# Instituto Tecnológico
# y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

## Departamento de Electrónica, Sistemas e Informática

## Maestría en Sistemas Computacionales



# SISTEMA DE ANÁLISIS DE COMPORTAMIENTO Y PARTICIPACIÓN ESTUDIANTIL PARA LA TOMA DE DECISIONES PEDAGÓGICAS MEDIANTE INTELIGENCIA ARTIFICIAL MULTIMODAL

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de
**MAESTRO EN SISTEMAS COMPUTACIONALES**


Presenta: **ISC. Miguel Alejandro Villa Guerrero**

Asesor: **Dra. Gabriela Calvario Sánchez**



San Pedro Tlaquepaque, Jalisco. Julio de 2026

# AGRADECIMIENTOS

**A mi esposa**, quien ha sido mi pilar inquebrantable durante estos dos años de maestría. Gracias por tu amor infinito, tu comprensión y por el increíble trabajo en equipo que hicimos en casa para que yo pudiera dedicarle tiempo a este proyecto. Sin tu soporte diario, tu paciencia y tu aliento en los momentos más pesados, alcanzar esta meta hubiera sido muy difícil.

**A mis padres**, por ser el motor principal de mi educación y otorgarme la invaluable oportunidad de cursar una carrera universitaria. Todo el esfuerzo, los desvelos y los sacrificios que invirtieron para sacarme adelante durante mi pregrado son los cimientos que hoy me permiten acceder y culminar esta maestría. Este logro es el resultado de todo lo que me han dado, es tan suyo como mío.

Finalmente, **a mi asesora de Trabajo de Obtención de Grado**, por su dedicación, paciencia y guía experta a lo largo de este último año y medio. Agradezco profundamente el trabajo en conjunto, el valioso tiempo invertido en nuestras juntas de revisión, su retroalimentación siempre precisa y todo el apoyo técnico y académico brindado para llevar el desarrollo de este proyecto a buen puerto.

# DEDICATORIA

*A mi familia, por su apoyo incondicional y su paciencia a lo largo de este trayecto académico.*

# RESUMEN

Tanto en clases presenciales como virtuales, registrar las conductas y actitudes de los estudiantes resulta fundamental para el seguimiento académico. Sin embargo, este proceso representa un desafío significativo al gestionar grupos numerosos. La situación se complejiza en entornos virtuales, populares tras la transición a la enseñanza en línea, donde la interacción visual directa disminuye al proyectar materiales didácticos o al limitarse el número de cámaras visibles simultáneamente en pantalla, dificultando la identificación oportuna de situaciones de riesgo académico.

Entre los problemas más comunes en la práctica docente en línea se encuentran la dispersión de la atención durante las explicaciones, ausencias temporales injustificadas, baja participación activa, inasistencias o la conexión pasiva sin interacción real. La diversidad de estos eventos sobrepasa la capacidad de monitoreo manual y simultáneo del profesorado durante la sesión.

El desarrollo de una plataforma web que integre modelos de inteligencia artificial facilita la automatización de este monitoreo, proporcionando información cuantitativa y cualitativa sobre tiempos de atención, ausencias y participación. Asimismo, el sistema brinda retroalimentación sobre la dinámica de clase (tiempos de exposición del docente, identificación de temas con mayor interacción y estadísticas de receptividad del grupo).

El enfoque principal de esta propuesta se centra en el análisis post-sesión de clases virtuales grabadas en video, extrayendo flujos de audio y fotogramas para su procesamiento local y offline mediante modelos de visión artificial, reconocimiento de voz y procesamiento de lenguaje natural. Los resultados se consolidan en tableros de control intuitivos orientados a la optimización del proceso de enseñanza y aprendizaje en entornos virtuales.

# TABLA DE CONTENIDO

# LISTA DE FIGURAS

# LISTA DE TABLAS

# LISTA DE ACRÓNIMOS Y ABREVIATURAS

**API** Application Programming Interface (Interfaz de Programación de Aplicaciones). 28, 32–34, 36

**ASGI** Asynchronous Server Gateway Interface. 33

**CNN** Convolutional Neural Network (Red Neuronal Convolucional). 22

**CPU** Central Processing Unit (Unidad Central de Procesamiento). 30, 35

**CRUD** Create, Read, Update, Delete. 34, 36

**FPS** Frames Per Second (Fotogramas por Segundo). 29, 34, 39

**GPU** Graphics Processing Unit (Unidad de Procesamiento Gráfico). 35

**IA** Inteligencia Artificial. 13, 25, 26, 41

**ITESO** Instituto Tecnológico y de Estudios Superiores de Occidente. 12, 14

**LBPH** Local Binary Patterns Histograms. 18

**LLM** Large Language Model (Modelo de Lenguaje Grande). 35, 41

**LSTM** Long Short-Term Memory. 18

**MTCNN** Multitask Cascaded Convolutional Networks. 18

**NLP** Natural Language Processing (Procesamiento de Lenguaje Natural). 32, 38

**OCR** Optical Character Recognition (Reconocimiento Óptico de Caracteres). 35, 37, 41

**ORM** Object-Relational Mapping (Mapeo Objeto-Relacional). 34

**SPA** Single Page Application (Aplicación de Página Única). 32, 34, 40, 41

# 1. INTRODUCCIÓN

## 1.1. Antecedentes

En el ámbito de la docencia en el Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), se ha identificado la necesidad de contar con una herramienta tecnológica orientada al seguimiento del desempeño estudiantil. Si bien plataformas como Microsoft Teams, Zoom y Google Meet proporcionan datos básicos de asistencia y permanencia en las sesiones, existe una brecha en cuanto a analíticas profundas sobre la atención y el comportamiento de los participantes. Una herramienta robusta de análisis permitiría realizar un monitoreo sistemático del estudiante, facilitando la detección temprana de riesgos académicos, indicadores de posible deserción o bajo rendimiento.

Dicha información resulta valiosa no solo para la toma de decisiones pedagógicas informadas, sino también para implementar intervenciones oportunas destinadas a mitigar el rezago. Un sistema con estas características incrementa la capacidad para ofrecer una educación personalizada y efectiva, adaptándose a las necesidades de cada estudiante a pesar de las barreras que pudiese presentar la modalidad virtual.

Actualmente, la diferencia entre los datos crudos de conexión y la información relevante sobre el proceso de aprendizaje representa una oportunidad para innovar en el campo de la tecnología educativa. La implementación de un sistema de análisis no solo beneficia la planeación docente diaria, sino que también contribuye a la calidad educativa y al éxito escolar de los estudiantes que cursan asignaturas en línea.

## 1.2. Definición del problema

Las plataformas de videoconferencia y colaboración utilizadas en la educación en línea ofrecen datos limitados. Si bien estos datos pueden ser suficientes en escenarios corporativos, resultan insuficientes en el ámbito educativo para comprender en profundidad el desempeño de cada estudiante en la sesión. Al emplear estas herramientas en la educación, la escasez de información sobre participación y atención dificulta la identificación temprana de riesgos académicos y limita la toma de decisiones pedagógicas oportunas.

En entornos virtuales, la falta de mediciones avanzadas sobre patrones de interacción, cambios en la conducta o lapsos de desconexión dificulta el monitoreo continuo del progreso estudiantil. Como resultado, el profesorado no dispone de las herramientas necesarias para adaptar la enseñanza de forma individualizada ni para intervenir de manera adecuada ante signos de rezago o desmotivación.

Por lo tanto, el problema radica en la ausencia de una solución integral que permita analizar datos relevantes sobre el comportamiento y la participación de los estudiantes durante las clases en línea. Sin estas analíticas, se limita la oportunidad de ofrecer una educación personalizada que atienda las necesidades específicas de cada alumno, lo cual impacta negativamente en la calidad de la formación y en los índices de éxito académico.

## 1.3. Objetivo general

El objetivo general de este trabajo consiste en diseñar e implementar un sistema que integre modelos de Inteligencia Artificial (IA) para facilitar la obtención de información analítica sobre el desempeño estudiantil y brindar retroalimentación a partir de los datos recopilados en las sesiones de clase virtual. Dicha solución busca integrar herramientas de recolección de información y análisis de comportamiento para ofrecer una perspectiva de la participación, atención y evolución de los estudiantes a lo largo de las sesiones educativas.

Para lograrlo, se propone desarrollar una aplicación web utilizando modelos de visión artificial, reconocimiento de voz y procesamiento de lenguaje natural enfocados en el seguimiento de estudiantes y en la generación de estadísticas docentes. El sistema incluye funcionalidades como el registro de tiempos de atención y participación, control de asistencias y ausencias durante la sesión, así como el análisis de cambios de interés o niveles de interacción ante temas específicos. Con esta información, se identifican patrones y conductas útiles para la gestión del proceso de enseñanza y aprendizaje.

Complementariamente, el sistema integra capacidades de procesamiento de lenguaje natural para la transcripción automática de las sesiones. A través del análisis de sentimiento de las intervenciones, se extraen métricas sobre el clima emocional y la dinámica de la conversación, proporcionando al docente una perspectiva cuantitativa y cualitativa sobre

el desarrollo de cada clase.

## 1.4. Objetivos específicos

Diseñar y desarrollar una aplicación web que emplee modelos de inteligencia artificial local para el análisis de sesiones de clase, cubriendo los siguientes requerimientos:

1. **Seguimiento de estudiantes:**
    - Registrar los tiempos de atención visual y participación en pantalla.
    - Controlar asistencias, ausencias y tiempos fuera de la sesión virtual.
    - Analizar comportamientos grupales, tales como cambios de interés durante explicaciones específicas.
2. **Retroalimentación docente:**
    - Generar estadísticas sobre la duración y distribución de las exposiciones.
    - Identificar las dinámicas que promueven un mayor nivel de interacción en el grupo.
    - Detectar los temas o momentos de la clase que generan variaciones de conducta.
    - Analizar el sentimiento verbal obtenido a partir de la transcripción de la clase.

## 1.5. Justificación

La facilidad para recopilar y analizar información sobre el desempeño estudiantil proporciona a los profesores de instituciones como el ITESO las herramientas necesarias para evaluar alternativas en la planeación y el seguimiento académico. Al transitar de un monitoreo basado exclusivamente en observaciones generales (las cuales se complejizan en un ámbito virtual) a un esquema fundamentado en datos y capacidades analíticas, es posible estructurar registros históricos que ayuden a evaluar la evolución de la participación de los alumnos y a organizar de manera más efectiva las estrategias de enseñanza y los tiempos de intervención pedagógica.

La base para lograr esta mejora en la calidad educativa radica en la administración de los datos, lo cual involucra la adquisición, limpieza, procesamiento y exploración de la información generada durante las clases virtuales. La adopción de un sistema integral ayuda a contar con una mejor planificación académica, a la vez que optimiza y automatiza los procesos de recolección de datos. Como resultado, al disponer de información detallada que anteriormente no se registraba, se incrementa la eficiencia en la detección de riesgos académicos y se mejora la calidad de la información disponible para la toma de decisiones.

Asimismo, se reconoce la diversidad de contextos educativos y la disponibilidad de recursos tecnológicos en las instituciones, puesto que no todas cuentan con el mismo nivel de infraestructura o acceso a herramientas de análisis avanzado. Sin embargo, la propuesta ofrece un enfoque flexible y adaptable, de tal forma que el profesorado, independientemente del nivel de infraestructura tecnológica inicial, pueda beneficiarse de una metodología basada en el análisis de datos. Esto se traduce en una mayor competitividad académica, una toma de decisiones informada y mejores resultados en el aprendizaje estudiantil.

# 2. ESTADO DEL ARTE O DE LA TÉCNICA

Existen diversos artículos [1], [2], [3] que analizan la satisfacción de los estudiantes con las clases virtuales, destacando que, en general, esta ha sido baja. De hecho, el alumnado suele mostrar una mayor preferencia por las modalidades presenciales o híbridas. Esto proporciona una perspectiva de las causas detrás de estas respuestas, ya que el principal problema parece ser la falta de adaptación de los cursos al formato virtual. Como ejemplo se presenta un estudio de la Universidad de Córdoba [3] en el cual se revelan desafíos significativos en la enseñanza en línea. La crisis sanitaria obligó a una rápida virtualización, lo que derivó en que los estudiantes manifestaran insatisfacción con la transición. Según dicho estudio, aplicado a 883 estudiantes, la mayoría percibió que la docencia no se ajustó al formato virtual, calificándola de manera negativa. Se evidenció que la preferencia general se inclina hacia clases presenciales o híbridas, especialmente en áreas como Ciencias Sociales y Jurídicas y Ciencias de la Salud, donde la virtualización fue menos efectiva. Sin embargo, en disciplinas técnicas como Ingeniería y Arquitectura, se observó una mejor adaptación y mayor aceptación de las herramientas digitales.

Dentro del mismo estudio [3] se explica que uno de los factores que contribuyeron a la insatisfacción fue la falta de uniformidad en la metodología docente, ya que cada profesor implementó una estrategia propia sin coordinación institucional. Mientras algunos optaron por sesiones síncronas mediante videoconferencias, otros emplearon un enfoque asíncrono, basado en materiales grabados y documentos de soporte. El alumnado valoró positivamente un modelo híbrido que combinara ambos formatos, puesto que las clases exclusivamente asíncronas fueron percibidas como menos efectivas para el aprendizaje. Además, aunque plataformas como Zoom y Blackboard Collaborate fueron las más utilizadas, su implementación no siempre fue la óptima, y la variedad de herramientas empleadas generó dificultades de adaptación.

Estos resultados coinciden con otras investigaciones [1], [2] sobre la satisfacción del alumnado con la enseñanza virtual, donde se establece que la percepción negativa se relaciona

con la falta de adecuación del curso al entorno digital. La preferencia por la presencialidad se mantiene debido a la interacción directa con el profesorado y la facilidad para resolver dudas en tiempo real. No obstante, se considera que la educación superior debe evolucionar hacia modelos flexibles e integradores, en los que la tecnología se utilice de manera estratégica para complementar y mejorar la experiencia de aprendizaje. La experiencia durante la pandemia ha dejado lecciones sobre la necesidad de planificación y formación docente en competencias digitales para garantizar una transición efectiva a escenarios de enseñanza híbrida o virtual.

Desde la identificación de este problema, se han desarrollado proyectos enfocados en la detección de emociones en clases virtuales. Un ejemplo se encuentra en el proyecto .ªnálisis de sentimientos con inteligencia artificial para mejorar el proceso enseñanza-aprendizaje en el aula virtual"[2], que aborda la transformación educativa por el avance de la digitalización y la necesidad de adaptar la enseñanza al entorno virtual. Este proyecto evidencia la importancia de comprender el estado emocional de los estudiantes para mejorar la experiencia educativa, y propone el uso de inteligencia artificial con redes neuronales para analizar en tiempo real las emociones de los alumnos mediante reconocimiento facial, permitiendo a los docentes evaluar la percepción de la clase y ajustar las estrategias de enseñanza.

Para el análisis, se desarrolló una herramienta basada en modelos de inteligencia artificial como DeepFace, ArcFace y FaceNet. Estos algoritmos permiten identificar expresiones faciales y clasificar emociones como felicidad, tristeza, miedo, enojo o neutralidad. La aplicación fue probada capturando imágenes en tiempo real y generando un informe del estado emocional general del aula. En lugar de centrarse en el estudiante individual, la herramienta proporciona un panorama grupal que ayuda al profesorado a tomar decisiones metodológicas informadas.

Los resultados obtenidos [2] muestran que este sistema facilita la adaptación de las estrategias pedagógicas al contexto emocional del grupo. Se comprobó que los docentes pueden acceder a un monitoreo constante del estado de ánimo sin interrumpir la dinámica de la clase. El sistema destaca por su viabilidad y compatibilidad con plataformas de educación virtual como Zoom, Teams o Blackboard. El estudio también señala la posibilidad de ampliar el uso de esta herramienta para la medición de participación en clase y la evaluación de la motivación en entornos digitales.

Otro desarrollo relevante es el proyecto de investigación del Instituto Tecnológico Metropolitano (ITM) de Medellín, Colombia [1], el cual consistió en entrenar modelos de reconocimiento facial y detección de emociones utilizando dos bases de datos diseñadas específicamente para el proyecto.

La primera base de datos contenía videos de 9 estudiantes mientras observaban contenido diseñado para provocar emociones específicas: felicidad, neutralidad y enojo. A partir de esos videos, grabados a 30 fotogramas por segundo, se extrajeron 9873 imágenes distribuidas en 2425 muestras para "neutral", 3907 para "felicidad", y 3539 para .enojo"[1].

La segunda base de datos registró secuencias de video de 15 estudiantes durante videoconferencias. Cada estudiante participó en dos conferencias: una de interés personal y otra de libre elección. Se recolectaron 1228 muestras de .ªtenciónz 438 de "no atención".

Para el desarrollo del reconocimiento facial se eligió el modelo Haar-cascade, un algoritmo de detección de objetos que identifica rostros. Este modelo, implementado con OpenCV [4], convierte las imágenes a escala de grises y utiliza características Haar (bordes y líneas) para clasificar imágenes con rostros (positivas) y sin ellos (negativas), ajustando pesos de manera iterativa.

Para el reconocimiento de emociones se aplicó el algoritmo Local Binary Patterns Histograms (LBPH). Este operador analiza las texturas de la imagen etiquetando los píxeles según su vecindario y generando números binarios, cuyos histogramas resultantes se comparan para clasificar las emociones.

