imágenes de diferentes personas mostrando las siete diferentes emociones que reconocerá el sistema, felicidad, neutral, tristeza, asco, enojo, miedo y sorpresa, una vez que se encontraron 3 bases de datos grandes se comenzó a analizar el contenido de las fotografías para ver si cumplían ciertos puntos necesario:

- Contener las siete emociones necesarias.
- Las imágenes deben estar etiquetadas.
- Debe haber un conjunto grande de cada emoción.

La única base de datos que cumplía estos puntos fue FER-2013 [37], esta es una base de datos de imágenes que cumple los 3 puntos anteriores, contiene un mínimo de 1000 imágenes en casi todas las categorías excepto por 'asco' pero

como es una emoción difícil de capturar es lo mejor que se pudo encontrar, además las imágenes se encuentran etiquetadas por carpeta. Es una base de datos abierta.

Una vez seleccionada la base de datos con la que se va a entrenar se comienza a aplicar filtros a las diferentes imágenes de entrenamiento y pruebas, como las imágenes son de 48x48 píxeles se comienza redimensionándolas a 150x150 píxeles, después se crean **lotes** de un tamaño predefinido, dado que es una base de datos muy grande estos lotes son importantes para que la red procese los datos por separado y esto hace más eficiente el entrenamiento, y por último se realiza la conversión de las imágenes de entrenamiento y pruebas a escala de grises.

Una vez las imágenes están listas, se procede a realizar el entrenamiento con la función ***fit*** que tiene Keras. Esta función recibe los siguientes parámetros:

Datos de entrenamiento: Son las imágenes que se procesaron para entrenar.

Etiquetas de los datos de entrenamiento: Son las etiquetas correspondientes a las imágenes de entrenamiento.

Batch_size: Corresponde a la cantidad de imágenes que se enseñan a la misma vez a la red neuronal. Este valor se definió como 200.

Épocas: Es el número de veces que se repite el entrenamiento.

validation_data: Son los datos con los que se evaluará la pérdida del modelo al final de cada época.

callbacks: un vector donde se especifican los callbacks de Keras [38].

shuffle: Definido como true, esto hace que los datos se mezclen en cada iteración lo que cambia el orden de estos. Esto se hace para reducir el sobreentrenamiento de la red neuronal.

3.2.3 Desarrollo del algoritmo para el reconocimiento facial

En esta sección se explica el desarrollo del algoritmo para el reconocimiento facial y el funcionamiento de este para agregar el rostro de un nuevo usuario a los rostros reconocibles.

3.2.3.1 Codificación del rostro de una nueva persona

Utilizando el algoritmo de detección facial descrito en el punto 3.1.2.1 se obtienen los rostros y se tienen que codificar para poder relacionarse con un nombre para su posterior reconocimiento. A continuación, se describe el algoritmo que se realizó para decodificar el rostro y guardar su vector para el reconocimiento de dicho rostro a futuro.

Utilizando la librería `face_recognition` y su función `face_encodings` se le da como entrada el rostro a codificar y se obtiene como respuesta un vector de 128 elementos que contiene una descripción única de las características faciales. Este vector se guarda en la base de datos en el registro correspondiente al usuario al que pertenece la imagen, de esta forma el rostro puede ser relacionado con el nombre al que pertenece para su futuro reconocimiento en el sistema.

3.2.3.2 Ejemplo de cómo agregar un nuevo rostro para su reconocimiento

En este apartado se describe el proceso para agregar un nuevo rostro para su reconocimiento en un futuro, se muestra los pasos a realizar en el sistema y un ejemplo del vector de características obtenido del rostro.

Agregar rostro siendo Maestro:

1. Iniciar sesión con las credenciales de maestro
2. Dirigirse al apartado “Registrar foto”
3. Seleccionar la materia a la que pertenece el alumno al que se le quiere asignar su fotografía
4. Seleccionar el alumno al que se le quiere agregar la fotografía

5. El nombre del alumno se llenará automáticamente, si el usuario no tiene una foto agregada se mostrará una foto de ejemplo, si el usuario ya tiene una foto agregada se mostrará su foto.

6. Colocar al alumno delante de la cámara mirando a la misma.

7. Presionar el botón “Tomar foto”.

8. Se mostrará la foto tomada, si el maestro aprueba la foto deberá presionar “Guardar foto” y la foto se guardará en el perfil del usuario, así como el vector de características de su rostro.

9. Repetir los pasos del 4 al 8 para tomar la fotografía a otro usuario de la misma materia o del 5 al 8 para tomar la fotografía de otro usuario de otra materia.

Agregar rostro siendo usuario:

1. Iniciar sesión con las credenciales de usuario o administrador

2. Dirigirse al apartado “Registrar foto”

3. El nombre del usuario se llenará automáticamente, si el usuario no tiene una foto agregada se mostrará una foto de ejemplo, si el usuario ya tiene una foto agregada se mostrará su foto.

4. Colocarse delante de la cámara mirando a la misma.

5. Presionar el botón “Tomar foto”.

6. Se mostrará la foto tomada, si el usuario aprueba la foto deberá presionar “Guardar foto” y la foto se guardará en el perfil del usuario, así como el vector de características de su rostro.

El proceso se divide en dos partes ya que un Maestro no puede agregar una foto para él mismo, solo puede agregar fotos para sus alumnos, mientras que un administrador solo puede agregar la foto para su propio perfil. A continuación, se muestra un ejemplo de un rostro, como se recorta el rostro y también el vector obtenido perteneciente a este rostro.

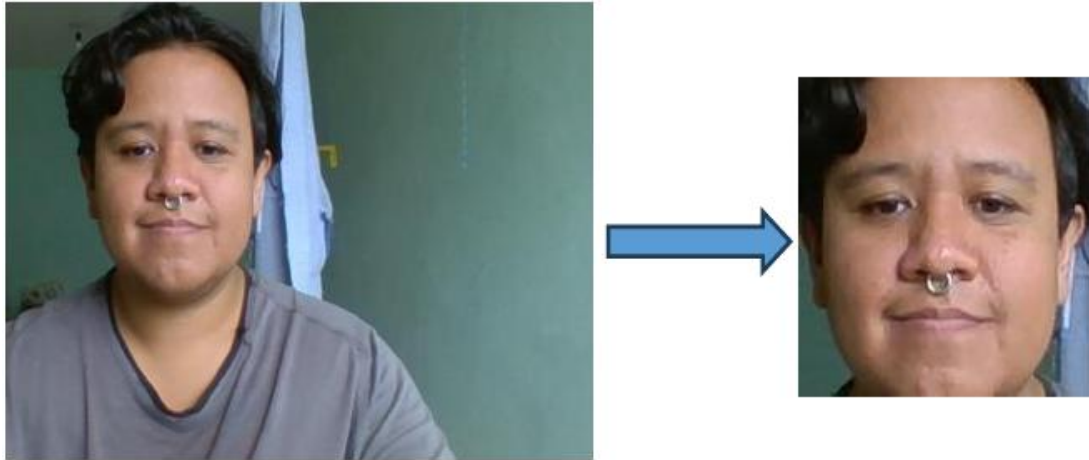


Fig. 13 La primer imagen la foto tomada y la segunda imagen es el rostro reconocido al que se le extraerá el vector de características.

```
[ -0.04082341  0.02222829  0.05258732 -0.02327272 -0.05173363 -0.06887461
-0.01608595 -0.13334163  0.11972953 -0.16553183  0.27157414 -0.04172546
-0.14529659 -0.1126314  0.01297567  0.18887576 -0.1268007  -0.11979634
-0.02991596 -0.01431327  0.06628613 -0.07289405 -0.01074866  0.01679712
-0.10477586 -0.32212457 -0.0803803  -0.14108318 -0.08141044 -0.04970947
-0.0205569  0.02078171 -0.16463913 -0.02520066  0.00897129  0.03964179
-0.05075607 -0.05028171  0.17963406  0.00608109 -0.20927538 -0.04929547
 0.0429699  0.22098014  0.14243126  0.0550229  0.01217872 -0.10058454
 0.16322708 -0.19816934  0.09936512  0.0154504  0.16434604  0.01936679
 0.05914638 -0.12278414  0.07827007  0.09009363 -0.21213196  0.04170052
 0.08281729 -0.04654858 -0.08177521 -0.01889667  0.3338044  0.17532589
-0.13263996 -0.09117936  0.15745163 -0.10092872 -0.07945953 -0.0084404
-0.13662022 -0.20265071 -0.29468361  0.0213593  0.38664204  0.06419919
-0.25525931 -0.05366677 -0.04950103  0.03621998  0.19124679  0.11628804
-0.03190005 -0.03020861 -0.09010592 -0.06569601  0.2125562  -0.08224593
-0.06295141  0.19752935 -0.06766225  0.00563328 -0.0064963  -0.03218972
-0.07181251  0.06775571 -0.12779599 -0.05825019  0.08180321 -0.04557351
-0.03350078  0.14282674 -0.21661527  0.07685406 -0.02697115  0.0214708
 0.08854987  0.00873288 -0.1701744  -0.12089907  0.07936343 -0.19382067
 0.17743038  0.1345697  0.04448419  0.1005656  0.09233767  0.0704858
 0.01425242 -0.02470944 -0.27368826  0.01314098  0.12927401  0.0045325
 0.04387645 -0.00987464]
```

Fig. 14 Vector de características resultante del rostro de la Fig. 13.

3.2.4 Desarrollo de la base de datos

En esta sección se explica cómo se desarrolló la base de datos y su estructura para el correcto funcionamiento del sistema.

3.2.4.1 Estructura de la base de datos.

Para el desarrollo de la base de datos, el primer paso es diseñar la estructura, desde las tablas, sus campos y como se conectarán unas con otras.

Las tablas necesarias son:

Usuarios: Para guardar las credenciales y datos como el nombre, nombre de usuario, rol, correo, etc., de los usuarios registrados y de los de posibles nuevos usuarios.

FotoUsuario: Para guardar la ubicación de la foto de cada usuario y su respectivo vector de características.

Materia: Para guardar los datos como el nombre, la descripción, el maestro que la imparte y el NRC, de las materias registradas y las posibles nuevas materias.

Materia_alumnos: Esta tabla es la conexión entre los usuarios y las materias que cursan.

ArchivoCSV: Para guardar la ubicación del CSV y guardar los datos de los reconocimientos faciales y de emociones, la fecha del registro y el usuario y materia a la que este registro pertenece.

A continuación, se muestra el diagrama Entidad-Relación de la base de datos desarrollada.

3.2.4.2 Desarrollo de la base de datos.

Una vez que se tiene la estructura de la base de datos se procede a su creación, para esto se utilizó el módulo “models” de Django, en este se puede definir la

estructura de una base de datos usando clases de Python. Se optó por implementarla de esta forma porque se puede crear la base de datos usando el lenguaje Python en lugar de instalar el Sistema gestor de bases de datos de MySQL y escribirlo en SQL. A continuación, se mostrará el código con el que se crearon las tablas:

La tabla “Usuarios” ya existe desde que se realiza la conexión del proyecto a la base de datos. Así que solo se conectaron las vistas de inicio de sesión y crear cuenta con esta tabla para su funcionamiento.

Tabla FotoUsuario:

```
class FotoUsuario(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    foto = models.ImageField(upload_to='fotos_usuarios/')
    vectorRostro = models.TextField(blank=True) # Almacena el vector como texto

    def __str__(self):
        return f'Foto del usuario {self.user_id}'
```

Fig. 15 Clase con la que se creó la tabla FotoUsuario.

Tabla Materia

```
class Materia(models.Model):
    id = models.AutoField(primary_key=True)
    nrc = models.CharField(max_length=10)
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField()
    maestro = models.ForeignKey(User, on_delete=models.CASCADE, limit_choices_to={'is_staff': 1},
    related_name='materias_iniciosesion', blank=True, null=True)
    class Meta:
        managed = False #Evita que Django intente crearla de nuevo
        db_table = 'iniciosesion_materia'
    def __str__(self):
        return self.nombre
```

Fig. 16 Clase con la que se creó la tabla Materia.

Tabla Materia_Alumnos

```

class MateriaAlumnos(models.Model):
    id = models.AutoField(primary_key=True)
    materia = models.ForeignKey(Materia, on_delete=models.CASCADE, related_name='materias_alumnos')
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='alumnos_materias', limit_choices_to=
{'is_staff': False, 'is_superuser': False})

    class Meta:
        verbose_name = "Agregar alumnos a materia"
        verbose_name_plural = "Agregar alumnos a materia"
        managed = False #Evita que Django intente crearla de nuevo
        db_table = 'iniciosesion_materia_alumnos'

    def __str__(self):
        return f"Materia ID: {self.materia.nombre} - Alumno ID: {self.user.get_full_name}"

```

Fig. 17 Clase con la que se creó la tabla *Materia_Alumnos*.

Tabla ArchivoCSV

```

class ArchivoCSV(models.Model):
    id = models.AutoField(primary_key=True)
    nombre_archivo = models.CharField(max_length=255)
    contenido_csv = models.TextField()
    fecha_subida = models.DateTimeField(auto_now_add=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='archivosCSV') #No limito las opciones
    porque todos los usuarios pueden usar el reconocimiento de emociones
    materia = models.ForeignKey(Materia, on_delete=models.CASCADE, related_name='archivosCSV')
    managed = False #Evita que Django intente crearla de nuevo

    def __str__(self):
        return self.nombre_archivo

```

Fig. 18 Clase con la que se creó la tabla *ArchivoCSV*.

3.2.5 Funcionamiento en conjunto para el reconocimiento facial y de emociones

En esta sección se explica cómo se combinó el funcionamiento de los métodos de reconocimiento facial y reconocimiento de emociones para poder utilizar ambas herramientas al mismo tiempo y se muestra un ejemplo del funcionamiento del sistema.

3.2.5.1 Algoritmo que combina el reconocimiento facial y de emociones

Los algoritmos de reconocimiento facial y reconocimiento de emociones son dos algoritmos diferentes, ambos reciben como entrada una imagen, pero la procesan de forma diferente. Para lograr que ambos algoritmos trabajen en conjunto se deben seguir los siguientes pasos:

1. Cargar los vectores de características de la materia correspondiente.
2. Obtener la imagen que envía el servidor.
3. Procesar la imagen, esto se logra convirtiéndola a un arreglo numpy y redimensionándola.
4. Procesar la imagen para detectar los rostros, las ubicaciones y las predicciones.
5. Iterar sobre cada cara detectada con las ubicaciones obtenidas en el paso 3.
6. Comparar el vector del rostro actual con los vectores de los usuarios cargados en el paso 1. Si algún vector coincide se guarda el id y el nombre para rellenar los arreglos que se enviarán al servidor.
7. Verificar que existen predicciones disponibles, si existen se guarda la predicción correspondiente y su porcentaje, si no existen predicciones disponibles se guarda una etiqueta de “Error en la predicción”.
8. Se guarda la hora de la predicción
9. Se crea un registro con los datos, y se guarda en el arreglo que se enviará al servidor.
10. Repetir del paso 5 hasta el 9 hasta que ya no haya más rostros.
11. Se envía el arreglo al servidor para mostrar los resultados.
- 12.

