



Tecnológico
de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

“Yo, como integrante de la comunidad estudiantil del Tecnológico de Monterrey, soy consciente de que la trampa y el engaño afectan mi dignidad como persona, mi aprendizaje y mi formación, por ello me comprometo a actuar honestamente, respetar y dar crédito al valor y esfuerzo con el que se elaboran las ideas propias, las de los compañeros y de los autores, así como asumir mi responsabilidad en la construcción de un ambiente de aprendizaje justo y confiable”

“Inteligencia artificial avanzada para la ciencia de datos II”

**Sistema de pase de lista y detección de participación
basado en Machine Learning y visión computacional**

Equipo 1:

Frida Cano Falcón A01752953

Jorge Javier Sosa Briseño A01749489

Guillermo Romeo Cepeda Medina A01284015

Daniel Saldaña Rodríguez A00829752

Profesores:

Ivan Mauricio Amaya Contreras

Hugo Terashima Marín

Edgar Covantes Osuna

Gerardo Ibarra Vázquez

Julio Antonio Juárez Jiménez

Fecha de entrega: 28 de noviembre de 2023

Índice

Índice	2
Abstract	3
1. Introducción	3
2. Fundamentos	6
3. Descripción general del proyecto	6
3.1. Descripción	6
3.2. Requerimientos	7
4. Propuesta de solución	8
4.1. Justificación del método seleccionado	8
4.2 Viabilidad de integración en el Tecnológico de Monterrey	11
4.3. Análisis FODA	12
5. Metodología	14
5.1. Plan de trabajo	14
Análisis de Negocios:	14
Portal de registro de asistencia:	15
Medir participación en el aula:	15
Evaluar algoritmos:	16
Portal de administración:	16
6. Validación e implementación	17
6.1 Validación del modelo de reconocimiento facial	17
6.1 Algoritmo de sistema de asistencia	20
6.2 Sistema de registro de participaciones	21
6.3 Base de datos	23
6.4 Página Web	26
6.5 Dashboard	32
7. Conclusiones	33
8. Referencias	34
9. ANEXOS	35
9.1 Link del repositorio:	35
9.2 Algoritmo de visión computacional	36

Abstract

Este estudio presenta la implementación de una página web destinada al registro y gestión de asistencia y participación en el aula. El objetivo central es desarrollar un sistema autónomo capaz de automatizar la toma de asistencia y rastrear la participación de los estudiantes, generando informes pertinentes para los profesores. Se emplearon la librería face_recognition para el reconocimiento facial y el modelo yolov8_pose de Ultralytics para estimar la pose de los estudiantes, complementado con una función para el seguimiento individual por frame. Los resultados demuestran un valor diferencial significativo para las comunidades educativas, la sociedad y los tomadores de decisiones. La automatización de la toma de asistencia no solo simplifica una tarea repetitiva y agotadora, sino que también libera valiosos minutos de clase para actividades más provechosas. La captura automática de la participación genera datos cruciales que, tras un análisis detallado, proporcionan información distintiva sobre las características de una clase en comparación con otras. En la evaluación del algoritmo face_recognition, entrenado con el extenso dataset "Faces in the Wild" (3,000,000 caras), se logró una impresionante precisión del 99.3%. Sin embargo, se observó una disminución en la precisión cuando se enfrenta a contextos dinámicos y frames no estáticos, señalando áreas de mejora para futuras implementaciones. Estos resultados destacan el potencial de esta solución tecnológica en el ámbito educativo y su relevancia para la toma de decisiones informadas.

1. Introducción

En el ámbito educativo, la toma de asistencia y el seguimiento de la participación de los estudiantes son procesos cruciales pero a menudo tediosos y propensos a errores. La utilización de tecnologías avanzadas puede ofrecer soluciones eficientes y precisas a estos desafíos. Este reporte presenta un enfoque innovador basado en visión computacional con aprendizaje profundo para desarrollar un sistema automatizado de toma de asistencia y seguimiento de participaciones en un entorno de aula. El objetivo principal de este proyecto es no solo optimizar el proceso de registro de asistencia, eliminando la pérdida de tiempo asociada, sino también proporcionar a los educadores una valiosa retroalimentación sobre la dinámica de sus clases para fomentar mejoras continuas.

La implementación de este sistema implica la utilización de algoritmos de aprendizaje profundo que permiten la identificación única de cada estudiante mediante análisis de imágenes capturadas en tiempo real durante las clases. Esto no solo simplifica la toma de asistencia, sino que también habilita la recopilación de datos detallados sobre la participación individual de los estudiantes. Además, se desarrollará un dashboard en una plataforma web que permitirá a los profesores acceder fácilmente a la información recopilada y recibir retroalimentación en tiempo real sobre la asistencia y la participación de los estudiantes.

La importancia de esta propuesta radica en la optimización de los recursos y la eliminación de la carga administrativa asociada a la toma de asistencia manual, permitiendo a los profesores centrarse más en la calidad de la enseñanza. Asimismo, al proporcionar una herramienta que ofrece retroalimentación detallada sobre la dinámica en clase, se abre la puerta a oportunidades de mejora continua y personalización del proceso educativo. Este proyecto busca no solo abordar la problemática de la pérdida de tiempo en la toma de asistencia, sino también contribuir al desarrollo de entornos educativos más eficientes y centrados en el estudiante.

2. Fundamentos

En esta sección, se proporciona una explicación a los fundamentos teóricos que respaldan el abordaje del problema en cuestión. Al investigar sobre el estado del arte, fue en la sección de la detección facial y de pose que se encontró una gran cantidad de proyectos existentes, de algunos de estos se tomó inspiración mientras que otros fueron utilizados para respaldar la validez de la implementación del proyecto.

En general estas soluciones existentes se enfocan en llevar un control de alumnos y profesores, es en ambiente teórico donde se encontraron proyectos desarrollados para ser utilizados en el aula.

Basandonos en el articulo “Student Attendance System in Crowded Classrooms using a Smartphone Camera”[3], en el algoritmo de detección de rostros se incorpora una verificación manual para garantizar la precisión de la detección, donde un operador humano

revisa las imágenes para confirmar y corregir posibles errores del algoritmo automático. Posteriormente, cada rostro detectado se somete a una función de descripción, que extrae características distintivas como la posición de los ojos, la forma de la nariz y la disposición de puntos clave. Estas descripciones son fundamentales para la comparación en la fase posterior del proceso, facilitando la coincidencia de rostros y contribuyendo a la precisión del sistema en la identificación facial

Durante este proyecto se seleccionó el algoritmo `dlib_face_recognition_resnet_model_v1`. Este modelo, basado en ResNet con 29 capas convolucionales, se destaca por su alta precisión en la identificación de rostros. Aborda la localización de rostros en imágenes y el emparejamiento con identidades específicas. Así mismo, utiliza técnicas como el algoritmo Viola-Jones o enfoques basados en aprendizaje profundo para identificar rostros dentro de una imagen. Esto asegura la identificación de regiones faciales, sentando las bases para el procesamiento posterior.

En cuanto a la detección de Pose, el algoritmo YOLOv8-Pose fue el seleccionado ya que se destaca por su rendimiento sobresaliente en términos de precisión y velocidad. YOLO (You Only Look Once) es una arquitectura de red neuronal convolucional diseñada para la detección de objetos en imágenes según el paper “the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection” [4]. La versión YOLOv8, utilizada en este contexto específico para la detección de poses, mejora la precisión y eficiencia mediante la incorporación de capas y técnicas avanzadas de aprendizaje profundo.

El funcionamiento de YOLOv8-Pose, en base a la bibliografía anteriormente mencionada, implica la predicción de poses mediante la identificación de keypoints o puntos clave en el cuerpo. Estos keypoints son puntos específicos que representan articulaciones, como codos, rodillas y tobillos. El algoritmo utiliza una estructura de red neuronal convolucional profunda para aprender representaciones robustas de estas articulaciones, permitiendo la inferencia precisa de la posición y orientación del cuerpo en la imagen.

A nivel matemático, YOLOv8-Pose utiliza técnicas de regresión para estimar las coordenadas de los keypoints. La red aprende a mapear las características de la imagen a las coordenadas

espaciales de las articulaciones, optimizando los pesos de la red mediante funciones de pérdida que cuantifican la discrepancia entre las predicciones y las coordenadas reales.

Por otra parte, en el desarrollo web, se incorpora el framework Django [5], basado en Python. Este framework no solo promueve una metodología de trabajo eficiente y código limpio, sino que también ofrece características como un sistema de autenticación integrado, la capacidad para definir y modificar esquemas de bases de datos de manera intuitiva, y un sistema de plantillas versátil.

Para la gestión de datos, se elige la base de datos relacional de código abierto PostgreSQL [6], reconocida por su estabilidad, escalabilidad y robustez. PostgreSQL es especialmente adecuada para manejar grandes volúmenes de datos y transacciones complejas, garantizando altos estándares de integridad y consistencia de datos mediante el cumplimiento de las propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

La infraestructura en la nube desempeña un papel esencial, y en este proyecto se opta por Azure de Microsoft [7]. Esta plataforma de servicios en la nube destaca por su capacidad para escalar recursos según las necesidades empresariales, su integración con otras herramientas de Microsoft y su capacidad para manejar grandes cargas de trabajo con alta disponibilidad y acceso remoto. La elección de Azure asegura una plataforma confiable y escalable para la implementación y operación del proyecto.

3. Descripción general del proyecto

3.1. Descripción

El proyecto que se busca desarrollar a lo largo de esta concentración es propuesto y asesorado por la empresa socio-formadora **NDS Cognitive Labs**. En su documento de propuesta [8] se incluye la siguiente descripción general del proyecto, la cual se tomará como base para determinar los objetivos y metodología del equipo:

La propuesta consiste en desarrollar un sistema de Machine Learning altamente especializado para medir la eficiencia de los cursos impartidos por una universidad a través del seguimiento

de objetos y la reidentificación de estudiantes y profesores en el aula. Este sistema revolucionario se basa en la recopilación de datos de video en tiempo real de múltiples aulas de clases, donde se utilizarán algoritmos de reconocimiento de rostros para identificar y seguir a los participantes de manera única. Con la capacidad de llevar un registro automático de la asistencia, medir la participación de los estudiantes y detectar la atención en clase, este sistema permitirá a la universidad evaluar de manera precisa y objetiva el rendimiento de sus cursos.

Además de su funcionalidad central, la propuesta incluye características avanzadas como la configuración flexible de clases y horarios, el seguimiento automático de las clases programadas y la integración con plataformas de aprendizaje. Con una interfaz de control de asistencia intuitiva y la generación automática de informes y estadísticas, este proyecto busca simplificar y mejorar significativamente la gestión de cursos universitarios, brindando información valiosa para la toma de decisiones y permitiendo una experiencia de aprendizaje más efectiva y personalizada para los estudiantes.

Se busca optimizar el control de asistencia y participación en las aulas para obtener los siguientes beneficios:

- Perder el menor tiempo posible pasando lista entre los alumnos.
- Tomar acción cuando se detecta una baja participación entre todos los alumnos.
- Identificar solo a los alumnos registrados dentro del curso.
- La participación y asistencia detectada es visible a los involucrados en el curso, es decir, alumnos y profesores.

3.2. Requerimientos

Así mismo, el socio-formador propone los siguientes requerimientos, como objetivos esenciales a cumplir del proyecto:

1. Recolección de Datos:

- El sistema debe ser capaz de recopilar datos de video al menos el mismo día en batch (near real-time) de las aulas de clases.
- Los datos de video pueden venir de una cámara IP que puede ser fácilmente emulada en tiempo real o pueden venir de un video estático (por ejemplo hecho con Unity).

