



**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: 1 de noviembre de 2023**

# Índice

<b>Índice</b>	<b>2</b>
<b>Descripción general del proyecto</b>	<b>3</b>
<b>Requerimientos</b>	<b>4</b>
<b>Propuesta de solución</b>	<b>5</b>
Arquitectura de la propuesta	5
Vialidad en el campus	6
Integración en el campus	6
Análisis FODA	6
<b>Metodología de trabajo (SMART) para cada una de las etapas</b>	<b>8</b>
Kanban de organización	8
Análisis de Negocios	8
Portal de registro de asistencia:	9
Medir participación en el aula:	9
Evaluar algoritmos:	10
Portal de administración:	10
<b>Implementación</b>	<b>10</b>
Sistema de asistencia	11
Sistema de registro de participaciones	12
Base de datos	13
Página Web	15
Dashboard	21
<b>Referencias</b>	<b>22</b>
<b>Link del repositorio</b>	<b>22</b>

## Descripción general del proyecto

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 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.

## 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.

## **Propuesta de solución**

### **Arquitectura de la propuesta**

La solución que nuestro equipo propone está compuesta por varios módulos y herramientas que permiten la correcta unificación de varios métodos que son necesarios para los requerimientos del socio-formador. En primera instancia necesitamos reconocer a los alumnos de la clase para cumplir con el requerimiento de asistencia en clase, para esto utilizaremos la librería de OpenCV con la librería de face recognition, la implementación de estas librerías en python nos permitirán reconocer rostros y compararlos con un dataset de imágenes que nos darán como resultado quien está presente en clase en tiempo real. Además necesitamos reconocer la acción de participación en clase, para esto utilizaremos un algoritmo de pose detection el cual nos da la ubicación en tiempo real de las articulaciones de las personas presentes en el salón de clases, esto nos permitirá desarrollar un algoritmo que con base en un criterio de levantamiento de mano, nos dirá si alguien está realizando una participación o no. Estas dos acciones se realizan en dos computadoras independientes debido al poder computacional que requieren cada uno de los algoritmos que utilizamos. No obstante estamos usando una misma cámara a través de un aplicación de celular que nos permite hostear una ip local para poder acceder a la misma imagen sin necesidad de estar anclados a una computadora. Otro requerimiento que buscamos cumplir es la implementación del conocimiento estadístico que hemos recolectado a través de nuestras carreras universitarias para desplegarlo en un dashboard, este dashboard busca compartir la información digerible recolectada en clase para mostrarla a los alumnos y al profesor por separado, para esto implementaremos una página web con uso de servidor que muestre en vivo la participación y detección de pose en vivo, así como la asistencia.

Para medir el funcionamiento de nuestra solución, hemos desarrollado algunos indicadores que nos permitan 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 reconocimiento facial**

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

#### **Algoritmo reconocimiento facial**

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

#### **Vialidad en el campus**

La implementación óptima para un salón de clases con lo que planeamos será el recurso computacional que requerimos, necesita de una computadora con una tarjeta gráfica de al menos 8-16 GB. Así como también una CPU poderosa que realice las tareas lógicas de la tarea de participación estudiantil. También necesitaremos de una concepción a internet para compartir nuestra plataforma. Sin olvidar que necesitamos un servidor que de respuesta a los requisitos de la página web que hospeda la participación de los alumnos para ser visualizada por el maestro. El precio de dicha computadora se estima entre los \$25,000 a los \$50,000 pesos mexicanos, además del costo de los servidores que dependiendo del tráfico que tenga la plataforma costaría entre \$700 a \$10,000 pesos mexicanos mensualmente. Este servicio con la implementación que contamos en el momento puede no ser lo más óptimo para brindar la solución que buscamos, por lo que hay que encontrar una manera más rentable de entregar la solución que se nos pide.

#### **Integración en el campus**

El plan para integrar nuestra solución en campus requeriría de una integración embebida para el sensor que necesitamos así como un microcontrolador que pueda integrar los modelos de inteligencia artificial que utilizamos para subirlos a la nube, procesarlos allá y mostrar la web que queremos para todos los alumnos y profesores que lo requieran en el aula.

#### **Análisis FODA**

##### *Fortalezas:*

Tenemos 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 nuestros 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 necesitamos para entrenar redes neuronales que puedan reconocer acciones en un salón de clases, pero 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 nuestro 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 que no tomamos en cuenta, etc.

*Amenazas:*

El proyecto tiene varias amenazas que no hemos 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.

## Metodología de trabajo (SMART) para cada una de las etapas

La metodología de trabajo se ve organizada a través de objetivos SMART (Específicos, Medibles, Alcanzables, Relevantes, Temporales) y distribuida en un Kanban en donde se utilizó la plataforma de Miro para dar seguimiento a las actividades intermedias que nos permitirán alcanzar los objetivos.

### Kanban de organización

En el siguiente link se presenta el Kanban de organización, en este dividimos las tareas por áreas (filas) y momentos de realización (columnas): Backlog, In progress, Revision, Done (lista, en progreso, en revisión, realizadas).