Los resultados [1] mostraron tasas de acierto del 95.0 % para el modelo de emociones y del 96.7 % para el de atención. La implementación de la aplicación web permitió a los docentes obtener información en tiempo real sobre el estado emocional y la atención de los estudiantes, lo cual contribuye a mejorar los procesos de enseñanza-aprendizaje en entornos virtuales.

La precisión obtenida en las pruebas se alcanzó tras un proceso de entrenamiento en el cual los modelos fueron entrenados con el 80 % de los datos, destinando el 20 % restante para prueba y validación. En el caso del modelo de atención, se utilizó una red neuronal de tipo LSTM para procesar secuencias multivariadas generadas a partir de puntos fiduciales del rostro capturados con MTCNN.

Este enfoque demostró ser eficaz tanto en el ámbito técnico como en el práctico, al facilitar a los docentes herramientas para el monitoreo del compromiso emocional y cognitivo del alumnado en clases virtuales [1].

## 2.1. Librería GazeTracking [5]

Existen soluciones enfocadas en analizar la atención que un estudiante presta a la pantalla [5], [6]. Entre ellas destaca GazeTracking, una biblioteca desarrollada en Python para el rastreo de la mirada en tiempo real, la cual proporciona datos sobre la posición de la pupila y determina la dirección visual de la persona. Esto resulta útil para analizar los movimientos oculares y detectar momentos de distracción.

El proyecto está publicado bajo la licencia MIT (código abierto), lo que permite utilizar, modificar y distribuir la librería libremente. Gracias a esta flexibilidad, es posible adaptarla e integrarla en entornos de investigación para profundizar en el estudio de la interacción del estudiante con la interfaz.

La herramienta GazeTracking se basa en técnicas de visión por computadora que analizan, fotograma a fotograma, los rasgos faciales y la posición de la pupila. Utiliza la cámara web para capturar el rostro del usuario en tiempo real y, mediante algoritmos de reconocimiento facial, localiza los ojos dentro de la región facial.

Posteriormente, realiza un procesamiento de la pupila para analizar su posición exacta y determinar si el usuario mira a la izquierda, a la derecha, hacia arriba o hacia abajo. Al finalizar el procesamiento, devuelve información sobre la posición y dirección de la mirada, lo cual se integra en la aplicación general. En la Figura 2.1 se ilustra el funcionamiento de esta herramienta.


Figura 2.1: Ejemplo de estimación visual mediante la herramienta GazeTracking.

Esta tecnología permite determinar si un estudiante desvía la mirada de la pantalla, lo que puede indicar una pérdida de interés. Con los datos recopilados, se obtiene el tiempo total que el alumno mantiene la vista al frente, sirviendo como métrica de concentración sin necesidad de métodos invasivos.

A partir de la revisión de GazeTracking, se identifican otras herramientas orientadas al procesamiento visual en videollamadas, las cuales contribuyen al análisis de emociones, puntos de referencia faciales y estimación de la atención.

## 2.2. Librería EmotiEffLib [7]

EmotiEffLib es una biblioteca ligera diseñada para el reconocimiento eficiente de emociones faciales en imágenes y secuencias de video. Entre sus características principales

destaca el procesamiento en tiempo real, lo cual resulta útil para optimizar la obtención de resultados.

Esta biblioteca ofrece implementaciones en Python y es compatible con PyTorch, lo que facilita su integración con otros desarrollos. Dispone de la capacidad de clasificar múltiples emociones faciales, sirviendo de base para analizar el estado emocional del grupo durante la videollamada. Sin embargo, EmotiEffLib no incluye de manera nativa funciones para la estimación de la mirada o la atención visual, por lo que requiere complementarse con herramientas como GazeTracking.

## 2.3. Framework DeepFace [8]

DeepFace es un framework de análisis facial híbrido para Python que integra modelos de visión artificial como VGG-Face, Google FaceNet y OpenFace en una sola interfaz. A diferencia de bibliotecas específicas, ofrece una solución integral que abarca el reconocimiento de emociones, la verificación de identidad y el análisis de atributos demográficos (edad, género y raza). Su arquitectura de alto nivel permite obtener resultados complejos con pocas líneas de código, agilizando el desarrollo del módulo de análisis de comportamiento.

Para los propósitos del presente proyecto, DeepFace resulta adecuado para procesar fotogramas de clases grabadas y extraer métricas sobre el estado anímico del grupo, destacando por su robustez en la detección de rostros bajo diferentes ángulos en videollamadas. No obstante, al envolver modelos de aprendizaje profundo pesados, su consumo de recursos es mayor en comparación con opciones optimizadas para CPU. Dado que carece de funciones de rastreo ocular, su uso se complementa con librerías específicas de mirada. En la Figura 2.2 se muestra un ejemplo del funcionamiento de este framework.


Figura 2.2: Ejemplo de detección facial y análisis de atributos con DeepFace.

## 2.4. Facial Emotion Recognition con OpenCV y Deepface [4], [9]

Este desarrollo implementa la detección de emociones faciales en tiempo real utilizando las bibliotecas DeepFace y OpenCV [4]. Sus características principales incluyen la captura de video desde la cámara web, la localización del rostro y la clasificación de emociones, superponiendo etiquetas visuales en los fotogramas, tal como se detalla en el flujo de la Figura 2.3.

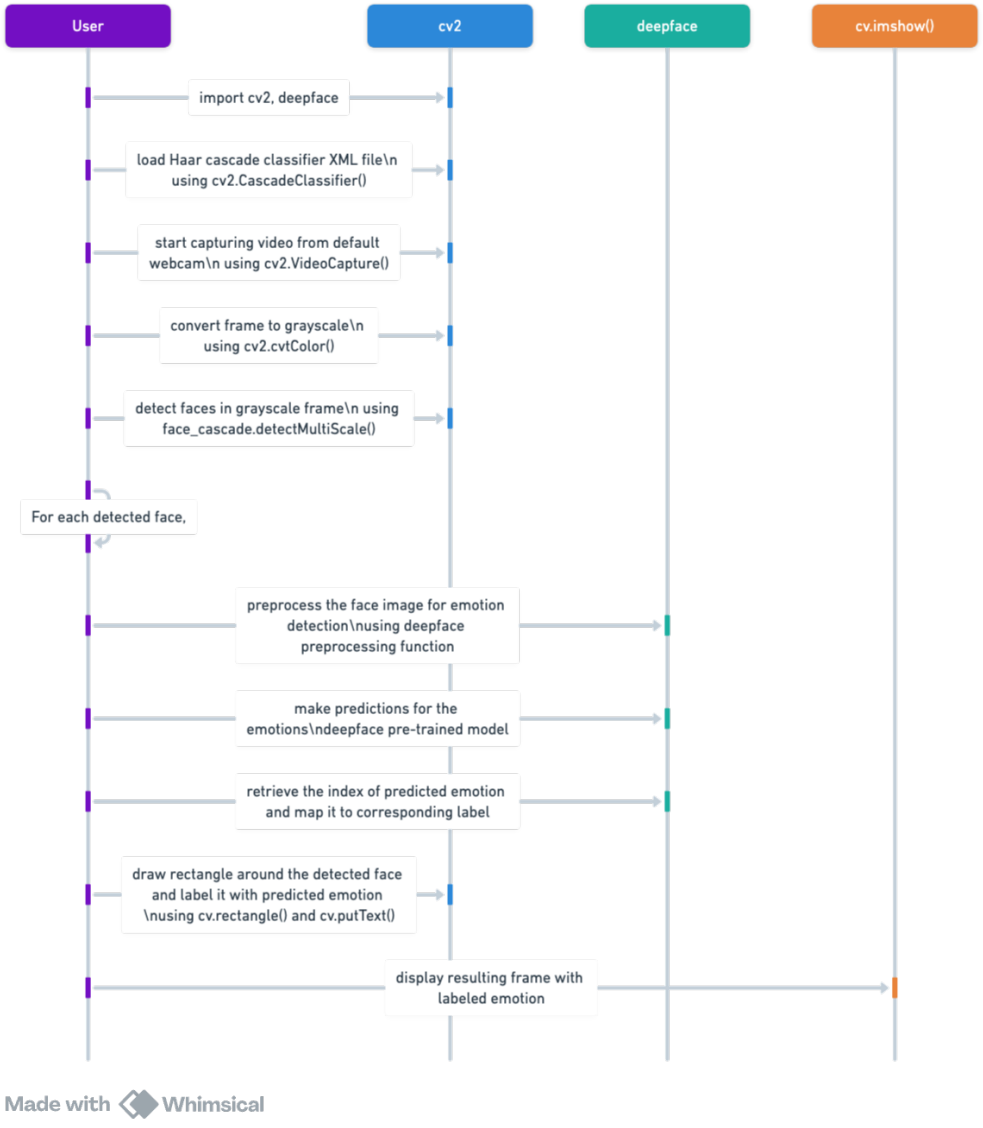


Figura 2.3: Flujo de procesamiento para el reconocimiento de emociones con OpenCV y DeepFace.

La rapidez de este procesamiento es útil para obtener métricas agregadas al concluir la sesión. La disponibilidad de modelos preentrenados en DeepFace evita la necesidad de realizar entrenamientos desde cero. No obstante, al igual que EmotiEffLib, se concentra en emociones y no aborda la estimación de la mirada.

## 2.5. Emotion-detection [10]

Este proyecto clasifica las emociones del rostro en siete categorías: enojo, disgusto, miedo, felicidad, neutralidad, tristeza y sorpresa. Utiliza redes neuronales convolucionales (CNN) entrenadas con el conjunto de datos FER-2013, el cual consta de 35,887 imágenes en escala de grises de expresiones faciales.

El sistema detecta rostros en tiempo real empleando un clasificador en cascada de Haar, basado en el método de Viola y Jones [11]. Una vez localizado el rostro, la región de interés se redimensiona a 48x48 píxeles y se introduce a la red neuronal convolucional, que produce una distribución probabilística mediante una función Softmax. La emoción con mayor probabilidad es mostrada en pantalla.

El desarrollo fue realizado en Python con TensorFlow, y proporciona instrucciones para el uso de modelos preentrenados o su entrenamiento desde cero. La implementación está diseñada para procesar múltiples rostros en una imagen, lo cual resulta relevante para futuros análisis presenciales en el salón de clases.

## 2.6. Librería OpenSeeFace [6]

OpenSeeFace está diseñada para el seguimiento en tiempo real de rostros y puntos de referencia faciales (landmarks), contando con soporte para su integración con el motor Unity. Su función principal consiste en detectar características faciales clave (ojos, cejas, nariz y boca), permitiendo el análisis geométrico detallado de las facciones.

Esta herramienta se sitúa como opción secundaria debido a que su enfoque principal no está alineado directamente con el análisis de emociones o de atención visual requerido. No obstante, es posible emplear sus coordenadas de tracking facial para entrenar modelos complementarios que infieran emociones a partir del desplazamiento de los puntos fiduciales. Un aspecto relevante a destacar es su robustez para reconocer rostros en condiciones deficientes de iluminación, como se ilustra en la Figura 2.4, o bajo ángulos extremos de inclinación de la cabeza, mostrados en la Figura 2.5.

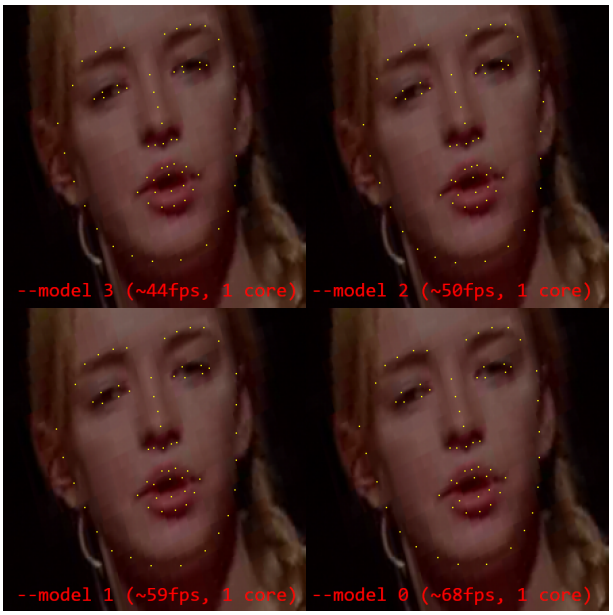


Figura 2.4: Detección de puntos fiduciales con OpenSeeFace en condiciones de baja iluminación.

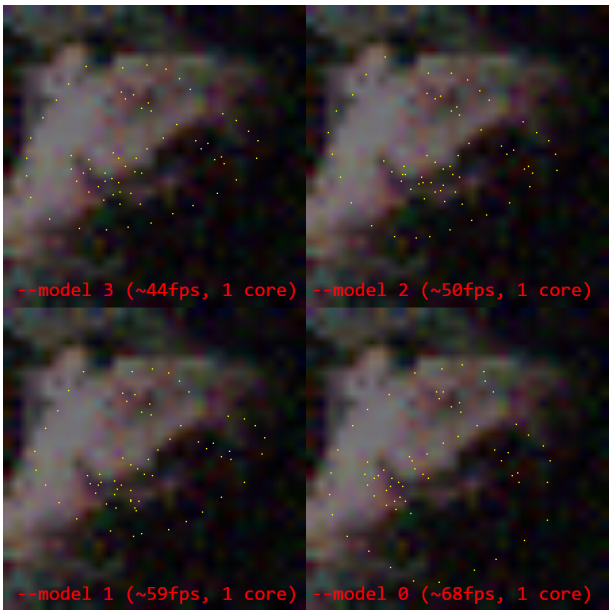


Figura 2.5: Tracking facial con OpenSeeFace ante rotaciones severas de la cabeza.

## 2.7. Comparativa de Modelos y Herramientas

A partir del análisis individual de los proyectos, se estructuró una comparación basada en criterios técnicos y funcionales, resumida en la Tabla 2.1. Esto facilita la selección de la combinación óptima de modelos para el diseño del sistema.

| **Herramienta** | **Propósito Principal** | **Características Clave** | **Licencia / Backend** |
|---|---|---|---|
| GazeTracking | Estimación de mirada y seguimiento de pupila. | Detección de la pupila, estimación de dirección izquierda/derecha/centro. | MIT / OpenCV, Dlib |
| EmotiEffLib | Clasificación eficiente de emociones y engagement. | Ligera, optimizada para tiempo real, análisis de frames y video. | Apache 2.0 / PyTorch, TensorFlow |
| DeepFace | Framework híbrido para análisis facial. | Integración de múltiples modelos (VGG-Face, FaceNet), análisis de emociones, género y edad. | MIT / Keras, TensorFlow |
| OpenSeeFace | Seguimiento de rostro y puntos fiduciales. | Tracking preciso de puntos clave en condiciones de iluminación deficiente, integración con Unity. | MIT / PyTorch, ONNX |
| Emotion-detection | Clasificación de emociones faciales en 7 clases. | Redes convolucionales (CNN) entrenadas con el dataset FER-2013. | Código Abierto / TensorFlow, Keras |

Tabla 2.1: Comparativa de modelos y herramientas de visión artificial analizados en el estado del arte.

En conclusión, dado que ninguno de los proyectos [5], [6], [7], [8], [9], [10] resuelve de manera aislada todos los requerimientos de atención y análisis afectivo, el diseño metodológico del sistema se fundamenta en la complementariedad de las herramientas. Al combinar el análisis afectivo global de DeepFace con la estimación de atención de GazeTracking y el procesamiento de audio y texto, es posible consolidar una solución integral para el análisis del aula virtual.

# 3. MARCO TEÓRICO Y METODOLÓGICO

## 3.1. Marco teórico

### 3.1.1. Conceptos Generales

La Inteligencia Artificial (IA) se refiere a sistemas informáticos capaces de realizar tareas que normalmente requieren inteligencia humana, tales como el aprendizaje, la percepción y la toma de decisiones. Dentro de la IA, se encuentran áreas específicas como el Machine Learning y el Deep Learning, que utilizan grandes volúmenes de datos para mejorar continuamente su desempeño en tareas concretas.

### 3.1.2. Inteligencia Artificial Aplicada a la Educación

La aplicación de la IA en la educación ha abierto nuevas posibilidades para personalizar el aprendizaje y mejorar la gestión educativa. Entre estas aplicaciones se encuentran los sistemas de tutoría inteligente, plataformas de aprendizaje adaptativo y herramientas para el análisis del comportamiento estudiantil. Estas tecnologías permiten recopilar información sobre el progreso de los alumnos, identificar áreas de oportunidad y proporcionar retroalimentación tanto a estudiantes como a docentes.

La detección de emociones en entornos educativos virtuales parte de la premisa de que la expresión facial es un componente esencial de la comunicación humana. Con este conocimiento se adopta la inteligencia artificial y el aprendizaje profundo para desarrollar soluciones que permitan, mediante el análisis de imágenes, identificar las emociones predominantes en los rostros de los estudiantes.

Se emplean redes neuronales para extraer características relevantes de las imágenes y clasificarlas en categorías. Estos modelos son capaces de identificar patrones y asociarlos con estados emocionales como neutralidad, alegría o enojo. Existen diversos desarrollos para reconocer rostros y clasificar emociones y rasgos faciales de forma local, optimizando el consumo de recursos computacionales durante la sesión de clase.