2. Almacenamiento Seguro de Datos:

- Garantizar la seguridad y privacidad de los datos de los estudiantes y sus registros de asistencia.

3. Identificación de Alumnos y Profesor:

- Debe incluir un módulo de reconocimiento de rostros para identificar a cada estudiante y al instructor de manera única.
- El rostro del estudiante funge como su ID para ligarlo al sistema de control de asistencia.

4. Registro de Asistencia:

- El sistema debe llevar un registro automático de la asistencia de cada estudiante en cada clase.

5. Medición de la Participación:

- Debe ser capaz de cuantificar la participación de los estudiantes durante la clase. Para esto es necesario que el sistema identifique y haga el tracking de la persona, es decir, en una de las cámaras se puede hacer la identificación y en la otra el tracking.
- La participación es diferente a la asistencia y se mide a criterio de cada equipo. Por ejemplo, se puede medir en base a si pasaron a resolver un problema al pizarrón o levantaron la mano (pose-detection).
- El porcentaje de participación también es medible a través de hacer diarización sobre el audio, por ejemplo con AWS Transcribe. Es razonable asumir que la clase tiene 10 personas o menos. También es posible hacer la asignación del nombre del speaker manualmente.
- La participación de las personas que no registran asistencia (e.g. la del profesor) debe ser excluida.

6. Transcripción de la clase (nice-to-have):

- Convertir el audio a texto.
- Usar un LLM como GPT 3.5 sobre LangChain para hacer consultas (QA) sobre las clases.

7. Informes y Estadísticas:

- Generar informes y estadísticas automatizados sobre la asistencia y participación de los estudiantes.
- Esta información debe estar disponible para los estudiantes ya sea a través de un rol específico en el portal o por correo electrónico.

8. Interfaz de Administración:

- Debe contar con una interfaz de administración para configurar y supervisar el sistema o las clases.
- Esta interfaz debe estar disponible únicamente para el profesor.

9. Configuración de Clases y Horarios:

- Debe permitir la configuración de las clases, horarios y ubicaciones de las aulas de manera flexible.
- Esto aplica para el rol de profesor.
- Múltiples alumnos, múltiples cursos, cardinalidad muchos a muchos.

10. Capacidad de Escalabilidad:

- El sistema debe ser escalable para manejar un número creciente de aulas y estudiantes.

4. Propuesta de solución

4.1. Justificación del método seleccionado

La solución está compuesta por varios módulos y herramientas que permiten la correcta unificación de varios métodos que son necesarios para los requerimientos establecidos por el socio-formador. Para ello se propone la estructura de la Figura:

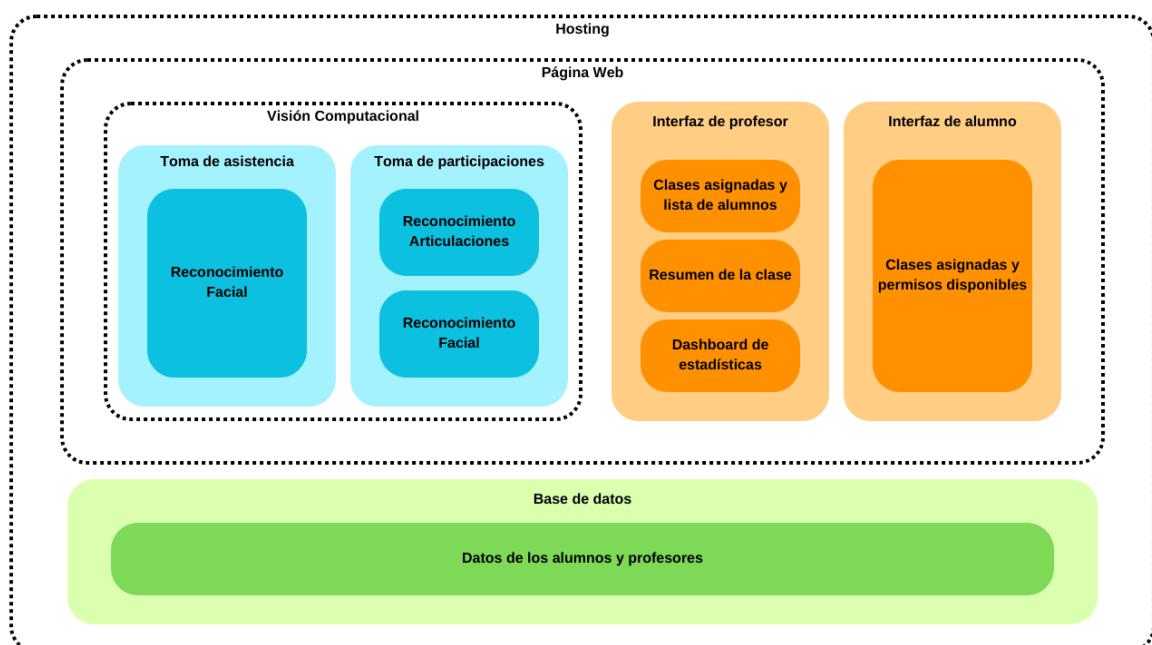


Fig 1.1 . Overview de la solución

En primera instancia es preciso detectar y reconocer a las personas asistentes en la clase para cumplir con el requerimiento de pase de lista en tiempo real, esto se logra con visión computacional a través de reconocimiento facial. Como ya se vio existen múltiples métodos para detectar y clasificar rostros, de ellos el más preciso ha sido el modelo de redes neuronales. Es por ello que se indagó en la literatura las estructuras con mejor tasa de precisión en ambientes controlados como lo es un salón de clases, en donde la iluminación se considera constante y los movimientos de las personas a reconocer son mínimos, por lo tanto no es necesario tener una detección y reconocimiento facial constante (aplicado en todos los fotogramas de un video streaming). El modelo más apegado a los requerimientos antes mencionados es un modelo de red ResNet, *dlib_face_recognition_resnet_model_v1*,[9] y [10] debido a que garantiza una precisión del 99.38% en el estándar LFW (Labeled Faces in the Wild), un benchmark comúnmente utilizado en reconocimiento facial. El modelo mencionado es una red ResNet con 29 capas convolucionales. Esencialmente, es una versión de la red ResNet-34 del paper "Deep Residual Learning for Image Recognition" de He, Zhang, Ren y Sun, con algunas capas eliminadas y la reducción a la mitad del número de filtros por capa. Se utiliza una red con la métrica de pérdida para su entrenamiento.

De acuerdo con David King, creador de la red, esta se entrenó desde cero con un conjunto de datos de aproximadamente 3 millones de rostros, derivado de varias fuentes, como el conjunto de datos face scrub (<http://vintage.winklerbros.net/facescrub.html>), el conjunto de datos VGG (http://www.robots.ox.ac.uk/~vgg/data/vgg_face/) y un gran número de imágenes obtenidas de internet. Se realizaron esfuerzos para limpiar el conjunto de datos eliminando errores de etiquetado, y aproximadamente la mitad de las imágenes provienen de VGG y face scrub. Además, se aseguró de evitar la superposición de identidades con el conjunto de datos LFW.

El entrenamiento de la red comenzó con pesos inicializados aleatoriamente y utilizó una pérdida métrica estructurada que intenta proyectar todas las identidades en bolas no superpuestas de radio 0.6. La pérdida es esencialmente un tipo de pérdida de bisagra par-a-par que se aplica a todas las parejas en un mini-lote e incluye minería de negativos difíciles a nivel de mini-lote. El modelo resultante obtiene un error medio de 0.993833 con una desviación estándar de 0.00272732 en el benchmark LFW.

En cuanto a la detección de la acción de participación en clase, se requiere un modelo que permita localizar no solo a las personas que se encuentran en el aula sino también a sus articulaciones esto se conoce como estimación de posición (en inglés pose-estimation). Actualmente existen múltiples librerías con modelos preentrenados de redes neuronales que detectan articulaciones, entre los más conocidos están Mediapipe y Yolo. Mediapipe es un modelo con bastante precisión sin embargo su framework solo permite detectar los puntos clave de una persona por fotograma. Por otro lado, el modelo de modelos de red neuronal convolucional de Yolov8-pose, por sus siglas en inglés You Only Look Once versión 8 [11], otorga como salida un conjunto de 17 puntos clave (articulaciones, puntos de referencia de rostro, entre otras) representadas por coordenadas de múltiples personas por fotograma. Dicho modelo fue pre entrenado utilizando la base de datos etiquetada de COCO (Common Objects in Context), la cual consta de más de 118 mil imágenes de entrenamiento, 15 mil de validación y 41 mil de testeо. Esto permitirá desarrollar un algoritmo que con base en un criterio de levantamiento de mano, indicará si alguien está realizando una participación o no. Así mismo esta elección resulta pertinente por la función de rastreo de personas en el fotograma (tracking) la cual permite otorgar un identificador único a cada persona reconocida en el fotograma y que facilita la tarea de asignación de participaciones.

Para medir el funcionamiento de la solución propuesta, se desarrollaron algunos indicadores que permiten evaluar los diferentes componentes de la misma.

Página Web:

- Conexión con la nube
- Identificación de Alumno y Profesor
- Portal de registro de asistencia
- Uso de protocolo HTTP
- Página web en Django

Algoritmo de reconocimiento facial

- Confianza del 99%
- Framerate mayor a 1 FPS
- Detección mayor o igual a 10 personas por frame

Algoritmo de participación

- Confianza mayor a 25%
- Framerate mayor a 15 FPS
- Detección mayor o igual a 10 personas por frame

4.2 Viabilidad de integración en el Tecnológico de Monterrey

Para implementar eficazmente este producto en el campus, se requiere una infraestructura integral que abarque tanto aspectos físicos como virtuales. En cuanto a la infraestructura física, se requiere una infraestructura en el salón donde se coloque una cámara en un sitio privilegiado para la detección de rostros con las menores obstrucciones posibles. Mientras que también se prioriza la iluminación en el salón para el uso de los algoritmos de visión computacional. A su vez se requiere de una computadora con conexión a internet para acceder al software por medio de un url. Mientras que en el apartado de la nube las especificaciones son menores debido a la amplia gama de servicios y escalabilidad que ofrecen estos tipos de servicios. Dependiendo del uso, distribución y disponibilidad que se requiera de la aplicación varía el costo de servidor para la misma. A continuación se muestra una tabla de precios que ofrece un servidor de web donde esta aplicación se encuentra servida.

Name	ACU/vCPU	vCPU	Memory (GB)	Remote Storage (GB)	Scale (instance)	SLA	Cost per hour (instance)	Cost per month (instance)
▼ Dev/Test (For less demanding workloads)								
Free F1	60 minutes/day...	N/A	1	1	N/A	N/A	Free	Free
Basic B1	100	1	1.75	10	3	99.95%	0.017 USD	12.41 USD
<input checked="" type="checkbox"/> Basic B2	100	2	3.5	10	3	99.95%	0.034 USD	24.82 USD
Basic B3	100	4	7	10	3	99.95%	0.067 USD	48.91 USD
▼ Production (For most production workloads)								
Premium v3 P0V3	195*	1	4	250	30	99.95%	0.101 USD	73.73 USD
Premium v3 P1V3	195	2	8	250	30	99.95%	0.155 USD	113.15 USD
Premium v3 P2V3	195	4	16	250	30	99.95%	0.31 USD	226.30 USD
Premium v3 P3V3	195	8	32	250	30	99.95%	0.62 USD	452.60 USD
Standard S1	100	1	1.75	50	10	99.95%	0.095 USD	69.35 USD
Standard S2	100	2	3.5	50	10	99.95%	0.19 USD	138.70 USD
Standard S3	100	4	7	50	10	99.95%	0.38 USD	277.40 USD
Premium v2 P1V2	210	1	3.5	250	30	99.95%	0.101 USD	73.73 USD
Premium v2 P2V2	210	2	7	250	30	99.95%	0.202 USD	147.46 USD
Premium v2 P3V2	210	4	14	250	30	99.95%	0.403 USD	294.19 USD

Fig 1.2 Costos mensuales servidor Web App

Además del costo del servidor Web App, se agregan consigo el costo del servidor de base de datos, así como las conexiones y servicios de seguridad entre los mismos. Sin embargo dichos costos al igual que este servidor son flexibles y escalables conforme crezca el proyecto. La viabilidad del proyecto es sustentada en la confianza y recurso que se le permita

invertir al mismo, dado que este es costeable conforme su utilidad sea de importancia, por lo que si se cuenta con un servidor de este tipo, lo óptimo es que el mismo esté trabajando la mayoría del tiempo y genere un valor regularmente.