Link: [https://miro.com/app/board/uXjVMjhQZtg=?share\\_link\\_id=427279822761](https://miro.com/app/board/uXjVMjhQZtg=?share_link_id=427279822761)

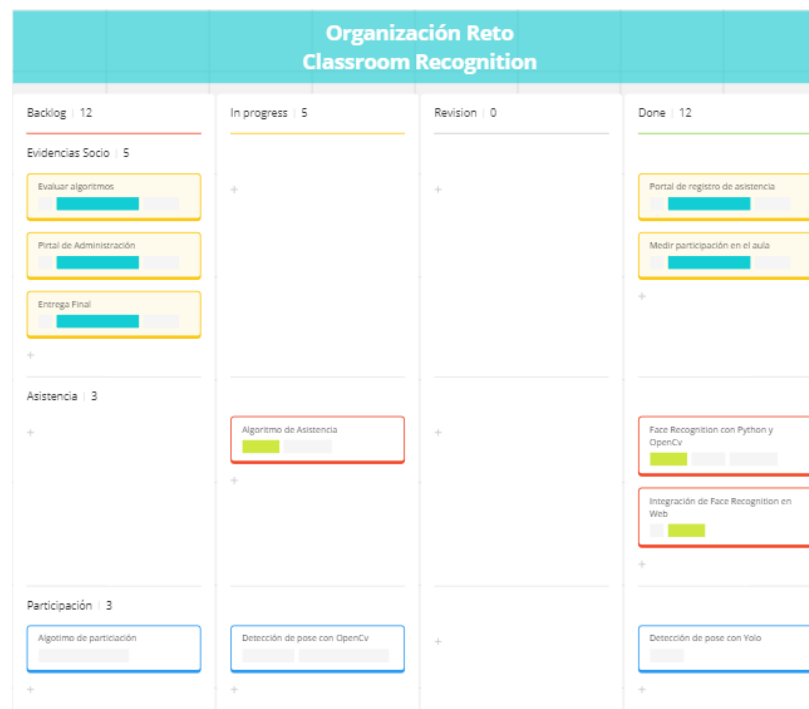


Fig #. Vistazo del [Kanban](#) de organización

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 buscamos.
- R: Esta investigación es relevante debido a que al tomar en cuenta lo contribuido a la comunidad científica por nuestros compañeros así como por otros miembros de la comunidad científica, esto nos permitirá no cometer los mismos errores que otros investigadores, así como tomar la ruta más óptima para completar nuestro objetivo y solucionar nuestro problema.
- T: La investigación debe ser realizada en los primeros pasos del negocio, dado que fundamenta todo nuestro desarrollo presente y futuro. Por lo que debe estar terminada para el 10 de octubre 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 necesitamos tomar los datos de los alumnos para poder acceder a ellos desde el backend y mostrar nuestra solución .
- T: La implementación del registro de alumnos y maestros debe estar terminada para el lunes 2 de octubre.

#### **Medir participación en el aula:**

- S: Medir la asistencia y participación del salón de clases utilizando los algoritmos que tenemos a nuestra 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 lunes 16 de Octubre.

### **Evaluar algoritmos:**

- S: Al haber completado una primera implementación del sistema realizado para medir la participación. Esto nos permitirá evaluar cada uno de nuestros algoritmos implementados. Esto nos permitirá mejorar nuestros algoritmos a través del cambio de los parámetros y pesos de los mismos.
- M; Tener una métrica de evaluación que nos permita evaluar los algoritmos nos dará algo con que compararlos cuando vayamos cambiando hyper parámetros y así elegir el mejor con base en esta métrica .
- A: A lo largo del semestre hemos 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 compararla y utilizar la que más información nos aporte.
- R: Una vez completados los objetivos previos, la implementación de la métrica para evaluar el algoritmo, así como su mejora serán un tema que abordaremos con doctores especialistas en el tema, la colaboración con ellos será clave para mejorar nuestro algoritmo.
- T: La implementación de la métrica de evaluación de algoritmos así como la mejora del mismo deberá ser realizada antes del 31 de octubre.

### **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 tengamos en el equipo para el desarrollo web. por lo que para conseguirlo debemos concentrarnos 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, decidimos utilizar un framework complejo pero muy completo como lo es Django, puesto que nos permite desarrollar forzosamente una página bien hecha, aunque sea más complejo.
- T: La implementación del UX y UI completo debe estar terminada para el 15 de noviembre.

### **Implementación**

## Sistema de asistencia

Para lograr tomar asistencia es necesario realizar un reconocimiento facial de los alumnos y profesores en el aula en el horario de la clase. En primera instancia se buscó una arquitectura de red neuronal convolucional que, gracias a su precisión (accuracy), nos permitiera tener no solo detección de rostros sino también reconocimiento de ellos. Se encontró una librería compatible con Python llamada *face\_recognition* la cual otorga una precisión del 99.38% en la plataforma de Labeled Faces in the Wild (Geitgey, 2020) y se basa en el estado del arte de reconocimiento facial propuesto por la librería de C++ *Dlib*. Un feature que ofrece esta librería es el procesamiento de imágenes, en lo que concierne al reconocimiento facial los colaboradores de C++ generaron un modelo pre-entrenado de face recognition (*dlib\_face\_recognition\_resnet\_model\_v1*) haciendo uso de aprendizaje profundo utilizando la base de datos, Labeled Faces in the Wild.