### 3.1.3. Retos Éticos y Privacidad en el uso de IA

El uso de tecnologías de inteligencia artificial y análisis de datos en la educación plantea retos en cuanto a la privacidad y el manejo ético de la información. La recolección de datos sensibles, como imágenes de rostros y grabaciones de voz, requiere de políticas claras sobre el consentimiento informado y el almacenamiento seguro de los datos. Además, es fundamental evitar sesgos en los modelos de análisis y garantizar que el uso de estas herramientas no derive en discriminación hacia los estudiantes.

Para mitigar estos riesgos, se deben implementar medidas como:

- Anonimización de datos sensibles antes de su almacenamiento.
- Limitación del acceso a la información únicamente al personal autorizado.
- Mecanismos de consentimiento informado para estudiantes.
- Auditorías periódicas y transparencia en el uso de los datos.

La adopción de estas soluciones tecnológicas no sustituye la labor docente, sino que la complementa. El reto radica en combinar la automatización del análisis de atención con la empatía y capacidad de respuesta pedagógica del profesorado. Al hacerlo, se diseñan estrategias más personalizadas e inclusivas, reforzando la interacción y fomentando la participación estudiantil.

El análisis del aula mediante inteligencia artificial representa una oportunidad para transformar el proceso educativo, brindando herramientas avanzadas para comprender la participación, la atención y las emociones de los estudiantes en entornos virtuales. La implementación de tecnologías como el reconocimiento facial, el análisis emocional y el seguimiento ocular permite no solo detectar de manera temprana el rezago académico, sino también adaptar las estrategias pedagógicas a las necesidades específicas del grupo.

No obstante, esta aplicación presenta desafíos relevantes, entre los cuales destacan las limitaciones técnicas asociadas a las condiciones del entorno (iluminación o calidad de imagen) y los retos éticos relacionados con la gestión de datos sensibles. Por ello, es fundamental establecer lineamientos institucionales rigurosos que garanticen un uso ético, seguro y transparente de la IA.

En definitiva, aunque el uso de la inteligencia artificial en la educación posee un potencial considerable para mejorar la experiencia educativa, su implementación debe ser planificada, contemplando principios éticos y priorizando la privacidad y el bienestar del alumnado y del cuerpo docente.

## 3.2. Desarrollo metodológico

Este capítulo describe la metodología para transformar video, audio y metadatos generados en clases virtuales en información cuantitativa y cualitativa útil para el docente. La metodología se alinea con los objetivos de registrar la atención, ausencias y participación de los estudiantes, y ofrecer retroalimentación sobre la dinámica de clase y tiempos de presentación. En la Figura 3.1 se presenta el esquema general de las fases de la metodología.

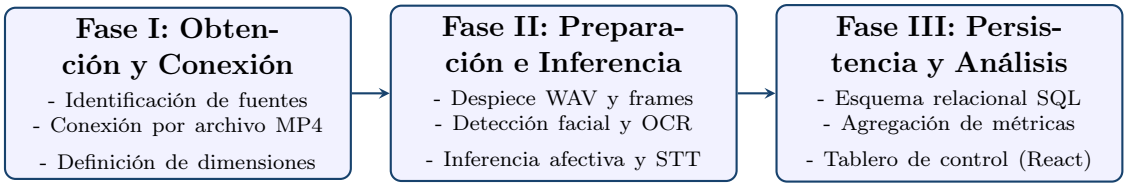


Figura 3.1: Esquema general de las tres fases metodológicas del proyecto.

### 3.2.1. Fase de Obtención de datos

Esta fase cubre la recolección de insumos necesarios para la construcción del proyecto, la determinación de métricas y dimensiones, la identificación de fuentes y la selección del tipo de conexión a dichas fuentes.

#### 3.2.1.1. Determinación de métricas y dimensiones

**Dimensiones:**

- Curso, asignatura, grupo, docente, periodo escolar.
- Sesión, fecha, hora de inicio y fin, modalidad de interacción.
- Identificador de alumno (anonimizado).
- Tiempo (segmentos de sesión).
- Contexto técnico (calidad de frame, iluminación estimada).

**Métricas:**

- Atención: porcentaje de fotogramas con rostro detectado, estimación de la dirección de mirada y estabilidad visual.
- Participación: tiempo de habla por alumno y por docente, número de intervenciones en chat o audio.
- Presencia/ausencia: tiempo de conexión activa, permanencia en pantalla y detección de ausencias.
- Expresiones/emociones: distribución porcentual de categorías emocionales.
- Dinámicas de clase: tiempo de exposición del docente frente a la interacción grupal y efectos en la atención del grupo.
- Transcripción: texto generado a partir del flujo de audio.

#### 3.2.1.2. Identificación de las fuentes de datos

Las fuentes principales del sistema corresponden a grabaciones de video de clases virtuales (procedentes de plataformas como Microsoft Teams, Zoom o Google Meet), audio de la sesión, logs de asistencia, eventos de chat y metadatos de la plataforma.

#### 3.2.1.3. Selección del tipo de conexión

Se contemplan dos estrategias principales para la ingesta de datos:

- Extracción por archivo (procesamiento asíncrono de grabaciones en formato MP4).
- Extracción por API (conectores directos de la plataforma cuando sea viable).

Para la implementación inicial y validación del piloto, se prioriza la extracción mediante archivos cargados de forma directa por el docente debido a su menor fricción técnica.

### 3.2.2. Fase de preparación de los datos

En esta fase se busca normalizar, alinear y transformar los datos crudos multimedia para habilitar análisis consistentes. Se apoya en un pipeline de procesamiento multimodal detallado en la Figura 3.2.

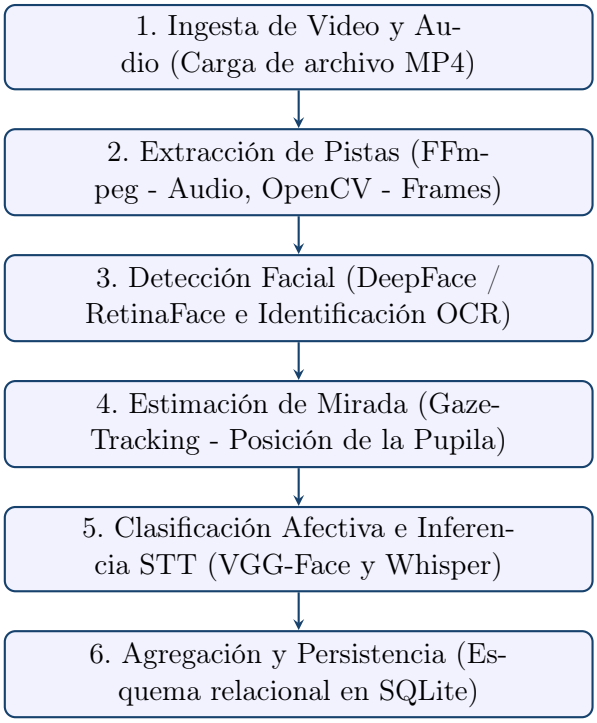


Figura 3.2: Diagrama de flujo del pipeline de preparación y procesamiento de datos en la sección 3.2.2.

#### 3.2.2.1. Unificación de los datos con identificadores

Se normalizan los nombres y tipos de campos (identificador de sesión, identificador anonimizado de alumno, marcas de tiempo y calidad del fotograma) y se establecen claves relacionales para la unión entre las fuentes de datos (video, audio y eventos).

#### 3.2.2.2. Sincronización multimodal y timestamp

Se alinean las pistas de video, audio y logs mediante marcas de tiempo unificadas, corrigiendo derivas temporales y segmentando la sesión en bloques de tiempo regulares para la agregación de métricas.

#### 3.2.2.3. Normalización de señales

En el flujo de video se aplica remuestreo a una tasa de FPS fija, redimensionado de imágenes y compresión para optimizar el almacenamiento. En la pista de audio se realiza remuestreo (a 16 kHz) y reducción de ruido de fondo para optimizar el reconocimiento de voz.

#### 3.2.2.4. Detección y seguimiento de rostro

Para la detección y seguimiento facial se utiliza el framework de visión artificial **DeepFace** [8]. Este permite integrar de manera dinámica múltiples motores de detección en función de la precisión y recursos disponibles:

- **RetinaFace:** Empleado por defecto debido a su alta precisión, resistencia a oclusiones y cambios de iluminación severos.
- **MediaPipe (BlazeFace):** Seleccionado para ejecuciones rápidas y de bajo consumo en CPU.
- **MTCNN:** Utilizado cuando se requiere alinear rasgos faciales basados en puntos fiduciales.
- **YOLOv8 y Haar Cascades:** Soportados como alternativas ligeras adicionales.

Este módulo localiza las coordenadas espaciales de los rostros de los estudiantes presentes en cada fotograma, registrando métricas de presencia y verificando la asistencia a la sesión.

#### 3.2.2.5. Extracción de características de atención y mirada

La estimación de la mirada se ejecuta utilizando la biblioteca especializada **GazeTracking** [5]. El método analiza la región ocular aislada por el detector facial, localizando la posición relativa de las pupilas mediante el cálculo del centroide en una máscara de umbral binario. A partir de la posición relativa de la pupila respecto a las esquinas del ojo, se determina si el estudiante mira directamente a la pantalla (centro) o si desvía la mirada (izquierda, derecha, abajo), generando métricas agregadas de atención visual.

#### 3.2.2.6. Reconocimiento de expresiones/emociones

Una vez aislado el rostro, el framework **DeepFace** [8] ejecuta la clasificación de emociones empleando el modelo convolucional preentrenado **VGG-Face**. El método toma la región facial recortada y normalizada y devuelve un vector probabilístico distribuido en siete emociones estándar: neutralidad, alegría, tristeza, enojo, miedo, disgusto y sorpresa. Estos datos se promedian por bloques temporales para analizar la receptividad anímica colectiva del grupo de forma anónima.

#### 3.2.2.7. Diarización del habla y cuantificación de participación

El flujo de audio es procesado utilizando el modelo de transcripción local **Whisper.cpp** [12], que implementa el algoritmo de codificador-decodificador de Whisper para generar

una transcripción estructurada con marcas de tiempo a nivel de palabra y oración. La participación oral se cuantifica mediante la segmentación del audio y la detección de actividad de voz, promediando el tiempo de intervención del docente frente a los estudiantes.

#### 3.2.2.8. Agregación temporal, reglas y métricas derivadas

Se consolidan los datos de las fases anteriores mediante algoritmos basados en reglas. Se calcula el puntaje de atención individual (correlacionando la detección facial y la mirada frente a la pantalla) y se estiman las dinámicas de participación cruzando las intervenciones de voz con el historial de chat de la clase.

#### 3.2.2.9. Anonimización, cumplimiento y persistencia

Como medida de privacidad, los datos personales se anonimizan mediante identificadores numéricos únicos que reemplazan el nombre de los estudiantes antes del procesamiento. Una vez extraídas las métricas vectoriales y de texto, los archivos de video y audio temporales pueden ser eliminados, conservando únicamente las estadísticas estructuradas agregadas.

### 3.2.3. Fase de almacenamiento para análisis

Los datos resultantes se estructuran en un esquema analítico relacional en SQLite [13]. Las métricas se almacenan en tablas de hechos (Atención, Participación, Presencia) vinculadas a dimensiones (Alumno, Sesión, Tiempo, Curso), optimizando la velocidad de respuesta del frontend al generar los tableros de control.

# 4. ARQUITECTURA E IMPLEMENTACIÓN DEL SISTEMA

Este capítulo detalla la construcción del sistema, una aplicación web diseñada para la gestión, procesamiento y análisis de sesiones de clase basadas en video. El código fuente de la aplicación desarrollada se encuentra disponible públicamente en el repositorio de GitHub: `https://github.com/villami-school/MastersProject.git`.

Se expone la arquitectura general del proyecto, las tecnologías adoptadas para el frontend y backend, y se describe el funcionamiento de los canales de procesamiento (pipelines) encargados de la extracción de métricas mediante Inteligencia Artificial.

## 4.1. Arquitectura del Sistema

El sistema opera bajo un modelo cliente-servidor, separando la interfaz de usuario de la lógica de negocio y el procesamiento intensivo. La arquitectura se compone de tres bloques principales:

- **Frontend (Cliente):** Aplicación de página única (SPA) responsable de la interacción con el usuario, visualización de estadísticas y gestión de las sesiones.
- **Backend (Servidor/API):** Núcleo del sistema expuesto a través de una API RESTful. Orquesta el almacenamiento, las peticiones HTTP y la delegación de tareas pesadas a procesos en segundo plano.
- **Módulos de Inteligencia Artificial:** Conjunto de herramientas y modelos locales (independientes de servicios en la nube) que ejecutan la transcripción de audio, detección de rostros, análisis de emociones y procesamiento de lenguaje natural (NLP).

## 4.2. Herramientas y Tecnologías Utilizadas

Para garantizar un rendimiento óptimo en el procesamiento de video y la inferencia de modelos, se seleccionó un conjunto de tecnologías robustas. En la Figura 4.1 se detalla el diagrama de interconexión de los componentes tecnológicos y el flujo de datos entre el cliente, el servidor y los motores locales de inteligencia artificial.

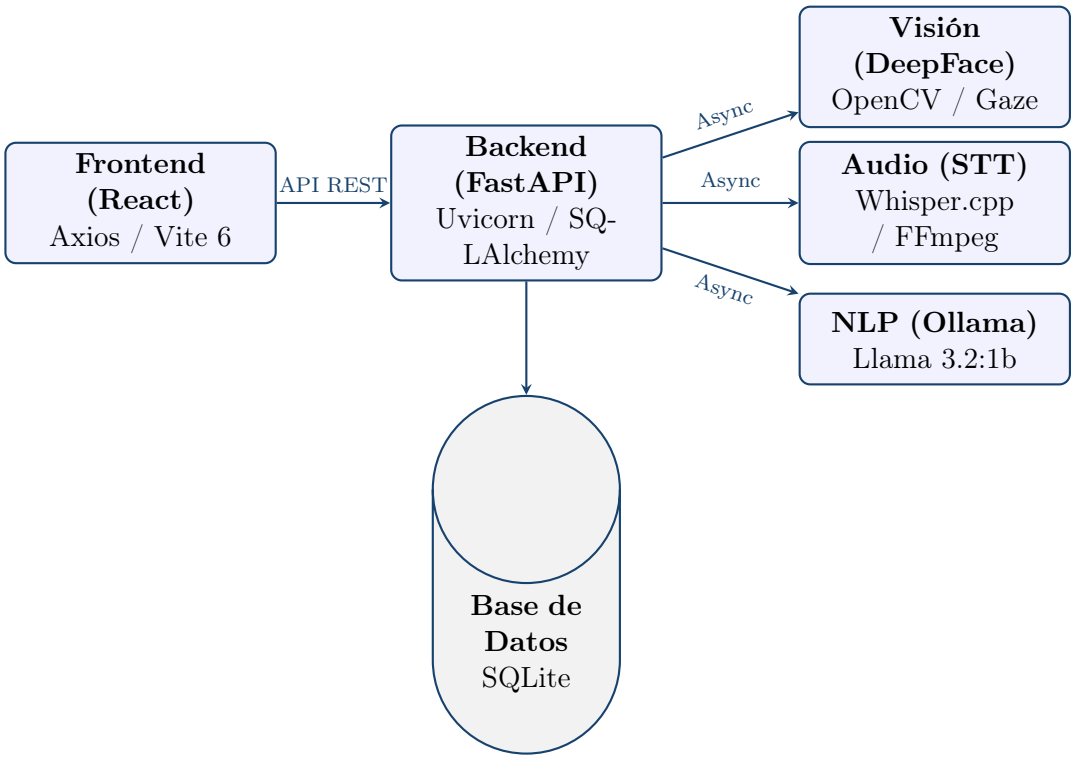


Figura 4.1: Diagrama de interconexión de componentes, tecnologías y flujos de datos del sistema.

**Backend y API** - FastAPI [14] y Uvicorn [15]

El núcleo del servidor está construido con FastAPI [14], un framework web para Python que permite el desarrollo de APIs RESTful. Su elección se basa en su soporte nativo para operaciones asíncronas y validación automática de datos. Todas las rutas se agrupan bajo el prefijo `/api` para mantener una clara separación con los recursos estáticos. Para desplegar y ejecutar esta API, se utiliza Uvicorn [15], un servidor web ASGI que maneja las peticiones concurrentes, lo cual es vital cuando se lanzan múltiples tareas en segundo plano.

**Base de Datos** - SQLite [13] y SQLAlchemy 2.x [16]

La persistencia de la información se gestiona mediante SQLite [13], un motor de base de datos relacional ideal para manejar las relaciones entre cursos, sesiones y resultados de análisis. Para interactuar con esta base de datos desde el código Python, se emplea

SQLAlchemy 2.x [16], un ORM. Esta herramienta permite manipular las tablas y registros como objetos de Python, facilitando operaciones CRUD de manera segura y protegiendo al sistema de vulnerabilidades como la inyección SQL.

**Frontend** - React 19 [17], Vite 6 [18] y Axios [19]

La interfaz de usuario es una aplicación de página única (SPA) desarrollada con React 19 [17], lo que permite construir componentes visuales dinámicos e interactivos (como los tableros de control y gráficas) que reaccionan a los cambios de estado. Como herramienta de empaquetado y construcción se seleccionó Vite 6 [18], debido a su rapidez en el entorno de desarrollo. Finalmente, la comunicación entre esta interfaz y el backend se realiza mediante Axios [19], un cliente HTTP basado en promesas que facilita el consumo de la API REST, el manejo de errores y la carga de archivos multimedia.