4.3. Análisis FODA

Fortalezas:

Se tiene un equipo altamente capacitado en diseño web, Inteligencia artificial, uso de herramientas computacionales así como implementación de nuevas tecnologías. Este equipo también tiene las herramientas cognitivas para resolver problemas de índole lógica y matemática con el uso de la teoría y apoyo en la comunidad científica. Varios de los integrantes cuentan con experiencia en el uso de visión computacional e implementación de sensores, interpretación de los mismos y despliegue de la información procesada en diferentes dashboards. El proyecto es una implementación de conocimientos novedosos para crear una solución a un problema antiguo con una aproximación científica.

Oportunidades:

El proyecto puede escalar a una implementación en escuelas para el reconocimiento de acciones en un salón de clases, el proyecto en la actualidad tiene una limitación de poder de procesamiento así como de acceso a datasets para recolectar la información que se necesita para entrenar redes neuronales que puedan reconocer acciones en un salón de clases, sin embargo este proyecto tiene la oportunidad de escalar a un reconocimiento de otras acciones como bullying, aburrimiento, distracción, entusiasmo, etc.

Debilidades:

El proyecto tiene la debilidad incesante de no cumplir con un grado de aceptación en el que pueda reconocer con confianza y precisión las acciones de los estudiantes en un salón de clases debido a la volatilidad de las diferentes acciones que puede realizar el humano con su cuerpo, por lo que a pesar de que el proyecto sea exitoso en el scope de reconocimiento, podría no ser relevante a la hora de medir con un criterio humano lo que es una participación, alguien podría hablar sin levantar la mano, alguien podría participar con otros criterios no tomados en cuenta, etc.

Amenazas:

El proyecto tiene varias amenazas que no se han tomado en cuenta como lo es la privacidad y la ética del proyecto que es recolectar información, procesarla y tomar decisiones con ella, así como lo es una amenaza el uso de decisiones de una inteligencia artificial para recolectar información de algo tan subjetivo como lo es la participación. Los reclamos y quejas por parte de los alumnos por el uso de métricas basadas en la detección de una IA sin duda no faltarán.

5. Metodología

5.1. Plan de trabajo

La metodología de trabajo se ve organizada a través de objetivos generales SMART (Específicos, Medibles, Alcanzables, Relevantes, Temporales), estos objetivos se distribuyeron en actividades más específicas con fechas de realización deseables que se pueden visualizar en un [GANTT](#).

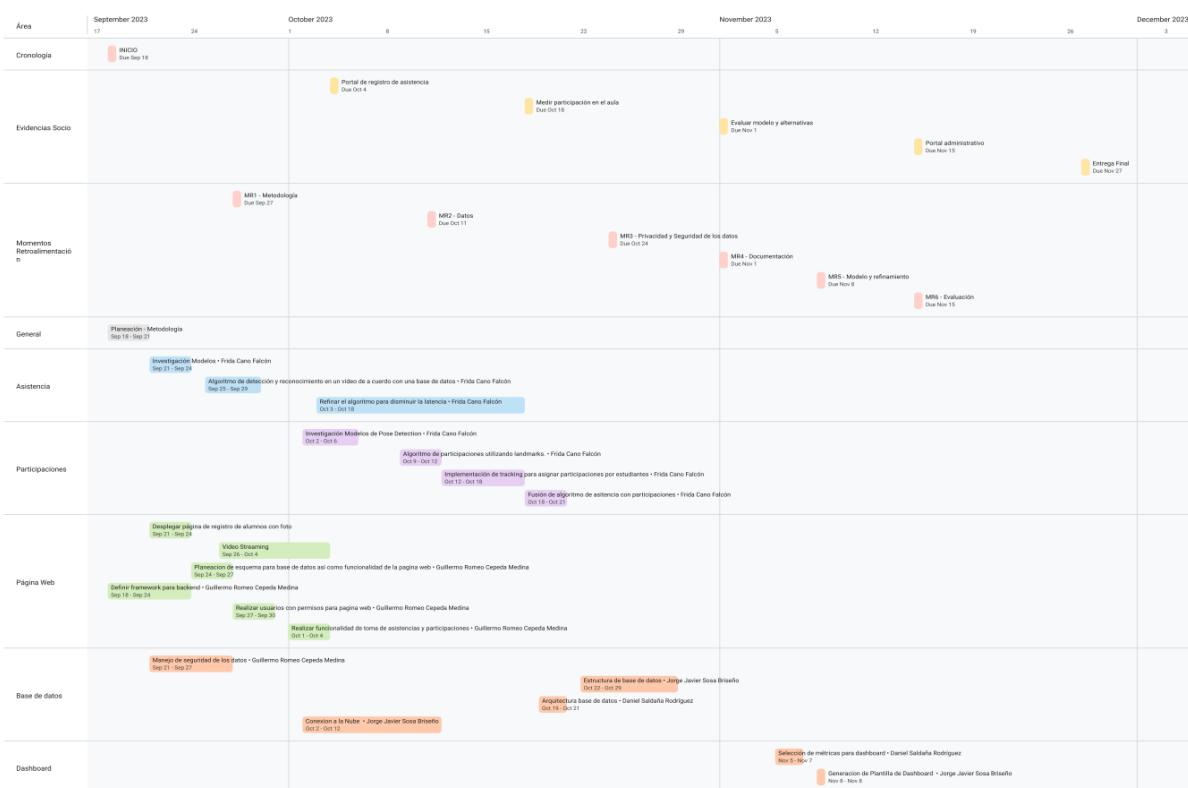


Fig 2.1. Previsualización de GANTT

En seguida se presentan los objetivos SMART por áreas:

Análisis de Negocios:

- Realizar una investigación exhaustiva de documentos de investigación científica en el tema a abordar, en este caso se tienen que recolectar papers de visión computacional, específicamente artículos con investigación en el área de detección de acciones, detección de poses y detección de rostros.

- S: Recolectar, leer e interpretar artículos científicos relevantes en visión computacional para la detección de poses, acciones y rostros.
- M; Recolectar 10 artículos científicos de investigación
- A: Dentro del Tecnológico de Monterrey existen varias publicaciones tanto por maestros como por alumnos que sobrepasan las 10 publicaciones que se buscan.
- R: Esta investigación es relevante debido a que al tomar en cuenta lo contribuido a la comunidad científica por compañeros de clase así como por otros miembros de la comunidad científica, esto permite no cometer los mismos errores que otros investigadores, así como tomar la ruta más óptima para completar el objetivo planteado y solucionar la problemática.
- T: La investigación debe ser realizada en los primeros pasos del negocio, dado que fundamenta todo el desarrollo presente y futuro. Por lo que debe estar terminada para el 31 de septiembre de 2023.

Portal de registro de asistencia:

- S: Realizar un portal de registro de asistencia en el que ambos alumno y profesor puedan ingresar y se guarde en una base de datos su información. Esta página web será implementada con el framework de Python Django utilizando una base de datos SQL y hosteando sus servicios a la nube para poder acceder a ella por internet.
- M; Página web con tres apartados /home, /admin (maestro), /alumno para que estos puedan registrarse.
- A: A pesar de que Django es un framework complejo, la implementación de dos aplicaciones donde se puedan tomar los datos de los usuarios no es tan compleja, solo hay que pedir los datos con un modelo llamado alumnos y crear una página html que los pida.
- R: Esta página web con la implementación del registro de maestros y alumnos es importante debido a que se necesitan tomar los datos de los alumnos para poder acceder a ellos desde el backend y mostrar la solución .
- T: La implementación del registro de alumnos y maestros debe estar terminada para el 4 de octubre de 2023.

Medir participación en el aula:

- S: Medir la asistencia y participación del salón de clases utilizando los algoritmos que se tienen a disposición de face recognition y pose estimation.
- M; Tener un criterio de evaluación de una participación para implementar un condicional que conteste la pregunta de si alguien está participando en clase o no.
- A: Esto es alcanzable utilizando un árbol de decisión utilizando los criterios motrices de una persona, y tomando en cuenta la posición y movimiento de las articulaciones de un humano cuando levanta la mano.
- R: Lo más complejo de esta parte es definir el criterio de evaluación con base en la motricidad de las articulaciones de las personas.
- T: La implementación del registro de alumnos y maestros debe estar terminada para el 18 de octubre de 2023.

Evaluar algoritmos:

- S: Al haber completado una primera implementación del sistema realizado para medir la participación. Esto permitirá evaluar cada uno de los algoritmos implementados y así mejorar los algoritmos a través del cambio de los parámetros y pesos de los mismos.
- M; Tener una métrica de evaluación que permita evaluar los algoritmos dará algo con que compararlos cuando se vayan cambiando los hiper parámetros y así seleccionar el mejor con base en esta métrica .
- A: A lo largo del semestre se han implementado tanto como en retos como en actividades diferentes métricas de rendimientos para algoritmos de machine learning, por lo que no debería ser complejo implementar la métrica, pero si es necesario compararlas y utilizar la que más información aporte y transmita.
- R: Una vez completados los objetivos previos, la implementación de la métrica para evaluar el algoritmo, así como su mejora serán un tema que se abordara con doctores especialistas en el tema, la colaboración con ellos será clave para mejorar el algoritmo.
- T: La implementación de la métrica de evaluación de algoritmos así como la mejora del mismo deberá ser realizada antes del 1 de noviembre de 2023.

Portal de administración:

- S: Terminar la página web con la interfaz de usuario y la experiencia de usuario, esto implica tener una base de datos, un desarrollo de backend completo que implemente las acciones que el usuario realizará en la web. Además se tiene que hacer un desarrollo completo de la aplicación para que sea robusta y no falle con casos “extraños”.
- M; Página web con tres apartados /home, /admin (maestro), así como las acciones de visualización en tiempo real del salón de clases con la detección de pose, el reconocimiento facial, la toma de asistencia y participación. Así como también gráficas que representen las estadísticas de la participación en clase .
- A: Esta entrega es sumamente dependiente del conocimiento que se tenga en el equipo para el desarrollo web. por lo que para conseguirlo el equipo se debe enfocar en desarrollar una plataforma estable para el proyecto, además de tener implementado ya los algoritmos de visión computacional y participación en clase.
- R: Para que la implementación sea realista, se decidió utilizar un framework complejo y completo como lo es Django, puesto que permite desarrollar forzosamente una página bien elaborada.
- T: La implementación del UX y UI completo debe estar terminada para el 15 de noviembre de 2023.