Las principales razones para elegir la librería *face\_recognition* son las siguientes: en primer lugar, su flexibilidad de implementación, dado que ofrece funciones intuitivas compatibles con los datos de salida generados por OpenCV durante el procesamiento de imágenes. En segundo lugar, destaca su impresionante precisión, con una tasa de exactitud superior al 99%. Esto significa que no es necesario entrenar el modelo con una base de datos extensa; de hecho, el algoritmo puede trabajar con una base de datos que tenga una sola imagen por persona que se desee reconocer. No obstante, es relevante mencionar que la calidad de las imágenes es crucial para obtener registros faciales de alta calidad.

Una vez se seleccionó la biblioteca adecuada, 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 estamos trabajando con procesamiento de imágenes. Dado 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.

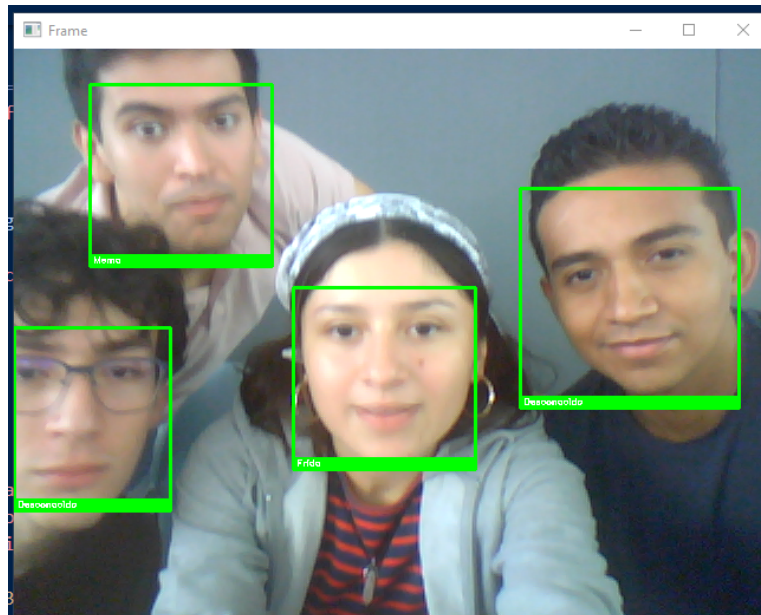


Fig. Pruebas con *face\_recognition*

A partir de esto, generamos un arreglo de contadores que llevan el número de apariciones del rostro de un alumno registrado en el tiempo de “asistencias a tiempo”. Esto nos permite tener mayor seguridad de la asistencia ya que el reconocimiento puede tener falsos positivos, que son mínimos, pero que por ellos no tomamos 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': 'Katio', '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': 'Visu', 'attendance': 0, 'delay': 0, 'participations': 0}}
[PROCESO] Cerrando programa...
[PROCESO] Tiempo de ejecución 65.98000001987549
```

Fig. Resultados de algoritmo de asistencia

## Sistema de registro de participaciones

En primera instancia, para el desarrollo del algoritmo de detección de participaciones se buscaron herramientas que nos permitieran obtener la posición y las características de las extremidades de personas. Entre las múltiples librerías existentes se encontraron los algoritmos de Mediapipe y de Yolo (You Only Look Once). La librería de Mediapipe brinda los marcadores de las articulaciones de una persona de forma muy precisa y sin ralentizar el proceso de video, sin embargo este algoritmo está configurado para una persona y se

consideró que no era óptimo para la solución del reto. Por otro lado, el algoritmo de Yolov8 (por su versión 8) permite no solo hacer la estimación de la posición de varias personas, sino que nos permite llevar un rastreo de las mismas a través de los distintos fotogramas del video con su función de *tracking*.

Nuestro 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 tener falsos positivos mínima.

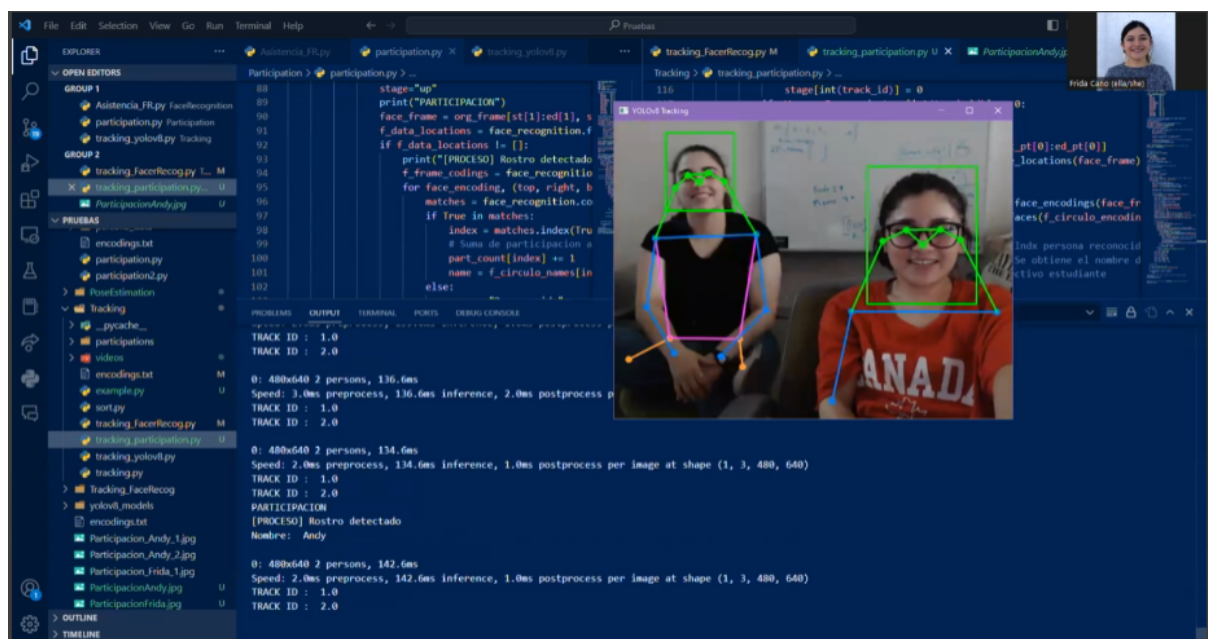


Fig. Resultados de algoritmo de participación

## Servicios en la Nube

### 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.

#### 1. Tabla personas:

Esta tabla actúa como una base general para las matrículas de la universidad, permitiendo una clasificación generalizada entre 'alumno' y 'profesor'.

