Identificador Facial

**6ºB**

**Profesor: Emiliano, Vena**

**Alumnos:**

* **Lemos, Lisandro**
* **Schiano, Silvestre**

**Descripción:**

El proyecto para los dos primeros meses de la materia Instalaciones y Aplicaciones de la Energía, va a ser una caja de tamaño cuaderno, aproximadamente tamaño N5, con un Arduino ESP32K, con el cual se escaneara el rostro de la persona para que el Arduino lo detecte a través de una cámara, para próximamente enviar esta información a la base de datos, que reside en la red de internet del colegio, y los plasmará en una interfaz gráfica en java y así poder devolverlos a la placa Arduino y mostrarlos en la placa “Display Oled 0.96 Azul 128x64 I2c Ssd1306 Arduino Ubot”, con el fin de mostrar la lista de alumnos, los que asistieron(en tiempo o tarde) y los ausentes. Con el fin de pasar la información de manera automática y así facilitar el trabajo.

**----------------------------------------------------------------------------------------------------------------------------------------------**

**Lenguaje a usar:**

* C++: Usado para programar el ESP32-CAM
* TypeScript: Creación de API
* SQL (SQLite): Almacén de datos
* JavaScript / TypeScript (React): Para el frontend del sistema web
* **Tailwind CSS**: Para estilizar la interfaz, en diferentes dispositivos.

**----------------------------------------------------------------------------------------------------------------------------------------------**

**Ya hechos:**

Seguridad y vigilancia:

* Identificación de personas en espacios públicos (China).
* FaceID.

Comercio y Marketing:

* Coca-Cola Personalización de experiencias y publicidad por persona.
* “Smile to pay” en tiendas de Alibaba (China).

Bancos y finanzas:

* Bancos como BBVA permite abrir una cuenta en el banco con identificación facial.
* HSBC permite el acceso seguro a sus apps de banca en línea.

Salud:

* Hospitales reconocimiento facial para pacientes accidentados.
* Apps de bienestar emocional.

Educación:

* Reconocimiento facial en las universidades de India y China.
* Confirmación de asistencia en exámenes en línea.

**----------------------------------------------------------------------------------------------------------------------------------------------**

Ejemplo Ideal: **Diseño 3D (Por nosotros):**



**Materiales:**

Módulos electrónicos principales:

* ESP32-CAM AI-Thinker (con cámara OV2640)
* Display LCD gráfico 128x64 con controlador ST7920 (interfaz SPI)
* Teclado con botones físicos (idealmente: números del 1 al 9, confirmar, cancelar, flechas ↑ ↓ ← →, y botón de encendido/apagado)
* Módulo lector de tarjetas microSD (interfaz SPI)

Alimentación:

* Batería recargable Li-ion 3.7V (como celda 18650 o similar)
* Módulo de carga TP4056 (con microUSB)
* Interruptor de encendido/apagado
* (Opcional) Módulo Step-Up (boost converter) de 3.7V a 5V, como el MT3608

Conectividad y montaje:

* Placa base (protoboard o PCB)
* Cables Dupont macho-hembra
* Resistencias de 10kΩ (para pull-up o pull-down en botones y SD)
* Caja o carcasa (impresa en 3D, el diseño que hice)

Otros para desarrollo y pruebas:

* PC con Wi-Fi (para programación del ESP32 y para ejecutar el backend)
* Cable USB a serial TTL (como CP2102, CH340 o FTDI) si el ESP32-CAM no tiene puerto USB
* Tarjeta microSD (4 GB o más)
* (Opcional) Lector de pendrives USB (si el diseño incluye entrada USB para cargar listas)

Software y librerías necesarias:

* **Arduino IDE**, con:
  + Librería U8g2 (para el display ST7920)
  + Soporte para placas ESP32
  + Librerías: WiFi.h, HTTPClient.h, SD.h, FS.h
* **Edge Impulse** (para entrenar y exportar la red neuronal de reconocimiento facial)
* **Node.js + Express** (para el backend del servidor de asistencia)
* **React + TailwindCSS** (para la interfaz web)
* **SQLite** (para guardar los datos localmente)

**----------------------------------------------------------------------------------------------------------------------------------------------**

**Procedimiento:**

1. Dispositivo de asistencia con ESP32-CAM

**Lenguaje:** C++ (Arduino)  
**Funcionalidades principales:**

1.1. Inicio del dispositivo mediante botón físico.  
1.2. Menú inicial con dos opciones:

* Toma de lista.
* Enviar registros a una computadora por Wi-Fi.

1.3. Selección de curso: Año, división y turno (por ejemplo, 6°B Tarde).  
1.4. Validación con contraseña del docente.  
1.5. Reconocimiento facial:

* Detecta el rostro del alumno con la cámara.
* Si lo reconoce, lo marca como presente.
* Si no lo reconoce, permite:
  + Reintentar.
  + Saltar al siguiente alumno.
  + Tomar una nueva foto y guardarla para futuro reconocimiento.

1.6. Confirmación del fin de lista por parte del docente (con clave).  
1.7. Guarda los registros en memoria interna o tarjeta SD.  
1.8. Envío de registros por Wi-Fi:

* Sube los datos a un backend conectado a la misma red local.

2. Backend web (API REST)

**Lenguaje:** TypeScript (Node.js + Express)  
**Base de datos:** SQLite  
**Endpoints principales:**

2.1. POST /login – Login de docentes.  
2.2. /students – Gestión de alumnos (CRUD):

* Crear, leer, actualizar y eliminar alumnos por curso.

2.3. /attendance – Toma de asistencia:

* Registrar presencia o ausencia.
* Consultar historial por curso o fecha.

2.4. POST /upload – Carga de datos desde el ESP32-CAM.  
2.5. GET /export – Exportar asistencia en formato CSV.  
2.6. (Opcional) Configuración del sistema (a definir si se hace con interfaz web).

3. Interfaz web (Frontend)

**Lenguaje:** React + TailwindCSS  
**Funcionalidades principales:**

3.1. Página de inicio (Home):

* Botón para iniciar sesión.
* Resumen de asistencias del día.

3.2. Login para docentes.  
3.3. Gestión de cursos y alumnos:

* Ver lista de alumnos por curso.
* Editar datos.
* Agregar o eliminar alumnos.

3.4. Visualización de asistencias:

* Por curso, fecha o alumno.
* Filtros y búsqueda avanzada.

3.5. Exportación de registros:

* Descargar registros en formato CSV (compatible con Excel).

3.6. Sincronización con el ESP32-CAM:

* Ver qué datos fueron subidos recientemente.