6. Validación e implementación

6.1 Validación del modelo de reconocimiento facial

Anteriormente se ha establecido que para la toma de asistencia y participaciones a través de visión computacional es necesario hacer uso de redes neuronales. Por un lado, para la toma de asistencia se plantea un algoritmo de detección y reconocimiento facial durante un periodo determinado en un video en tiempo real. Por ello se propone usar el framework **face-recognition versión 1.3.0** proporcionado por Python Package Index, el cual emplea el modelo antes explicado (***dlib_face_recognition_resnet_model_v1***), que promete un 99.38% de precisión en detección y reconocimiento. A pesar de ser este un valor prometedor, es necesario comprobarlo en el contexto en el cuál se realizará la aplicación.

El objetivo es probar el rendimiento y precisión del modelo en el contexto que acontece, en este caso es un salón de clase, por lo que el dataset que se utilizó para realizar la validación del modelo está situado en un contexto donde se encuentran condiciones de luz y elementos de fondo comunes en un salón de clase (sillas, computadoras, pizarrones, papelería). El dataset es una recopilación propia de 200 fotografías de 4 personas, específicamente, 50 fotografías por persona considerando los siguientes casos:

	Lentes	Sin Lentes	Distancia lejos
Iluminación alta	10	10	10
Iluminación baja	10	10	

Tabla 1. Condiciones de fotografías



Fig 3.1 . Extracto del dataset

Se aplica entonces el modelo de detección y reconocimiento facial a estas imágenes, buscando obtener las etiquetas de las clasificaciones posibles de acuerdo con los siguientes casos:

- Reconocimiento exitoso
 - Nombre de las personas: *ariadna, daniel, frida, guillermo.*
- Detección fallida:
 - Detección de múltiples rostros en la imagen: ***MR.***
 - Ningún rostro detectado: ***NR.***
- Reconocimiento fallido
 - Se detectó un rostro pero no se relaciona con los de la base de datos: ***Desconocido.***

Para visualizar los resultados obtenidos por las predicciones se generó una matriz de confusión.

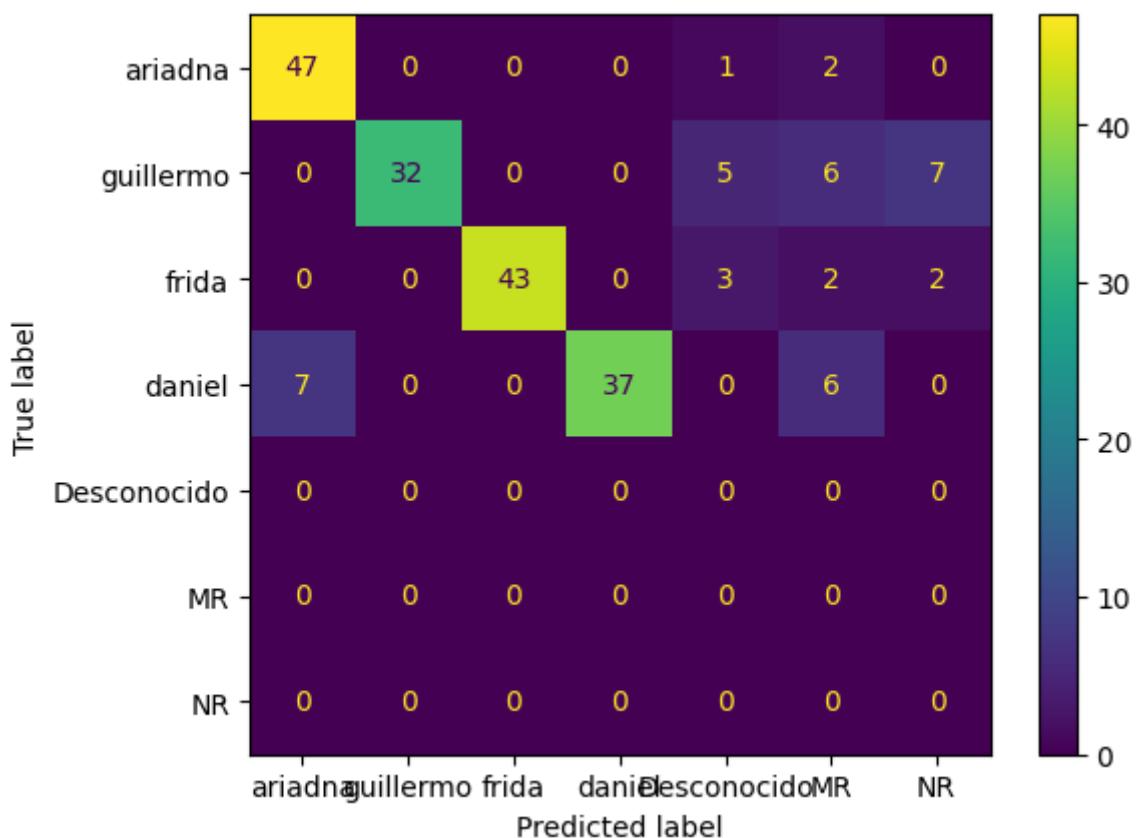


Fig 3.2. Resultados de las predicciones

Estos resultados se dividen en dos procesos diferentes la detección de rostros y el reconocimiento de estos, ya que la detección solo consiste en verificar si hay algún rostro en la imagen y proporciona sus coordenadas y el reconocimiento facial es comparar dichos rostros con los de la base de datos. De acuerdo a estos resultados se calculan las métricas pertinentes para comprobar el rendimiento: exactitud, precisión, sensibilidad, F1 score.

En cuanto a la detección se considera que:

1. Exactitud: el número de rostros correctamente detectados (181) dividido por el número total de fotografías (200).
2. Precisión: los verdaderos positivos son las detecciones correctas (181), y los falsos positivos serían el número de veces que el modelo predijo un rostro cuando no lo había.
3. Sensibilidad: Los falsos negativos serían las veces que el modelo no detectó un rostro cuando sí lo había (19 en tu caso).
4. F1 Score: El F1 Score es una medida que combina precisión y recuperación en un solo número, proporcionando una evaluación general del rendimiento.

Los resultados para las detecciones son los siguientes:

```
Detection Accuracy: 0.875
Detection Precision: 0.8101851851851852
Detection Recall: 0.8373205741626795
Detection F1: 0.8235294117647058
```

Fig 3.3. Métricas de rendimiento detección de rostros

Los valores que fueron considerados relevantes son la precisión ya que indica la proporción de instancias correctamente identificadas en relación con todas las instancias identificadas como positivas por el modelo, mide la capacidad del modelo para evitar falsos positivos. En el caso de las detecciones se obtiene un porcentaje de falsos positivos menor al 20%. Esto nos permite reconocer que el modelo de detección es lo suficientemente fiable para abordar la problemática.

En cuanto al reconocimiento (comparación de las 128 características únicas de los rostros), se obtuvieron los siguientes resultados:

```
Accuracy: 0.795
Precision: 0.5529100529100529
Recall: 0.4542857142857143
Fone: 0.49423426488051136
```

Fig 3.4 Métricas de rendimiento

Se aprecia que los falsos positivos no rebasan el 50%, contemplando que en estos cálculos se tomaron en cuenta también las múltiples detecciones donde verificando con la imagen, el rostro detectado sí era el correcto, por lo que se asume que este valor sube aún más.

En la parte teórica estos son los resultados, sin embargo cuando revisamos

6.1 Algoritmo de sistema de asistencia

Una vez validados los modelos adecuados para el contexto, se procedió al desarrollo del algoritmo diseñado para llevar a cabo el registro de asistencia. Este algoritmo se integró en el contexto del sistema de clases tradicional del Tecnológico de Monterrey en México, en particular, en su modelo de estudios Tec21. En este modelo, la duración de las clases varía desde una hora hasta hora y media como mínimo, y hasta cuatro a cinco horas como máximo. Se determinó que la mejor aproximación sería realizar el registro de asistencia por intervalos de una hora, dividiendo los primeros 15 minutos para capturar la “asistencia en tiempo”, y después de ese período, registrar la “asistencia con retardo”. Esta estrategia también permite gestionar de manera eficiente el uso de recursos de procesamiento de la computadora encargada de ejecutar el algoritmo, lo cual es esencial considerando que se está trabajando con procesamiento de imágenes y dada la abundante información contenida en cada fotograma, el proceso puede ralentizarse en función de la cantidad de rostros detectados.

La implementación de esta solución requirió de múltiples iteraciones que se ven reflejadas y documentadas a detalle en el repositorio del proyecto. Las principales son:

- Reconocimiento facial de una persona con base en una imagen:
root_FaceRecognition.

- Reconocimiento facial de una persona en un video en tiempo real: ***root_FaceRecognition***.
- Reconocimiento facial de más de una persona en un video en tiempo real: ***multiple_FR***.
- Optimización de la obtención de los registros de los rostros de las personas en la base de datos para realizar el reconocimiento: ***multiple_FR***.
- Implementación del algoritmo de asistencia: ***attendance_FR***.

Se obtuvieron los siguientes resultados, se logró hacer detección de más de un rostro de manera precisa en un video en tiempo real.

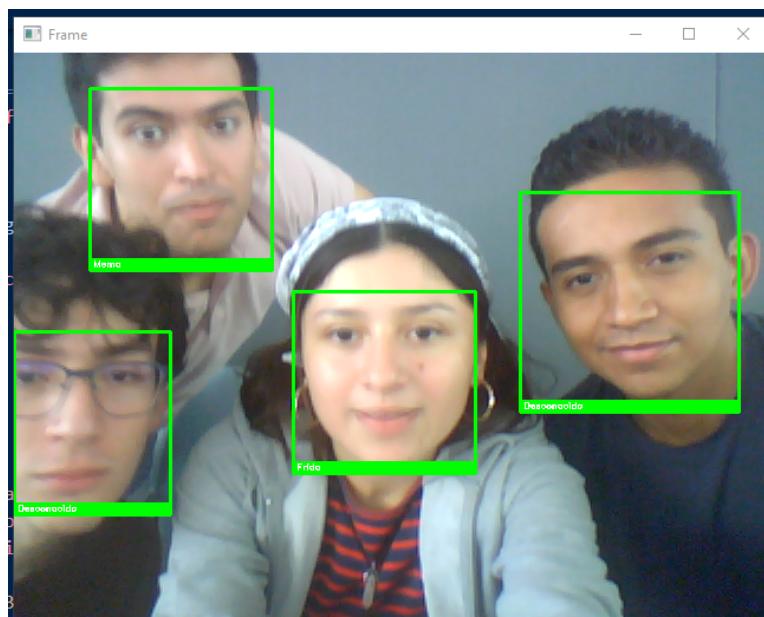


Fig 4.1.. Pruebas con modelo *face_recognition*

A partir de esto, se generó un arreglo de contadores que llevan el número de apariciones del rostro de un alumno registrado en el tiempo de “asistencias a tiempo”. Esto permite tener mayor seguridad de la asistencia ya que el reconocimiento puede tener falsos positivos, que son mínimos, pero que por ellos no se toma una asistencia con el primer registro de personas en un solo fotograma, sino es un promedio de las apariciones.

```
-----RESUMEN-----
[PROCESO] Camara liberada
[0: {'name': 'Andy', 'attendance': 0, 'delay': 0, 'participations': 0}, 1: {'name': 'Ari', 'attendance': 0, 'delay': 0, 'participations': 0}, 2: {'name': 'Frida', 'attendance': 1, 'delay': 1, 'participations': 5}, 3: {'name': 'Gerardo', 'attendance': 0, 'delay': 0, 'participations': 0}, 4: {'name': 'Iair', 'attendance': 0, 'delay': 0, 'participations': 0}, 5: {'name': 'Katia', 'attendance': 0, 'delay': 0, 'participations': 0}, 6: {'name': 'Memo', 'attendance': 0, 'delay': 0, 'participations': 0}, 7: {'name': 'Sebas', 'attendance': 0, 'delay': 0, 'participations': 0}, 8: {'name': 'Yisus', 'attendance': 0, 'delay': 0, 'participations': 0}]
[PROCESO] Cerrando programa...
[PROCESO] Tiempo de ejecucion 69.0800000198749
```

Fig 4.2. Resultados de algoritmo de asistencia

Por otro lado, en los resultados obtenidos en el proceso de validación existen errores de detección: se detectan rostros en figuras que se dibujan por la textura del suelo. Para evitar estos falsos positivos durante el proceso de toma de asistencia, se decidió fusionar el modelo de detección de yolov8n-pose, el cual otorga los cuadros delimitadores de las personas ubicadas en el fotograma de forma muy precisa, con el modelo de reconocimiento facial.

