

---

## Asistente de clases virtuales

---

**Gerson Oswaldo Ruiz Ramirez - 202011405**  
**Paulo Fernando Merida Salazar - 202002042**  
**Byron Estuardo Solís González- 201906588**  
**Victor Abdiel Lux Juracán - 201403946**  
**Juan Carlos David Gomez Chanax - 201612106**  
**Anthony Alexander Aquino Santiago - 202001923**

### I. INTRODUCCIÓN

En ocasiones, al analizar cámaras de vigilancia o videos específicos, es importante extraer de las grabaciones, datos relevantes que permitan generar información. La IoT es aplicable para realizar análisis en diversas situaciones, una de las cuales es la detección de rostros en grabaciones. En este caso se aplicó la IoT en una app de escritorio, la cual analiza la imagen que proporciona la cámara web local con el fin de iniciar sesión en una videollamada de Google Meet al reconocer un rostro permitido, es decir, no cualquier rostro sino sólo los autorizados podrán acceder a la llamada. En la llamada, el dispositivo electrónico es capaz de seguir al rostro de la persona, con el fin de mantener siempre centrada la cara en la cámara. Así mismo, envía una alerta a un dispositivo electrónico, encendiendo un led y sonando una alarma. indicando si se detectó o no un rostro. La app de escritorio se comunica con un dashboard web para mostrar un registro de los inicios y cierres de sesión, indicando la hora de entrada o salida según sea el caso. Éste tipo de app, puede ser útil para implementarla en apps para instituciones educativas que imparten clases en línea o para empresas que realizan sus juntas virtuales.

### II. Objetivos

#### A. General

- Desarrollo de un sistema capaz de monitorear la presencia de un rostro determinado, diferenciándolo de otros rostros u objetos en una cámara.

#### B. Específicos

- Alertar al usuario de la app, el reconocimiento de rostros, utilizando un dispositivo electrónico.
- Permitir el acceso a una videollamada mediante reconocimiento facial.

- Visualizar en un dashboard web el registro histórico de los inicios de sesión o salidas que reporte la app.

### III. Desarrollo de la práctica

#### STACK DESIGN FRAMEWORK

##### A. Infraestructura del Producto

##### a. Materiales físicos:

- Arduino Uno con ATmega328
- Protoboard
- Leds
- Bocina
- Resistencias
- Servomotores

## b. Materiales digitales

- Base de datos en MySQL
- App de escritorio utilizando Python con la librería OpenCV para la detección de rostros.
- App web desarrollada en ReactJS
- Api en JS para la conexión entre la base de datos y la app web
- Máquina virtual en AWS para alojar la base de datos y app web.

## B. Sensores

- No se utilizaron sensores en éste proyecto.

## C. Conectividad

El dispositivo está compuesto por un Arduino, el cual por medio de un puerto serial se comunica con la laptop que ejecuta la app de escritorio. A través de ese puerto, la app de escritorio envía alertas si detecta o no rostros, utilizando la detección de rostros de la librería OpenCV. Si detecta un rostro, se procede a realizar el reconocimiento del rostro, para determinar si se le da acceso a una videollamada o no, se envía un valor y el arduino enciende un led y emite un sonido de aprobación sí se reconoció un rostro autorizado, si no, envía otro número para encender otro led y emitir una alerta de no autorización. Si se autorizó, la app de Python enviará coordenadas del rostro en la cámara, para mover servomotores que se encargarán de mover la misma para mantener centrado al rostro.

Si se registra un login o una salida de la videollamada, se envía a una base de datos MySQL el registro de salida o entrada. La base

de datos será consultada por medio de una api desarrollada en NodeJS. Dicha api será consumida finalmente por la app web desarrollada en ReactJS, generando un registro histórico de entradas y salidas al sistema. La api y la app web se alojan en la máquina virtual en la nube, la misma en la que se aloja la base de datos.