- **matricula (VARCHAR(9))**: Representa el identificador único de la persona en la universidad.
- **tipo (ENUM)**: Permite identificar si la persona es un 'alumno' o 'profesor'.

## 2. Tabla alumnos:

Almacena información específica sobre los estudiantes.

- **matricula1 (VARCHAR(9))**: Identificador único del alumno. Relacionado con la tabla personas.
- **nombre y apellido (VARCHAR(255))**: Nombres y apellidos del alumno.
- **carrera (VARCHAR(255))**: Carrera que cursa el alumno.
- **hora\_de\_inicio y hora\_de\_FIN (TIME)**: Horario en que el alumno inicia y finaliza sus clases.

## 3. Tabla profesores:

Guarda datos específicos sobre los profesores de la universidad.

- **matricula2 (VARCHAR(9))**: Identificador único del profesor. Relacionado con la tabla personas.
- **nombre y apellido (VARCHAR(255))**: Nombres y apellidos del profesor.
- **edad (INT)**: Edad del profesor.

## 4. Tabla participacion:

Esta tabla relacional registra las participaciones de los alumnos en eventos o actividades.

- **participacion\_id (INT)**: Identificador único de la participación.
- **fecha (DATE)**: Fecha de la participación.
- **comentario (VARCHAR(500))**: Detalles o comentarios sobre la participación.
- **alumno\_id (VARCHAR(9))**: Relaciona la participación con un alumno específico.

## 5. Tabla asistencia:

Registra la asistencia tanto de alumnos como de profesores.

- **asistencia\_id (INT)**: Identificador único de la asistencia.
- **fecha (DATE)**: Fecha de asistencia.
- **matricula (VARCHAR(9))**: Matrícula de la persona que asistió.
- **asistio (BOOLEAN)**: Indica si la persona asistió (true) o no (false).
- **tipo (ENUM)**: Identifica si la persona es un 'alumno' o 'profesor'.

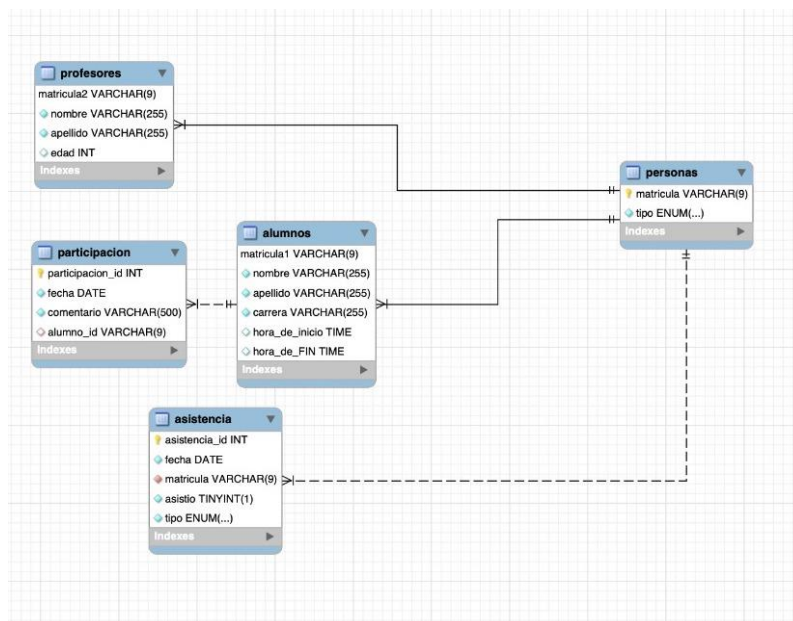
Relaciones entre Tablas:

- Las tablas alumnos y profesores están relacionadas con la tabla personas a través de sus respectivas matrículas (matricula1 y matricula2).
- La tabla participacion está relacionada con la tabla alumnos mediante el campo alumno\_id.
- La tabla asistencia se relaciona con la tabla personas a través del campo matricula.

### 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.

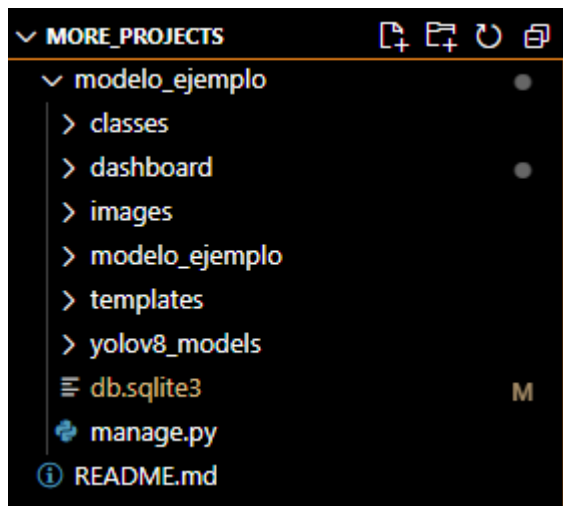
Con esta estructura, el sistema tiene la capacidad de gestionar información detallada sobre alumnos y profesores, así como registrar eventos importantes como participaciones y asistencias. Es una estructura escalable y flexible que podría ampliarse en función de las necesidades futuras. Así mismo, a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



### 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:



El archivo manage.py nos permite correr comandos que actualizan, migran y corren la base de datos, así como crear nuevas aplicaciones y otras utilidades. El archivo db.sqlite3 es donde se encuentra la base de datos, pero fácilmente puede ser reemplazada por otro engine como PostgreSQL o MYSQL. 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.