Teniendo pues que cuando se inicia el proceso de toma de asistencia se comienza a grabar la clase y el primer modelo que se activa es el de yolov8n-pose el cual detecta a todas las personas del aula, de este modelo se utiliza la salida de los “puntos clave” o keypoints para obtener las coordenadas de los rostros de dichas personas y hacer aplicar el reconocimiento facial a los rectángulos de los rostros por persona. Esto permitió no solo tener resultados más precisos sino también mejorar la latencia del video, ya que el proceso de detección ya no es tan pesado al no tener que buscar más de una cara por fotograma. Este procedimiento se puede apreciar en la figura 4.4.1.

6.2 Algoritmo de toma de participaciones

El algoritmo consiste en realizar un rastreo de las personas en el fotograma y detectar el cambio de movimiento de sus manos, si esta subió por arriba de sus ojos y después bajó, se toma una participación. Cabe mencionar que este es una primera propuesta de algoritmo, se buscará desarrollar un algoritmo más robusto que implemente más condiciones para tener una tasa de falsos positivos mínima.

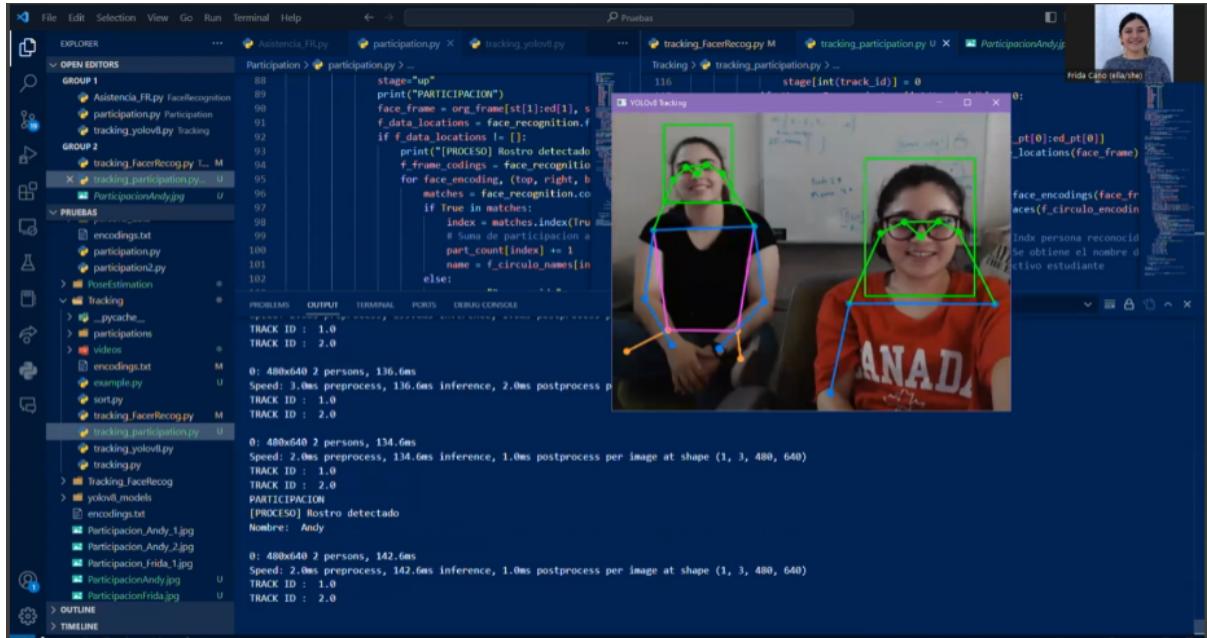


Fig 4.3. Resultados de algoritmo de participación

Ambos procesos se relacionan siguiendo el siguiente algoritmo, en donde se consideró que después de la toma de asistencia se inicia el rastreo de articulaciones de las personas para detectar las participaciones

El algoritmo detallado se encuentra en el Anexo 2, sin embargo aquí se presenta un overview en la siguiente figura:

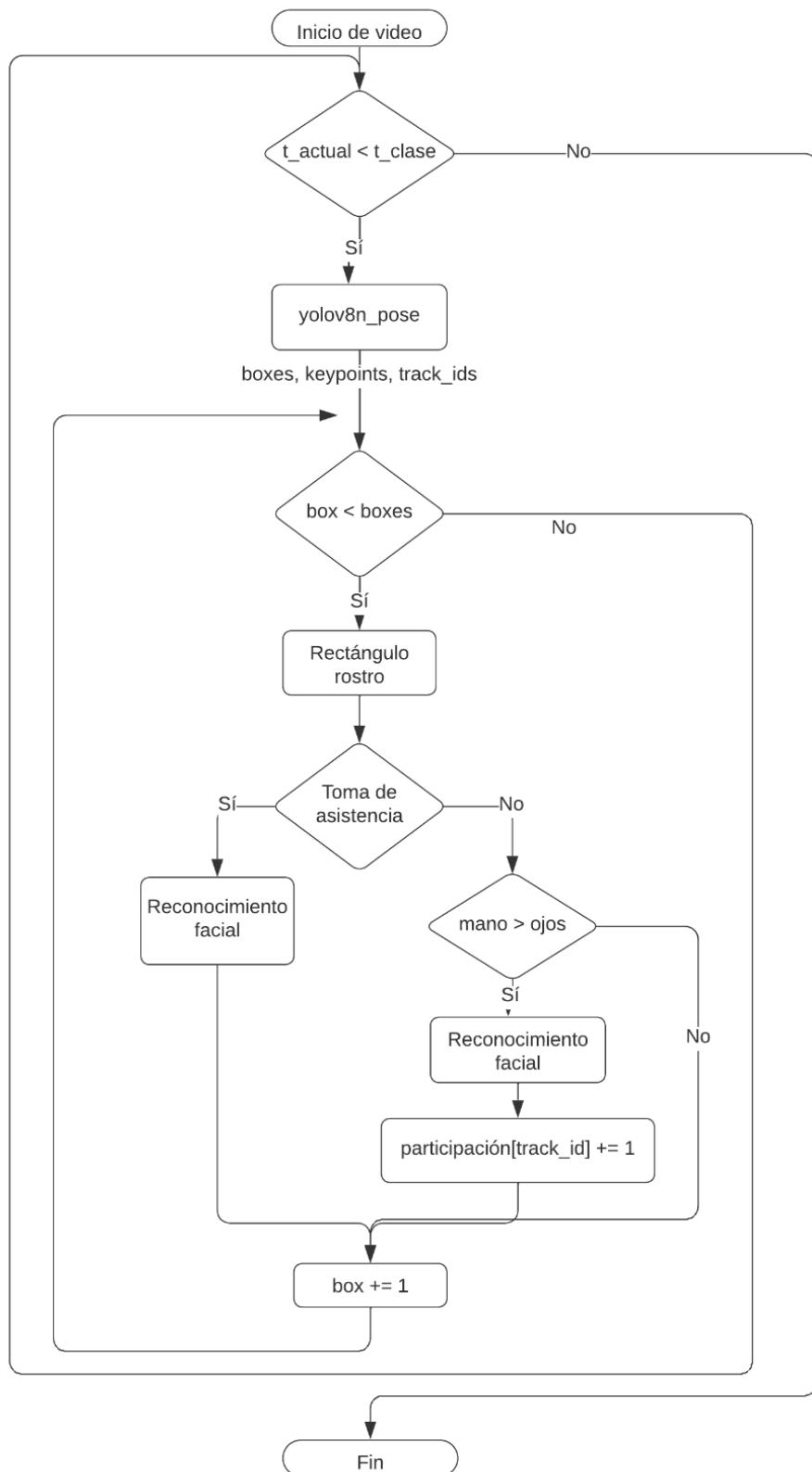


Fig 4.4.1 Algoritmo de toma de asistencia y participación

En el algoritmo se generó una función específica para realizar el reconocimiento facial,

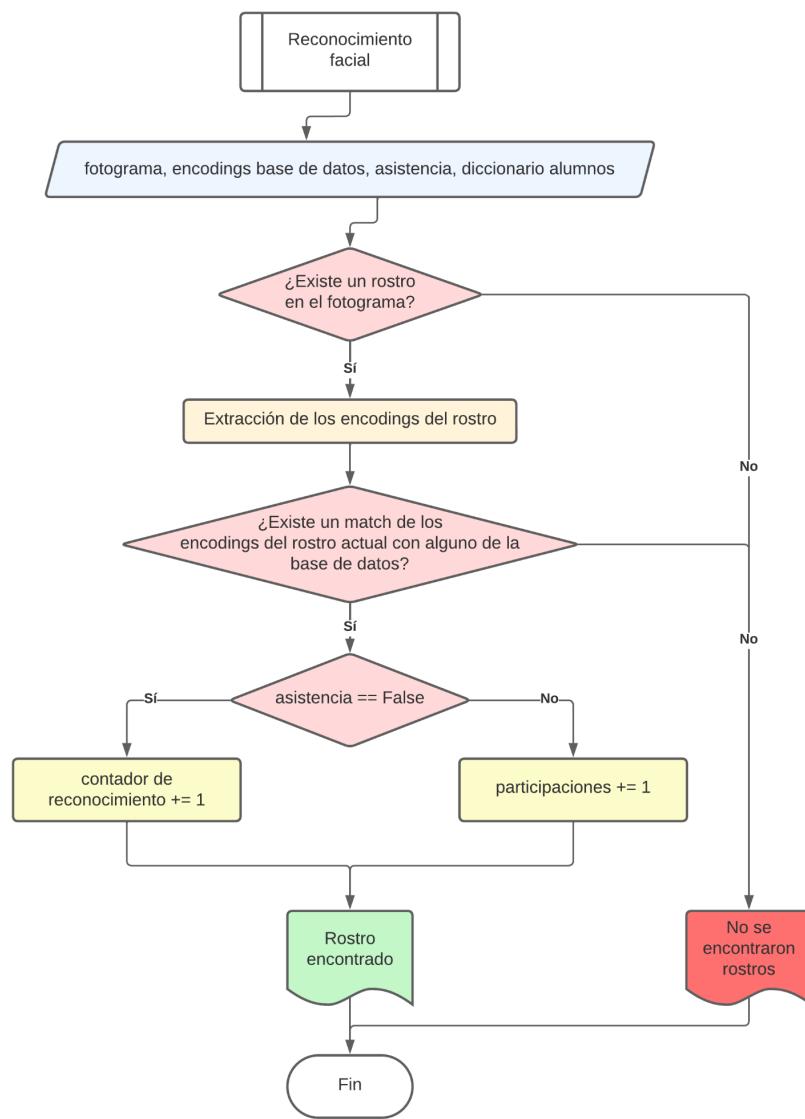


Fig 4.4.2 Diagrama de la función de reconocimiento facial

6.3 Servicio de plataforma en la nube

Para que la plataforma de toma de asistencia pueda ser utilizada de manera eficiente y en cualquier parte del mundo es necesario crear un grupo de recursos en un proveedor de servicios en la nube que sea capaz de sustentar el proyecto. Para esto se decidió utilizar Azure en vista de su calidad de soporte así como la posibilidad de implementar nuevas aplicaciones de la inteligencia artificial debido a su relación con OpenAI. Para este proyecto se utilizaron los siguientes servicios:

1. Failure Anomalies - pruebasdjango-ow4feh6pcs6xe-appinsights

- **Tipo:** Failure Anomalies
- **Descripción:** Regla de alerta para el detector inteligente en Azure Application Insights. Su propósito es identificar y notificar anomalías relacionadas con fallos en la aplicación.