Figura 1: Conectividad del dispositivo

## D. Analítica

Para la parte de analítica del dispositivo, se utilizan los datos obtenidos mediante la cámara en Python, para hacer sonar una alarma y encender un LED según el dato obtenido.

La forma en que se obtienen las medidas a través de la cámara es:

- Bandera si es un rostro autorizado o no: Se envía al arduino un caracter para que lo lea y determine si debe sonar la alarma de confirmación de rostro o no autorización de rostros.

*caracter: 'g' si se reconoció un rostro autorizado o 'r' si no era un rostro autorizado o detección de otros objetos.*

- Coordenadas centrales para seguimiento de rostro: Desde Python se envían las coordenadas del punto central del rostro,

*coordenadaCentral:* coordenada que utilizará la librería map en arduino para mover los servomotores enviandole también como parámetros el rango de grados por los que la cámara se movilizará, la librería se encarga de retornar los grados que cada servomotor debe moverse.

### E. SmartApp

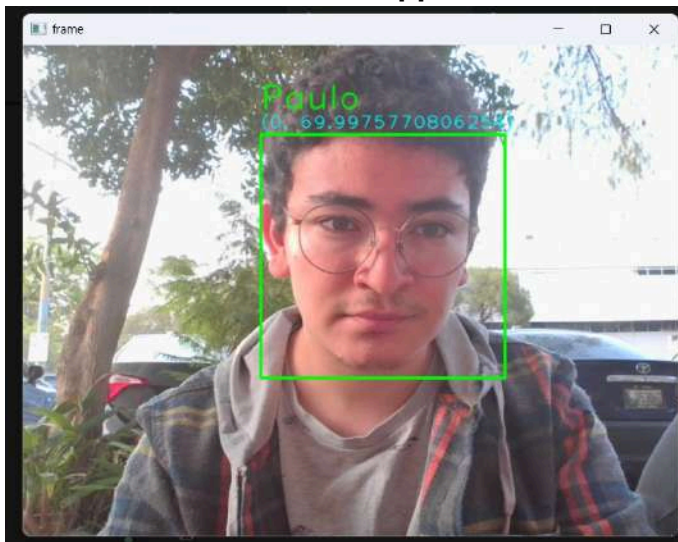


Figura 2: Registro de un rostro reconocido

### F. Boceto del prototipo

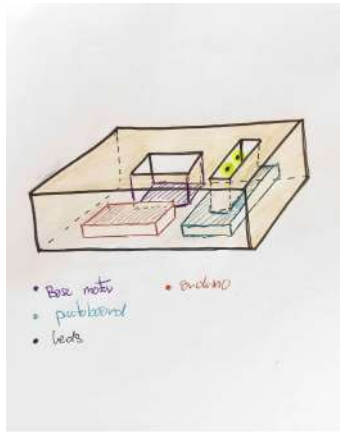


Figura 3: Boceto del prototipo

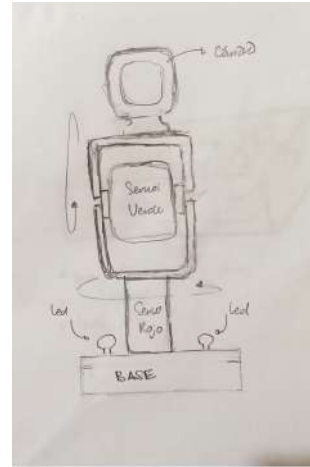


Figura 4: Boceto del prototipo

### G. Construcción del prototipo

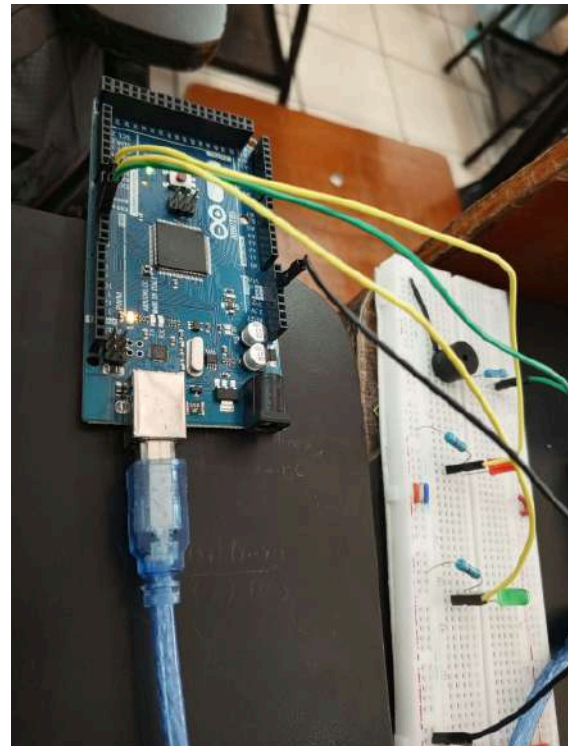


Figura 5: Construcción del prototipo

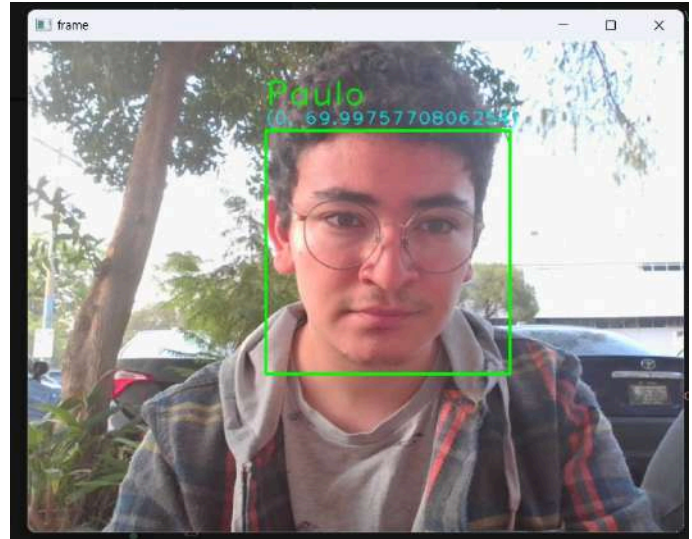


*Figura 6: Construcción del prototipo*

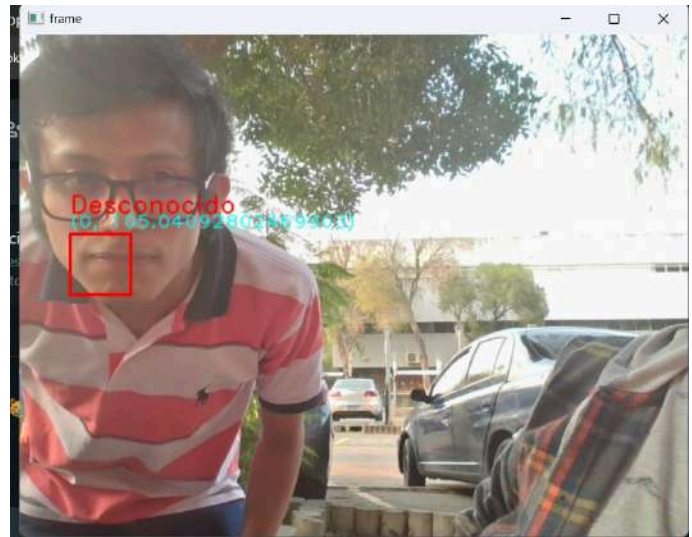


*Figura 7: Construcción de la base de la cámara web*

## H. Aplicación



*Figura 8: Cámara reconociendo un rostro autorizado*



*Figura 9: Cámara denegando el acceso*

**Link del repositorio:**

[https://github.com/pauloMerida/ACE2\\_1S24\\_G5.git](https://github.com/pauloMerida/ACE2_1S24_G5.git)