3.2.5.2 Ejemplo del funcionamiento del reconocimiento facial y de emociones

A continuación, se mostrarán algunas imágenes del funcionamiento del reconocimiento facial y de emociones, donde se reconocerá a una persona por su nombre y su emoción:

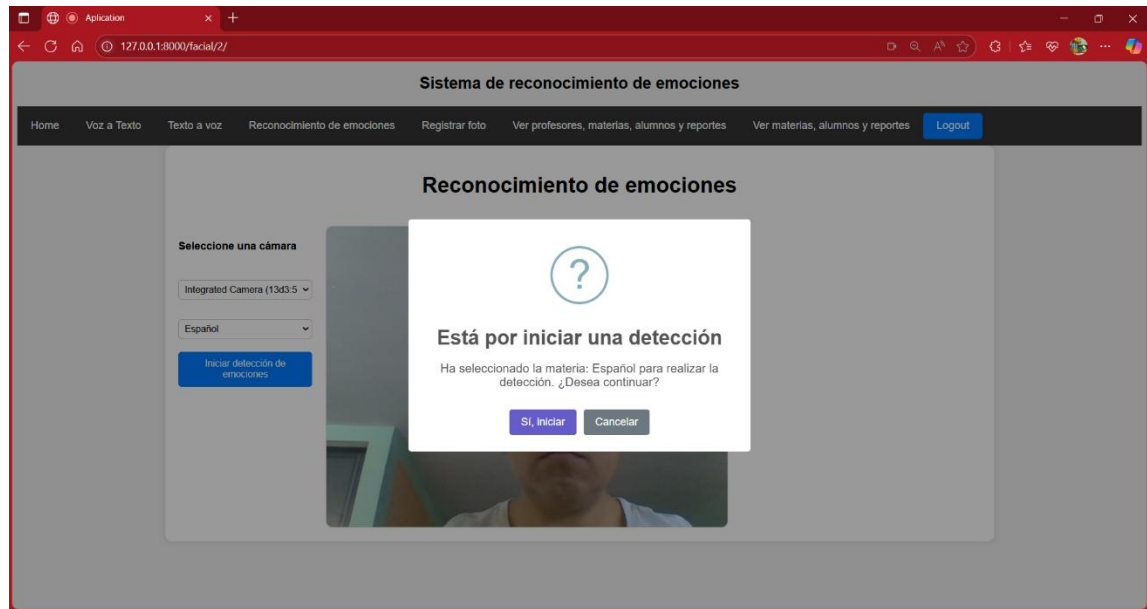


Fig. 19 Al presionar el botón iniciar detección se muestra un cuadro de confirmación.

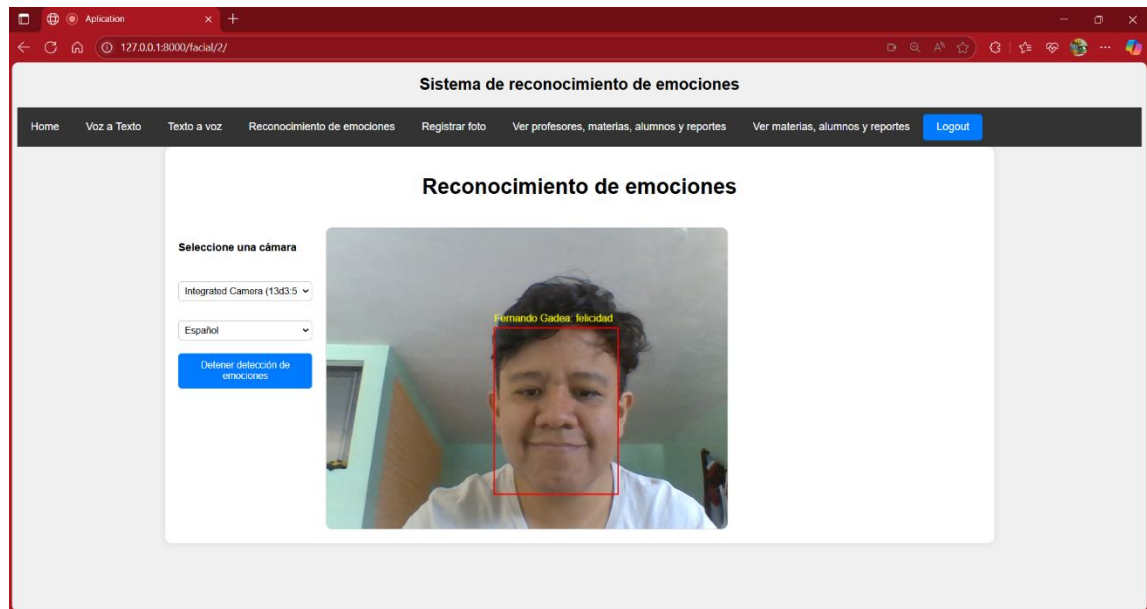


Fig. 20 Ejemplo de detección Facial y de emociones, emoción: Felicidad.

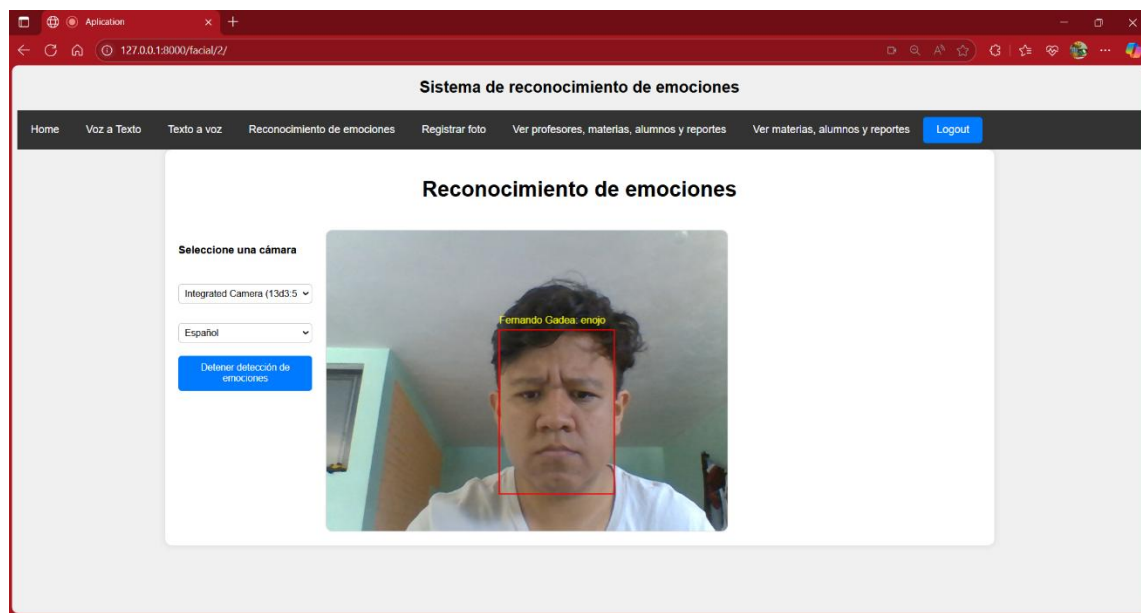


Fig. 21 Ejemplo de reconocimiento facial y de emociones, emoción: Enojo.

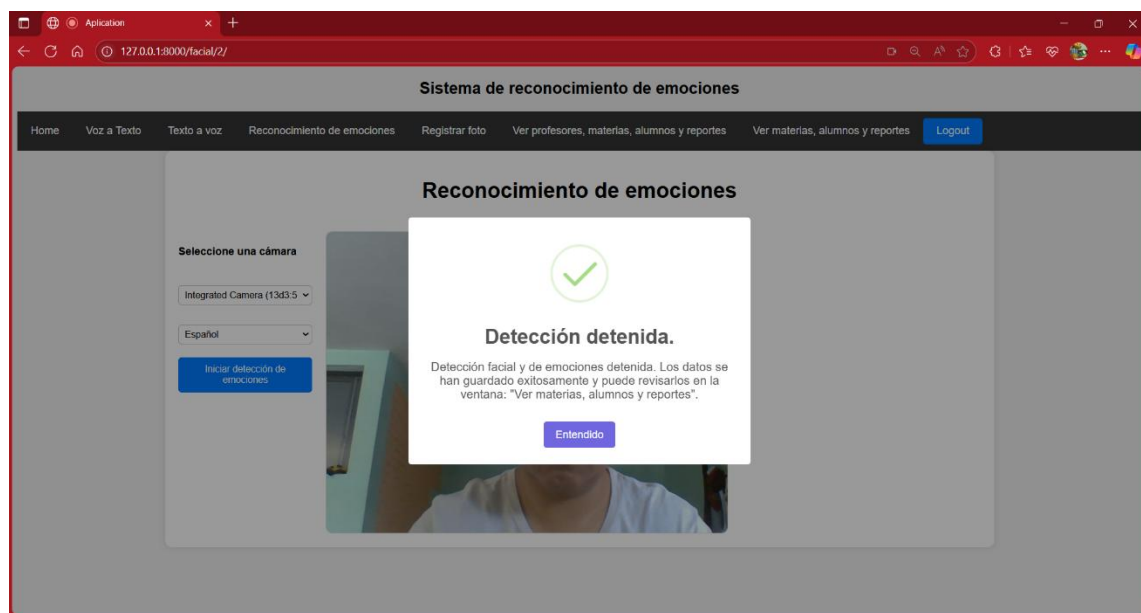


Fig. 22 Al presionar el botón detener detección se muestra un mensaje de aviso.

3.2.6 Diseño de la visualización de gráficas de barras

En esta sección se explica la solución a la que se llegó para poder obtener gráficas de barras legibles.

Se decidió implementar un cuadro para poder seleccionar el intervalo de tiempo que mostraría cada barra, ya que había dos posibles casos:

- a) Los datos obtenidos son muy pocos
- b) Los datos obtenidos son muchos

Y para cada caso no se puede utilizar el mismo intervalo de tiempo ya que, si son pocos datos y se agrupan esto podría resultar en pérdida de información, mientras que si son muchos datos y se muestran todos completos las etiquetas de tiempo se pueden amontonar al punto de ser ilegibles o las barras se distorsionan haciendo que la gráfica sea ilegible.

Los intervalos que se optaron por incluir son:

- a) 9 segundos: Ideal para sesiones de menos de 5 minutos.
- b) 30 segundos: Ideal para sesiones de 5 minutos a 20 minutos.
- c) 1 minuto: Ideal para sesiones de 20 minutos a 40 minutos.
- d) 2 minutos: Ideal para sesiones de 40 minutos a 80 minutos.
- e) 5 minutos: Ideal para sesiones de más de 80 minutos.

La forma de calcular la emoción por cada intervalo es sumando su porcentaje y eligiendo la emoción con mayor porcentaje durante ese intervalo, ya que el porcentaje representa la posibilidad de que la emoción sea la correcta entonces entre más alto porcentaje más seguro está el sistema.

3.2.7 Desarrollo del sistema de texto a voz

En esta sección se explica cómo se realizó el algoritmo para poder escuchar el texto que un usuario escriba.

Cómo se explicó anteriormente, para realizar el algoritmo que pueda convertir el texto escrito a un audio para su reproducción es necesario utilizar la librería **gTTS** (Google Text-To-Speech).

El primer paso para realizar el sistema de texto a voz es importar la librería **gTTS** con la que se usará el modulo **gTTS** para configurar el audio que se creará, los parámetros del módulo son:

- Texto: Es el texto para leer.
- Lenguaje: Es el lenguaje en el que se leerá el texto.
- Slow: Define si el texto será leído de forma lenta o no.

El valor del parámetro 'texto' se obtendrá a través de una solicitud **POST** cuando el usuario presione el botón en el **FrontEnd** para escuchar el texto, el lenguaje a leer el texto será español, y el valor de **slow** será '**False**' porque se requiere que el texto se escuche a velocidad normal. Esta configuración contiene el audio que se guarda en una variable llamada '**speech**' que se utilizará a lo largo del algoritmo.

El siguiente paso es crear un objeto de bytes en memoria para que funcione como un archivo donde se guardará el audio. Utilizando el módulo **write_to_fp** de la variable '**speech**' se guarda el audio contenido en esta variable en el objeto creado previamente y este audio se devuelve al **FrontEnd**, donde se reproduce automáticamente. El audio que se escucha es el texto que escribió el usuario.

Cada vez que el usuario presione el botón para escuchar el audio, el audio anterior se elimina y se crea uno nuevo, esto para hacer más eficaz el uso de memoria.

3.2.8 Desarrollo del sistema de voz a texto

En esta sección se explica cómo se realizó el algoritmo de reconocimiento de voz, para que el usuario pudiera hablar y el sistema escribiera lo que escuche.

Cómo se explicó anteriormente, para realizar el algoritmo que pueda convertir el audio de entrada del micrófono a texto, es necesario utilizar las librerías **Wave**, **pydub**, **speech_recognition**, en conjunto para poder analizar el audio y escribirlo.

El primer paso para crear el algoritmo es comenzar a grabar del lado del frontend un audio, cuando el usuario haya terminado de grabar el audio presionará un botón para leer lo que se dijo. Este audio se recibirá a través de una solicitud POST y se comprobará que esta solicitud contenga un archivo con la clave 'audio'. Si esto se cumple, el archivo se extraerá y se creará un archivo temporal en el servidor donde se escribirá el contenido del archivo de audio.

El siguiente paso es comprobar que el archivo temporal sea de tipo .WAV, si no lo es, se convertirá a este tipo de archivo, esto facilita la futura manipulación del archivo con **speech_recognition**. Después de convertir el archivo a .WAV o si el archivo ya tenía ese formato, se guardará el archivo temporal en el almacenamiento de Django.

Una vez se tiene el archivo en el formato deseado se hace el procesamiento de este. Se crea un objeto '**recognizer**' de la librería **speech_recognition**. Después se utiliza la librería Wave para abrir el archivo y asegurarse que sea .WAV. Dentro del bloque que abrió el archivo con Wave se vuelve a abrir el archivo esta vez con el objeto '**recognizer**', se ajusta el reconocimiento al ruido del ambiente con **recognizer.adjust_for_ambient_noise**, se graba el contenido de audio completo con **recognizer.record**, se envía el audio a la API de Google para su transcripción, especificando que el idioma es español, finalmente y si el audio se pudo transcribir se devuelve una respuesta de tipo JSON con el texto transcrito.

El texto que se devolvió en formato JSON se escribe en un cuadro de texto del **FrontEnd** para que el usuario pueda leerlo.

Capítulo 4 Pruebas del modelo y resultados

En esta sección del documento se mostrarán las pruebas realizadas para comprobar que el sistema funciona correctamente.

4.1 Pruebas del reconocimiento facial y de emociones

Existen una gran cantidad de factores que pueden afectar la ejecución de las pruebas, entre ellos se encuentran las condiciones del entorno, así como la cámara que se ha utilizada durante el desarrollo de estas, se realizaron dos pruebas con diferentes condiciones, las cuales fueron:

Prueba 1: Dentro del aula

- Luz de un foco fluorescente dentro del aula y luz natural entrando por cuatro ventanas grandes por 2 lados del aula , esta luz al ser natural variaba.
- Cámara integrada del dispositivo **IdeaPad L340 Gaming** HD 720p con micrófono de doble matriz.

Prueba 2: Al aire libre

- Luz natural al aire libre la cual variaba, siendo luz directa o sombra.
- Cámara integrada del dispositivo **IdeaPad L340 Gaming** HD 720p con micrófono de doble matriz.

Además, se ha seleccionado una ventana para el reconocimiento facial y de emociones de 1.5 segundos, esto quiere decir que cada 1.5 segundos se tomará un frame de la cámara para su procesamiento, este intervalo de tiempo se ha elegido ya que según una investigación realizada en 2020, un intervalo entre 1 y 2 segundos es óptimo para el reconocimiento de emociones, ya que permiten una mayor estabilidad para la extracción de características sin dejar pasar mucho tiempo para perder alguna emoción o muy poco para no ver ningún cambio [39]. Una vez

definidas las diferentes condiciones se mostrarán los resultados obtenidos en cada una de ellas.