2. pruebasdjango-ow4feh6pcs6xe-app-service

- **Tipo:** App Service
- **Descripción:** Azure App Service, plataforma que permite a los desarrolladores crear, implementar y escalar aplicaciones web. Ofrece características como escalabilidad automática, integración continua y opciones de alojamiento flexible.

3. pruebasdjango-ow4feh6pcs6xe-appinsights

- **Tipo:** Application Insights
- **Descripción:** Servicio de monitoreo y diagnóstico que provee información sobre el rendimiento y el uso de las aplicaciones.

4. pruebasdjango-ow4feh6pcs6xe-dashboard

- **Tipo:** Dashboard compartido
- **Descripción:** Un dashboard compartido en Azure que probablemente contiene información clave y métricas de varios servicios, una vista consolidada y personalizada para el monitoreo del proyecto.

5. pruebasdjango-ow4feh6pcs6xe-postgres-server

- **Tipo:** Azure Database for PostgreSQL flexible server
- **Descripción:** Servidor de base de datos PostgreSQL flexible en Azure, para implementar y escalar bases de datos PostgreSQL con facilidad, manteniendo altos niveles de seguridad y disponibilidad.

6. pruebasdjango-ow4feh6pcs6xe-postgres-server.private.database.azure.com

- **Tipo:** Zona de DNS privada
- **Descripción:** Una zona de DNS privada asociada con el servidor de base de datos PostgreSQL. Proporciona resolución de nombres de dominio dentro de una red privada. Hecha especialmente para resguardar la seguridad de los datos y servidores

7. pruebasdjango-ow4feh6pcs6xe-service-plan

- **Tipo:** Plan de servicio de App Service

- **Descripción:** Azure App Service, define la capacidad, las características y la escala de las aplicaciones web alojadas. Permite la asignación eficiente de recursos y el manejo de la demanda.

8. pruebasdjango-ow4feh6pcs6xe-vnet

- **Tipo:** Red virtual
- **Descripción:** Red virtual en Azure que permite a los usuarios crear redes privadas en la nube. Permite aislamiento, control de acceso y conectividad segura entre recursos.

9. pruebasdjango-ow4feh6pcs6xe-workspace

- **Tipo:** Espacio de Log Analytics
- **Descripción:** Un espacio de Log Analytics en Azure, que centraliza y analiza registros y datos operativos de diversos servicios.

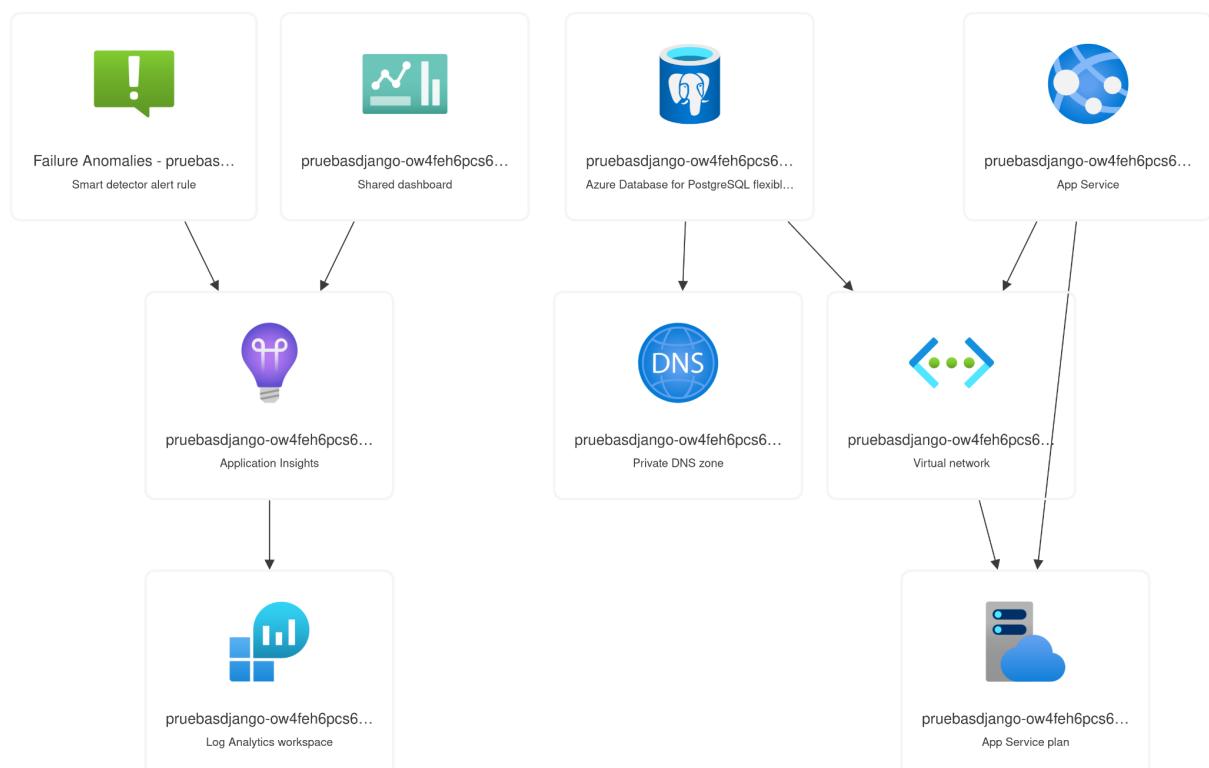


Fig 5.1. Relación entre servicios en la nube

link a la página web: pruebasdjango-ow4feh6pcs6xe-app-service.azurewebsites.net

6.4 Base de datos

Se ha diseñado una estructura de base de datos que permite gestionar información relacionada con las matrículas de las clases, clasificando a las personas en dos categorías principales: alumnos y profesores. Adicionalmente, se incorporan tablas relacionales que permiten registrar participaciones y asistencias de estas personas.

Modelo de Curso (Course):

- **Campos:**
 - name (CharField): Almacena el nombre del curso.
 - description (TextField): Almacena una descripción opcional del curso.
- **Funcionalidad:**
 - Registra información sobre los cursos ofrecidos, proporcionando nombres y descripciones.

Modelo de Usuario Personalizado (CustomUser):

- **Campos:**
 - age (PositiveIntegerField): Edad del usuario.
 - degree (CharField): Carrera del usuario.
 - semester (CharField): Semestre del usuario.
 - courses (ManyToManyField(Course)): Relación muchos a muchos con el modelo Course, permitiendo que los usuarios estén matriculados en varios cursos.
 - face_encoding (CharField): Almacena la codificación facial del usuario.
 - photo (ImageField): Almacena la foto de perfil del usuario.
- **Funcionalidad:**
 - Hereda de la clase de usuario de Django y agrega campos personalizados.
 - Permite a los usuarios estar matriculados en varios cursos.
 - Incorpora funcionalidad de reconocimiento facial para codificar las caras de los usuarios a partir de sus fotos de perfil.

Modelo de Asistencia (Attendance):

- **Campos:**

- user (ForeignKey a CustomUser): Relación con el modelo CustomUser para identificar al usuario.
- course (ForeignKey a Course): Relación con el modelo Course para identificar el curso.
- date (DateField): Almacena la fecha de la asistencia.
- is_attended (BooleanField): Indica si el usuario asistió (True) o no (False).

- **Funcionalidad:**

- Registra la asistencia de los usuarios a cursos en fechas específicas.
- Utiliza relaciones con otros modelos para asociar usuarios y cursos.

Modelo de Participación (Participation):

- **Campos:**

- user (ForeignKey a CustomUser): Relación con el modelo CustomUser para identificar al usuario.
- course (ForeignKey a Course): Relación con el modelo Course para identificar el curso.
- date (DateField): Almacena la fecha de la participación.
- amount (IntegerField): Almacena la cantidad de participación.

- **Funcionalidad:**

- Registra las participaciones de los usuarios en eventos o actividades relacionadas con los cursos.
- Utiliza relaciones con otros modelos para asociar usuarios y cursos.

Relación entre Tablas:

- La tabla de usuarios actúa como la tabla principal del esquema y de ella dependen todas las demás para estar relacionadas a través de llaves foráneas.

Características adicionales:

- Se ha implementado la propiedad "ON DELETE CASCADE" en las relaciones para asegurar la integridad referencial. Esto significa que si se elimina un registro en la tabla personas, también se eliminarán todos los registros relacionados en las tablas alumnos, profesores, participación y asistencia.

Con esta estructura, el sistema tiene la capacidad de gestionar información detallada sobre alumnos y profesores, así como registrar eventos importantes como participaciones y asistencias. Es una estructura escalable y flexible que podría ampliarse en función de las necesidades futuras. Así mismo, a continuación se muestra un esquema de cómo está estructurada la base de datos.