**Procesamiento Multimedia** - OpenCV [4] (cv2) y FFmpeg [20]

El preprocesamiento de los videos subidos a la plataforma recae en dos herramientas estándar. FFmpeg [20] se ejecuta mediante subprocesos para separar y decodificar los flujos multimedia, teniendo como tarea principal extraer la pista de audio del video original y convertirla a un formato estándar de 16 kHz necesario para los modelos de voz. Por otro lado, OpenCV [4] (a través de su librería cv2 en Python) se encarga del procesamiento visual, abriendo el archivo de video y extrayendo fotogramas precisos a intervalos regulares (basados en los FPS configurados) para su posterior análisis.

**Reconocimiento de Voz (STT)** - Whisper.cpp (whisper-cli [12])

Para la transcripción del audio a texto, el sistema implementa Whisper.cpp, una adaptación en C/C++ del modelo original de OpenAI. Se utiliza mediante su interfaz de línea de comandos (whisper-cli [12]) junto con un modelo en formato GGML. Esta elección técnica permite realizar una transcripción de precisión de forma estrictamente local y offline, evitando los costos y problemas de latencia asociados con el uso de APIs en la nube, y garantizando la privacidad de las conversaciones de la clase.

La implementación local de este módulo se fundamenta en su alta eficiencia y bajo consumo de recursos, facilitando su despliegue autónomo en el entorno de ejecución.

**Visión Artificial** - DeepFace [8]

El análisis del comportamiento no verbal de los estudiantes se delega a DeepFace, un framework híbrido de análisis facial para Python. Una vez que OpenCV [4] ha extraído los fotogramas del video, DeepFace se encarga de localizar los rostros en cada imagen y ejecutar una clasificación sobre ellos para identificar emociones básicas. Al integrar múltiples modelos de estado del arte en una sola librería, simplifica el proceso de extracción de métricas afectivas en cada iteración del video.

**Análisis de Sentimiento (LLM)** - Ollama [21] y el modelo llama3.2:1b [22]

Para evaluar el clima emocional del discurso, se integró procesamiento de lenguaje natural utilizando Ollama [21], una herramienta que permite ejecutar grandes modelos de lenguaje (LLM) de forma local. Específicamente, se utiliza el modelo llama3.2:1b [22], seleccionado por ser ligero pero lo suficientemente capaz para entender el contexto. El sistema toma la transcripción generada por Whisper, la divide en lotes lógicos y le solicita al LLM que analice la actitud o el sentimiento predominante en el texto, devolviendo métricas estructuradas que alimentan las estadísticas del profesor.

Para este módulo se evaluó inicialmente el uso de modelos tradicionales de procesamiento de lenguaje natural de librerías como spaCy o StanfordNLP; no obstante, los resultados no cumplieron con los estándares de calidad requeridos y el procesamiento de párrafos completos presentaba una latencia elevada.

**Reconocimiento Óptico (OCR)** - ocrmac [23] y thefuzz [24]

Como característica adicional para el control de asistencia o participación, se utiliza reconocimiento óptico de caracteres para leer los nombres de los estudiantes directamente de la pantalla de la videollamada. ocrmac se emplea para aprovechar los motores de visión nativos del sistema operativo y extraer texto de los fotogramas rápidamente. Dado que el texto extraído puede contener errores de lectura o tipográficos, se utiliza la librería thefuzz. Esta aplica algoritmos de distancia de Levenshtein (coincidencia difusa) para comparar el texto detectado con los nombres reales de los participantes, asegurando una identificación precisa incluso ante pequeñas imperfecciones en el OCR.

La incorporación de este módulo se decidió tras evaluar alternativas como OCRmyPDF y Tesseract, las cuales mostraron baja precisión de lectura o bien tiempos de procesamiento excesivamente altos para los fotogramas provistos.

## 4.3. Modelos de Detección Soportados

El sistema cuenta con una arquitectura modular que permite la selección de distintos motores de detección facial, adaptándose dinámicamente a las necesidades de precisión, las condiciones del entorno y los recursos computacionales (CPU o GPU) disponibles. Los modelos soportados en la aplicación para el análisis de fotogramas son los siguientes:

**OpenCV [11]:** Es la opción más ligera y veloz del sistema, basada tradicionalmente en clasificadores en cascada (Haar Cascades). Es ideal para ejecuciones en tiempo real sobre hardware con recursos muy limitados, ya que su costo computacional es mínimo. Sin embargo, al depender de patrones de contraste predefinidos, su precisión disminuye frente a rostros con oclusiones, variaciones en la iluminación o cuando los estudiantes no están mirando directamente a la cámara.

**MediaPipe [25]:** Desarrollado por Google, este framework (impulsado internamente por el modelo BlazeFace) ofrece el equilibrio entre robustez algorítmica y velocidad de pro-

cesamiento. Está optimizado para inferencia rápida, logrando detectar múltiples rostros con latencias mínimas incluso sin tarjetas gráficas dedicadas. Es la elección recomendada como estándar, ya que maneja de forma eficiente el seguimiento en escenarios académicos típicos.

**MTCNN [26]:** Este enfoque divide el problema de la detección en tres redes neuronales que trabajan de forma consecutiva (P-Net, R-Net y O-Net). Cada red va filtrando el ruido y refinando las cajas delimitadoras hasta detectar con exactitud los puntos de referencia faciales (ojos, nariz y comisuras de la boca). Esta estructura en cascada le otorga una alta precisión para alinear rostros e identificar a múltiples sujetos en un mismo cuadro, siendo muy útil cuando hay ruido visual en el fondo.

**YOLOv8 [27]:** Representa el estado del arte en arquitecturas de detección de un solo paso (single-shot detector). A diferencia de otros modelos, YOLOv8 analiza la imagen completa en una única pasada a través de su red neuronal convolucional, lo que le permite entender el contexto global del fotograma. Ofrece un rendimiento excepcional y es eficaz para detectar rostros a diferentes escalas sin sacrificar la velocidad, especialmente cuando el sistema cuenta con aceleración por hardware.

**RetinaFace [28]:** Es el modelo de aprendizaje profundo más avanzado e integrado en la plataforma. Utiliza una arquitectura de extracción de características que le permite realizar una localización facial precisa. Su mayor fortaleza radica en su resistencia ante rostros con oclusiones; es capaz de identificar a los estudiantes en condiciones extremas como contraluz severo, rostros parcialmente cubiertos o posiciones de cabeza inclinadas, garantizando la integridad de los datos.

## 4.4. Gestión de Datos y Endpoints Principales

El flujo de información se gestiona mediante rutas específicas en el backend. Los recursos principales operan bajo un modelo CRUD utilizando un sistema de borrado lógico para preservar la integridad histórica de los análisis.

### 4.4.1. Especificación de Endpoints de la API

La arquitectura de comunicación entre el cliente y el servidor se sustenta en una API RESTful, centralizada bajo el prefijo `/api`. Esta interfaz expone un total de diecisiete endpoints estructurados en tres módulos principales, encargados de orquestar desde la persistencia de datos hasta el procesamiento asíncrono con modelos de inteligencia artificial.

### 4.4.2. Módulo de Gestión de Cursos

Este apartado administra el ciclo de vida de las asignaturas mediante cuatro rutas principales.

La consulta del catálogo académico se realiza a través de `GET /api/courses/`, que recupera las entidades activas excluyendo aquellas marcadas con borrado lógico (soft delete).

Para inspecciones detalladas, `GET /api/courses/{course_id}` provee los metadatos de una entidad específica.

La creación de un nuevo curso ocurre mediante `POST /api/courses/create`, un servicio que persiste la información en la base de datos y aprovisiona automáticamente la estructura de directorios físicos (`uploads/<nombre>_<año>_<periodo>/`) necesaria para alojar el material multimedia.

Finalmente, la remoción de asignaturas se gestiona con `DELETE /api/courses/delete/{course_id}`, actualizando los estados de visibilidad para ocultar el registro sin comprometer la integridad de los archivos almacenados.

### 4.4.3. Módulo de Procesamiento de Sesiones

Este módulo constituye el núcleo operativo del sistema y se compone de ocho endpoints.

La navegación histórica se soporta con `GET /api/sessions/course/{course_id}` y `GET /api/sessions/{session_id}`, que permiten listar y detallar las clases respectivamente.

La ingesta de datos multimedia se maneja a través de `POST /api/sessions/create`, un servicio multipart que recibe el video, lo almacena, genera subcarpetas internas (images, video, audio) y actualiza el esquema relacional.

Su contraparte, `DELETE /api/sessions/delete/{session_id}`, aplica un borrado lógico a la sesión.

Para la orquestación de los motores de inteligencia artificial, `POST /api/sessions/analyze/{session_id}` delega a un trabajador en segundo plano (background worker) el análisis multimodal general, que incluye la extracción de audio, la transcripción con Whisper, el muestreo de fotogramas y el procesamiento OCR.

Paralelamente, el reconocimiento de expresiones faciales se dispara mediante `POST /api/sessions/analyze-frames/{session_id}` empleando DeepFace.

Dado el alto costo computacional de este proceso iterativo, el sistema expone `POST /api/sessions/stop-analyze-frames/{session_id}` para interrumpir la tarea de forma controlada modificando la bandera interna de ejecución.

Para mantener la interfaz actualizada, `GET /api/sessions/{session_id}/analysis-info` actúa como un mecanismo de polling, devolviendo al cliente el progreso en tiempo real y el estado de las tareas asíncronas.

### 4.4.4. Módulo de Resultados y Sentimiento

Este segmento consolida las métricas extraídas a través de cinco rutas especializadas.

La lectura y escritura directa de los objetos JSON generados se realizan con `GET /api/results/{result_id}` y `POST /api/results/create`.

Por otro lado, el procesamiento de lenguaje natural (NLP) se gestiona mediante tres rutas que interactúan con el modelo de lenguaje local a través de Ollama [21].

El endpoint `POST /api/results/{session_id}/analyze-sentiment` automatiza la evaluación de la transcripción completa, segmentándola en lotes (batches) de texto y persistiendo el sentimiento global en la base de datos.

Para interfaces que requieren mayor granularidad interactiva, `POST /api/results/{session_id}/analyze-sentiment-batch` procesa un único lote a la vez, permitiendo al frontend visualizar barras de progreso precisas.

Al concluir esta iteración, el cliente invoca `POST /api/results/{session_id}/save-sentiment` para consolidar y guardar de manera definitiva el vector completo de emociones de la sesión.

## 4.5. Pipelines de Procesamiento e Inteligencia Artificial

Las tareas de procesamiento se ejecutan en segundo plano a través de subprocesos asíncronos (background tasks) gestionados por FastAPI, lo que evita bloqueos en el hilo del servidor principal. La arquitectura del sistema está diseñada para ser asíncrona: una vez iniciado el análisis, el docente puede cerrar la aplicación web, apagar el navegador o desconectarse de la sesión y regresar posteriormente sin interrumpir el proceso de cómputo en el backend. Los resultados de cada fotograma y lote de texto se guardan de forma incremental en la base de datos relacional (SQLite), lo que garantiza la re-entrada del sistema, la persistencia en tiempo real y la disponibilidad del reporte parcial o total en cualquier momento.

Los tiempos de procesamiento típicos estimados en un procesador de propósito general (CPU de gama estándar) son los siguientes:

- **Reconocimiento de voz (Whisper.cpp):** Aproximadamente 0.15x de la duración

del video (por ejemplo, transcribir una sesión de 30 minutos requiere cerca de 4.5 minutos).

- **Análisis de sentimiento verbal (Ollama con Llama 3.2:1b):** Entre 2 y 3 minutos para el análisis y etiquetado de la sesión completa.
- **Análisis afectivo visual (DeepFace):** Aproximadamente 1.5x de la duración del video de la clase empleando una tasa de muestreo de fotogramas del 50 % (por ejemplo, analizar una sesión de 30 minutos toma cerca de 45 minutos).

El sistema se divide en tres pipelines operativos:

### 4.5.1. Pipeline de Análisis General Multimodal

Este flujo se activa mediante el endpoint `POST /api/sessions/analyze/{session_id}` y realiza el despiece del archivo multimedia:

- **Extracción de Audio:** Mediante subprocesos de sistema, FFmpeg convierte la pista de audio del video original a un formato estándar (pcm s16le, 16 kHz, mono), optimizado para el reconocimiento de voz.
- **Transcripción Automática:** El sistema verifica la existencia del binario de Whisper.cpp. Al ejecutarse, procesa el audio y genera un archivo de salida en formato JSON que es interpretado y concatenado por el backend para generar un registro textual en la base de datos (Results).
- **Extracción de Fotogramas:** Utilizando OpenCV [4], el sistema avanza a través del video basándose en una tasa de fotogramas por segundo (FPS) configurable por el docente. Las imágenes extraídas se guardan de forma secuencial en disco.
- **Identificación de Participantes:** Si la configuración lo requiere, se aplica un muestreo sobre los fotogramas guardados. Mediante ocrmac [23] y coincidencias difusas (librería thefuzz [24]), se identifican los nombres en pantalla para generar métricas de asistencia o participación.

### 4.5.2. Pipeline de Análisis de Emociones en Video

Ejecutado a través del endpoint `POST /api/sessions/analyze-frames/{session_id}`, este proceso se dedica a la lectura del comportamiento no verbal:

- **Muestreo:** Se indexan las imágenes generadas en el paso anterior y se aplica un filtro según el parámetro de muestreo (sample rate) para reducir la carga de procesamiento sin perder representatividad.

- **Inferencia Facial:** Cada imagen seleccionada se procesa mediante OpenCV [4] y se envía al framework DeepFace [8]. Este último realiza el tracking geométrico de los rostros en pantalla y ejecuta la clasificación de emociones.
- **Persistencia Incremental:** Los datos extraídos no se guardan al final, sino que se acumulan progresivamente en la base de datos, lo que permite observar la evolución del estado de la clase en tiempo real. Este bucle es interrumpible a petición del usuario.

### 4.5.3. Pipeline de Análisis de Sentimiento en Texto

Una vez completada la transcripción, se evalúa el clima general de la conversación oral mediante el endpoint `POST /api/results/{session_id}/analyze-sentiment`:

- **Segmentación (Batching):** El backend toma el texto completo de la transcripción y lo divide en bloques lógicos basándose en la puntuación y un límite aproximado de palabras, agrupándolos para optimizar la ventana de contexto del modelo de lenguaje.
- **Procesamiento mediante LLM:** Cada lote es enviado al servidor local de Ollama [21] (modelo llama3.2:1b [22]) junto con un prompt diseñado específicamente para detectar variaciones anímicas en contextos educativos.
- **Consolidación:** El modelo devuelve los resultados en formato estructurado, los cuales se ensamblan para proporcionar un puntaje de sentimiento global de la sesión, almacenándose permanentemente para su visualización en los tableros de control.

## 4.6. Diseño de Interfaz y Experiencia de Usuario (UI/UX)

La interacción del docente con el sistema se centraliza en una Aplicación de Página Única (SPA) desarrollada con React 19 [17] y optimizada mediante Vite 6 [18]. El diseño de la interfaz fue concebido bajo principios de usabilidad enfocados en reducir la carga cognitiva del profesor, asegurando que procesos computacionales complejos sean accesibles a través de un entorno visual responsivo e intuitivo.

### 4.6.1. Arquitectura de Navegación y Flujo de Trabajo

La navegación del sistema, gestionada internamente a través de react-router-dom, obedece a una estructura jerárquica adaptada al contexto educativo. El usuario interactúa con un flujo que va de lo general a lo particular: inicia en un panel principal donde selecciona un

curso activo, transita hacia el catálogo de clases históricas de dicha materia, y finalmente puede instanciar una nueva sesión subiendo un archivo de video. Al tratarse de una SPA, las transiciones entre estas vistas ocurren sin recargas completas de la página, ofreciendo una experiencia fluida e ininterrumpida.

### 4.6.2. Gestión de Tareas Asíncronas y Retroalimentación Visual

Dado que el procesamiento multimedia y la ejecución de modelos de IA requieren tiempos prolongados, el sistema implementa un manejo de la retroalimentación visual para evitar la frustración del usuario. Al lanzar un análisis, la interfaz gráfica no se bloquea; en su lugar, el cliente inicia un proceso de consultas periódicas (polling) hacia los endpoints del servidor para monitorear el progreso de las tareas en segundo plano. La interfaz traduce esta información en componentes dinámicos como barras de progreso e indicadores de estado por lote, brindando al docente transparencia sobre qué fase del análisis se está ejecutando (transcripción, evaluación facial o análisis de sentimiento).

### 4.6.3. Visualización de Datos y Tableros de Control (Dashboards)

La culminación del flujo de usuario ocurre en la pantalla de métricas (`/sessions/{sessionId}/results`). Esta vista transforma los datos crudos en formato JSON (generados por los pipelines del backend) en información pedagógica. El tablero de control se encarga de renderizar múltiples componentes analíticos, tales como:

- **Líneas de tiempo interactivas:** Permiten observar la evolución y las fluctuaciones del estado emocional del grupo minuto a minuto a lo largo del video.
- **Gráficos de distribución o resumen:** Sintetizan el clima emocional global y la métrica de sentimiento general de la conversación extraída por el modelo de lenguaje (LLM).
- **Tablas y listas de asistencia:** Alimentadas por el procesamiento de OCR, las cuales facilitan la revisión de qué participantes estuvieron presentes en pantalla.