Caso 1: Dentro del aula

En este caso se realizó la prueba con 20 alumnos de la preparatoria Benito Juárez. En este caso la prueba se realizó mientras su profesora se encontraba dando una clase de matemáticas, a cada alumno se le tomó su respectiva fotografía para obtener sus datos faciales y registrarlos en el sistema, una vez hecho el registro de los datos la prueba comenzó. A continuación, se muestran los gráficos obtenidos, la gráfica de barras mostrará la emoción predominante de cada minuto, mientras que la de pastel mostrará los porcentajes totales, también una pequeña encuesta que se realizó al terminar la detección para poder tener una idea de la tasa de aciertos del sistema:

4.1 Pruebas del reconocimiento facial y de emociones

86

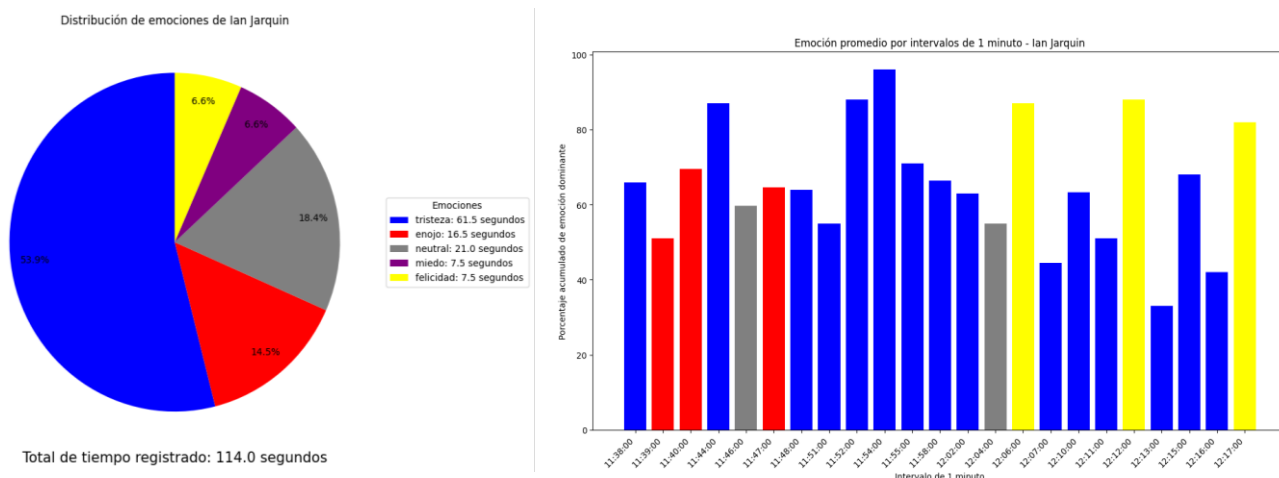


Fig. 23 Emociones de Ian Jarquin en el transcurso de 114 segundos.

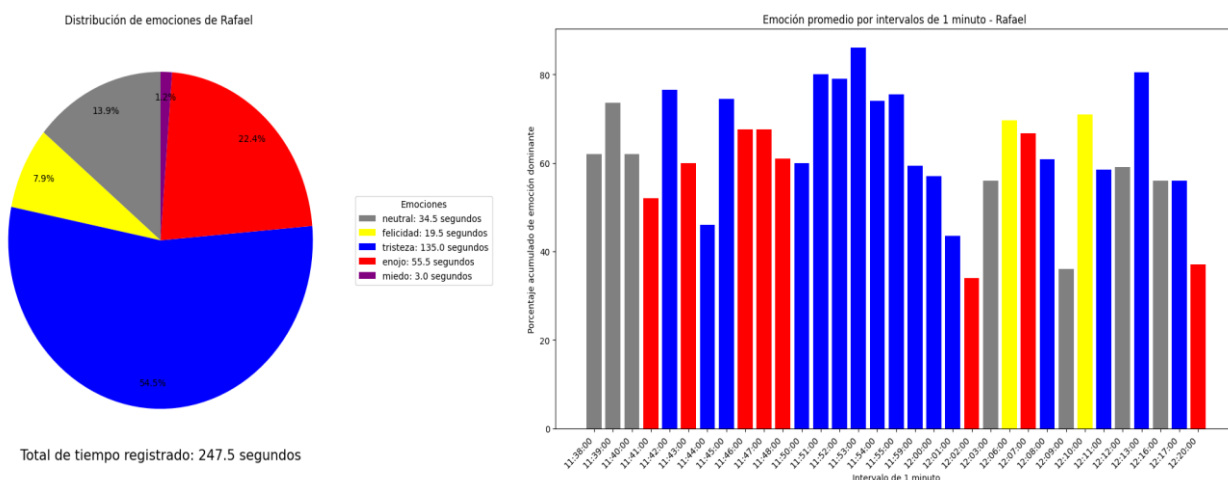


Fig. 24 Emociones de Rafael en el transcurso de 247.5 segundos.

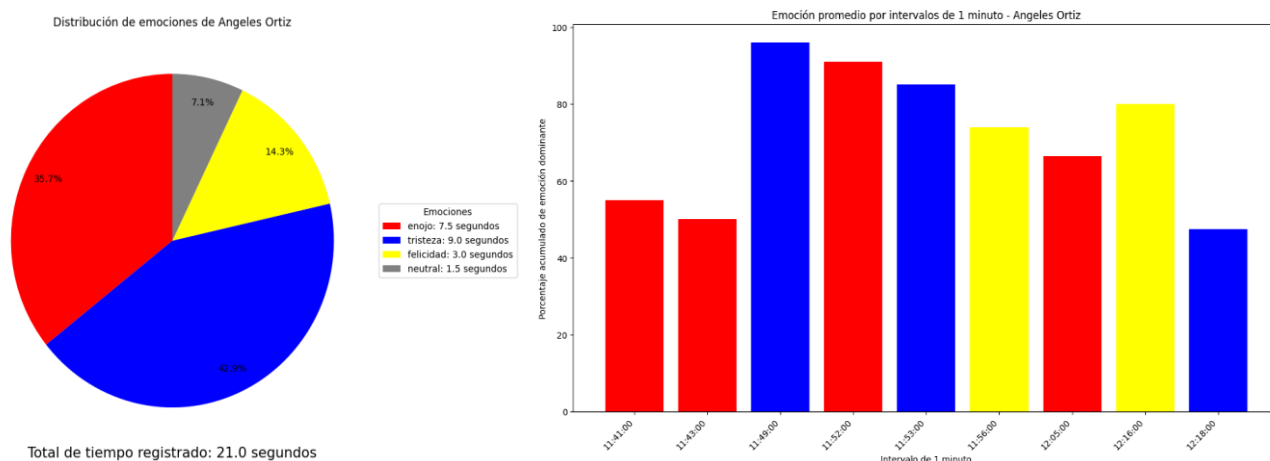
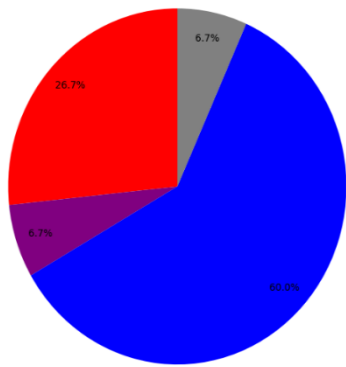


Fig. 25 Emociones de Ángeles Ortiz en el transcurso de 21 segundos.

Distribución de emociones de Dania García Hernández



Total de tiempo registrado: 22.5 segundos

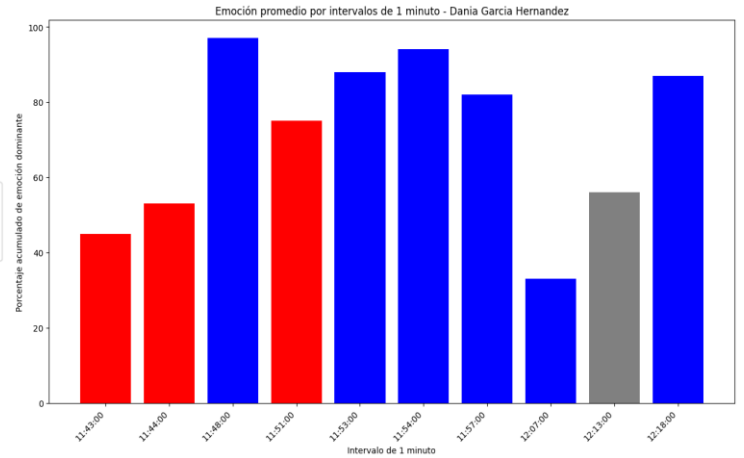
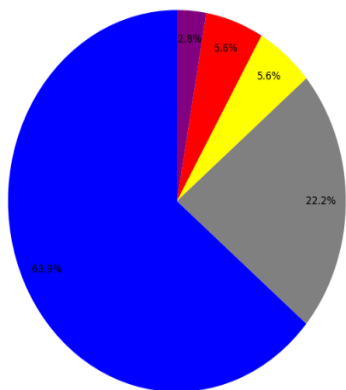


Fig. 26 Emociones de Dania García Hernández en el transcurso de 22.5 segundos.

Distribución de emociones de Karla Aguilar



Total de tiempo registrado: 54.0 segundos

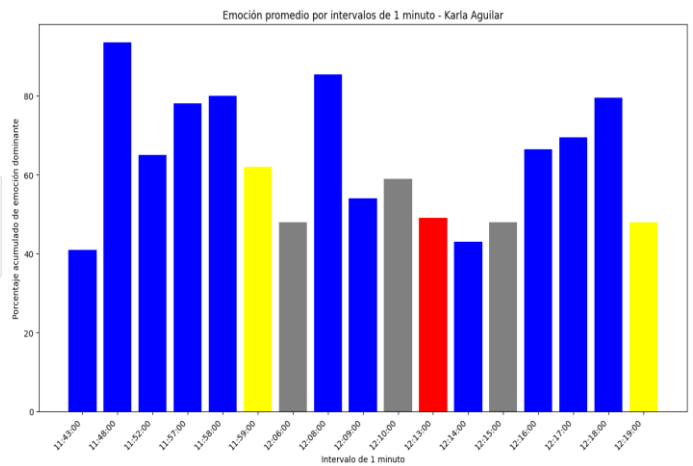
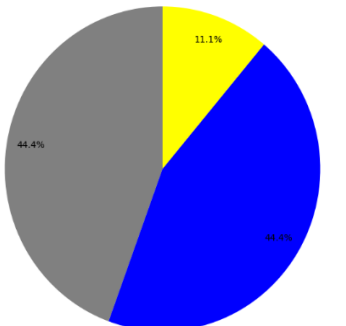


Fig. 27 Emociones de Karla Aguilar en el transcurso de 54 segundos.

Distribución de emociones de Gianelli



Total de tiempo registrado: 13.5 segundos

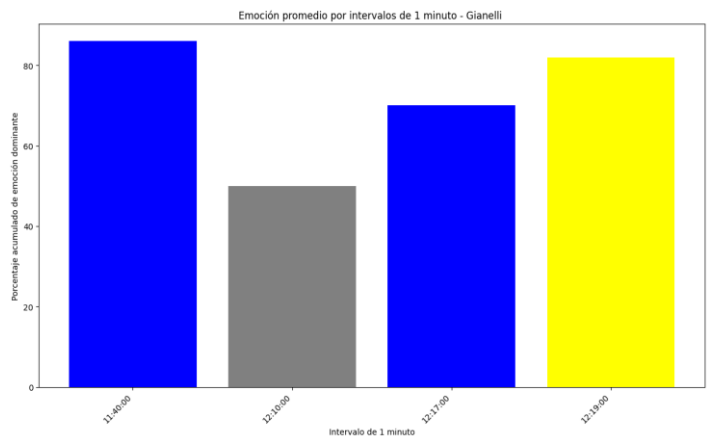


Fig. 28 Emociones de Gianelli en el transcurso de 13.5 segundos.

4.1 Pruebas del reconocimiento facial y de emociones

88

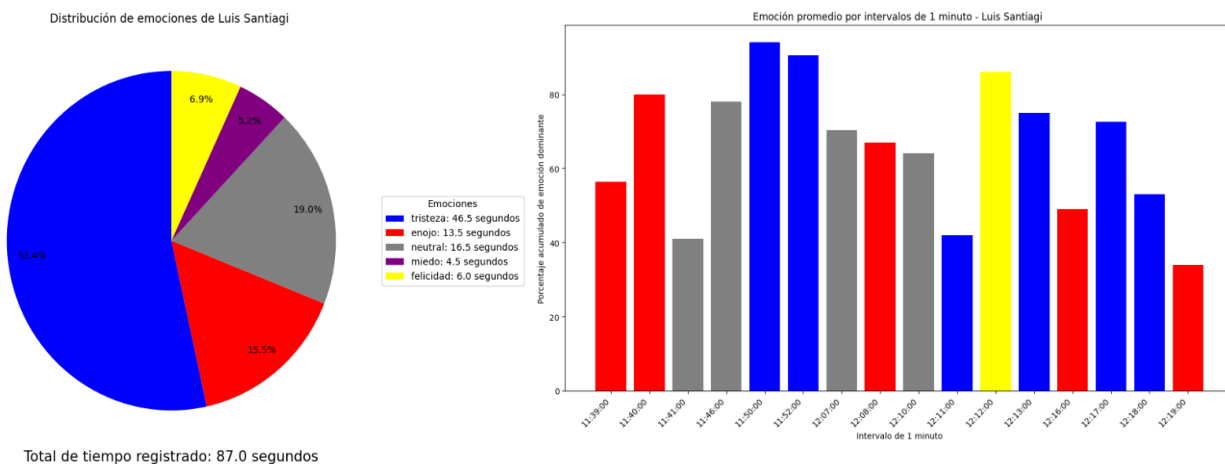


Fig. 29 Emociones de Luis Santiagi en el transcurso de 87 segundos.

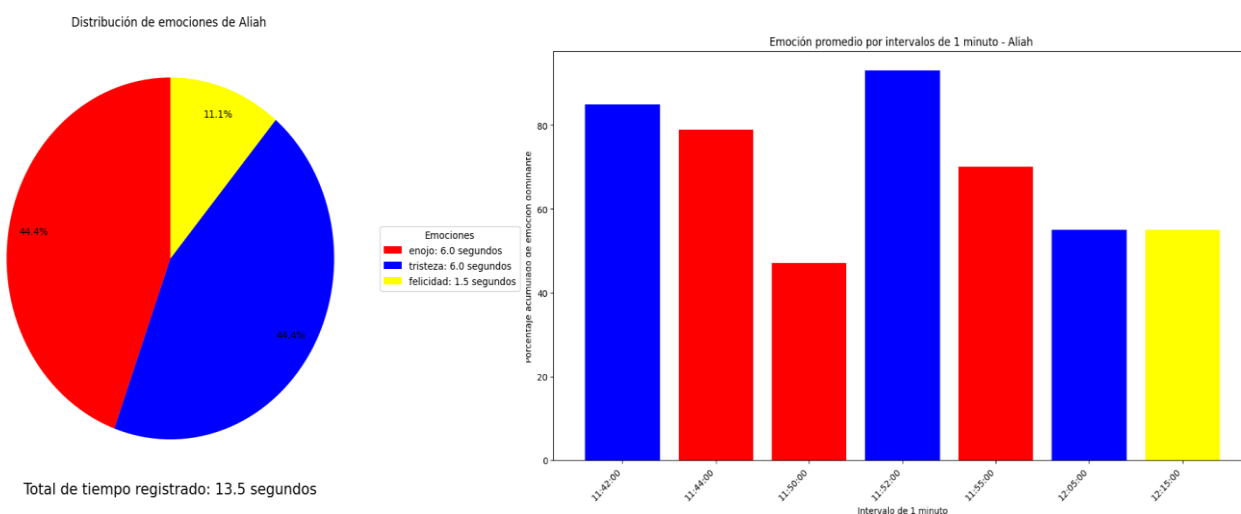


Fig. 30 Emociones de Aliah en el transcurso de 13.5 segundos.

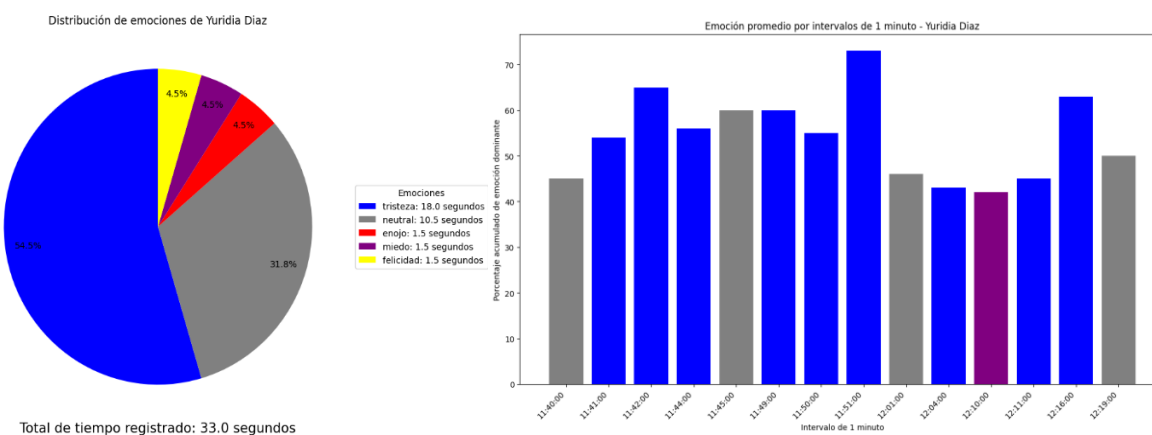


Fig. 31 Emociones de Yuridia Díaz en el transcurso de 33 segundos.

