

## Codigos de la aplicacion:

### Front\_end:

#### Index.html:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Asistencia de Estudiantes</title>

  <link rel="stylesheet" href="styles.css">

  <style>

    body {

      font-family: Arial, sans-serif;

      background-color: #f4f4f4;

      margin: 0;

      padding: 20px;

    }

    h1 {

      text-align: center;

      color: #333;

      margin-bottom: 30px;
```

```
}
```

```
#students-list {  
    max-width: 600px;  
    margin: 0 auto;  
}
```

```
.student {  
    background-color: #fff;  
    border-radius: 5px;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
    padding: 20px;  
    margin-bottom: 20px;  
}
```

```
.student h3 {  
    color: #007bff;  
    margin-top: 0;  
}
```

```
.student p {  
    margin: 5px 0;  
    color: #666;
```

```
    }  
  </style>  
</head>  
<body>  
  <div class="container">  
    <h1>Registro de Asistencia de Estudiantes</h1>  
    <div id="students-list"></div>  
  </div>  
  
  <script src="/js/script.js"></script>  
</body>  
</html>
```

### Script.js:

```
document.addEventListener('DOMContentLoaded', function () {  
  // Hacer una solicitud GET a la API para obtener la lista de estudiantes  
  fetch('http://127.16.5.77:5000/api/students')  
    .then(response => response.json())  
    .then(data => {  
      const studentsList = document.getElementById('students-list');  
  
      // Iterar sobre la lista de estudiantes y crear elementos HTML para  
      mostrarlos  
      data.data.forEach(student => {  
        const studentDiv = document.createElement('div');
```

```
studentDiv.classList.add('student');

studentDiv.innerHTML = `

    <h3>${student.nombre}</h3>

    <p>ID: ${student.id_estudiante}</p>

    <p>Clase: ${student.clase}</p>

`;

studentsList.appendChild(studentDiv);

});

})

.catch(error => console.error('Error:', error));

});
```

### Styles.css:

```
body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 20px;

}
```

```
h1 {

    text-align: center;

}
```

```
.student {
```

margin-bottom: 10px;

}

## Back\_end:

### App\_Final.py

```
from flask import Flask, jsonify, request
import mariadb
import sys

# Importar configuración de acceso a la base de datos
from config import DATABASE_CONFIG

app = Flask(__name__)

try:
    conn = mariadb.connect(**DATABASE_CONFIG)
except mariadb.Error as e:
    print(f"Error on connection: {e}")
    sys.exit(1)

cursor = conn.cursor()

# Ruta de prueba
@app.route('/api/hello', methods=['GET'])
def hello_world():
    return jsonify({'message': '¡Hola, mundo con Flask!'})

# Obtener todos los estudiantes
@app.route('/api/students', methods=['GET'])
def get_students():
    cursor.execute("SELECT * FROM Estudiantes")
    students = cursor.fetchall()
    student_list = []
    for student in students:
        student_list.append({
            "id_estudiante": student[0],
            "nombre": student[1],
            "clase": student[2]
        })
    response = jsonify({"data": student_list})
    response.headers.add("Content-type", "application/json")
    response.headers.add("Access-Control-Allow-Origin", "*")
```

```

        return response

# Registrar la asistencia de un estudiante
@app.route('/api/attendance', methods=['POST'])
def register_attendance():
    datos = request.json
    id_estudiante = datos.get('id_estudiante')
    fecha = datos.get('fecha')
    asistio = datos.get('asistio')

    cursor.execute("INSERT INTO Asistencias (id_estudiante, fecha, asistio)
VALUES (?, ?, ?)",
                  (id_estudiante, fecha, asistio))
    conn.commit()

    response = {"message": "Attendance recorded"}
    return jsonify(response), 200

# Obtener la lista de asistencias de un estudiante específico
@app.route('/api/attendance/<int:id_estudiante>', methods=['GET'])
def get_student_attendance(id_estudiante):
    cursor.execute("SELECT * FROM Asistencias WHERE id_estudiante = ?",
(id_estudiante,))
    attendances = cursor.fetchall()
    attendance_list = []
    for attendance in attendances:
        attendance_list.append({
            "id_asistencia": attendance[0],
            "id_estudiante": attendance[1],
            "fecha": attendance[2],
            "asistio": attendance[3]
        })
    response = jsonify({"data": attendance_list})
    response.headers.add("Content-type", "application/json")
    response.headers.add("Access-Control-Allow-Origin", "*")
    return response

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

```

## Base de Datos:

### ControlAsistencias:

```
CREATE DATABASE ControlAsistencias;
```

```
USE ControlAsistencias;
```

```
CREATE TABLE Estudiantes (  
    id_estudiante INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(255),  
    clase VARCHAR(50)  
);
```

```
CREATE TABLE Profesores (  
    id_profesor INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(255),  
    asignatura VARCHAR(100),  
    horario VARCHAR(100)  
);
```

```
CREATE TABLE Asistencias (  
    id_asistencia INT PRIMARY KEY AUTO_INCREMENT,  
    id_estudiante INT,  
    fecha DATE,  
    asistio BOOLEAN,  
    FOREIGN KEY (id_estudiante) REFERENCES Estudiantes(id_estudiante)  
);
```

```
INSERT INTO Estudiantes (nombre, clase) VALUES
```

('Juan Pérez', 'Matemáticas'),  
('María Gómez', 'Historia'),  
('Carlos Rodríguez', 'Ciencias');

INSERT INTO Profesores (nombre, asignatura, horario) VALUES  
('Ana López', 'Matemáticas', 'Lunes y Miércoles 8:00 - 10:00'),  
('Pedro Martínez', 'Historia', 'Martes y Jueves 10:00 - 12:00'),  
('Laura Fernández', 'Ciencias', 'Lunes y Miércoles 13:00 - 15:00');

INSERT INTO Asistencias (id\_estudiante, fecha, asistio) VALUES  
(1, '2024-05-01', true),  
(1, '2024-05-02', false),  
(2, '2024-05-01', true),  
(2, '2024-05-02', true),  
(3, '2024-05-01', true),  
(3, '2024-05-02', false);