```
from django.db import models
from django.contrib.auth.models import AbstractUser
from cryptography.fernet import Fernet
import cv2
import os
import face_recognition
import time
from django.conf import settings

class Course(models.Model):
    name = models.CharField(max_length=50)
    description = models.TextField(("Description"), blank=True, null=True,
help_text=("Description"))
    def __str__(self):
        return self.name

class CustomUser(AbstractUser):
    age = models.PositiveIntegerField(null=True, blank=True)
    degree = models.CharField(max_length=50, null=True, blank=True)
    semester = models.CharField(max_length=10, null=True, blank=True)
    courses = models.ManyToManyField(Course, related_name= 'courses')
    face_encoding = models.CharField(max_length=150, null=True, blank=True)
    photo = models.ImageField(upload_to='images/', null=True, blank=True)
    def __str__(self):
        return self.username

    def face_encode(self, photo):
        if photo:
            #show image
            image_ = face_recognition.load_image_file(photo)

            face_locations = face_recognition.face_locations(image_)[0]
# Obtiene las coordenadas del rostro en la imagen
            face_encoding = face_recognition.face_encodings(image_,
known_face_locations=[face_locations])[0] # Obtenemos las características
del rostro encontrado
```



```

        encoding_str = ",".join(map(str, face_encoding))
        return encoding_str
    else:
        print("No photo")
        return None

    def save(self, *args, **kwargs):
        if self.photo:
            self.face_encoding = self.face_encode(self.photo)

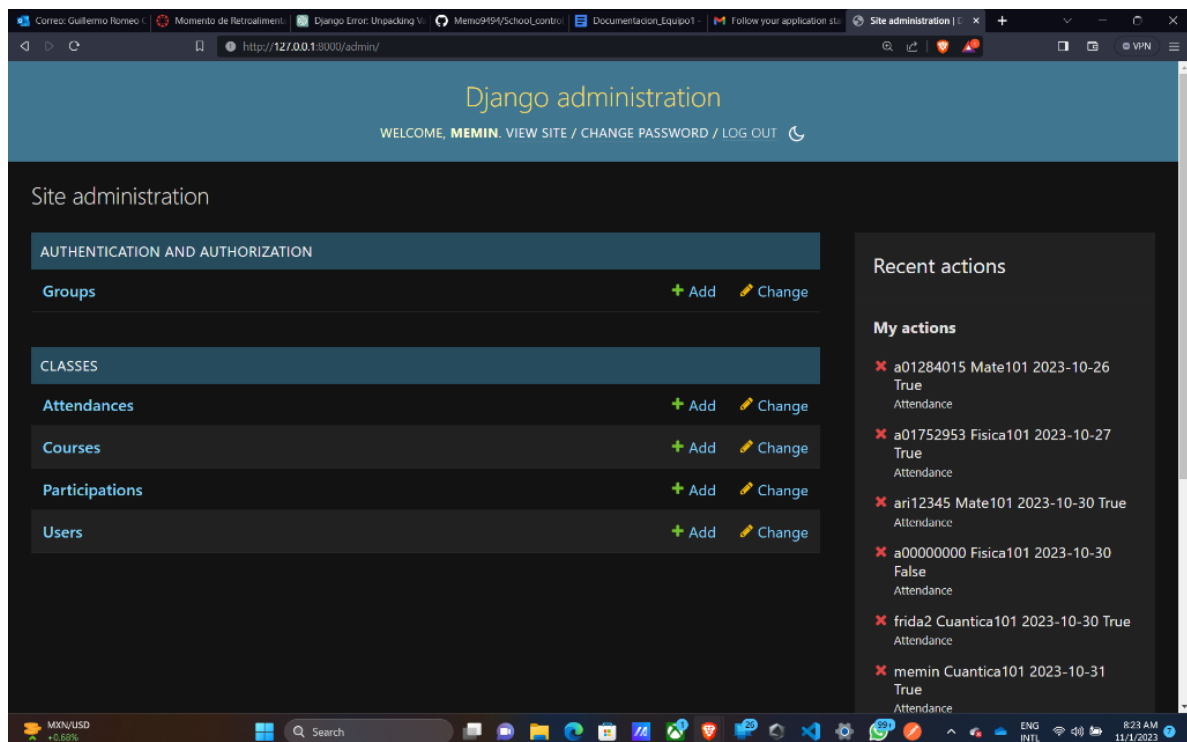
        super().save(*args, **kwargs)