De este modo, la interfaz actúa como un puente entre la ingeniería de datos subyacente y la toma de decisiones educativas, permitiendo al profesor identificar patrones de atención y ajustar sus estrategias didácticas sin necesidad de poseer conocimientos técnicos avanzados.

# 5. PRUEBAS Y VALIDACIÓN DEL SISTEMA

Este capítulo describe la fase de validación del sistema, detallando los escenarios de prueba ejecutados y los resultados obtenidos tras la implementación de los modelos de inteligencia artificial. El objetivo principal de esta etapa consiste en comprobar la estabilidad técnica del software y su capacidad para generar información útil, precisa y estructurada para el docente a partir de las grabaciones de las clases.

## 5.1. Escenarios de Prueba y Gestión Continua de Sesiones

Para asegurar la robustez de la arquitectura del sistema, las pruebas iniciales se enfocan en la carga y administración de datos. La validación de la plataforma se realizó utilizando una muestra de datos consistente en 5 sesiones de clase grabadas en video. Dichas sesiones contaron con las siguientes características:

- **Tamaño del grupo evaluado:** Entre 3 y 6 estudiantes activos por sesión, lo que permitió validar la capacidad del algoritmo de reconocimiento facial sobre un grupo variable de participantes simultáneos en pantalla.
- **Duración de las sesiones:** Rango de entre 30 minutos y 1 hora de duración por sesión, sumando un volumen acumulado de aproximadamente 3.5 horas de grabación multimedia.

Se valida con éxito la capacidad de la plataforma para gestionar múltiples sesiones de forma concurrente, permitiendo subir diversos archivos de video correspondientes a diferentes fechas y horarios.

El sistema demuestra una correcta persistencia de datos y organización de archivos. La base de datos relacional y el sistema de carpetas mantienen la integridad estructural (separando videos, audios e imágenes extraídas) sin generar conflictos de sobreescritura, lo que permite comprobar que la plataforma es escalable y apta para llevar el seguimiento longitudinal de un curso completo durante un periodo académico.

## 5.2. Validación Funcional de los Módulos de Inteligencia Artificial

Una vez validada la infraestructura de carga, se procede a evaluar el rendimiento de los pipelines de procesamiento sobre videos reales de sesiones de clase, obteniendo los siguientes resultados:

- **Transcripción y Sincronización Multimedia:** El modelo Whisper.cpp realiza la transcripción del audio extraído de las sesiones de manera íntegra. En la interfaz se implementa un módulo de reproducción de audio sincronizado con la transcripción, permitiendo al docente auditar el texto generado, hacer clic en fragmentos específicos y escuchar el momento exacto de la clase, validando la fiabilidad del reconocimiento de voz.
- **Análisis de Sentimiento por Fotograma (No Verbal):** El motor de visión artificial basado en DeepFace se ejecuta de manera iterativa. El sistema permite aislar los rostros y clasificar sentimientos por fotograma. Esto facilita el mapeo de la reacción visual de los estudiantes a lo largo del tiempo, demostrando que el bucle asíncrono en segundo plano funciona de manera estable y puede ser interrumpido sin comprometer la integridad de los datos acumulados.
- **Análisis de Sentimiento de la Transcripción (Verbal):** El procesamiento de lenguaje natural ejecutado a través del modelo local de Ollama [21] realiza el análisis de los lotes de texto transcrito, identificando la polaridad y el clima emocional general de la conversación sostenida durante la sesión.

## 5.3. Análisis de Resultados y Generación de Estadísticas

El resultado de las pruebas de extracción de datos permite consolidar el tablero de control (Dashboard), respondiendo a los objetivos planteados en la metodología. El sistema procesa los datos crudos para entregar los siguientes indicadores clave:

- **Mapeo y Detalle del Tiempo en Clase:** El sistema provee una segmentación temporal de la sesión. Se correlacionan los tiempos de duración del video con los momentos de intervención oral y las fluctuaciones emocionales, permitiendo al profesor identificar en qué minuto exacto (por ejemplo, durante la explicación de un tema complejo) ocurrieron cambios significativos en la dinámica del grupo.
- **Distribución Porcentual de Sentimientos:** A partir de la agregación de los datos obtenidos tanto de los fotogramas (DeepFace) como del texto (Ollama [21]), el sistema genera un desglose estadístico. La interfaz renderiza gráficas que muestran el porcentaje total de sentimientos predominantes durante la clase (ej. 60 % neutralidad, 25 % atención/felicidad, 15 % confusión/estrés). Esta cuantificación transforma datos abstractos en una métrica directa de receptividad.

## 5.4. Conclusión de la Fase de Pruebas

Los resultados confirman que el sistema cumple con los objetivos trazados. La integración entre la carga de múltiples sesiones, la extracción de emociones fotograma a fotograma, el análisis de sentimiento del discurso y la sincronización audiovisual demuestra que la plataforma funciona como una herramienta analítica estructurada. La información estadística generada aporta retroalimentación que facilita al docente la comprensión del nivel de atención y el estado anímico del alumnado, validando la pertinencia y viabilidad del proyecto.

# 6. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO

En este capítulo se presentan las capturas de pantalla de la interfaz de la aplicación, detallando el funcionamiento de cada una de las secciones y componentes del sistema desarrollado.

## 6.1. Interfaces de la Aplicación

### 6.1.1. Creación de Curso

En esta sección se ilustra el formulario para el registro de un nuevo curso académico. La interfaz, mostrada en la Figura 6.1, requiere ingresar el nombre de la asignatura, descripción, periodo escolar, año, días de la semana, horarios de inicio y fin, y el profesor titular. Una vez creado el curso, la plataforma actualiza el listado global de asignaturas activas, como se observa en la Figura 6.2.

### 6.1.2. Creación de Sesión

Para instanciar una nueva clase, se debe seleccionar el curso correspondiente y completar el formulario de registro de sesión que se ilustra en la Figura 6.3. La interfaz permite asociar la fecha de la sesión y cargar el archivo de video en formato MP4. Una vez procesada la carga, la sesión aparece en el catálogo histórico (Figura 6.4). Al seleccionar una sesión específica, se habilita la visualización del reproductor y los controles de análisis (Figura 6.5).

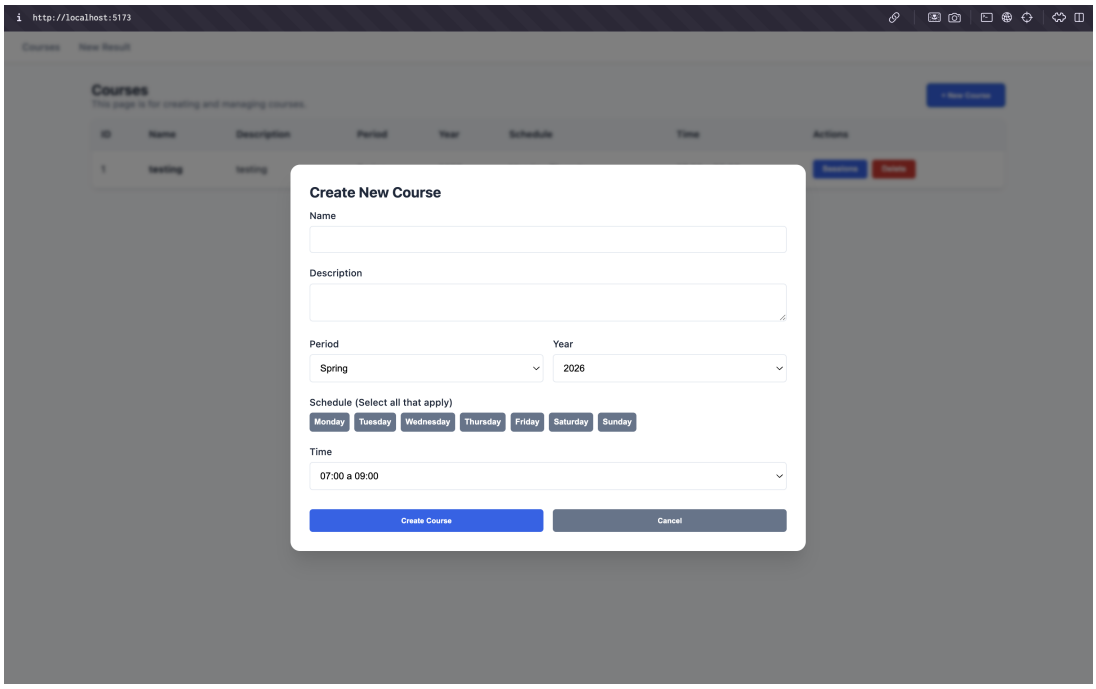


Figura 6.1: Formulario para el registro de un nuevo curso académico.

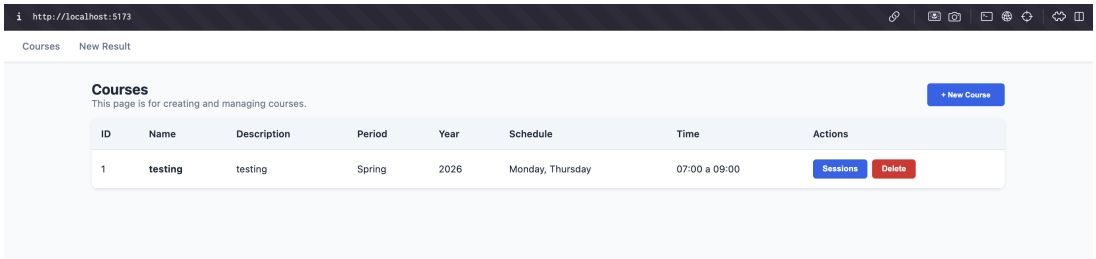


Figura 6.2: Listado de cursos registrados en la plataforma.

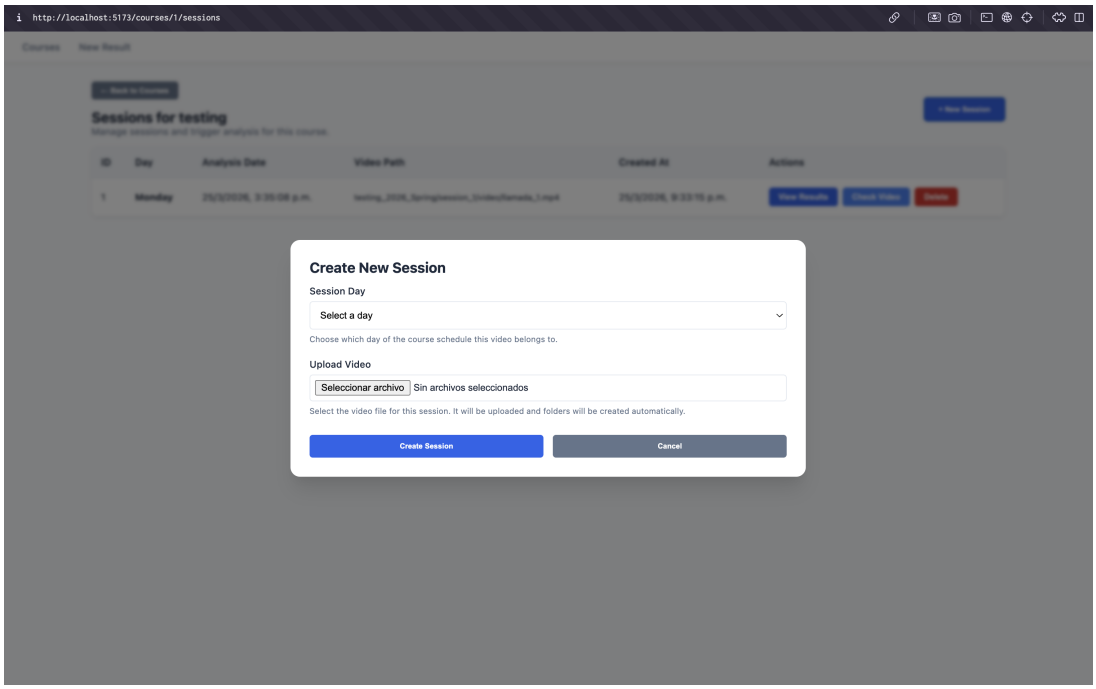


Figura 6.3: Formulario para la carga de video y creación de sesión.

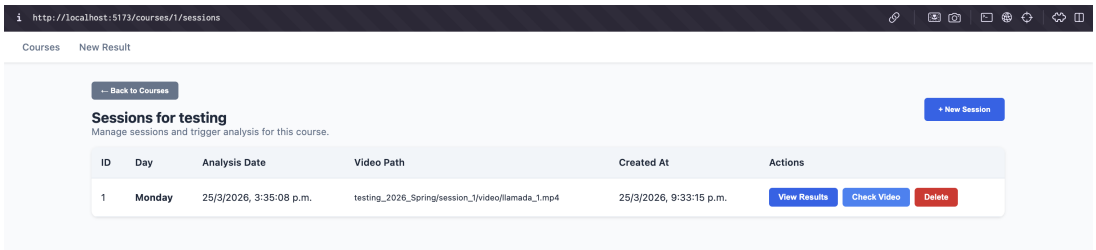


Figura 6.4: Listado de sesiones de clase asociadas a un curso.

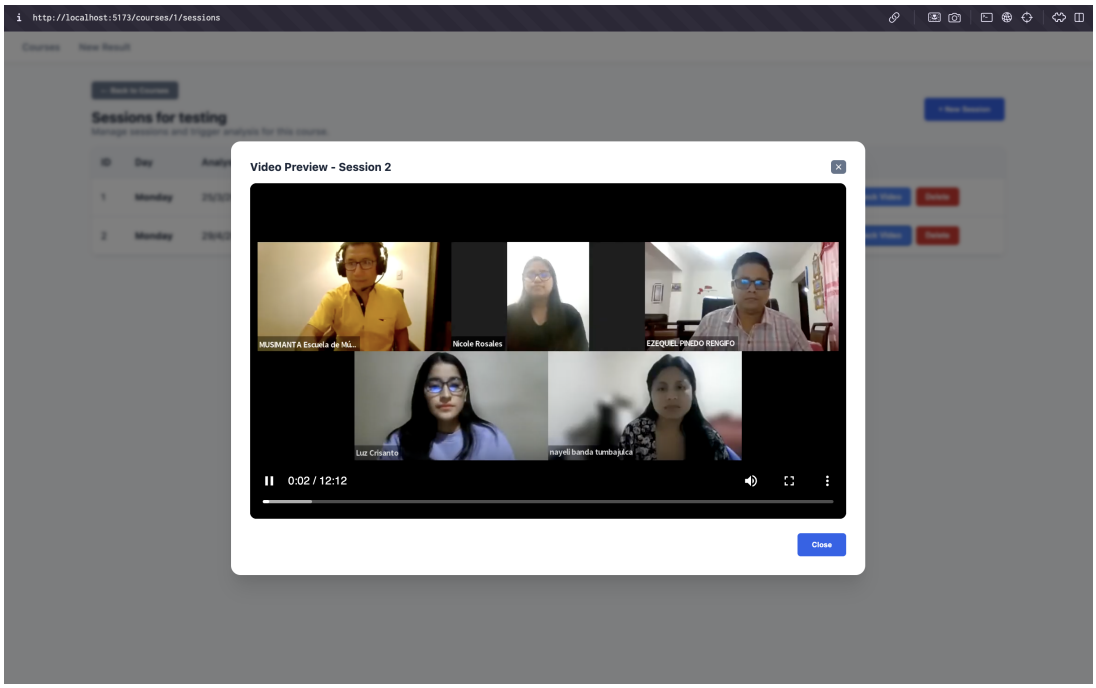


Figura 6.5: Interfaz de visualización de video y control de análisis de una sesión.

### 6.1.3. Primer Análisis (Ingesta y Procesamiento General)

A continuación se describen las fases del primer análisis de la sesión. El proceso requiere configurar en primer lugar la tasa de extracción de fotogramas por segundo y, posteriormente, el porcentaje de frames que se utilizará para el análisis de asistencia OCR, como se ilustra en la Figura 6.6. Dicha parametrización es relevante ya que reduce los tiempos de cómputo al omitir fotogramas redundantes.

Los pasos secuenciales ejecutados por el servidor se detallan visualmente en las siguientes interfaces:

- **Extracción de audio:** Separación de la pista de sonido usando FFmpeg (Figura 6.7).
- **Transcripción de voz:** Ejecución local del modelo Whisper.cpp (Figura 6.8).
- **Segmentación de fotogramas:** Extracción de imágenes a disco mediante OpenCV (Figura 6.9).
- **Detección OCR y Asistencia:** Extracción del texto en pantalla para identificar los nombres de los estudiantes activos, aplicando distancia de Levenshtein para resolver variaciones tipográficas (Figuras 6.10 y 6.11).