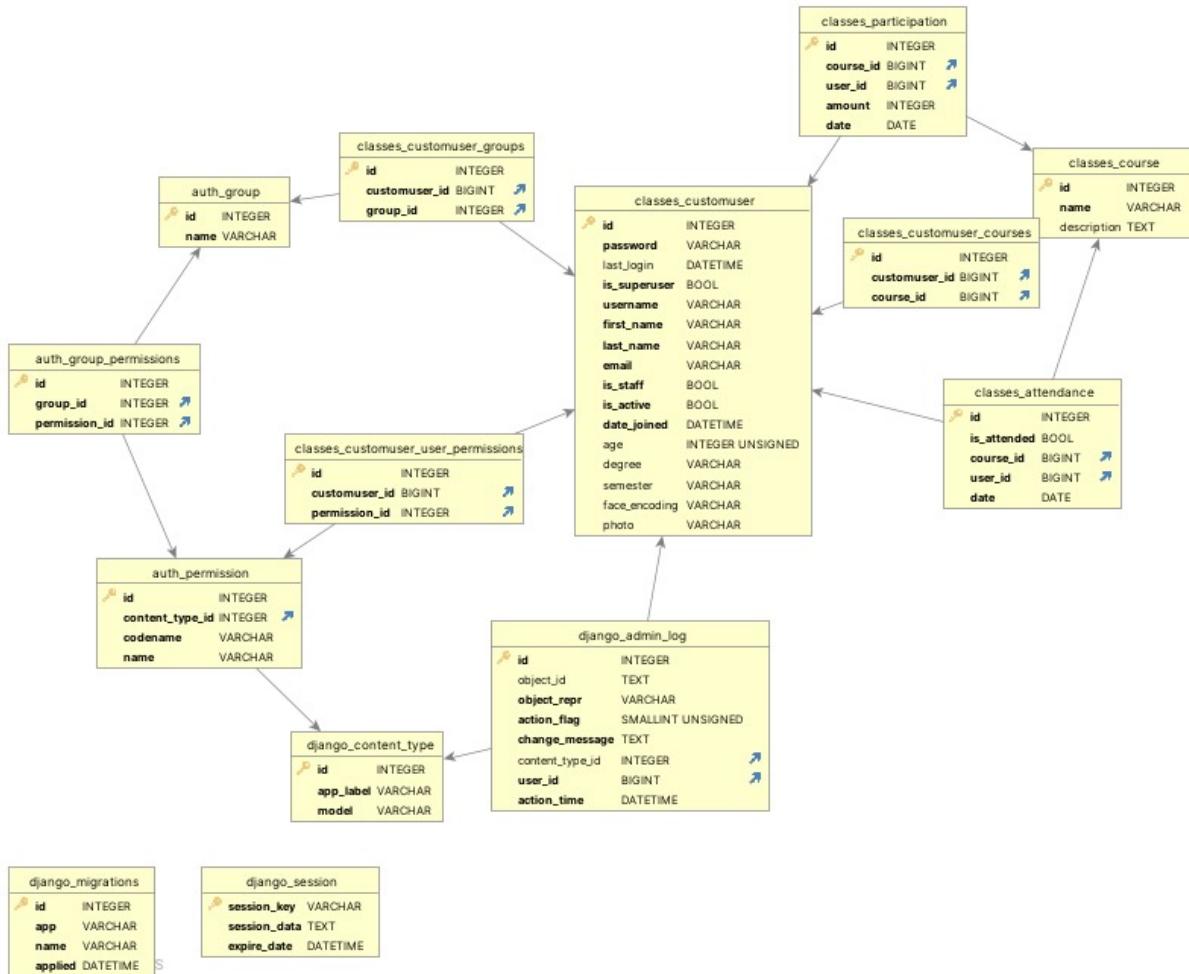


Fig 6.1. Estructura de la base de datos

6.5 Página Web

La estructura de la página web tiene sus raíces en el framework de Django, este framework se realizó en un principio para realizar servicios de noticias, por lo que tiene algunas facilidades para crear y manejar usuarios, así como otras amenidades de la base de datos, diferentes vistas, etc.

El proyecto se estructura de la siguiente manera:

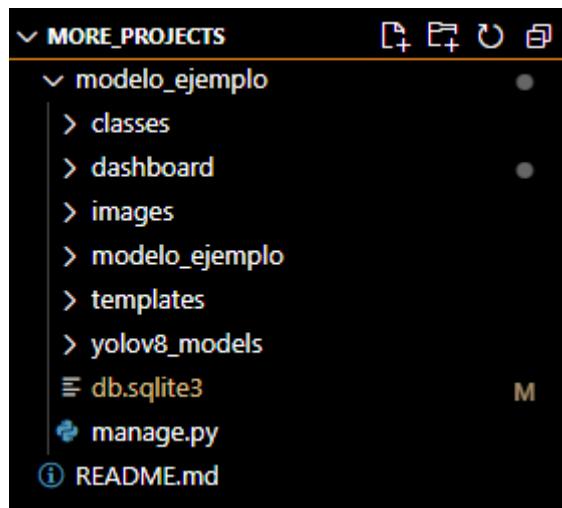


Fig 7.1. Estructura de la base de datos

El archivo manage.py permite correr comandos que actualizan, migran y corren la base de datos, así como crear nuevas aplicaciones y otras utilidades. La base de datos utilizada es PostgreSQL. Dentro de la carpeta de clases se encuentran los modelos que definen la base de datos. Entre ellos se encuentran los usuarios, cursos, participaciones y asistencias.

Y dentro de los archivos de admin.py y forms.py se encuentran las especificaciones de lo que el superusuario puede crear y editar, así como también se define lo que se despliega en el ‘admin site’.

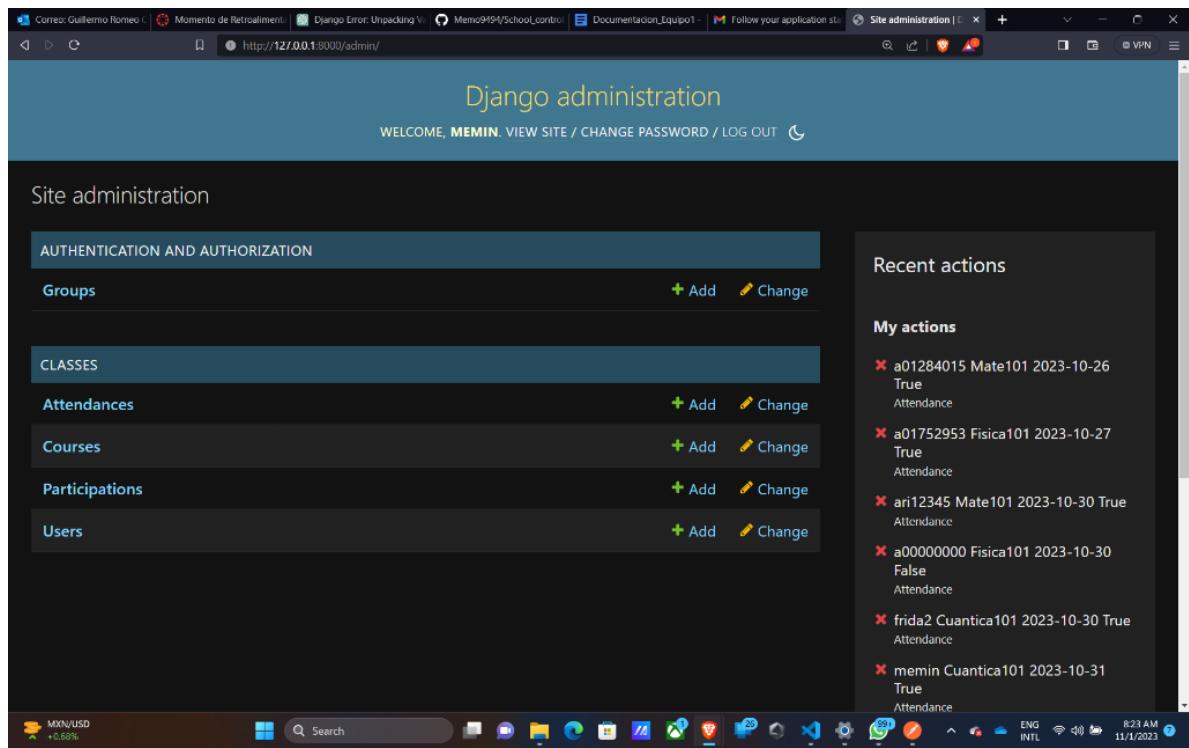


Fig 7.2. Estructura de la base de datos

Dentro del admin site, el superusuario puede crear y modificar usuarios, cursos, asistencias y participaciones, así como revisar detalladamente cada una de ellas.

Asistencias:

This screenshot shows the 'Attendances' list view in the Django admin. The left sidebar has 'CLASSES' selected. The main area displays a table of attendances with columns for 'ACTION', 'ATTENDANCE', and 'NAME'. Each row shows a checkbox, the attendance ID, the course name, and the date. An 'ADD ATTENDANCE' button is at the top right. At the bottom, it says '6 attendances'.

Action	Attendance	Name
<input type="checkbox"/>	a01752953	Cuantica101 2023-10-31 False
<input type="checkbox"/>	a01177555	Cuantica101 2023-10-31 False
<input type="checkbox"/>	memin	Cuantica101 2023-10-31 True
<input type="checkbox"/>	a01752953	Cuantica101 2023-10-31 False
<input type="checkbox"/>	a01177555	Cuantica101 2023-10-31 False
<input type="checkbox"/>	memin	Cuantica101 2023-10-31 True

Fig 7.3. Estructura de la base de datos

Participaciones:

The screenshot shows a Django admin interface for managing user participation data. On the left, there's a sidebar with navigation links: 'Start typing to filter...', 'AUTHENTICATION AND AUTHORIZATION' (Groups, Add), 'CLASSES' (Attendances, Courses, Participations, Add), and 'Users' (Add). The main area is titled 'Select user to change' and displays a list of users with columns: USERNAME, FIRST NAME, LAST NAME, and EMAIL ADDRESS. The list includes the following entries:

USERNAME	FIRST NAME	LAST NAME	EMAIL ADDRESS
A01222222	Pablo	Ruiz	pablo@gmail.com
a00000000	Andrea	Cepeda	andrea@gmail.co
a01177555	Maria Guadalupe	Rey Astinza	a01177555@itesr
a0123422			
a0128400	Daniel	Salinas	daniel@gmail.cor
a01284014	mejor	amigo	
a01284015	Guillermo	Cepeda	memin@gmail.co
a01285233	Eduardo	Hernandez	Eduardo@gmail.c

On the right, there's a 'FILTER' sidebar with sections for 'By staff status' (All, Yes, No), 'By superuser status' (All, Yes, No), and 'By active' (All, Yes, No).

Fig 7.4. Estructura de la base de datos

Este sitio también permite crear grupos de usuarios para otorgarles permisos en conjunto, se tomó ventaja de esta funcionalidad para diferenciar entre el staff de la escuela y los alumnos. Por ejemplo, el superusuario, que sería el administrador de la escuela tendría acceso a todo lo disponible en el sitio, sin embargo los maestros sólo podrían asignar cursos a alumnos, mientras que no podrían crear nuevos, cambiar participaciones pero no asistencias, etc. Solo miembros del staff pueden acceder a este sitio (Directores, administradores, maestros, etc).

Para los alumnos, creamos una vista llamada dashboard que les permite tener información de cada uno de los cursos a los que asisten por medio de un hipervínculo que los dirige al detalle del curso.

Hi memin! Welcome, have a look of your classes

Mate101

Fisica101

Cuantica101

diseno

You have None

Fig 7.5. Estructura de la base de datos

Al seleccionar alguno de los hipervínculos que se muestran (las clases asignadas) se encuentra el detalle de la clase

De esta vista, dependiendo de si el usuario es staff o no, se puede tomar la asistencia:

Alumnos	
Maria Guadalupe - Rey Astinza	
Frida - Cano	
Jose - Romeo	

Asistencia	Participación
memin - True	memin - 1
a01177555 - False	a01177555 - 0
a01752953 - False	a01752953 - 0
memin - True	memin - 1
a01177555 - False	a01177555 - 0
a01752953 - False	a01752953 - 0

[Tomar Asistencia](#)