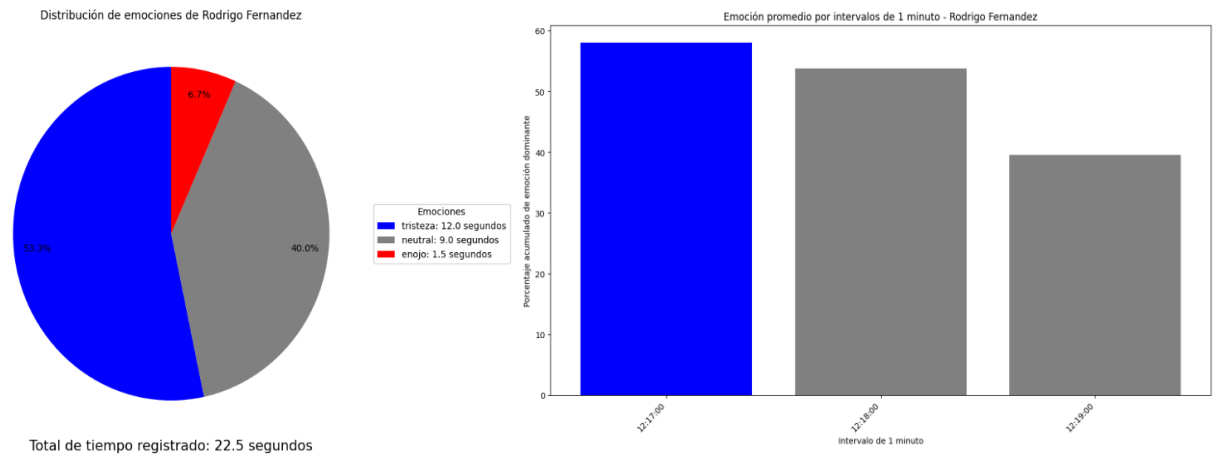


Fig. 32 Emociones de Rodrigo Fernández en el transcurso de 22.5 segundos.

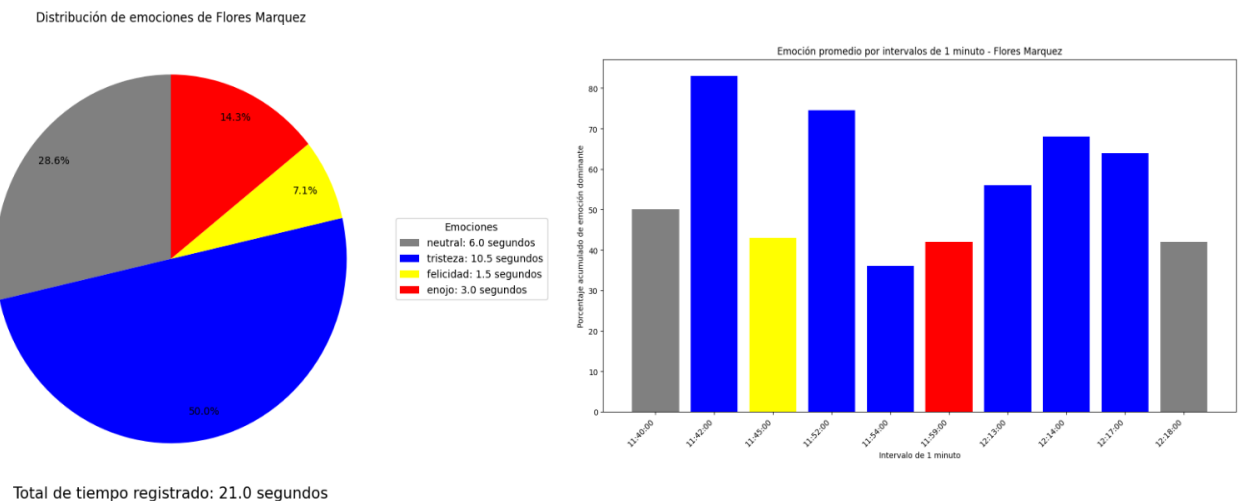


Fig. 33 Emociones de Flores Márquez en el transcurso de 21 segundos.

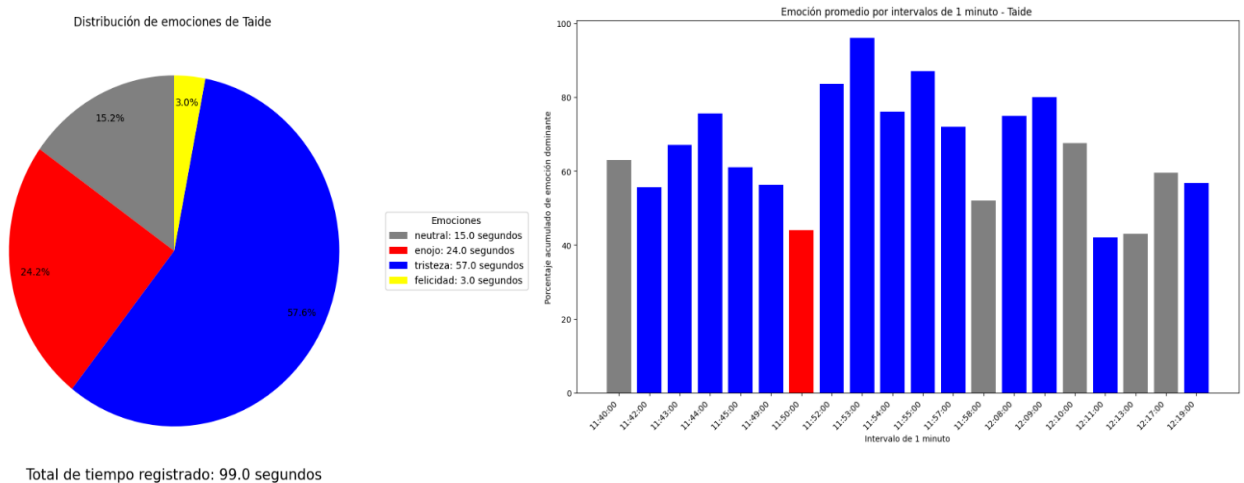


Fig. 34 Emociones de Taide en el transcurso de 99 segundos.

4.1 Pruebas del reconocimiento facial y de emociones

90

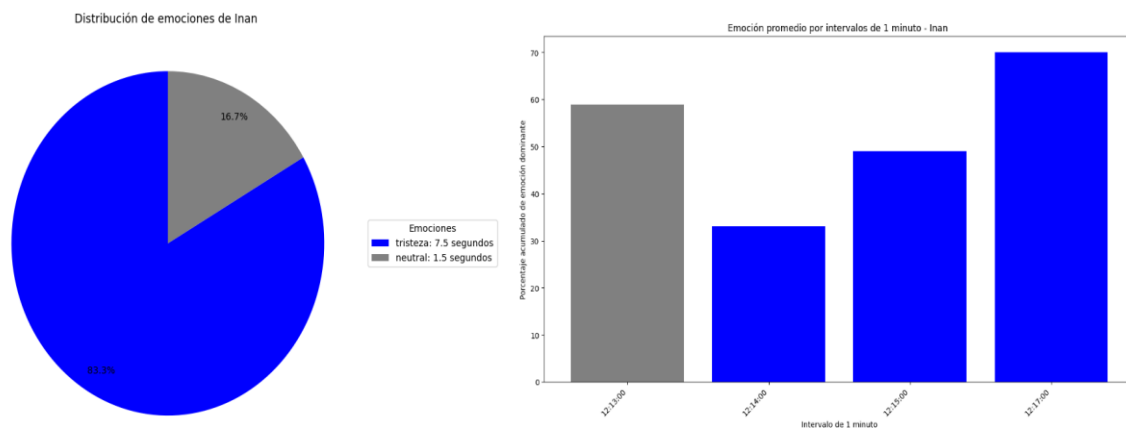


Fig. 35 Emociones de Inan en el transcurso de 9 segundos.

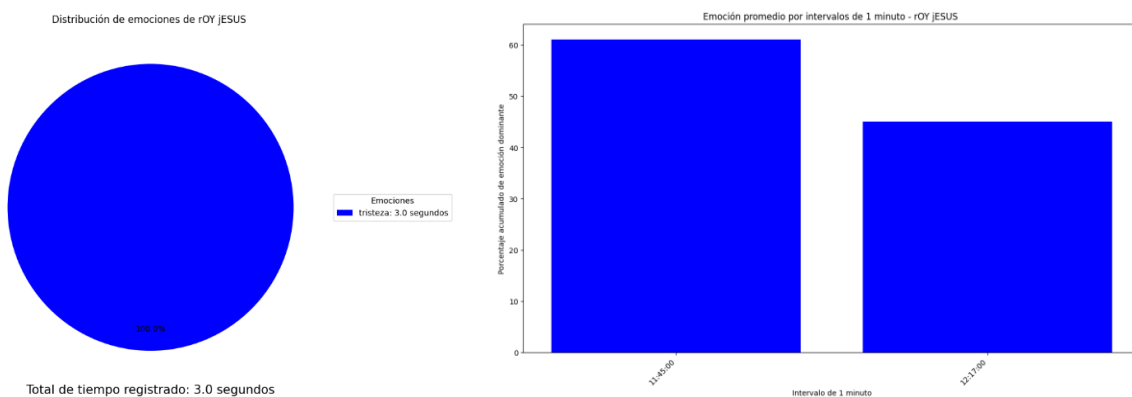


Fig. 36 Emociones de Roy Jesús en el transcurso de 3 segundos.

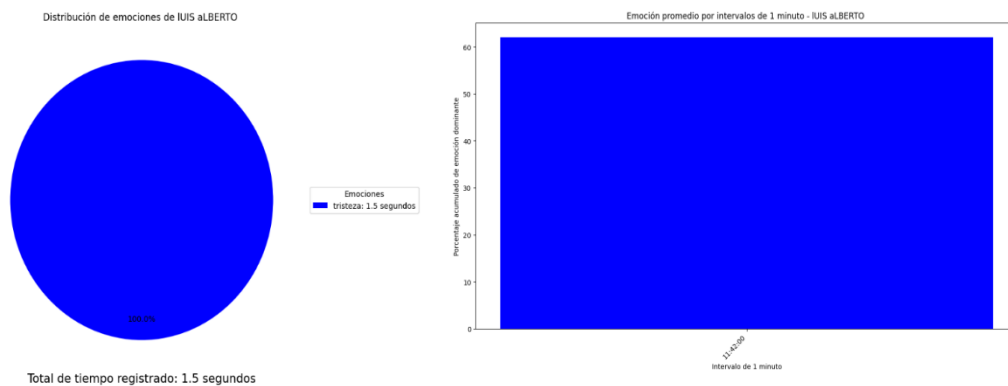


Fig. 37 Emociones de Luis Alberto en el transcurso de 1.5 segundos.

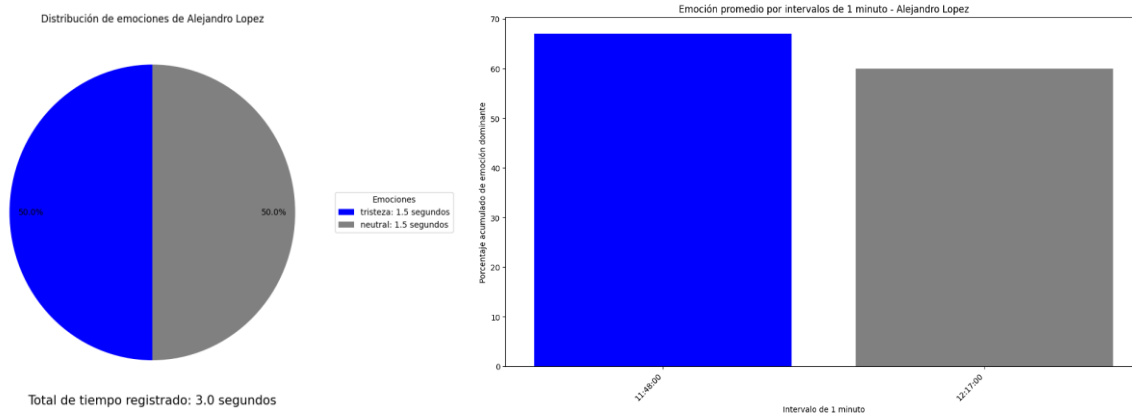


Fig. 38 Emociones de Alejandro López en el transcurso de 3 segundos.

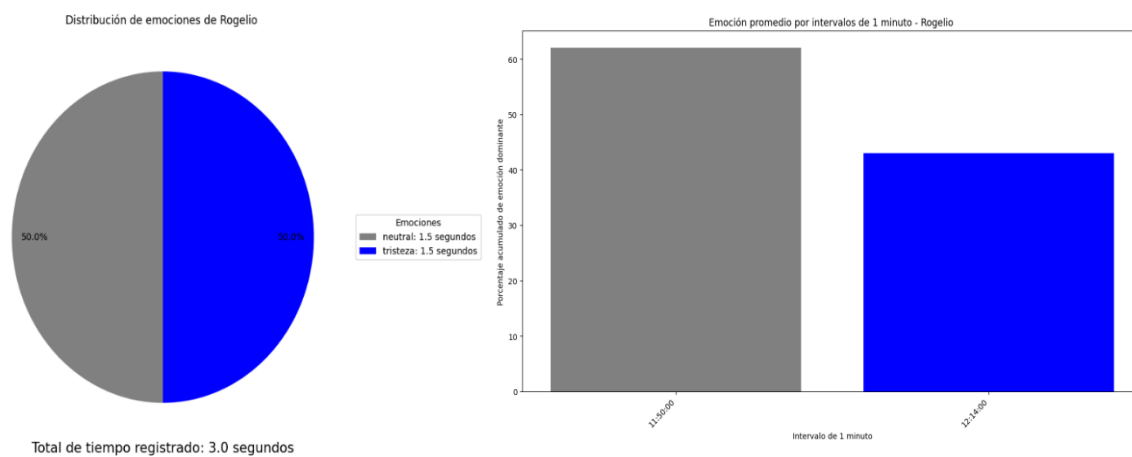


Fig. 39 Emociones de Rogelio en el transcurso de 3 segundos.

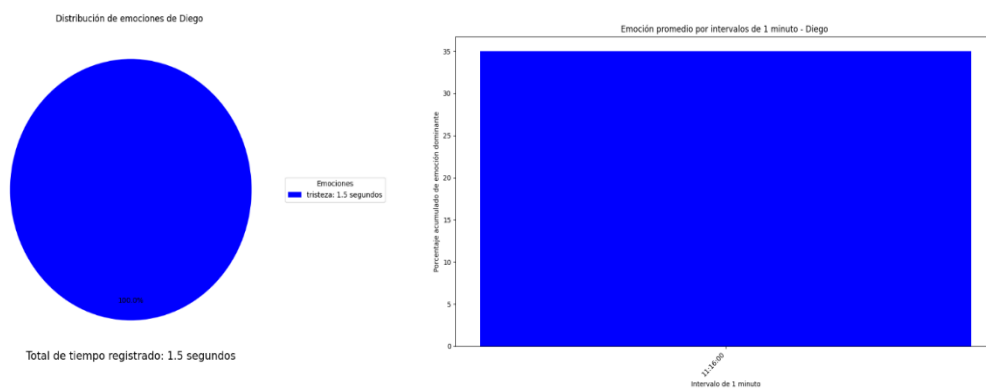


Fig. 40 Emociones de Diego en el transcurso de 1.5 segundos.