class Attendance(models.Model):
    user = models.ForeignKey(CustomUser, on_delete=models.CASCADE)
    course = models.ForeignKey('course', on_delete=models.CASCADE)
    date = models.DateField(unique_for_date=True)
    is_attended = models.BooleanField(default=False)
    def __str__(self):
        return str(self.user) + ' ' + str(self.course) + ' ' + str(self.date)
+ ' ' + str(self.is_attended)

class Participation(models.Model):
    user = models.ForeignKey(CustomUser, on_delete=models.CASCADE)
    course = models.ForeignKey('course', on_delete=models.CASCADE)
    date = models.DateField(unique_for_date=True)
    amount = models.IntegerField(default=0)
    def __str__(self):
        return str(self.user) + ' ' + str(self.course) + ' ' + str(self.date)
+ ' ' + str(self.amount)

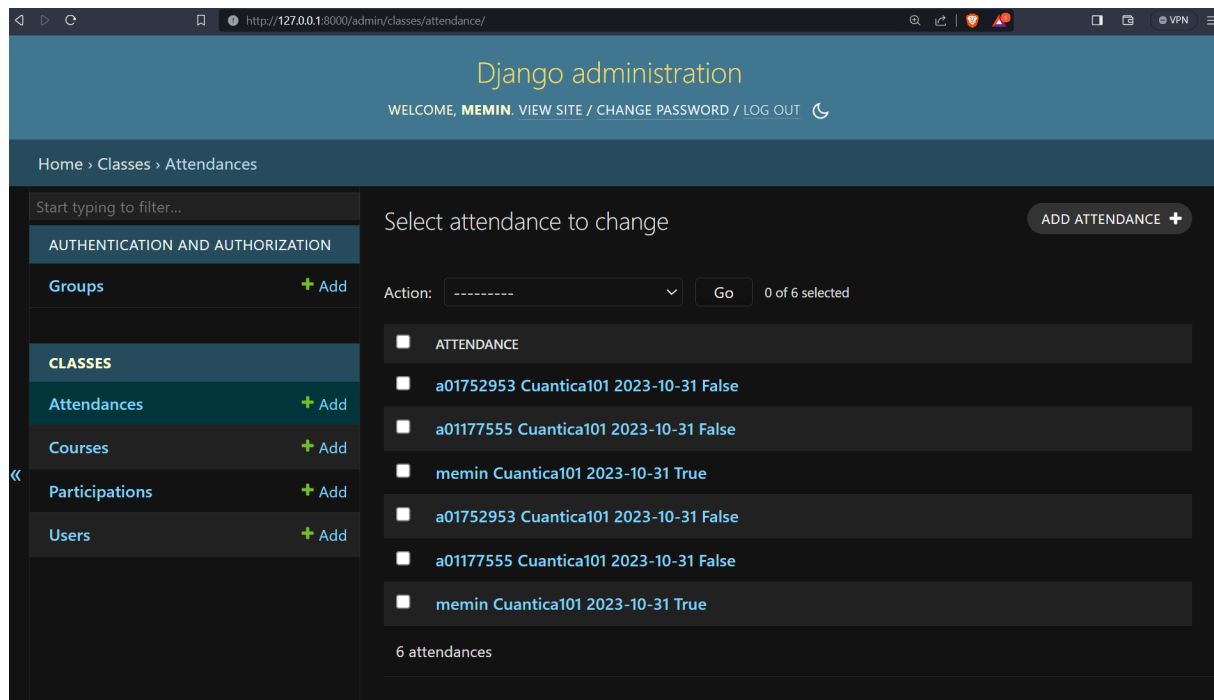
```

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’.

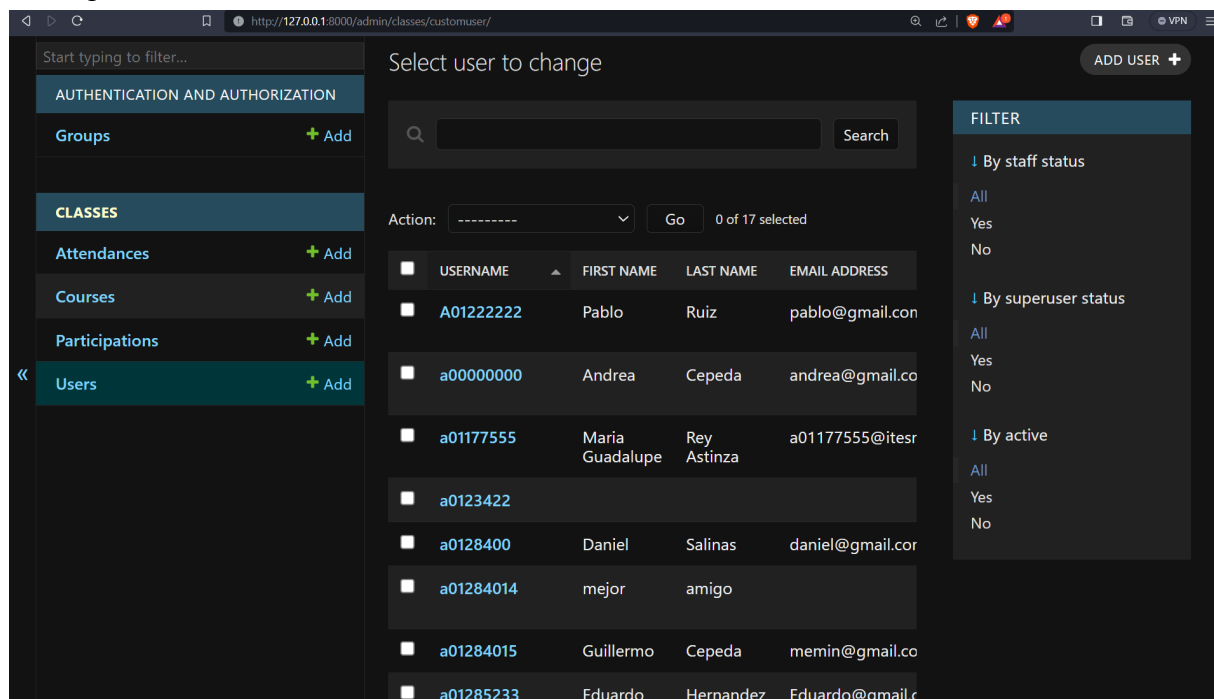


Dentro del admin site, el superusuario puede crear y modificar usuarios, cursos, asistencias y participaciones, así como revisar detalladamente cada una de ellas.

Asistencias:



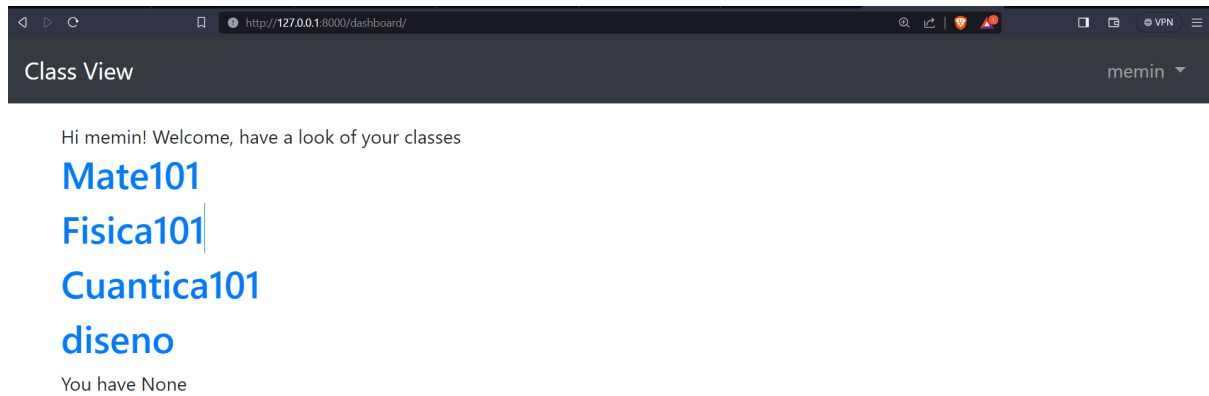
Participaciones:



Este sitio también nos permite crear grupos de usuarios para otorgarles permisos en conjunto, tomamos 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.



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

## Información del curso

### Cuantica101

Esta es una clase donde se realizan calculos cuanticos asi como otras cosas cuanticas

### Alumnos

Maria Guadalupe - Rey Astinza
Frida - Cano

### Asistencia

memin - True
a01177555 - False
a01752953 - False
memin - True
a01177555 - False

### Participación

memin - 1
a01177555 - 0
a01752953 - 0
memin - 1
a01177555 - 0

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

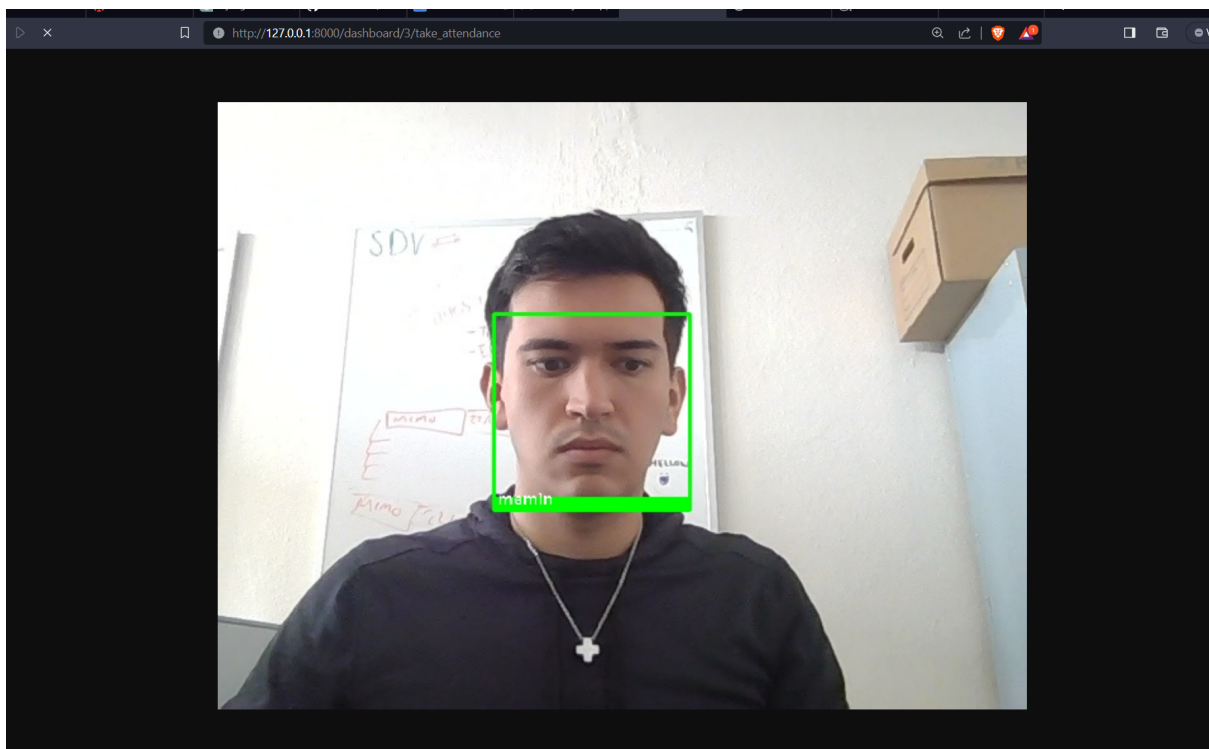
## Asistencia

memin - True
a01177555 - False
a01752953 - False
memin - True
a01177555 - False
a01752953 - False

[Tomar Asistencia](#)

## Participación

memin - 1
a01177555 - 0
a01752953 - 0
memin - 1
a01177555 - 0
a01752953 - 0



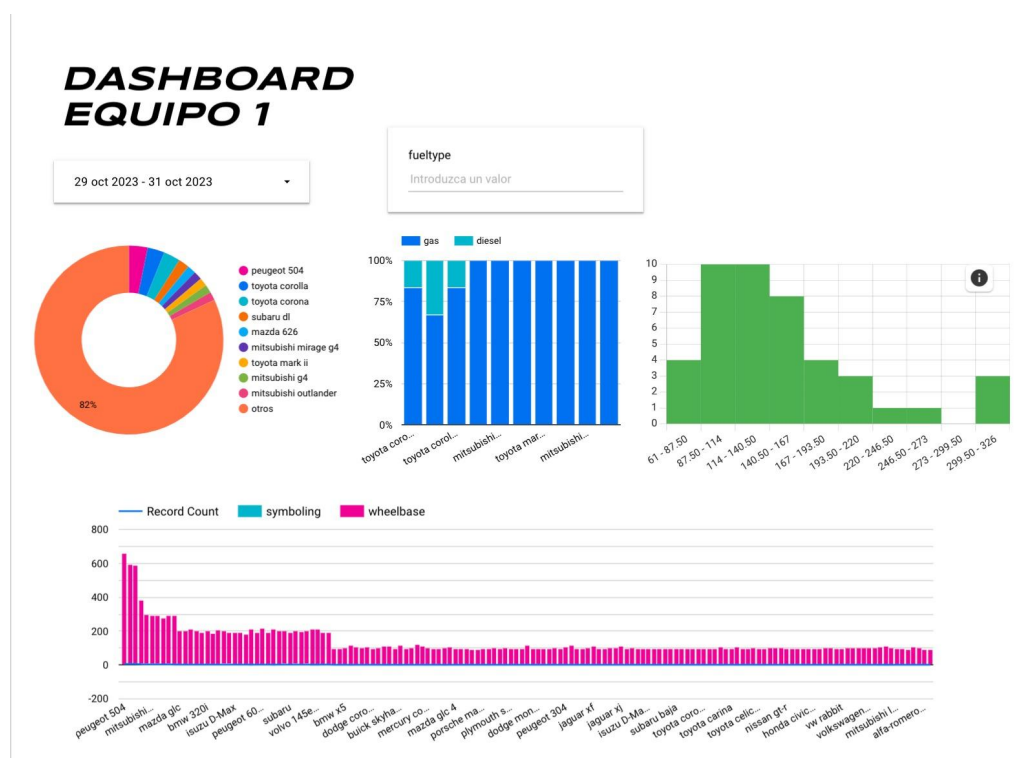
Las asistencias se toman en los primeros n segundos de la clase, después de esos segundos se empiezan a tomar las asistencias de los alumnos, estas también tienen un tiempo determinado para ser registradas y al final de este tiempo se registran en la base de datos y pueden ser accedidas por quienes tengan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

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

## Dashboard

Decidimos utilizar Looker Studio para nuestro 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 nos permite visualizar y analizar nuestros conjuntos de datos complejos con eficiencia. Su naturaleza colaborativa facilita compartir insights y fomentar una toma de decisiones basada en evidencia dentro de nuestro equipo de investigación, mientras que las sólidas opciones de personalización y seguridad garantizan que nuestras visualizaciones sean tanto precisas como protegidas. La facilidad de uso y la funcionalidad avanzada de Looker Studio lo convierten en una herramienta indispensable para traducir nuestros análisis de datos en historias impactantes y decisiones bien fundamentadas.

Asimismo, durante estas semanas hemos comenzando a diseñar el dashboard para nuestra entrega, a continuación les mostraremos un avance del mismo, con variables de prueba.



## Referencias

- 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/>

## Link del repositorio

- [https://github.com/Memo9494/classrecon\\_team1\\_TC3007C.501/tree/main](https://github.com/Memo9494/classrecon_team1_TC3007C.501/tree/main)