Al finalizar esta fase, los indicadores de estado cambian en la interfaz, desactivándose las opciones de configuración e ingesta primaria para dar paso a los tableros analíticos, como

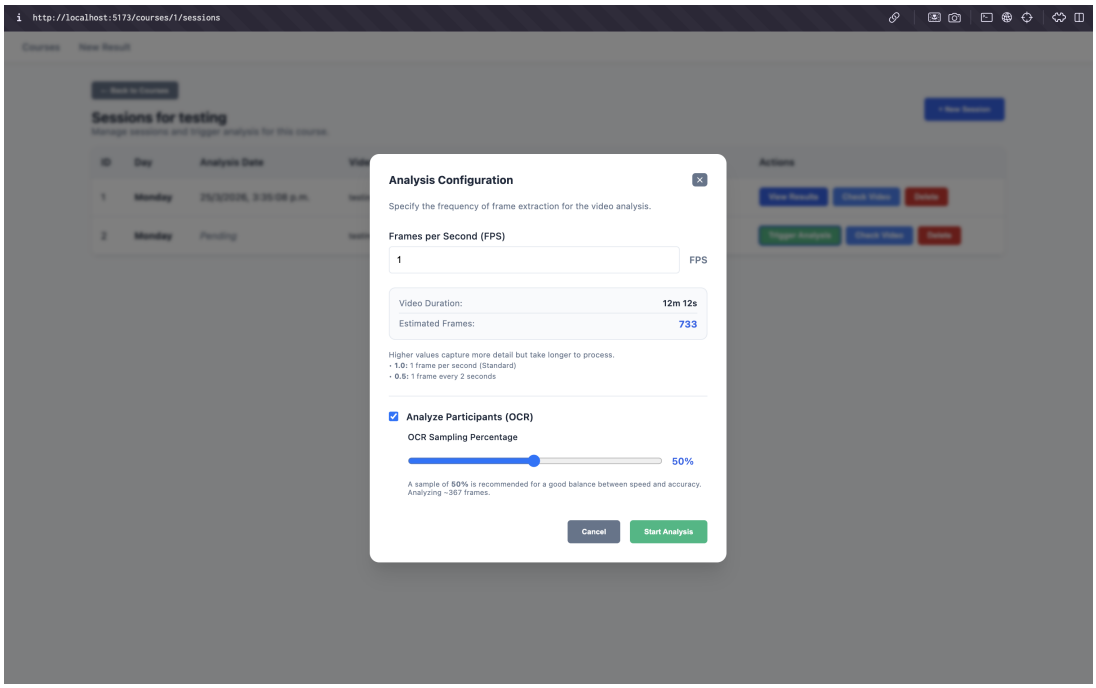


Figura 6.6: Configuración de parámetros para la ingesta y extracción de fotogramas.

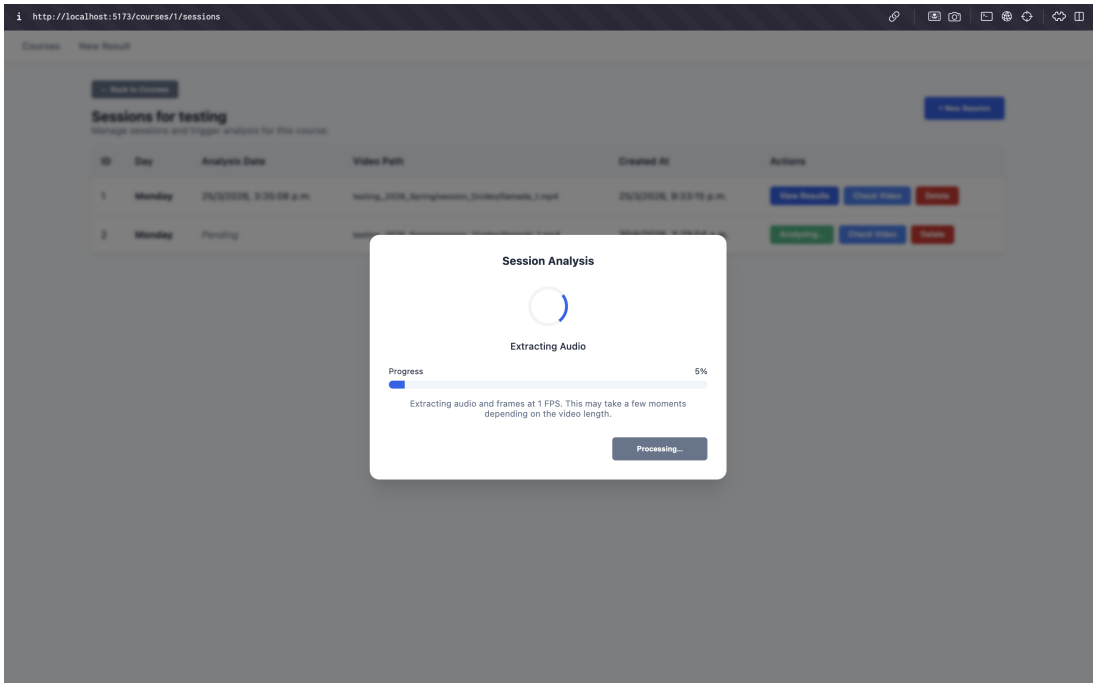


Figura 6.7: Ejecución del pipeline de audio: extracción de la pista WAV mediante FFmpeg.

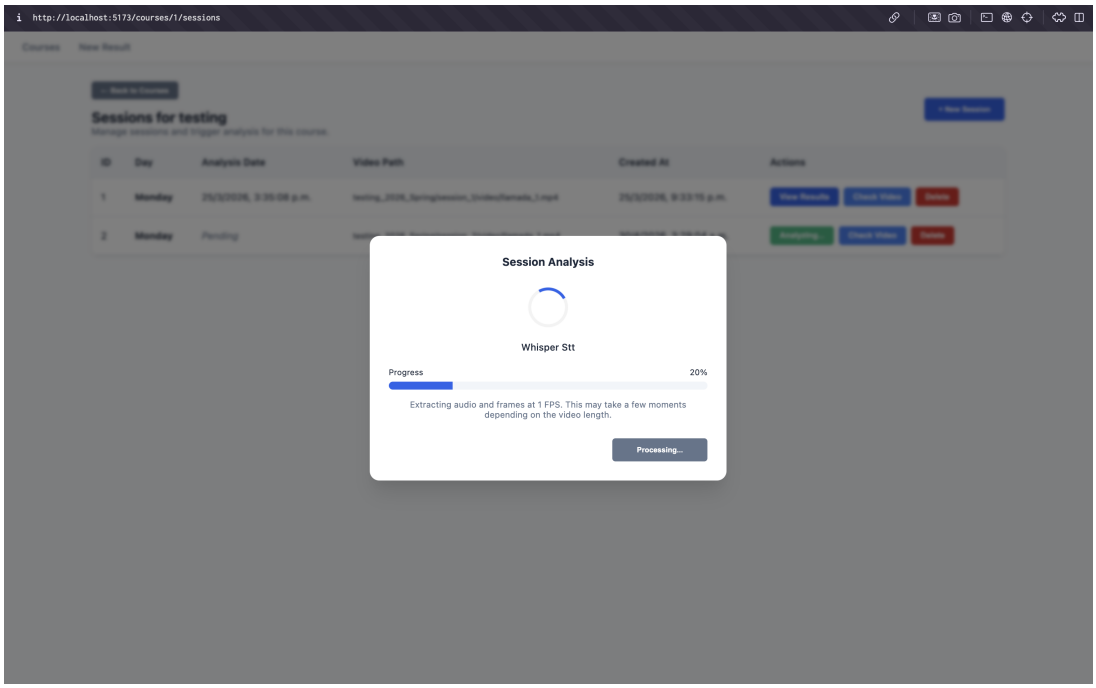


Figura 6.8: Ejecución de transcripción automática local utilizando Whisper.cpp.

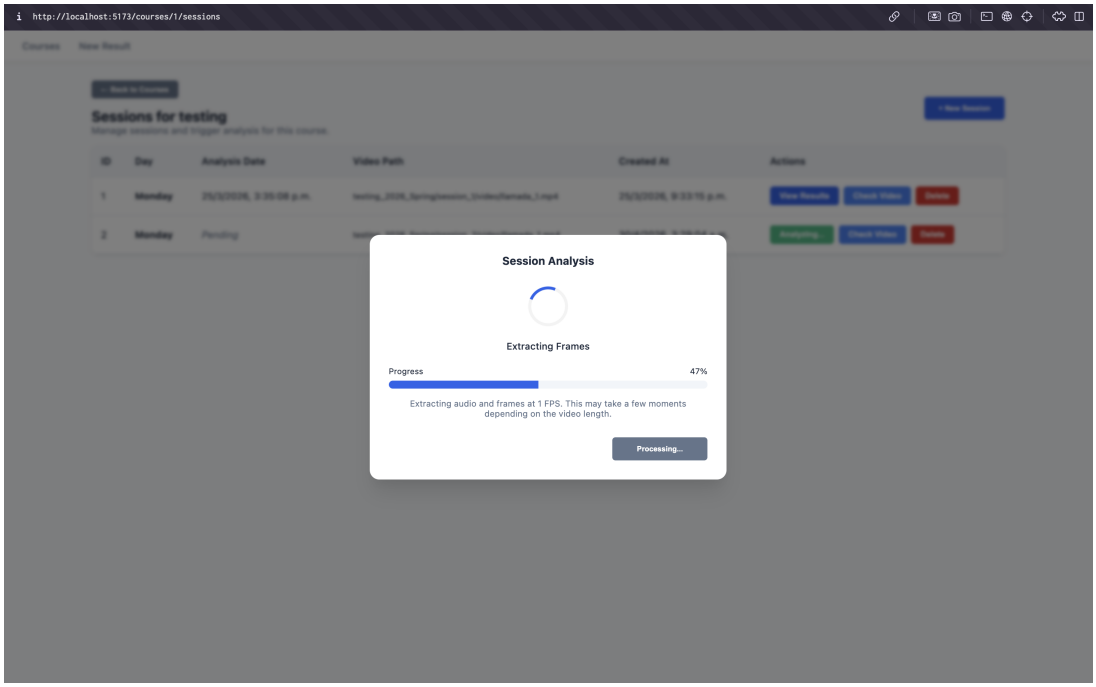


Figura 6.9: Segmentación y guardado de fotogramas del video en almacenamiento local.

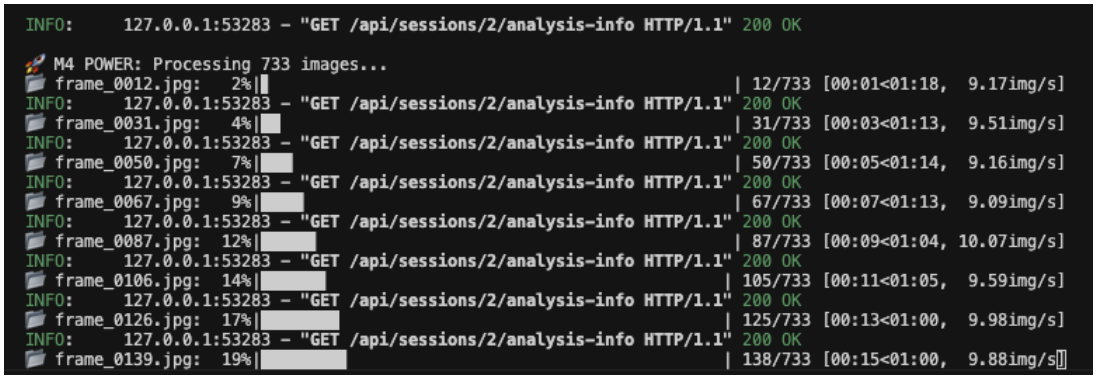


Figura 6.10: Descripción del proceso de agrupamiento de nombres y coincidencia difusa (Fuzzy Matching).

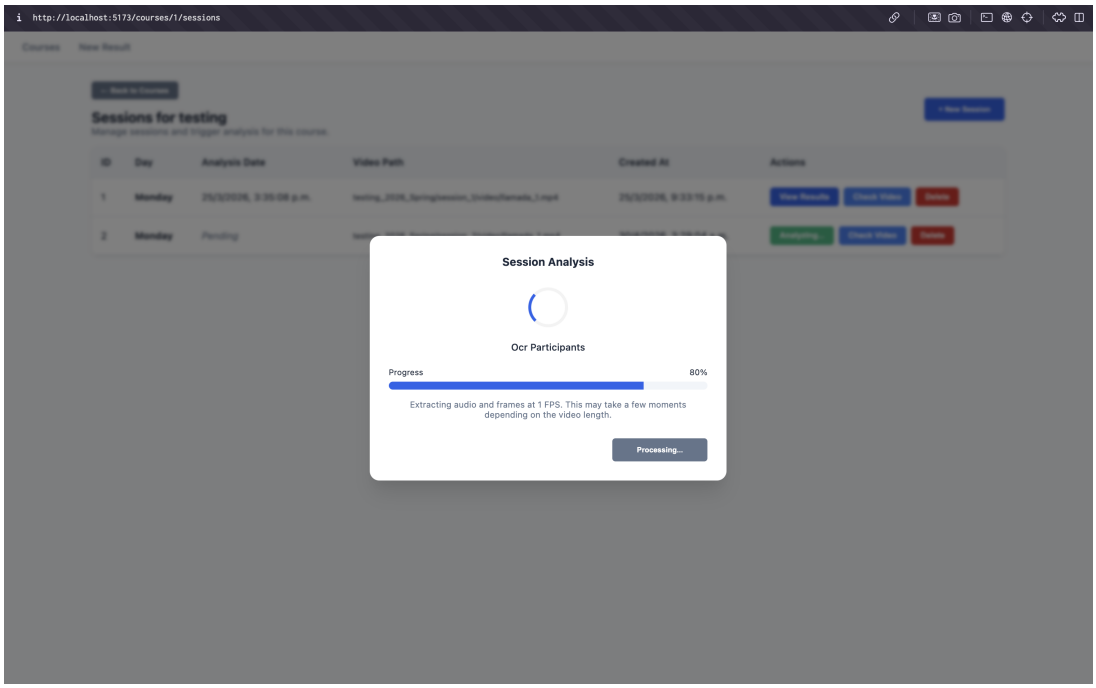


Figura 6.11: Procesamiento OCR en ejecución sobre los fotogramas muestreados.

se muestra en la Figura 6.12. El acceso a las métricas consolidadas se realiza mediante el botón de resultados (Figura 6.14).

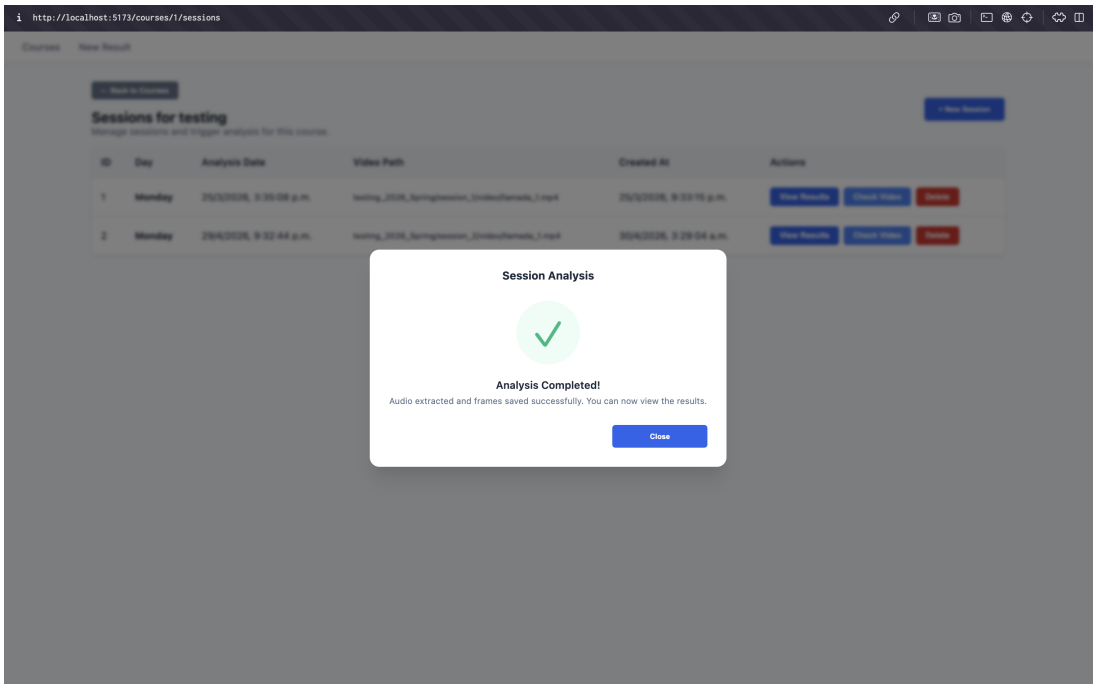


Figura 6.12: Confirmación en interfaz de la finalización del pipeline general.

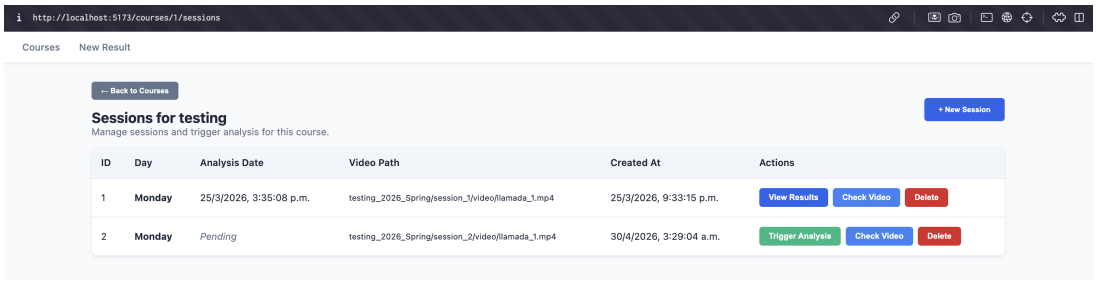


Figura 6.13: Actualización de controles de la sesión al concluir el preprocesamiento.