4.1 Pruebas del reconocimiento facial y de emociones

92

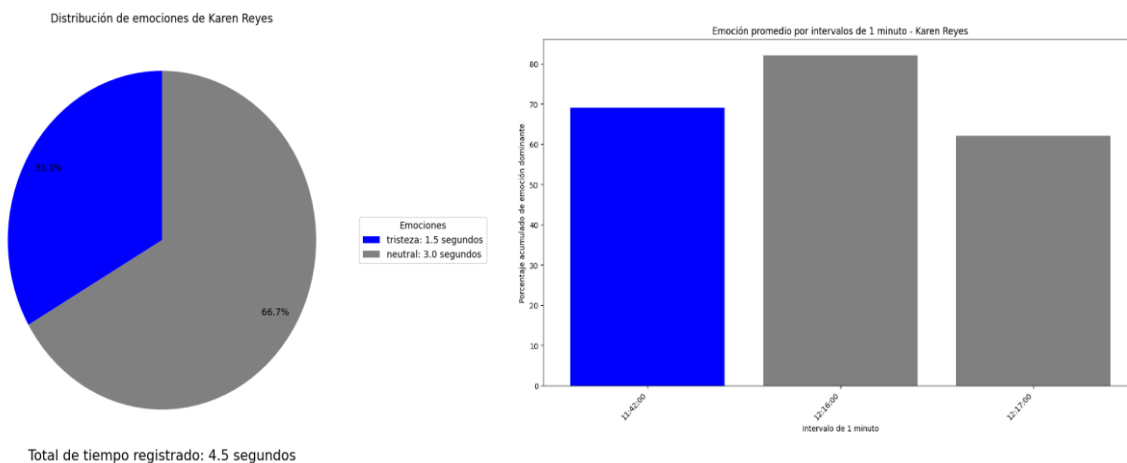


Fig. 41 Emociones de Karen Reyes en el transcurso de 4.5 segundos.



Obtenido de la encuesta



Obtenido del sistema



Ambos coinciden

Alumno	Emoción registrada en la encuesta						
	Alegría	Tristeza	Enojo	Miedo	Sorpresa	Neutral	Asco
Ian Jarquin							
Rafael							
Ángeles Ortiz							
Dania García							

Karla Aguilar							
Gianelli							
Luis Santiagi							
Aliah							
Yuridia							
Flores Márquez							
Taide							
Rodrigo							
Luis Alberto							
Roy Jesús							
Inan							
Diego Gael							
Alejandro López							
Rogelio							
Karen Reyes							

Tabla 6 Tabla resumen: Encuestas realizadas al terminar la detección

Una vez terminada la detección se crean dos gráficos, una gráfica de pastel donde se puede ver la emoción predominante durante el tiempo total de la detección y una gráfica de barras donde se puede ver la emoción que se registró y la hora en que se registró la emoción. Utilizando la gráfica de pastel es posible detectar a los usuarios que tienen un registro mayor de emociones negativas, mientras que utilizando la gráfica de barras es posible saber el momento en el que estas emociones se presentaron.

Se puede observar que el sistema puede diferenciar entre usuarios, ya que se crearon las gráficas para cada uno de ellos, con su respectivo nombre y emociones. Si este no fuera el caso no se habría registrado nada, ya que solo se guardan los registros de los usuarios conocidos, las emociones de los usuarios desconocidos se pueden ver al momento de la ejecución, pero no se guarda ninguna

gráfica ya que al pertenecer a diferentes usuarios que no están registrados esta información no serviría de nada.

En este caso la prueba duró aproximadamente 1 hora, pero al ser un aula grande y la cámara al no tener la mejor resolución, muchos alumnos no fueron captados por el sistema durante la sesión completa, especialmente los alumnos al fondo del aula, mientras que los alumnos sentados al frente sí fueron captados. Debido a que solo se captaban los alumnos que tenían la cámara frente a ellos se decidió que la computadora se comenzaría a mover para que tomara imágenes de la mayoría de los alumnos posibles, aunque por cortos periodos de tiempo, interrumpiendo lo menos posible la clase de la profesora o la concentración de los alumnos cuando estaban realizando el ejercicio del día.

Cómo se puede ver en las gráficas de algunos alumnos, como Gianeli Fig. 28, Flores Márquez Fig. 34, Rodrigo Fernández Fig. 32, Luis Alberto Fig. 35, Roy Jesús Fig. 37, Inan Fig. 36, Diego Fig. 38, Alejandro López Fig. 40, Rogelio Fig. 39 y Karen Reyes Fig. 41, hubo muy pocas detecciones ya que la cámara se estuvo moviendo para poder obtener por lo menos 1 detección por alumno, mientras que en los demás alumnos hubo detecciones más completas ya que estos se encontraban sentados al frente o en las orillas y era más fácil tomar detecciones de ellos sin interrumpir su clase.

Hubo 19 registros de diferentes alumnos, mientras que un alumno no se obtuvo registro, esto puede ser porque estuvo cubriendo su rostro mientras se movía la cámara, porque estuvo agachado todo el tiempo haciendo la actividad o porque no se pasó frente a él ya que estaba al fondo en medio del aula. Hay varios motivos por los que esto puede ocurrir, pero el sistema tuvo un 95% de éxito en reconocer a los diferentes alumnos que hubo.

Al analizar las encuestas se obtuvieron los siguientes resultados:

El porcentaje de aciertos del sistema es de un 61.11%. Esto refleja un rendimiento moderado del sistema, pues, aunque logra coincidir en una parte significativa de los casos, aun presenta un margen considerable de fallas.

Es importante considerar que las emociones no siempre se expresan de manera visible en el rostro, especialmente dentro del aula pues los alumnos pueden mantener expresiones neutras o disimular sus emociones. Además, factores ajenos al sistema como la postura, las actividades que están realizando, y las características de cada alumno pueden influir en el funcionamiento de este.

El porcentaje de lecturas erróneas, es decir, el sistema las detectó, pero los alumnos no las marcaron en su encuesta es de 25%. En este apartado la emoción predominante es la tristeza, esto puede deberse a que como los alumnos estaban realizando su trabajo de clase estaban agachados en su libreta o viendo un poco al frente con la mirada hacia abajo, lo que asemeja una mirada caída que puede estar acompañada de la falta de expresividad, el sistema interpretará como señal de tristeza, cuando en realidad los alumnos solo están concentrados. Esto muestra una limitación del sistema de reconocimiento facial y de emociones para distinguir entre emociones reales y expresiones derivadas de posturas o actividades específicas, destacando la importancia de considerar el contexto o la hora en que esta emoción está presente, por ejemplo, analizando las Fig. 31, Fig. 33, Fig. 35 y Fig. 37, podemos observar que la emoción tristeza está presente en el mismo rango de tiempo (11:50 a 12:00) para estos alumnos lo que indica que es más probable que sea porque estaban realizando la actividad de clase y no porque realmente estaban tristes.

El porcentaje de lecturas no detectadas, es decir, los alumnos las marcaron en la encuesta, pero el sistema no las detectó es de 13.88%. En este apartado la emoción predominante no detectada fue la sorpresa, esto sugiere que los alumnos sintieron sorpresa (posiblemente por la propuesta de utilizar un sistema para evaluar sus emociones), pero esta emoción no se reflejó en sus expresiones faciales. Esto indica una limitación en el sistema para reconocer emociones que, aunque estén presentes a nivel interno, pueden no mostrarse de manera externa.

En conjunto, estos resultados sugieren que el sistema puede resultar útil como una herramienta de apoyo para detectar las emociones en el aula, pero no debería ser utilizado de manera aislada para evaluar los estados emocionales

individuales sin considerar otros métodos o fuentes de información, como encuestas u observaciones.

Caso 2: Al aire libre

En este caso se realizaron varias pruebas con distintos usuarios en el evento Explora y descubre: aventuras científicas, llevado a cabo el viernes 19 de Julio del 2024. A este evento atendieron una gran variedad de niños cuyos padres estuvieron de acuerdo en que se les tomara una foto para el registro de su rostro y el uso del sistema. En este caso las pruebas realizadas fueron cortas ya que había muchas exposiciones. A continuación, se muestran los gráficos obtenidos además de una pequeña encuesta que se les realizó al terminar la detección:

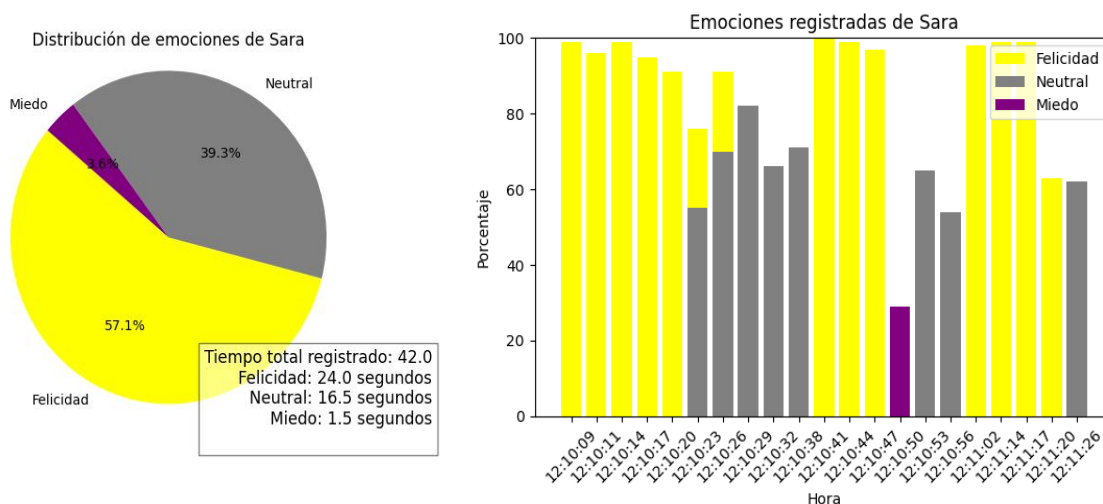


Fig. 42 Emociones de Sara en el transcurso de 42 segundos.

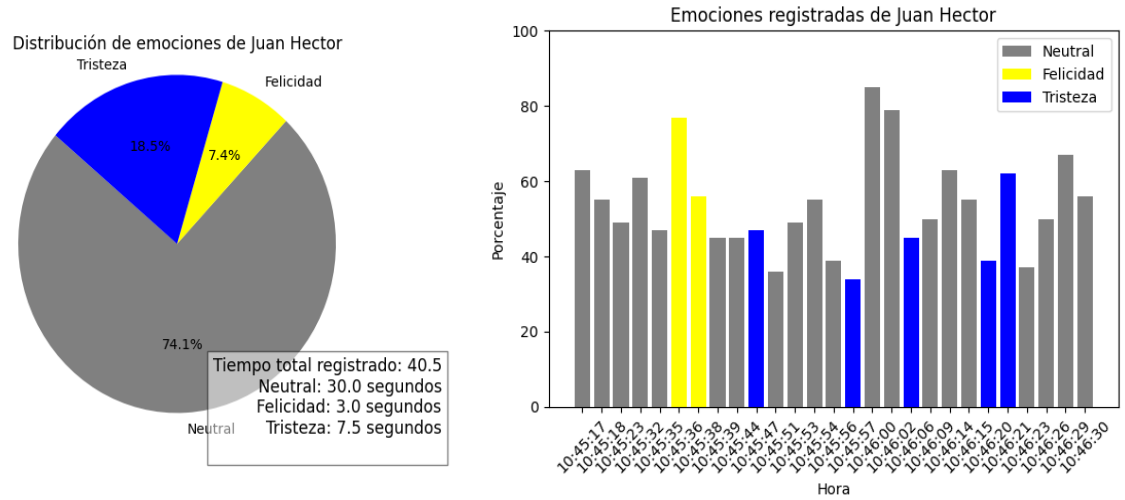


Fig. 43 Emociones de Juan Héctor en el transcurso de 40 segundos.

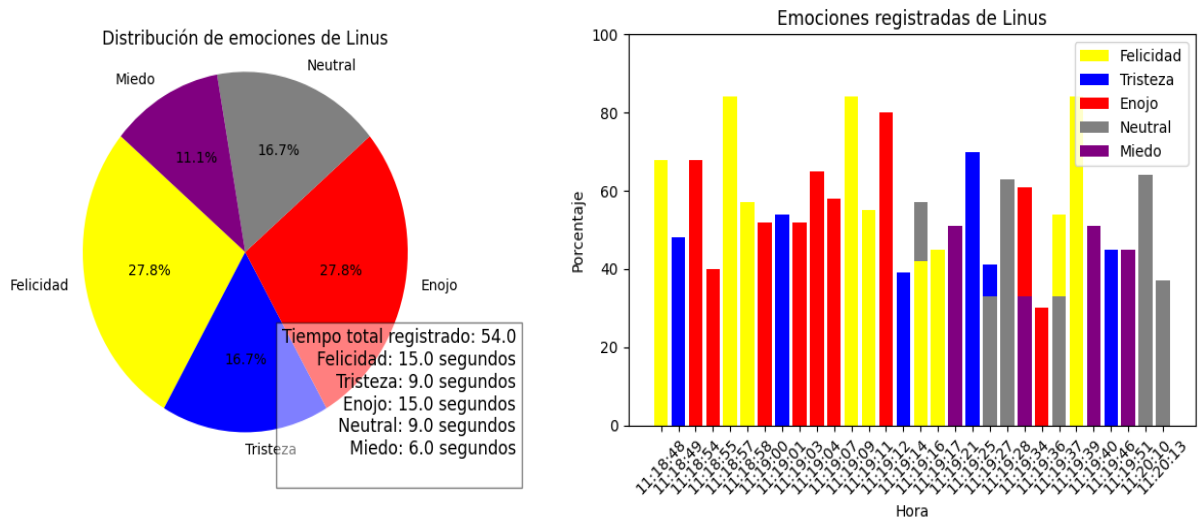


Fig. 44 Emociones de Linus en el transcurso de 54 segundos.

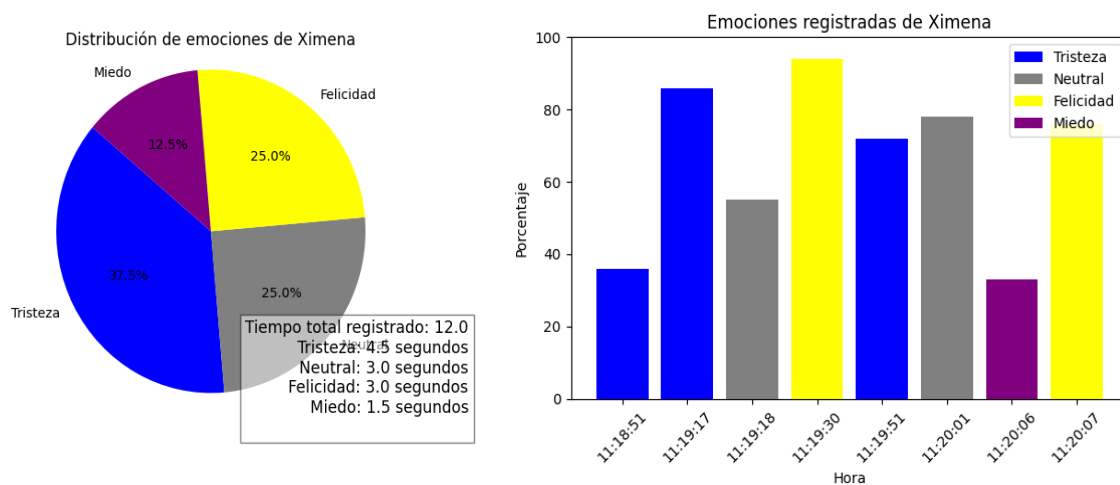


Fig. 45 Emociones de Ximena en el transcurso de 12 segundos.

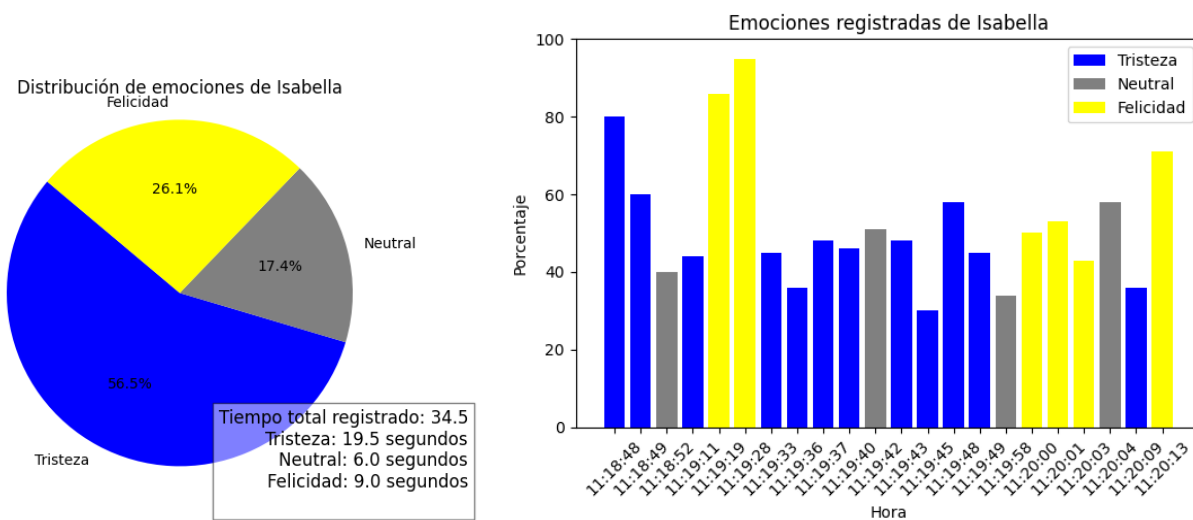


Fig. 46 Emociones de Isabella en el transcurso de 34 segundos.