Fig 7.6. Estructura de la base de datos

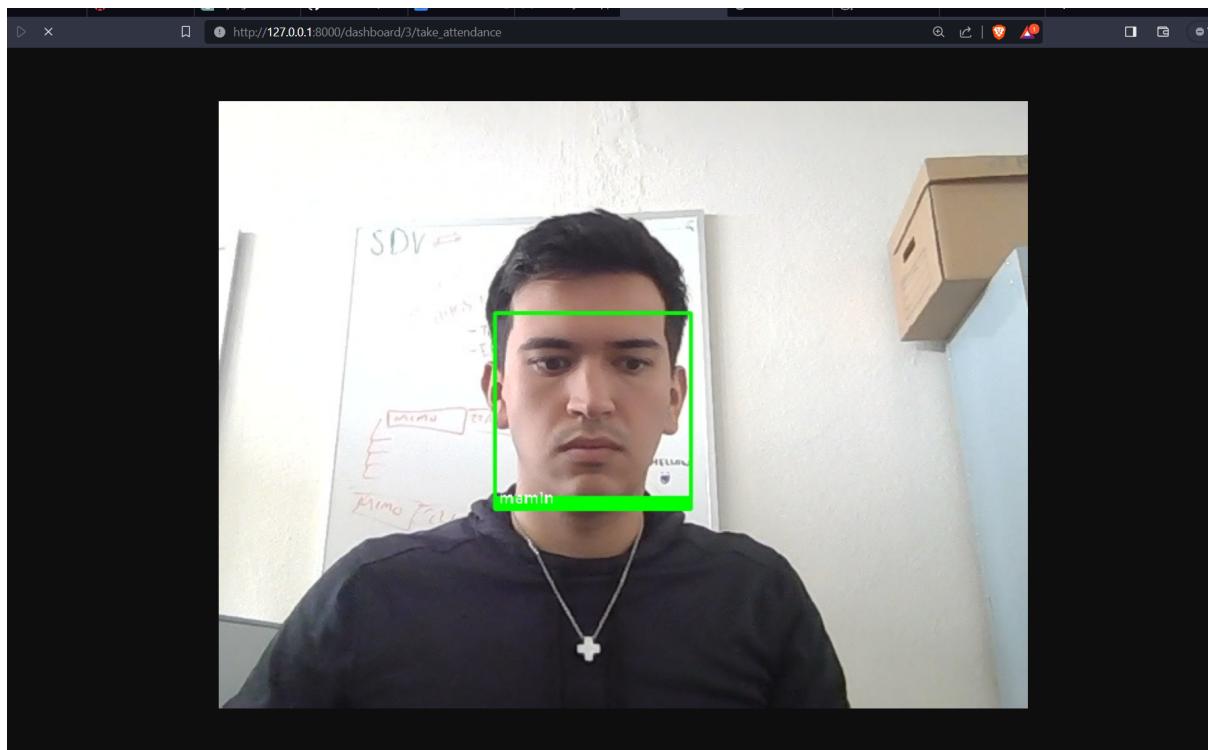


Fig 7.7. Estructura de la base de datos

Las asistencias se toman en los primeros segundos de la clase, después de esos segundos se empiezan a tomar las asistencias de los alumnos, estas también tienen un tiempo determinado para ser registradas y al final de este tiempo se registran en la base de datos y pueden ser accedidas por quienes tengan permiso.

Las funciones que permiten realizar estas acciones se encuentran detalladamente documentadas en el repositorio del reto.

El flujo del trabajo puede ser visualizado de la siguiente manera;

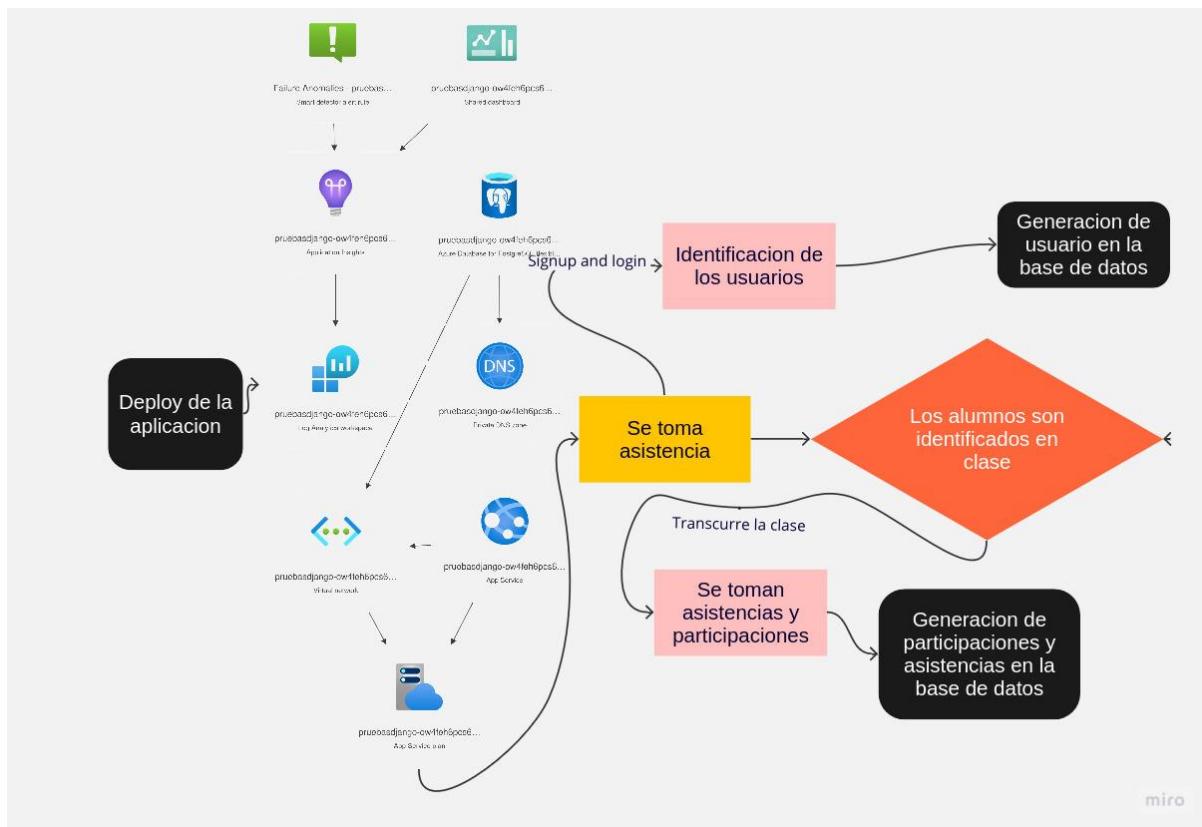


Fig 7.8 Flujo del proyecto

6.6 Dashboard

Se decidió utilizar HTML para el proyecto multidisciplinar de ciencia de datos debido a su interfaz intuitiva y su capacidad para integrarse sin problemas con múltiples fuentes de datos, lo que permite visualizar y analizar los conjuntos de datos complejos con eficiencia. Su naturaleza colaborativa facilita compartir insights y fomentar una toma de decisiones basada en evidencia dentro del equipo de investigación, mientras que las sólidas opciones de personalización y seguridad garantizan que las visualizaciones sean tanto precisas como protegidas. La facilidad de uso y la funcionalidad avanzada de HTML lo convierten en una herramienta indispensable para traducir el análisis de datos en historias impactantes y decisiones bien fundamentadas. A continuación se mostrará un avance del mismo, con variables de prueba.

Dashboard del Rendimiento de Clases

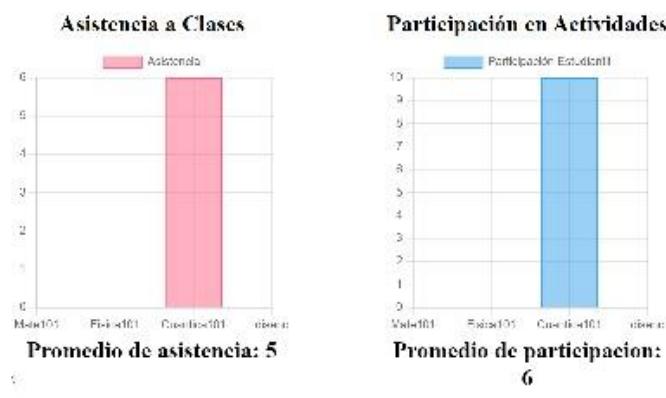


Fig 8.1. Visualización de datos en la web

7. Conclusiones

El proyecto ha logrado integrar con éxito tecnologías avanzadas de Machine Learning y reconocimiento facial en el contexto educativo. Se implementó un sistema de asistencia y participación basado en reconocimiento facial y seguimiento de movimientos, optimizando la gestión del tiempo y recursos en aulas universitarias. La solución, integrada en el modelo educativo Tec21 del Tecnológico de Monterrey, mejora significativamente la interacción y compromiso en el aula, al tiempo que mantiene altos estándares de privacidad y seguridad de los datos.

La precisión de la librería face_recognition permite un reconocimiento facial efectivo con una base de datos mínima, garantizando registros faciales de alta calidad. El sistema de participaciones, utilizando Yolov8, también ha mostrado resultados prometedores en la detección y seguimiento de movimientos. Además, la estructura flexible y escalable de la base de datos permite una gestión eficiente y adaptativa de la información de alumnos y profesores. La eficacia del sistema se refleja en su capacidad para contabilizar asistencias y participaciones en tiempo real, con una gestión de datos eficiente y una interfaz intuitiva en el dashboard, facilitando la revisión y análisis estadístico por parte de administradores y profesores.

A pesar de sus fortalezas, el proyecto presenta ciertas limitaciones. Por ejemplo, la dependencia de la calidad de las imágenes de la base de datos de las personas registradas en

el sistema para el reconocimiento facial podría ser un desafío en entornos con iluminación variable o calidad de cámara deficiente. Además, el reconocimiento facial y seguimiento de movimientos podrían generar falsos positivos, aunque mínimos. Para trabajos futuros, se podrían explorar algoritmos más robustos que implementen condiciones adicionales para minimizar los falsos positivos del algoritmo de participaciones. También sería beneficioso desarrollar mecanismos más sofisticados para la gestión de la privacidad y seguridad de los datos, teniendo en cuenta las preocupaciones éticas en torno al reconocimiento facial. La expansión del sistema para adaptarse a diferentes entornos educativos y la incorporación de funcionalidades adicionales para un análisis más profundo de los datos recopilados son otras áreas potenciales para el desarrollo futuro.

8. Citas y Referencias

- [1] *Material – Domingo Mery.* (s. f.). <https://domingomery.ing.puc.cl/material/>
- [2] Face recognition at distance <https://www.researchgate.net/publication/226823029>
- [3] Mery, D., Mackenney, I., & Villalobos, E. (año). Student Attendance System in Crowded Classrooms using a Smartphone Camera. Universidad Católica de Chile
- [4] Hussain, M. (2023, June 23). The Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection.
- [5] Django. (s. f.). Django Project. <https://www.djangoproject.com/>
- [6] PostgreSQL: documentation. (s. f.). The PostgreSQL Global Development Group. <https://www.postgresql.org/docs/>
- [7] Kcpitt. (s. f.). Azure Documentation. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/?product=popular>
- [8] NDS Cognitive Labs (Director). (s. f.). *Propuesta de Concentración Tec 2023* [Documento].
- [9] He, K. (2015, 10 diciembre). *Deep residual learning for image recognition.* arXiv.org. <https://arxiv.org/abs/1512.03385>

[10] Davisking. (s. f.). *GitHub - Davisking/Dlib-Models: Trained model files for Dlib example programs*. GitHub. <https://github.com/davisking/dlib-models>

[11] Ultralytics. (s. f.). *Pose*. Ultralytics YOLOv8 Docs. <https://docs.ultralytics.com/tasks/pose/>

[12] COCO - Common Objects in context. (s. f.). <https://cocodataset.org/#home>

Referencias

[13] C++. (s. f.). *Dlib C++ Library*. <http://dlib.net/>
Geitgey, A. (2020). Face-recognition [Software]. En PyPI (1.3.0).

<https://pypi.org/project/face-recognition/>

[14] *Going deeper into face detection: a survey*. (s. f.-b.). ar5iv. <https://ar5iv.labs.arxiv.org/html/2103.14983>

[15] Odemakinde, E. (2023, 9 noviembre). *Human pose estimation with deep Learning – ultimate overview in 2024*. viso.ai. <https://viso.ai/deep-learning/pose-estimation-ultimate-overview/>

[16] *The elements of end-to-end deep face Recognition: A survey of recent advances*. (s. f.). ar5iv. <https://ar5iv.labs.arxiv.org/html/2009.13290>

[18] Kcpitt. (s. f.). *Azure Documentation*. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/?product=popular>

9. ANEXOS

9.1 Link del repositorio:

https://github.com/Memo9494/classrecon_team1_TC3007C.501/tree/main

9.2 Link jupyter notebook validacion del modelo:

9.2 Algoritmo de visión computacional