### 6.1.4. Panel de Resultados

La interfaz de resultados de la sesión está estructurada en un diseño de dos columnas, separando la reproducción multimedia de las métricas de análisis. Esta vista consolida la información del preprocesamiento y se compone de los siguientes elementos:

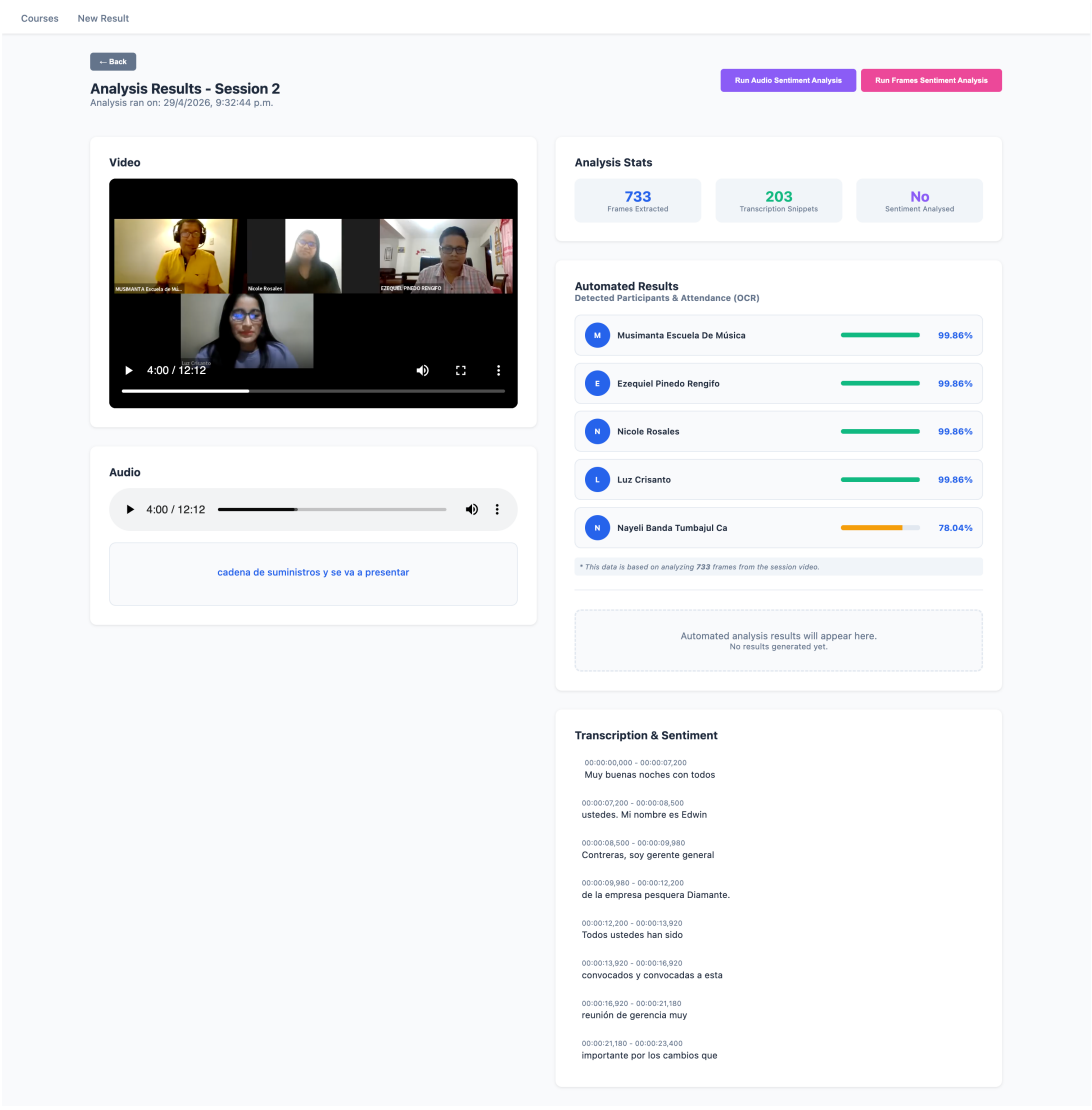


Figura 6.14: Acceso directo a la sección de reportes y visualización de resultados.

#### 6.1.4.1. Panel de Reproducción y Sincronización Multimedia (Columna Izquierda)

- **Video y Audio Desacoplados:** Reproductores independientes para la pista de video y la pista de audio WAV extraída.
- **Subtitulado Dinámico:** Justo debajo del reproductor de audio se ubica un componente de texto interactivo que resalta la frase en reproducción actual. Esto actúa como un sistema de subtítulos sincronizado, facilitando la auditoría del texto transcrito.

#### 6.1.4.2. Estadísticas Globales del Análisis (Analysis Stats)

En la parte superior derecha, se resume el volumen de datos procesados mediante métricas cuantitativas:

- **Frames Extracted:** Indica el volumen de imágenes guardadas en el almacenamiento.
- **Transcription Snippets:** Cantidad de oraciones segmentadas.
- **Sentiment Analysed:** Indicador binario sobre si se ha corrido el análisis emocional en la sesión actual.

#### 6.1.4.3. Reconocimiento de Asistencia mediante OCR (Automated Results)

El sistema muestra el catálogo de alumnos detectados mediante OCR junto con barras de progreso que indican su porcentaje de permanencia visual activa durante la clase (ej. 99.86 % para la mayoría de los usuarios, detectando ausencias relativas en casos puntuales de 78.04 %).

#### 6.1.4.4. Panel de Transcripción Estructurada (Transcription and Sentiment)

Lista detallada del discurso de la clase indexada con marcas de tiempo en formato de milisegundos (ej. 00:00:00,000 - 00:00:07,200), que sirve como base para segmentar el análisis de sentimiento secundario.

#### 6.1.4.5. Controles de Procesamiento Secundario

En la parte superior se exponen los botones de acción primarios (`Run Audio Sentiment Analysis` y `Run Frames Sentiment Analysis`). Estos controles coordinan la ejecución de

las tareas en segundo plano usando los modelos DeepFace y Ollama, una vez que la data de base está disponible.

### 6.1.5. Análisis de Sentimiento del Audio y la Transcripción

Para comprender el tono emocional del discurso, el sistema emplea procesamiento de lenguaje natural local mediante Ollama y el modelo Llama 3.2:1b [22]. El flujo funciona de manera iterativa por lotes (batches) para optimizar el consumo de memoria en el servidor. El modelo analiza el contexto de cada frase y clasifica la emoción del texto, proveyendo además una breve explicación textual del sentimiento detectado.

Los resultados del análisis se guardan en dos niveles:

- **Clasificación Granular:** Cada bloque de la transcripción es etiquetado con su emoción correspondiente (entusiasmo, felicidad, sorpresa), trazando una línea de tiempo emocional.
- **Consolidación Global:** Promedio porcentual agregado de las emociones verbales detectadas en toda la sesión.

En las Figuras 6.15 y 6.16 se muestra la ejecución de esta etapa y su posterior visualización.

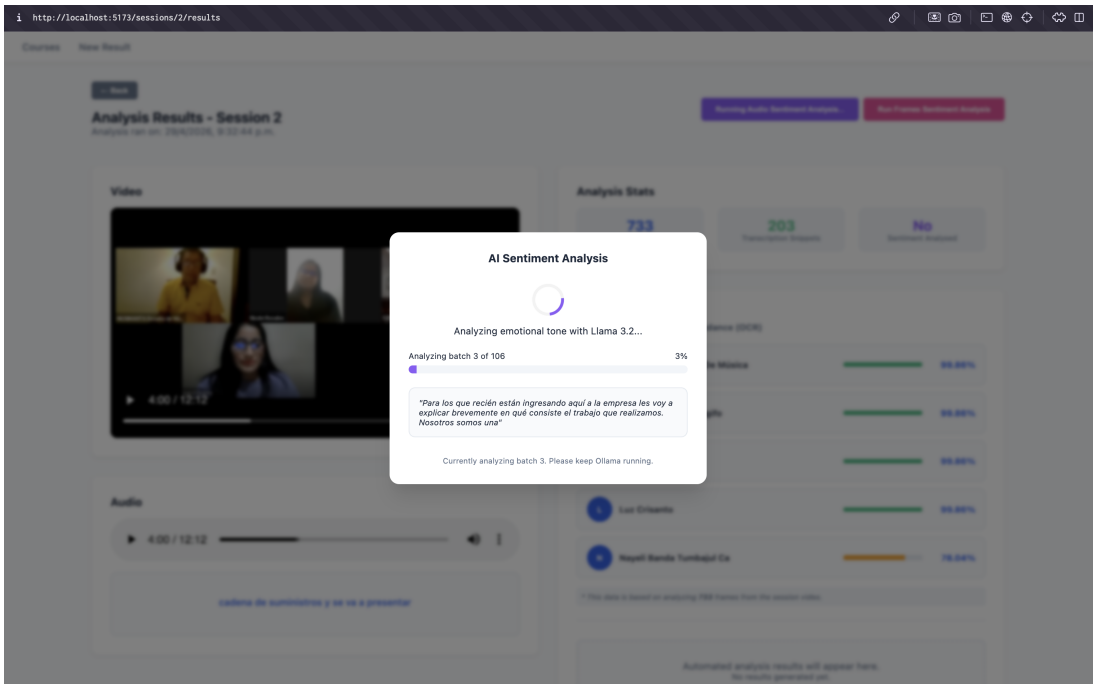


Figura 6.15: Ejecución por lotes del análisis de sentimiento sobre la transcripción.

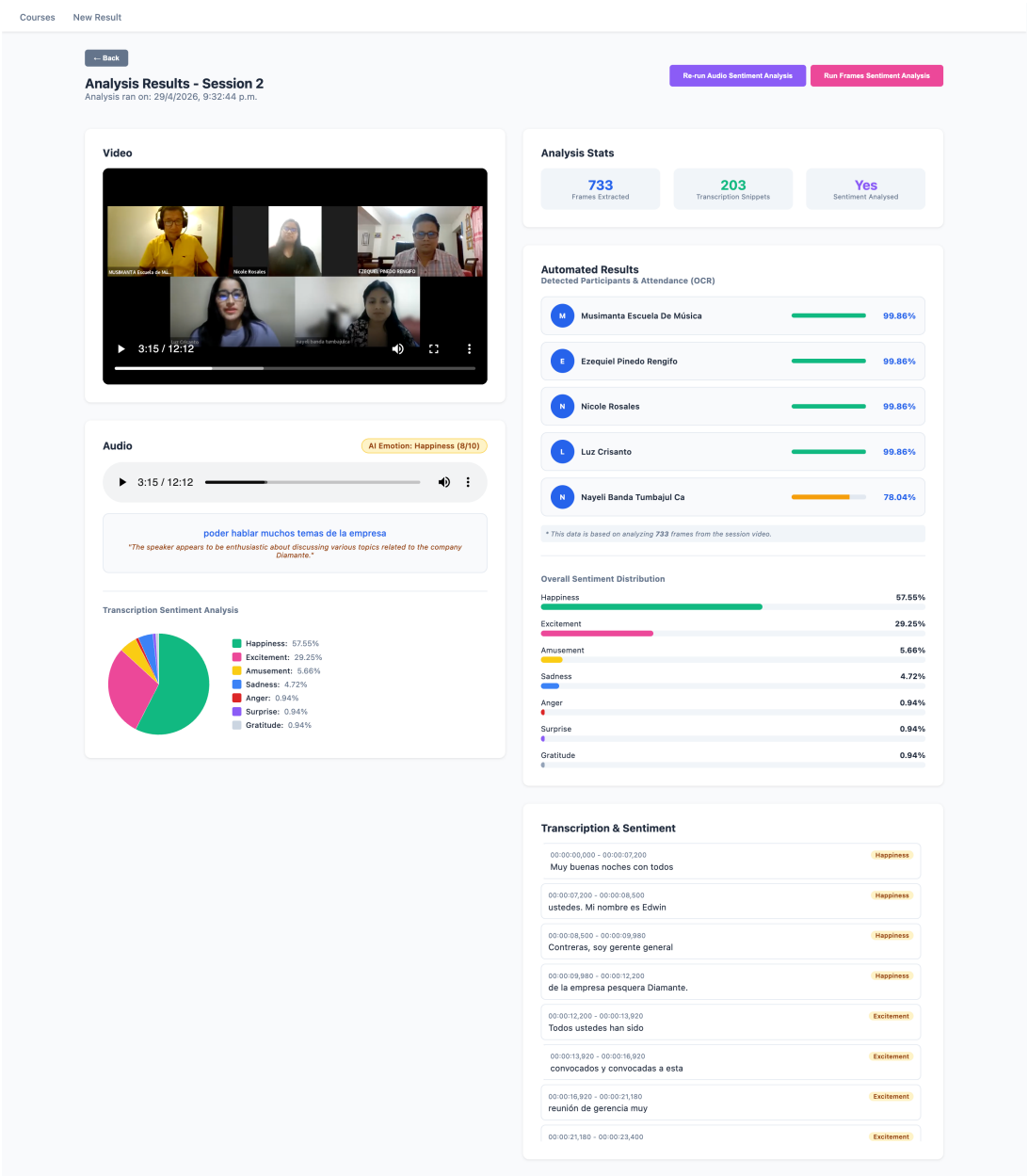


Figura 6.16: Visualización de resultados consolidados del análisis afectivo verbal.

### 6.1.6. Análisis de Sentimiento en Fotogramas (Frames)

La fase final del procesamiento corresponde al análisis de comportamiento no verbal en video. Se evalúan las expresiones faciales mediante DeepFace y el modelo detector configurado por el usuario.

Dado el costo computacional de la inferencia visual, se dispone de una barra de control (Figura 6.17) para definir el porcentaje de fotogramas a analizar (ej. 50 %). Esto optimiza la velocidad del proceso manteniendo la validez estadística de la muestra. La interfaz expone una barra de progreso y la opción de detener el análisis anticipadamente, como se ilustra en la Figura 6.18.

El tiempo de procesamiento requerido para este análisis en CPU estándar es de aproximadamente 1.5x de la duración del video (por ejemplo, 45 minutos para una sesión de 30 minutos). En contraste, el reconocimiento de voz (STT) toma cerca de 0.15x de la duración del video y el análisis semántico textual mediante el LLM local requiere de 2 a 3 minutos adicionales.

Debido a que el procesamiento se ejecuta en segundo plano a través de tareas asíncronas en el servidor web, no es necesario mantener la interfaz web abierta. El docente puede iniciar la tarea de análisis, cerrar la pestaña del navegador o apagar el computador y regresar más tarde para verificar el reporte. Los datos se persisten de forma incremental en la base de datos (SQLite) fotograma a fotograma; por lo tanto, la información procesada hasta el momento de una pausa o interrupción involuntaria se conserva y se muestra al instante al volver a ingresar al panel.

**Resultados del Análisis Visual:** Una vez concluido el análisis, se habilita un visor interactivo que presenta la información consolidada (Figura 6.19):

- **Mapeo y Etiquetado Visual:** Recuadros de detección espacial en el video indicando la emoción predominante (Neutral, Sad, Fear) y el porcentaje de confianza.
- **Navegación Secuencial:** Paginador para auditar los fotogramas individuales.
- **Estructura JSON:** Sección con el objeto JSON crudo arrojado por el modelo, detallando coordenadas, puntos fiduciales y el vector probabilístico de emociones.

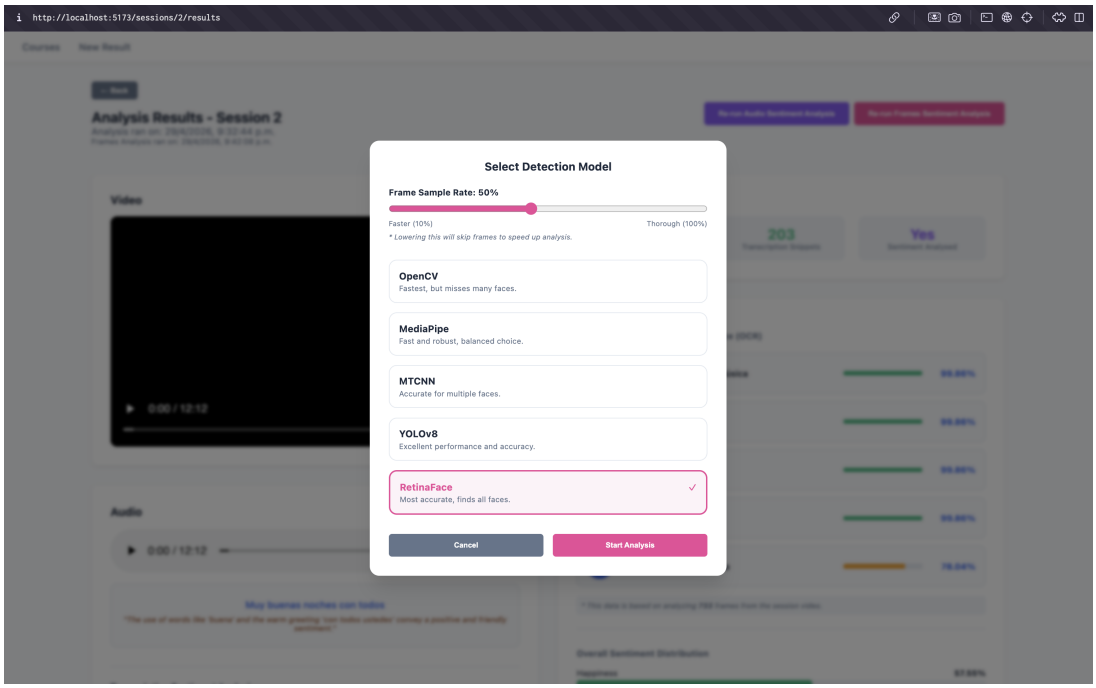


Figura 6.17: Configuración del muestreo de fotogramas para el procesamiento visual.