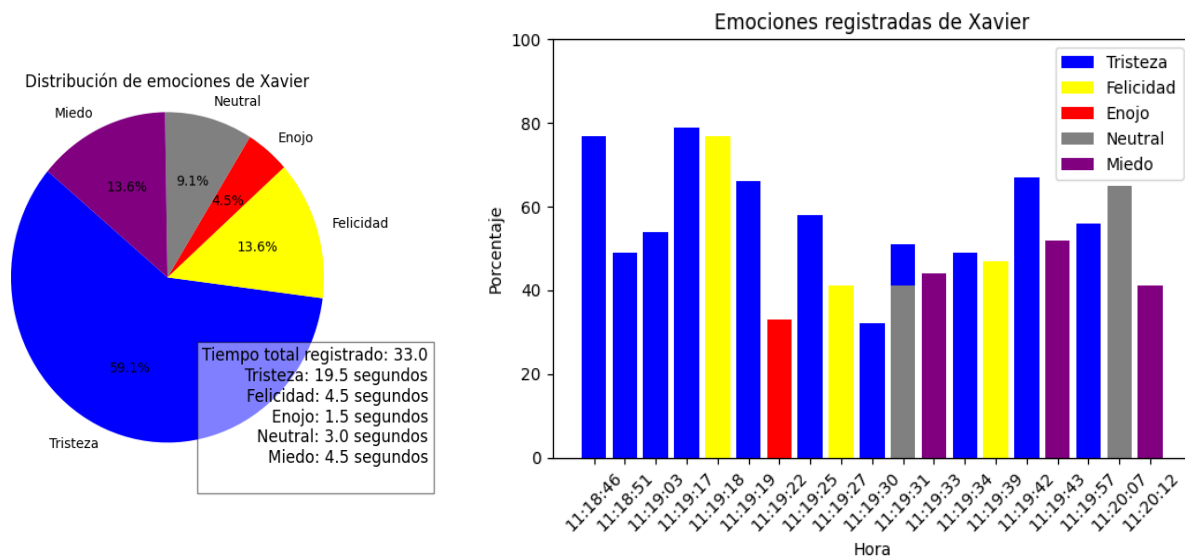


Fig. 47 Emociones de Xavier en el transcurso de 33 segundos.

	Obtenido de la encuesta
	Obtenido del sistema
	Ambos coinciden

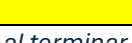
Usuario	Emoción registrada en la encuesta						
	Alegría	Tristeza	Enojo	Miedo	Sorpresa	Neutral	Asco
Sara							
Juan Héctor							
Linus							
Ximena							
Isabella							
Xavier							

Tabla 7 Tabla resumen: Tabla de las encuestas realizadas al terminar la detección.

En este caso también se crearon dos gráficos, la gráfica de pastel y la gráfica de barras.

Aunque el sistema también cuenta con algunas fallas en el reconocimiento facial, como se puede ver en la gráfica de barras de Linus Fig. 45, en la hora 11:19:14, 11:19:25, 11:19:28 y 11:19:36, existen dos gráficas a la misma hora, esto significa que el sistema reconoció a Linus dos veces y guardó el registro en el mismo campo de tiempo. En el momento en el que se tomó la imagen para su procesamiento había 2 o más usuarios, y que el sistema reconozca a ambos como la misma persona puede ser por diferentes factores externos, tales como, las condiciones del clima, en esos momentos pudo haber dado el sol directamente a la imagen, haberse nublado, etc., lo que afecta el funcionamiento del sistema. Además, es posible saber que fueron dos reconocimientos diferentes ya que son dos emociones diferentes y el sistema solo puede dar como resultado una emoción

por rostro en cada detección. Esto también sucede en el caso de Sara y Xavier, aunque pasó en menos ocasiones.

En el caso de Linus, Ximena, Isabella y Xavier, sus detecciones se realizaron al mismo tiempo, es decir, los cuatro usuarios estaban dentro de la visualización de la cámara, lo que quiere decir que el sistema es capaz de reconocer a varios usuarios a la vez, además de sus emociones y crear gráficos distintos para cada uno de ellos.

Los resultados de la comparación entre los gráficos obtenidos del sistema y la tabla obtenida de las encuestas son los siguientes:

El porcentaje de aciertos del sistema es de un 64.51%. Esto refleja un rendimiento moderado del sistema, pues, aunque logra coincidir en una parte significativa de los casos, aun presenta un margen considerable de fallas.

El porcentaje de lecturas erróneas, es decir, el sistema las detectó, pero los alumnos no las marcaron en su encuesta es de 22.50%.

El porcentaje de lecturas no detectadas, es decir, los alumnos las marcaron en la encuesta, pero el sistema no las detectó es de 12.90%.

Estos porcentajes son similares a los obtenidos en el caso 1, esto quiere decir que el sistema se comporta de manera similar siempre.

4.2 Ejemplo de un reporte del sistema de reconocimiento facial y de emociones

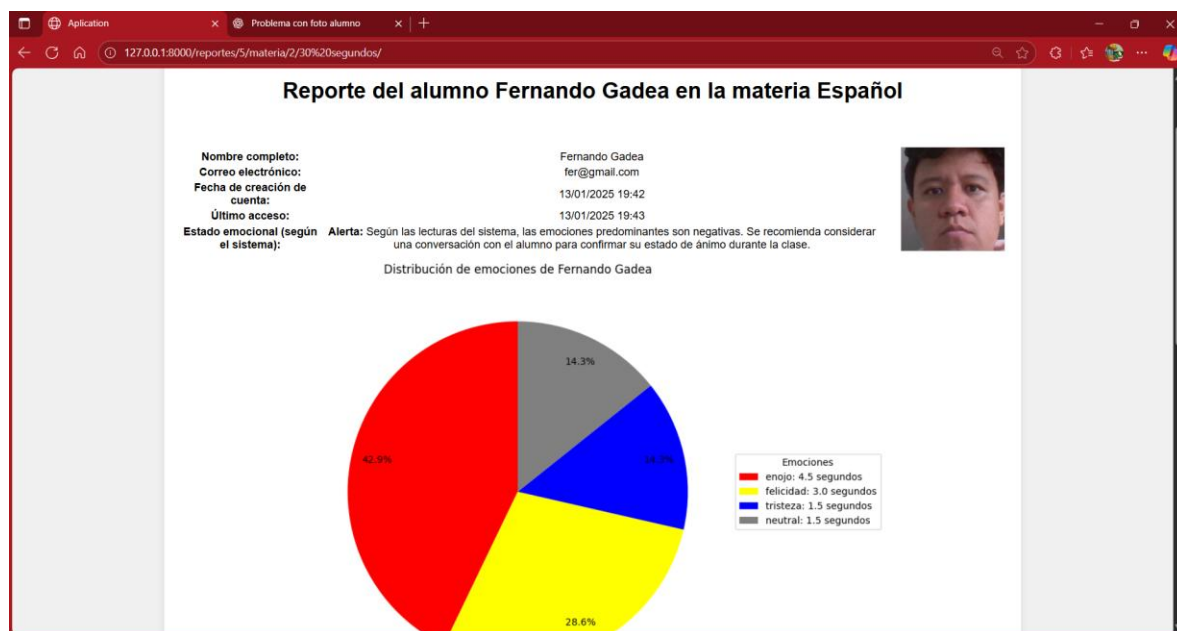


Fig. 48 Reporte de donde se muestran los datos, el estado emocional y las gráficas de un alumno.

Con los datos obtenidos del reconocimiento facial y de emociones crean gráficas, cómo las que se mostraron en el punto anterior, pero además estos datos se analizan para poder comprender la emoción predominante del alumno durante las sesiones y se clasifican los alumnos en tres categorías diferentes:

- **Alerta:** El alumno presenta en su mayoría emociones negativas, las cuales son enojo, asco, miedo y tristeza, y se le recomienda al profesor que hable con el alumno para confirmar si se siente mal durante las sesiones de clase o que le preste más atención durante las mismas.
- **Estable:** El alumno presenta en su mayoría emociones neutrales, por lo cual el profesor no debe preocuparse por el alumno, la atención que recibe es suficiente y debería concentrarse primero en los alumnos en alerta. Una vez que no haya alumnos en alerta podría intentar incluir o hacer participar más a los alumnos estables.

- Alegre: El alumno presenta en su mayoría emociones positivas, las cuales son alegría y sorpresa, por lo cual el profesor está haciendo un buen trabajo con estos alumnos y debería de continuar con el mismo.

Se espera que el profesor haga caso a estas recomendaciones para poder mejorar el aprendizaje de los alumnos.

Las clasificaciones se obtienen al sumar los tiempos presentados de cada emoción durante la sesión.

4.3 Pruebas del sistema de texto a voz

Para realizar las pruebas del sistema de texto a voz solo se requiere visitar la página, entrar a la sección “Voz a texto”, escribir un texto en la sección “Escribe tu texto aquí” y una vez que el texto está escrito presionar “Escuchar”, el sistema tardará más o menos tiempo dependiendo de la longitud del texto y al terminar de procesarlo se reproducirá automáticamente. A continuación, se muestra un ejemplo real del funcionamiento:

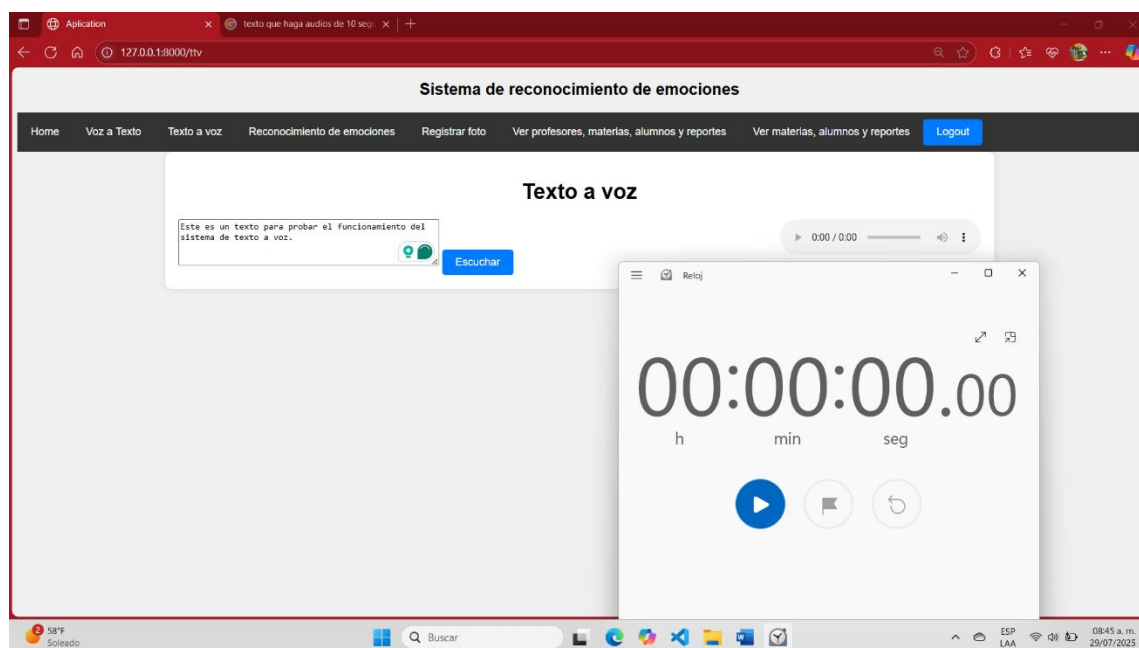


Fig. 49 Se inserta el texto en el recuadro de la izquierda y se presiona el botón "Escuchar".

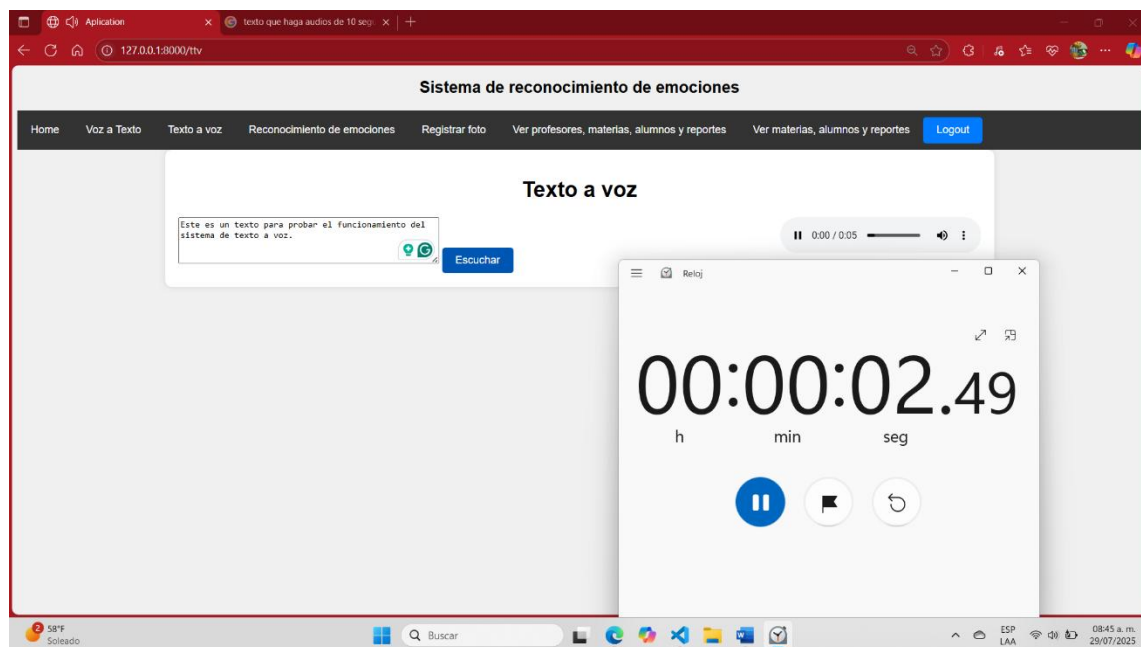


Fig. 50 Se procesa el texto y se comienza a reproducir el audio.

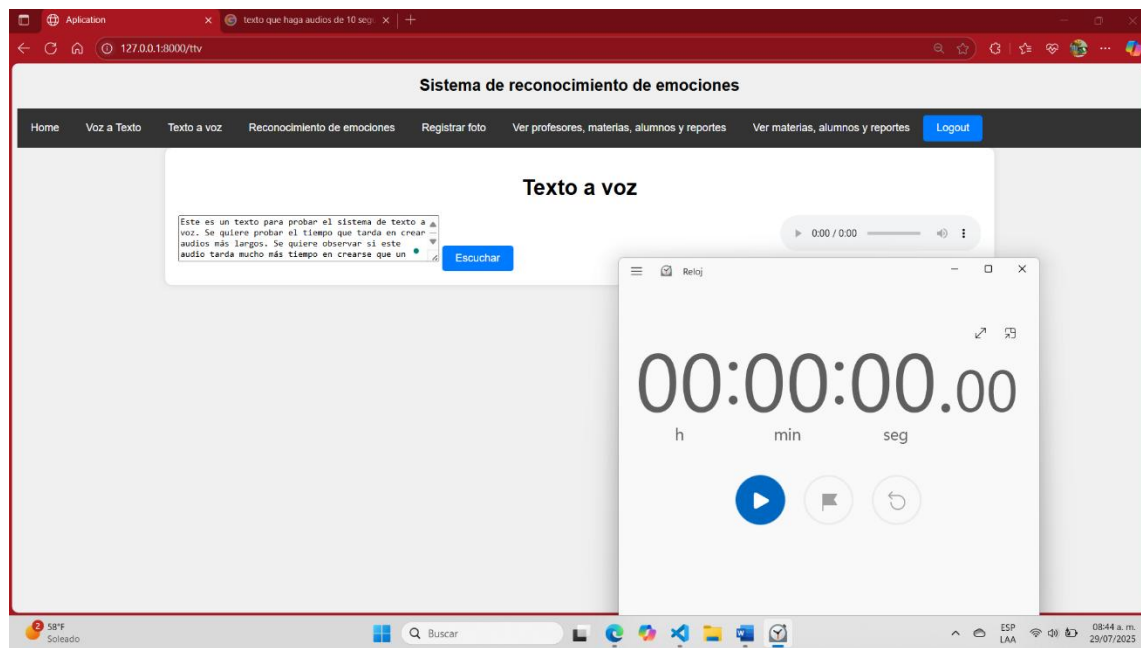


Fig. 51 Texto más largo para probar la optimización del sistema.

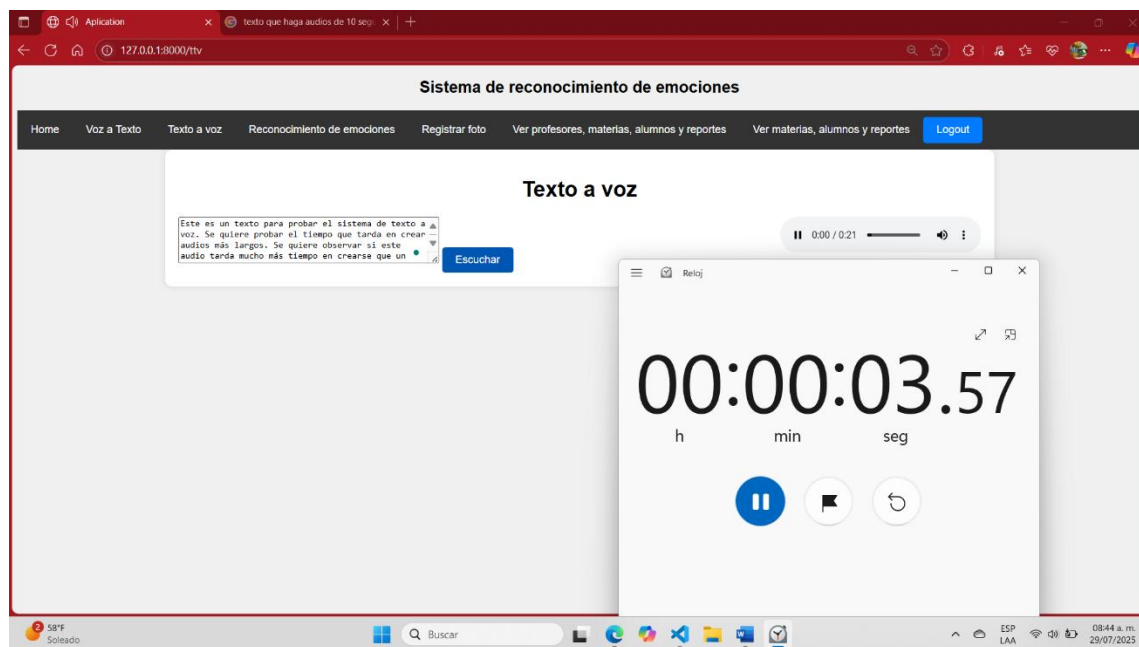


Fig. 52 Audio creado y tiempo que se tardó en crearse.

Como se puede observar en la Fig. 49 se introdujo el texto “Este es un audio para probar el funcionamiento del sistema de texto a voz” para poder probar el funcionamiento, después se presionó el botón “Escuchar” y en la Fig. 50 se puede observar que se creó un audio y este audio contenía el texto que se ingresó en el recuadro de la Fig. 49, por lo que se puede comprobar que el sistema funciona correctamente. En este caso el audio creado fue de 5 segundos y se creó en 2.49 segundos.

En la Fig. 51 se pone a prueba el tiempo que tarda en crearse un audio que sea mucho más largo que el de la Fig. 49, y en la Fig. 52 se puede observar que el audio creado fue de 21 segundos de duración y el tiempo que tardó en procesarse fue de 3.57 segundos. Es importante notar que la diferencia de tiempos entre un audio de 5 segundos y un audio de 21 segundos es pequeña, lo que significa que el sistema está bien optimizado para su uso.

4.4 Pruebas del sistema de voz a texto

Existen factores que podrían hacer que el sistema de voz a texto falle un poco, estos pueden ser el ruido de fondo, la calidad del micrófono o la claridad con la que el usuario esté hablando, se realizaron dos pruebas con diferentes condiciones, las cuales fueron:

Prueba 1: Con música de fondo y el usuario estaba sentado frente a la computadora.

Prueba 2: No había música de fondo y el usuario estaba sentado frente a la computadora.

Prueba 3: El usuario se alejó 1 metro del micrófono y no había música de fondo.

Prueba 4: El usuario se alejó 1 metro del micrófono y había música de fondo.

En todas estas pruebas la frase que se dirá cambiará un poco para cada prueba.

Caso 1: Música de fondo y el usuario frente a la computadora.

Se está realizando una prueba del sistema de voz a texto con diferentes condiciones, para esta prueba estoy sentado frente a la computadora y tengo música de fondo.

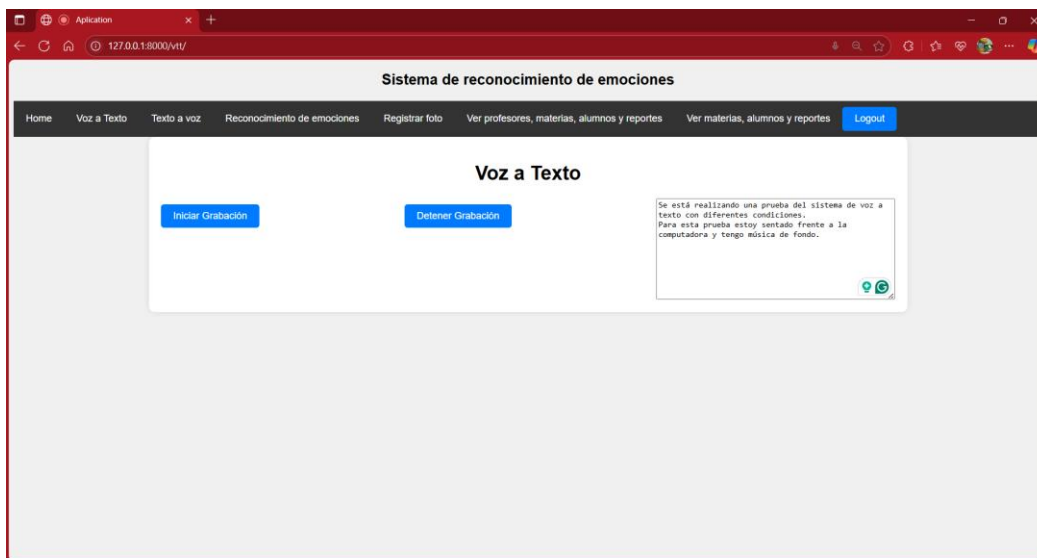


Fig. 53 Resultado del Caso 1: Música de fondo y el usuario frente a la computadora.

Caso 2: Sin música de fondo y el usuario frente a la computadora.

Se está realizando una prueba del sistema de voz a texto con diferentes condiciones, para esta prueba estoy sentado frente a la computadora y no tengo música de fondo.

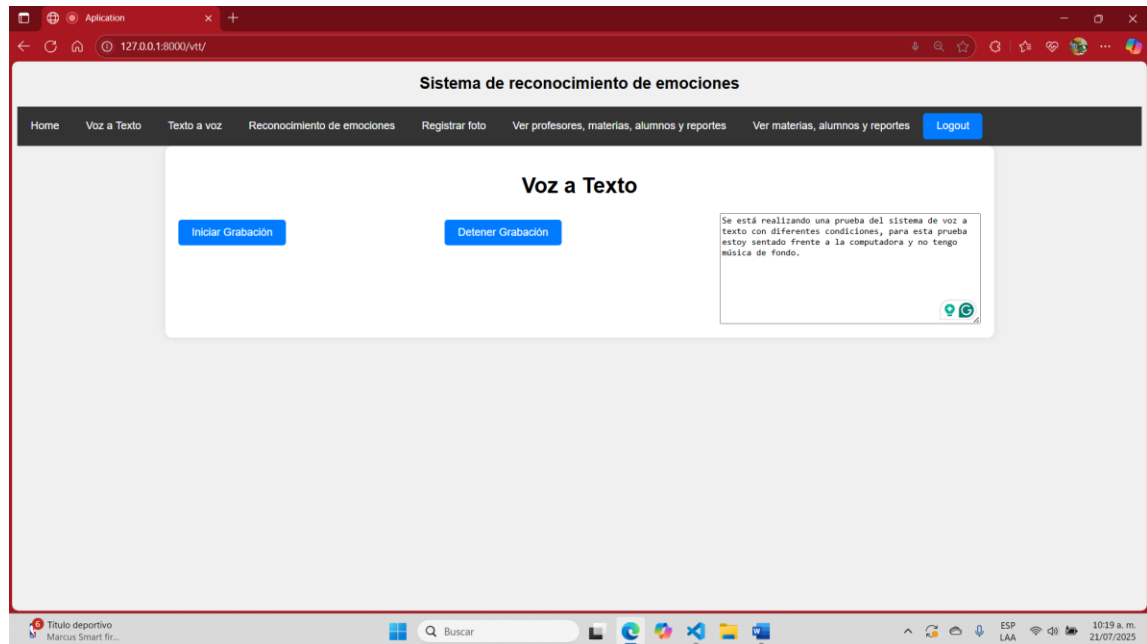


Fig. 54 Resultado del Caso 2: Sin música de fondo y el usuario frente a la computadora.

Caso 3: Sin música de fondo y el usuario a 1 metro de la computadora.

Se está realizando una prueba del sistema de voz a texto con diferentes condiciones, para esta prueba estoy alejado a 1 metro de la computadora y no tengo música de fondo.

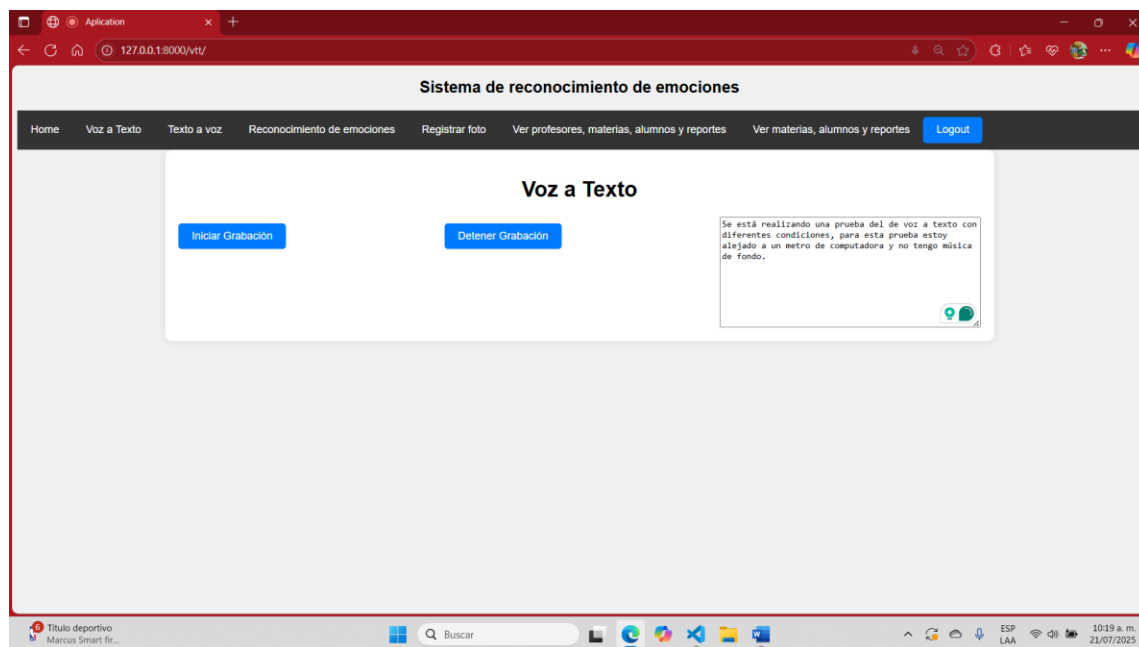


Fig. 55 Resultado del Caso 3: Sin música de fondo y el usuario a 1 metro de la computadora.

Caso 4: Con música de fondo y el usuario a 1 metro de la computadora.

Se está realizando una prueba del sistema de voz a texto con diferentes condiciones, para esta prueba estoy alejado a 1 metro de la computadora y tengo música de fondo.

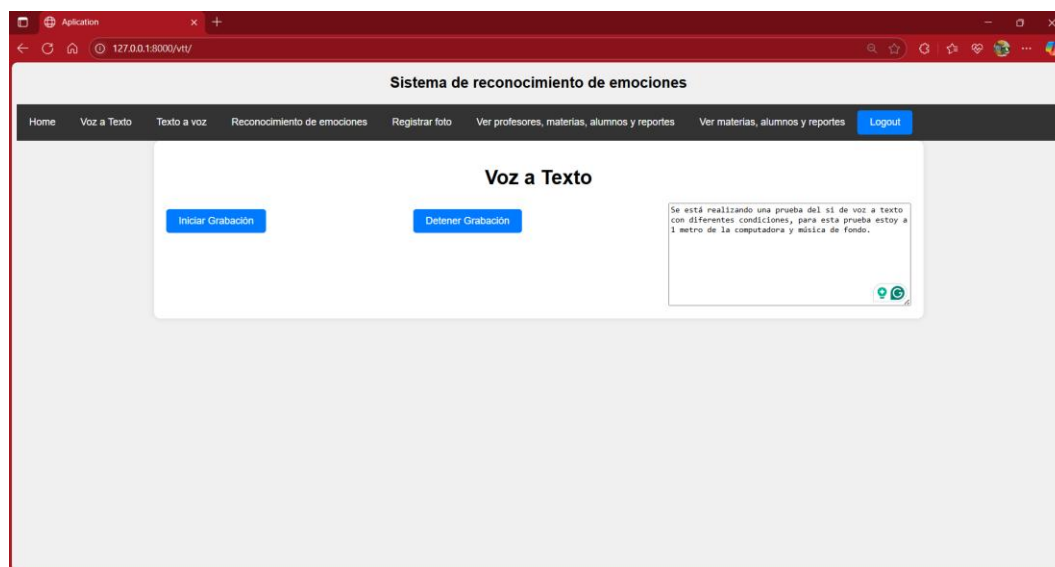


Fig. 56 Resultado del Caso 4: Conn música de fondo y el usuario a 1 metro de la computadora.

Como se puede observar en la Fig. 51 y Fig. 52, el texto obtenido de la grabación de audio del usuario es completamente correcto, aunque en los cuatro casos el sistema tardó un poco en procesar el audio, esto puede deberse a varios factores como los presentados en cada caso o lentitud en el servidor de la librería Google Text To Speech (gTTS). Mientras que en la Fig. 53 y Fig. 54, que son las condiciones menos recomendables para usar este sistema, se puede observar que se perdieron algunas palabras y tardó aún más en procesar este audio. Aunque se hayan perdido algunas palabras el texto aún es entendible, esto puede que no sea el caso para textos más complejos o con más palabras.

Se recomienda usar este sistema estando cerca de la computadora y evitando lo más posible el ruido de fondo, esto para el mejor funcionamiento posible.