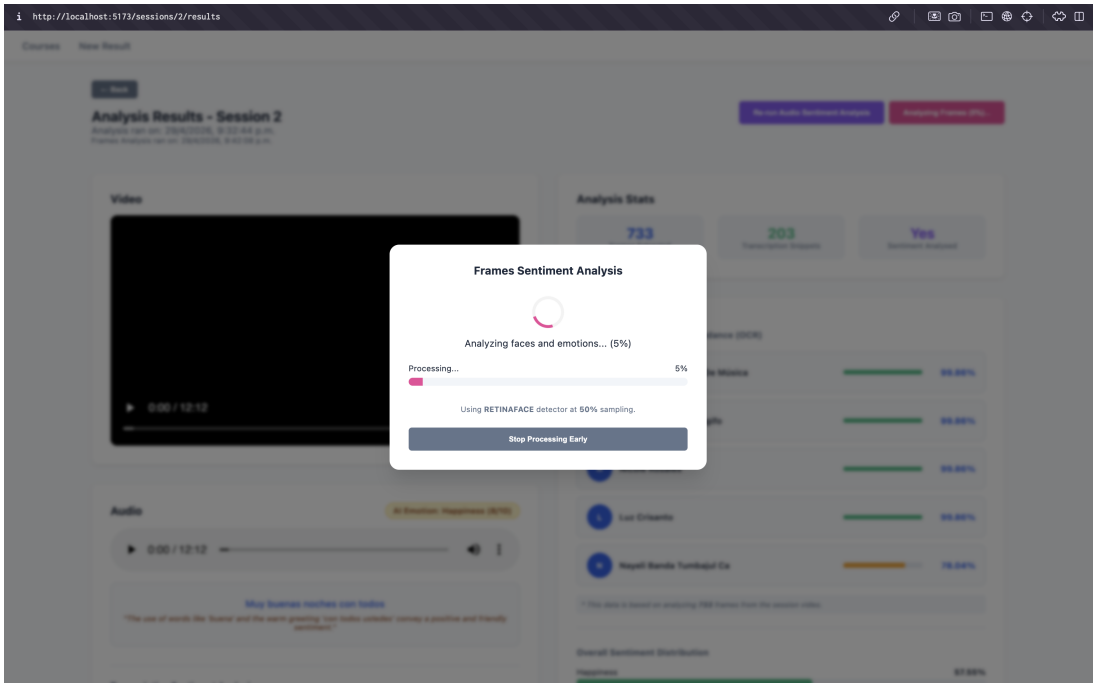


Figura 6.18: Monitoreo y progreso en vivo de la inferencia facial de la sesión.

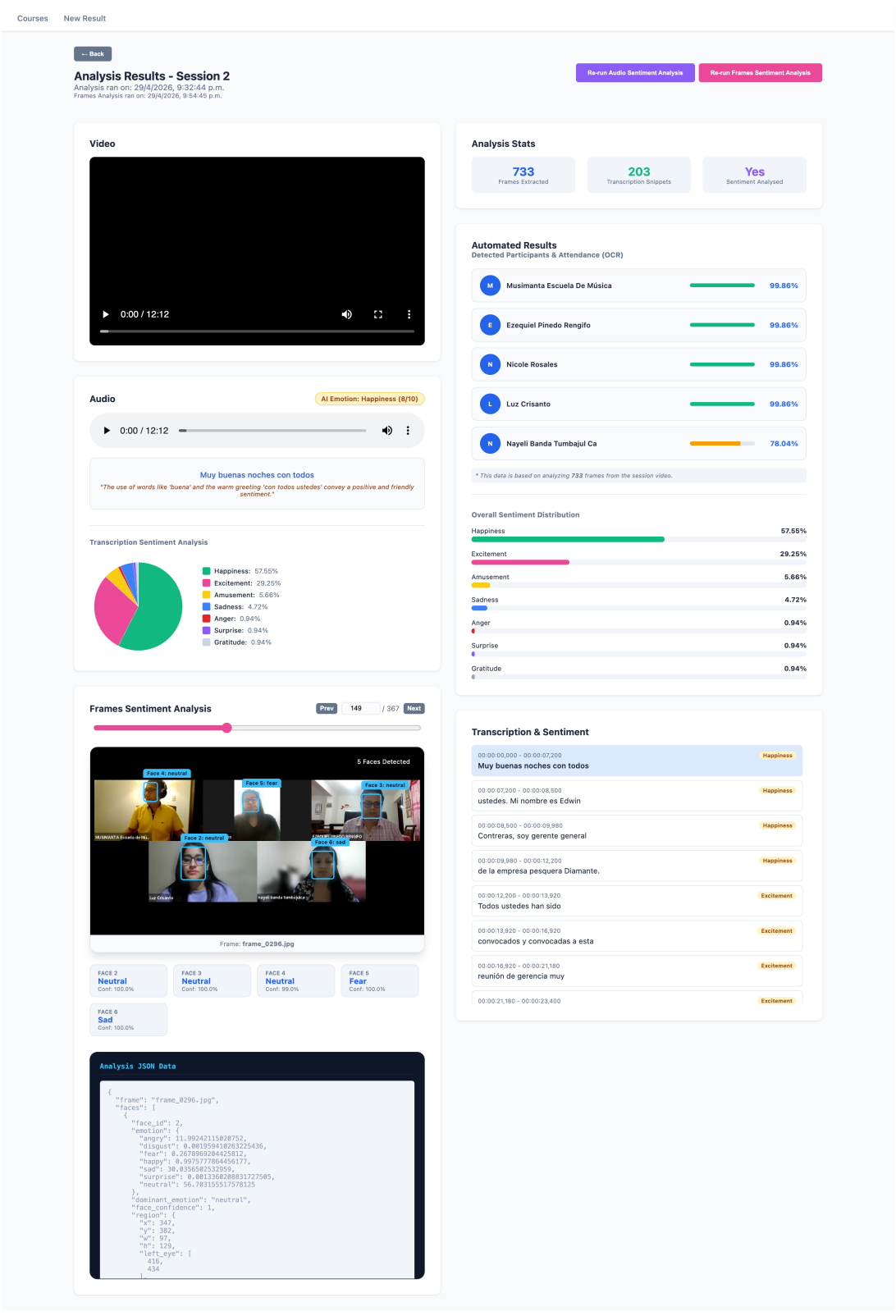


Figura 6.19: Resultados consolidados de la inferencia de emociones faciales sobre fotogramas.

# 7. CONCLUSIONES Y TRABAJO FUTURO

## 7.1. Conclusiones

El desarrollo e implementación del Sistema de Análisis de Comportamiento Estudiantil permite evaluar el potencial de integrar modelos de Inteligencia Artificial multimodal en el ámbito educativo. A partir de los resultados experimentales de validación, se presentan las siguientes conclusiones fundamentales estructuradas en función del cumplimiento de los objetivos planteados.

### 7.1.1. Cumplimiento del Objetivo General

Se consolida de manera exitosa una plataforma web que integra flujos de visión artificial, procesamiento de lenguaje natural y reconocimiento de caracteres para el análisis automático de clases grabadas en video. La solución desarrollada demuestra la viabilidad de transformar registros multimedia en conjuntos de datos estructurados para la toma de decisiones pedagógicas, resolviendo el problema de opacidad en la dinámica estudiantil de las sesiones virtuales y proporcionando indicadores claros tanto cuantitativos como cualitativos sobre el desempeño en el aula.

### 7.1.2. Cumplimiento de los Objetivos Específicos

**1. Seguimiento de Estudiantes:**

- *Registro de atención visual:* Fomentado mediante la integración del estimador de

dirección visual (GazeTracking), que registra los lapsos en los que los alumnos mantienen el enfoque en la pantalla frente a desvíos laterales o inferiores.

- *Control de asistencias y permanencia:* Automatizado mediante el procesamiento OCR aplicado a los fotogramas, logrando calcular el porcentaje exacto de permanencia en pantalla (asistencia continua) de forma no invasiva, superando las limitaciones del pase de lista convencional.
- *Análisis de conducta grupal:* Habilitado al promediar de forma temporal e incremental la distribución de emociones faciales colectivas detectadas por DeepFace, detectando oclusiones o ausencias en cámara.

**2. Retroalimentación Docente:**

- *Estadísticas de exposición:* El pipeline de análisis del habla cuantifica la participación oral y permite al docente auditar la proporción de tiempo dedicado a monólogos expositivos versus interacción dialogada.
- *Identificación de dinámicas reactivas:* La correlación temporal entre marcas de tiempo de video, transcripción y emociones grupales facilita al profesor identificar qué momentos específicos o temáticas generaron mayor interés o confusión en el grupo.
- *Clima emocional verbal:* Resuelto mediante el procesamiento por lotes del modelo local Llama 3.2:1b a través de Ollama, clasificando de manera semántica el sentimiento del discurso verbal del docente.

### 7.1.3. Aportaciones Técnicas y Viabilidad Operativa

**Optimización del Costo Computacional:** El procesamiento iterativo de video demanda recursos de hardware considerables. La metodología de muestreo de fotogramas (Frame Sample Rate) implementada en el sistema permite equilibrar la carga del procesador con la calidad analítica requerida. Al procesar únicamente fracciones muestreadas de la sesión (ej. 50 % de los fotogramas), se aceleran los tiempos de respuesta del análisis sin alterar la tendencia estadística del comportamiento visual del grupo.

**Privacidad y Procesamiento Local:** La ejecución estrictamente local y offline de los modelos (Whisper.cpp para transcripción, DeepFace para inferencia visual y Ollama para análisis semántico) constituye un factor crítico. Esta arquitectura garantiza el cumplimiento de las normativas de protección de datos en el entorno educativo, evitando el tráfico de imágenes o audio de los estudiantes hacia servidores en la nube y reduciendo los costos operativos del sistema a cero.

**Eficacia del OCR en Pantalla:** El empleo del módulo de OCR complementado con algoritmos de distancia de Levenshtein (thefuzz) representa una alternativa fiable frente a

la dependencia de las interfaces de programación propietarias de plataformas como Zoom o Teams. Esto permite aplicar la herramienta de análisis sobre cualquier archivo de video estándar independientemente de la plataforma de videoconferencia de origen.

## 7.2. Trabajos futuros y recomendaciones

A partir de los resultados y de la arquitectura implementada en el sistema, se proponen las siguientes líneas de investigación y desarrollo:

**Evolución de la Representación Visual y Dashboards Analíticos:** Se sugiere transitar de gráficos de distribución agregados hacia la superposición dinámica de series temporales en el reproductor de video. Visualizar gráficos de líneas interactivos en la línea de tiempo permitiría al docente ubicar con precisión los minutos exactos donde ocurrieron picos de confusión o dispersión grupal, optimizando la retroalimentación didáctica.

**Explotación de Estructuras de Datos de Inferencia Facial:** Los objetos JSON crudos devueltos por el framework DeepFace contienen coordenadas espaciales y vectores de probabilidad que no son aprovechados en su totalidad. Se recomienda desarrollar algoritmos heurísticos que sinteticen estas variables en un indicador unificado de Engagement o Nivel de Compromiso del estudiante, combinando presencia, estabilidad visual y valencia emocional.

**Fusión Multimodal Afectiva:** Un trabajo futuro relevante consiste en la correlación cruzada de datos verbales (NLP) y no verbales (visión artificial). Cruzar el análisis semántico de la transcripción con la reacción facial predominante del alumnado en el mismo segmento temporal permitiría verificar si el discurso del docente logra el impacto o la recepción esperada en la audiencia.

**Módulos de Alerta en Tiempo Real:** Aprovechando la base técnica de subprocesos asíncronos y polling, es factible optimizar la velocidad de inferencia de modelos ligeros como MediaPipe para diseñar un asistente en vivo. Este notificaría al ponente durante la videollamada si la atención general cae por debajo de un umbral establecido, permitiendo aplicar dinámicas de reactivación antes de concluir la sesión.

# BIBLIOGRAFÍA


[1] A. Piedrahíta-Carvajal, P. A. Rodríguez-Marín, D. F. Terraza-Arciniegas, M. Amaya-Gómez, L. Duque-Muñoz y J. D. Martínez-Vargas, "Aplicación web para el análisis de emociones y atención de estudiantes," *TecnoLógicas*, vol. 24, n.º 51, e1821, mayo de 2021. DOI: 10.22430/22565337.1821

[2] E. J. Flores Masias, J. H. Livia Segovia, A. García Casique y M. E. Dávila Díaz, "Análisis de sentimientos con inteligencia artificial para mejorar el proceso enseñanza-aprendizaje en el aula virtual," *PUBLICACIONES*, vol. 53, n.º 2, págs. 185-216, ene. de 2023. DOI: 10.30827/publicaciones.v53i2.26825

[3] J. Garres Díaz, J. Herrera Fernández, A. L. Albujer Brotons, M. Caballero Campos y T. Morales De Luna, "Análisis de la adaptación de la docencia virtual universitaria durante la COVID-19," *Revista de Innovación y Buenas Prácticas Docentes*, vol. 10, n.º 2, págs. 131-157, oct. de 2021. DOI: 10.21071/ripadoc.v10i2.13372

[4] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, págs. 120-123, 2000.

[5] A. Lame, *GazeTracking: Python library for eye tracking*, [GitHub repository], 2018. dirección: https://github.com/antoinelame/GazeTracking

[6] E. Vt, *OpenSeeFace: Robust Realtime Face and Facial Landmark Tracking on CPU with Unity Integration*, [GitHub repository], 2020. dirección: https://github.com/emilianavt/OpenSeeFace

[7] E. Churaev y A. Savchenko, *EmotiEffLib: Emotion Efficient Library for Emotion and Engagement Recognition*, [GitHub repository], 2023. dirección: https://github.com/av-savchenko/EmotiEffLib

[8] S. I. Serengil, *Deepface: A Lightweight Face Recognition and Facial Attribute Analysis Framework for Python*, [PyPI], 2020. dirección: https://pypi.org/project/deepface/

[9] M. Tiwari, *Facial Emotion Recognition using OpenCV and Deepface*, [GitHub repository], 2023. dirección: https://github.com/manish-9245/Facial-Emotion-Recognition-using-OpenCV-and-Deepface

[10] A. Balaji, *Emotion-detection: Real-time Facial Emotion Detection using deep learning*, [GitHub repository], 2019. dirección: https://github.com/atulapra/Emotion-detection

[11] P. Viola y M. Jones, "Rapid object detection using a boosted cascade of simple features," en *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, vol. 1, 2001, págs. I-511.

[12] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey e I. Sutskever, "Robust speech recognition via large-scale weak supervision," en *International Conference on Machine Learning*, PMLR, 2023, págs. 28 492-28 518.

[13] D. R. Hipp, *SQLite*, 2024. dirección: `https://www.sqlite.org`

[14] S. Ramírez, *FastAPI*, 2026. dirección: `https://fastapi.tiangolo.com`

[15] T. Christie, *Uvicorn: The lightning-fast ASGI server*, 2026. dirección: `https://www.uvicorn.org`

[16] M. Bayer, *SQLAlchemy*, 2024. dirección: `https://www.sqlalchemy.org`

[17] Meta Platforms, Inc., *React: The library for web and native user interfaces*, 2024. dirección: `https://react.dev`

[18] E. You, *Vite: Next Generation Frontend Tooling*, 2024. dirección: `https://vitejs.dev`

[19] M. Zabriskie, *Axios: Promise based HTTP client for the browser and node.js*, 2024. dirección: `https://github.com/axios/axios`

[20] FFmpeg Developers, *FFmpeg tool*, 2024. dirección: `https://ffmpeg.org`

[21] Ollama, *Ollama: Get up and running with large language models locally*, 2024. dirección: `https://github.com/ollama/ollama`

[22] AI@Meta, "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.

[23] M. Strauss, *ocrmac: Python wrapper for macOS native OCR*, 2026. dirección: `https://github.com/straussmaximilian/ocrmac`

[24] SeatGeek, *thefuzz: Fuzzy String Matching in Python*, 2026. dirección: `https://github.com/seatgeek/thefuzz`

[25] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran y M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *arXiv preprint arXiv:1907.05047*, 2019.

[26] K. Zhang, Z. Zhang, Z. Li e Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, n.º 10, págs. 1499-1503, 2016.

[27] G. Jocher, A. Chaurasia y J. Qiu, *Ultralytics YOLO*, ver. 8.0.0, 2023. dirección: `https://github.com/ultralytics/ultralytics`

[28] J. Deng, J. Guo, E. Ververas, I. Zafeiriou y S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," en *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, págs. 5203-5212.