Capítulo 5 Conclusiones

5.1 Conclusiones del proyecto

Se diseñó e implementó un sistema de reconocimiento facial y de emociones, además, se desarrolló un sistema que pudiera convertir el texto escrito a voz y otro que pudiera convertir un audio el audio que escuchara a través del micrófono a texto. Se realizaron diversas pruebas a estos sistemas en diferentes entornos, para el reconocimiento facial y de emociones las pruebas se realizaron en un aula y en actividades al aire libre, mientras que para el sistema de voz a texto y de texto a voz las pruebas se realizaron en un cuarto. A partir de los resultados obtenidos de estas actividades, se llegó a las siguientes conclusiones:

5.1.1 Reconocimiento facial y de emociones

- **Desempeño general:** Se obtuvo un porcentaje de aciertos del **61.11%** en las pruebas del sistema, es decir, logró reconocer correctamente las emociones de los estudiantes en algo más de la mitad de los casos en comparación con las emociones autodeclaradas en las encuestas. Este desempeño es aceptable considerando las condiciones reales y no controladas del entorno educativo, aunque revela un margen de mejora significativo.

- **Lecturas erróneas:** Se registró un **25% de lecturas erróneas**, es decir, emociones que el sistema detectó pero que no fueron reportadas por los alumnos en las encuestas. La emoción predominante en estos casos fue la **tristeza**, la cual fue una constante en todas las pruebas realizadas, y se llegó a la conclusión de que al realizar la detección si un usuario se encuentra agachado o mirando hacia abajo con una expresión neutra, el sistema lo interpretará como tristeza debido a los datos que se usaron como entrenamiento, ya que la tristeza normalmente se

reconoce por una expresión caída o mirada hacia abajo. Esto evidencia una limitación del sistema para diferenciar entre emociones reales y expresiones faciales asociadas a actividades específicas o a la postura corporal. En el trabajo a futuro se implementará una red neuronal que pueda diferenciar entre una persona agachada o con la mirada hacia abajo y una persona que está viendo de frente, esto para poder filtrar de mejor manera y evitar la ocurrencia a futuro de esta mala detección.

- **Emociones no detectadas:** El **13.88% de las emociones autodeclaradas** no fueron detectadas por el sistema. La emoción predominante en este grupo fue la **sorpresa**, lo cual puede explicarse porque los alumnos, aunque sorprendidos por el uso del sistema de reconocimiento durante sus clases, no manifestaron expresivamente esta emoción en sus rostros, manteniendo expresiones neutras debido a la concentración en sus actividades. Este hallazgo sugiere que existen emociones que, aunque presentes a nivel interno, no siempre se reflejan de manera visible y son, por tanto, difíciles de identificar solo mediante análisis facial.

- **Reconocimiento facial:** Se logró un reconocimiento facial del **95%** en condiciones de aula, aunque presentó limitaciones en la detección de alumnos ubicados en zonas lejanas o parcialmente cubiertos, debido a las restricciones de la cámara utilizada y las condiciones del entorno.

- **Limitaciones y mejoras futuras:** Los resultados muestran que, si bien el sistema puede servir como herramienta de apoyo para detectar emociones en contextos educativos, **no debería ser utilizado de manera aislada** para evaluar estados emocionales individuales. Es fundamental considerar factores contextuales, como postura, tipo de actividad y expresividad individual, para mejorar la precisión. Además, también es recomendable realizar un análisis profundo de las gráficas de barras de los alumnos para poder identificar patrones de emociones, especialmente el de la tristeza, aunque una repetición en la emoción de alegría puede indicar una buena concentración o aprendizaje, una repetición de enojo puede representar una mala concentración.

- **Otras limitaciones:** En la prueba realizada en el aula se llegó a la conclusión de que las condiciones de luz o la calidad de la cámara influyen de forma significativa en los resultados obtenidos de las detecciones. Condiciones de luz variantes, luz detrás de los alumnos o sombras muy pronunciadas pueden llevar al mal reconocimiento facial, desde confundir un alumno con otro, hasta no reconocerlo. Es importante tener bajo control las condiciones en las que se va a realizar la detección, por esto es recomendable que se realice en un aula aislada donde la luz no sea variable y los rostros de los alumnos no estén oscurecidos por sombras o gorros, también es recomendable tener una cámara de buena calidad para poder ver a los alumnos en la mejor resolución posible y aumentar la cantidad de alumnos reconocidos, ya que si la imagen es de mala calidad no será posible encontrar rostros. También se necesita colocar a los alumnos de forma estratégica para que no se cubran los unos a los otros, que haya el suficiente espacio para poder ver claramente los rostros de cada uno.

Una opción más radical y mucho más complicada, es que cada alumno tenga su propia cámara frente a él y así se mejoran la calidad de las imágenes obtenidas, al mismo tiempo mejorando la fiabilidad del reconocimiento sistema.

5.1.2 Texto a voz

- **Desempeño general:** Se pusieron a prueba algunos textos y el sistema fue capaz de convertir todos los textos a voz de forma exitosa, por lo que se podría decir que tuvo un porcentaje de aciertos del 100%. Se realizaron pruebas con textos más largos y el sistema tardó casi lo mismo en procesar textos que darían un audio de 3 segundos que textos que darían un audio de 20 segundos. Por lo que el sistema está bien optimizado.

5.1.3 Voz a texto

- **Desempeño general:** Se realizaron pruebas bajo diferentes conclusiones, y se concluyó su funcionamiento de manera adecuada ya que aún con ruido de fondo o el usuario lejos de la computadora el sistema fue capaz de reconocer lo que se estaba diciendo casi en su totalidad.

- **Limitaciones y mejoras futuras:** Se encontró que el sistema tiene un aspecto de mejora importante el cuál es la optimización en audios largos, con audios de 5 o 10 segundos este tiene una respuesta rápida, pero al ser audios más largos el tiempo de procesamiento para poder procesarlos se alarga demasiado, esto no es mejorable por la forma en la que funciona la librería Google Text To Speech pero podría implementarse un algoritmo diferente para mejorar el tiempo de respuesta del sistema.

En conclusión, el sistema desarrollado representa un avance significativo en el reconocimiento automático de emociones en entornos reales como aulas escolares, ofreciendo información valiosa que puede contribuir a entender mejor el estado emocional de los estudiantes. No obstante, los resultados obtenidos evidencian la necesidad de continuar perfeccionando los algoritmos de detección, así como tener un aula específicamente para utilizar este sistema, donde se tengan bajo control todas las condiciones que puedan hacer variar las detecciones del sistema, y de combinar estos sistemas con otras fuentes de información, como encuestas o análisis contextuales, para lograr una mayor precisión y fiabilidad en la interpretación de las emociones humanas. Además, en el apartado de texto a voz y voz a texto, este funciona correctamente, aunque este tenga posibles mejoras, como en la optimización de los algoritmos.

Capítulo 6 Trabajos futuros

En esta sección se describen posibles trabajos futuros, ya que el sistema tiene carencias en algunos aspectos que no se han cubierto ya que no formaban parte de la implementación de este trabajo.

6.1 Desarrollo de red neuronal para diferenciar entre usuarios agachados o inclinados y usuarios tristes

Para mejorar el porcentaje de aciertos del sistema, se podría implementar una red neuronal que pueda analizar a los usuarios y filtre a los que están agachados o inclinados y a estos no se les realice el reconocimiento para evitar registros erróneos.

6.2 Implementación de un nuevo campo para los usuarios que NO son maestro en el apartado “Reconocimiento facial y de emociones” para poder agregar los datos faciales de otros usuarios, usando su “Nombre de usuario” para su reconocimiento facial.

La realización del sistema estuvo enfocada en el reconocimiento de emociones en el aula, por lo que no hubo un gran enfoque para el uso de este sistema por usuarios que no fueran maestros. Una de sus limitaciones es que solo puede reconocer a más de un usuario siempre y cuando el maestro que realice el reconocimiento tenga a muchos usuarios en su clase. Pero para usuarios normales, el sistema solo puede identificar el rostro del usuario que registro sus datos faciales, es posible reconocer

las emociones de otros usuarios solo que los nombres serán desconocidos, ya que el usuario no tiene permiso de acceder a datos faciales que no sean los suyos.

Una posible solución es implementar un cuadro donde se pueda meter el nombre de usuario del usuario que quiere usar el sistema junto al dueño de la cuenta, de esta forma se cargan los datos faciales del usuario invitado y se pueden identificar los rostros de todos los usuarios que hayan cargado sus datos faciales a la vez.

6.3 Implementación de una mejora al método de voz a texto para que funcione en tiempo real

El método de voz a texto tiene un procesamiento lento entre más largo es el audio, este es un problema ya que normalmente los audios a transcribir son largos, se intentó dividir los audios, pero se perdía información. Un trabajo a futuro sería mejorarlo para que pueda dividir los audios en pequeñas secciones y los vaya transcribiendo prácticamente al momento de escucharlos para mejorar la experiencia del usuario.

Bibliografía

- [1] I Ibáñez, N. (2002). LAS EMOCIONES EN EL AULA. *Estudios Pedagógicos*, 28. <https://doi.org/10.4067/s0718-07052002000100002> Visitado el 22 de Enero del 2025
- [2] Amazon Web Services. (s. f.). *¿Qué es el machine learning?* Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/machine-learning/> Visitado el 27 de enero de 2025
- [3] Belcic, I., & Stryker, C. (2025, 26 noviembre). Aprendizaje supervisado. *¿Qué es el aprendizaje supervisado?* Recuperado 27 de enero de 2025, de <https://www.ibm.com/mx-es/topics/supervised-learning>
- [4] Bergmann, D. (2025, 26 noviembre). Aprendizaje profundo. *¿Qué es el aprendizaje profundo?* Recuperado 27 de enero de 2025, de <https://www.ibm.com/mx-es/topics/deep-learning>
- [5] McDuff, D., Mahmoud, A., Mavadati, M., Amr, M., Turcot, J., & Kaliouby, R. E. (2016b). AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. *ACM DL*, 3723-3726. <https://doi.org/10.1145/2851581.2890247> Visitado el 27 de enero de 2025
- [6] Franco, C. H. E., Ospina, C. T., Cuevas, E. E. S., & Capacho, D. A. V. (2015). Reconocimiento facial basado en Eigenfaces, LBHP y Fisherfaces en la BeagleBoard-xM. *ojs.unipamplona.edu.co*. <https://doi.org/10.24054/rcta.v2i26.395> Visitado el 27 de enero de 2025
- [7] Kristensen, S. (2023, 11 mayo). iMotions (Spanish). iMotions. <https://imotions.com/es/> Visitado el 10 de febrero de 2025
- [8] iMotions. (2024, 5 septiembre). Afectiva - Humanizing Technology with Emotion AI. Afectiva. <https://www.affectiva.com/> Visitado el 10 de febrero de 2025

- [9] OpenCV: *Introduction to OpenCV-Python Tutorials*. (s. f.). https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html Visitado el 10 de enero de 2025
- [10] Fernández, C. G. (2024, 6 noviembre). OpenCV: Introducción y su rol en la visión por computadora. *OpenWebinars.net*. <https://openwebinars.net/blog/opencv-introduccion-y-su-rol-en-la-vision-por-computadora/> Visitado el 10 de enero de 2025
- [11] *pillow*. (2026, 2 enero). PyPI. <https://pypi.org/project/pillow/> Visitado el 10 de enero de 2025
- [12] Python, R. (2022, 19 septiembre). *Instalar PIL / Pillow y aplicar efectos visuales - Recursos Python*. Recursos Python. https://recursospython.com/guias-y-manuales/instalar-pil-pillow-efectos/#google_vignette Visitado el 10 de enero de 2025
- [13] *scikit-image: Image processing in Python* — *scikit-image*. (s. f.). <https://scikit-image.org/> Visitado el 10 de enero de 2025
- [14] Introducción a TensorFlow. (s. f.). TensorFlow. <https://www.tensorflow.org/learn?hl=es> Visitado el 10 de febrero de 2025
- [15] *¿Cuáles son los pros y los contras de usar TensorFlow frente a PyTorch?* (2023, 1 septiembre). <https://es.linkedin.com/advice/0/what-pros-cons-using-tensorflow-versus-pytorch?lang=es&lang=es> Visitado el 10 de febrero de 2025
- [16] PyTorch. (s. f.). PyTorch. <https://pytorch.org/> Visitado el 27 de febrero de 2025
- [17] Contributors, P. (s. f.). *Torch.onnx*. <https://pytorch.org/docs/stable/onnx.html> Visitado el 27 de febrero de 2025
- [18] Kurama, V. (2024, 23 octubre). *PyTorch vs. TensorFlow: Key Differences to Know for Deep Learning*. Built In. <https://builtin.com/data-science/pytorch-vs-tensorflow#:~:text=PyTorch%20is%20ideal%20for%20research,high-performance%20and%20scalable%20models>. Visitado el 27 de febrero de 2025

[19] *Scikit-Learn, herramienta básica para el Data Science en Python*. (2023, 30 agosto). Máster En Data Science. <https://www.master-data-scientist.com/scikit-learn-data-science/> Visitado el 10 de febrero de 2025

[20] Team, K. (s. f.). Keras: Deep Learning for humans. <https://keras.io/> Visitado el 10 de febrero de 2025

[21] Pykes, K. (2023, August 2). *PyTorch vs Tensorflow vs Keras*. <https://www.datacamp.com/tutorial/pytorch-vs-tensorflow-vs-keras> Visitado el 10 de febrero de 2025

[22] Juanweb. (2023, 1 octubre). *Aprendizaje automático con PyBrain en Python*. CodigosPython. <https://codigospython.com/aprendizaje-automatico-con-pybrain-en-python/> Visitado el 10 de febrero de 2025

[23] GTTS. (2024, 10 noviembre). PyPI. <https://pypi.org/project/gTTS/> Visitado el 10 de febrero de 2025

[24] *IA de Text-to-Speech: conversión de texto a voz natural | Google Cloud*. (s. f.). Google Cloud. <https://cloud.google.com/text-to-speech?hl=es> Visitado el 10 de febrero de 2025

[25] *openai-whisper*. (2025, 26 junio). PyPI. <https://pypi.org/project/openai-whisper/> Visitado el 10 de febrero de 2025

[26] Fernández, Y. (2023, 13 noviembre). *OpenAI Whisper: qué es, cómo funciona y cómo puedes usar esta inteligencia artificial para transcribir*. . . Xataka. <https://www.xataka.com/basics/openai-whisper-que-como funciona-como-puedes-usar-esta-inteligencia-artificial-para-transcribir-audios> Visitado el 18 de febrero de 2025

[27] *SpeechRecognition*. (2025, 31 diciembre). PyPI. <https://pypi.org/project/SpeechRecognition/> Visitado el 18 de febrero de 2025

[28] *wave — Read and write WAV files*. (s. f.). Python Documentation. <https://docs.python.org/3/library/wave.html> Visitado el 18 de febrero de 2025

- [29] PyAudio: Cross-platform audio I/O for Python, with PortAudio. (s. f.). <https://people.csail.mit.edu/hubert/pyaudio/> Visitado el 18 de febrero de 2025
- [30] deepspeech. (2020, 10 diciembre). PyPI. <https://pypi.org/project/deepspeech/> Visitado el 18 de febrero de 2025
- [31] pydub. (2021, 10 marzo). PyPI. <https://pypi.org/project/pydub/> Visitado el 18 de febrero de 2025
- [32] OpenCV: Cascade Classifier. (s. f.). https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html Visitado el 18 de febrero de 2025
- [33] OpenCV. (n.d.). *opencv/data/haarcascades/haarcascade_frontalface_default.xml at master · opencv/opencv*. GitHub. https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml
- [34] *numpy.frombuffer* — NumPy v2.1 Manual. (s. f.). <https://numpy.org/doc/2.1/reference/generated/numpy.frombuffer.html>
- [35] OpenCV: cv::CascadeClassifier Class Reference. (s. f.). https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html#aaf8181cb63968136476ec4204ffca498
- [36] Sistemas Inteligentes. (2021, 16 junio). *Detección de emociones en tiempo real* 🤖 | *Redes neuronales convolucionales | Face emotion | Tensorflow* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Jaai8JLNllc>
- [37] FER-2013. (2020, July 19). <https://www.kaggle.com/datasets/msmbare/fer2013/code>
- [38] TensorFlow. (s.f.). tf.keras.Model [Documentación]. TensorFlow. Recuperado el 22 de julio de 2025, de https://www.tensorflow.org/api_docs/python/tf/keras/Model

[39] De Automática, C. E., Tejado, I., Pérez, E., José, C. G. A., Isaías, G. P., Merchán, P., Lozano, J., Salamanca, S., & Vinagre, B. M. (2018). *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018*. <https://ruc.udc.es/entities/publication/a1f62f5d-841f-4760-99db-75d9842c9e08>