<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Dashboard de Administrador</title>

<!-- Enlace a Google Fonts para la tipografía Margarine -->

<link href="https://fonts.googleapis.com/css2?family=Margarine:wght@400;600&display=swap" rel="stylesheet">

<!-- Enlace a Font Awesome para íconos -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>

body {

margin: 0;

font-family: 'Margarine', sans-serif;

display: flex;

min-height: 100vh;

background-color: #f7f7f7;

}

.sidebar {

width: 250px;

background-color: #2a2d34;

color: #fff;

position: fixed;

height: 100%;

top: 0;

display: flex;

flex-direction: column;

overflow-y: auto;

transition: left 0.3s ease;

}

.sidebar.active {

left: 0;

}

.nav-links {

list-style: none;

padding: 0;

margin: 0;

}

.nav-links a {

display: flex;

align-items: center;

padding: 15px;

color: #fff;

text-decoration: none;

}

.nav-links a:hover {

background-color: #3e4149;

}

.dropdown {

position: relative;

}

.dropdown-content {

display: none;

position: relative;

background-color: #3e4149;

list-style: none;

padding-left: 20px;

}

.dropdown-content a {

padding-left: 30px;

}

.dropdown:hover .dropdown-content {

display: block;

}

.header {

background-color: #4b6584;

color: #fff;

padding: 10px;

position: fixed;

top: 0;

left: 250px;

width: calc(100% - 250px);

display: flex;

justify-content: space-between;

align-items: center;

z-index: 1000;

}

.profile-header {

display: flex;

align-items: center;

position: relative;

}

.profile-header img {

border-radius: 50%;

width: 50px;

height: 50px;

cursor: pointer;

}

.profile-header:hover .dropdown-menu {

display: block;

}

.dropdown-menu {

display: none;

position: absolute;

top: 50px;

right: 0;

background-color: #fff;

z-index: 1001;

overflow: hidden;

transition: opacity 0.3s;

}

.dropdown-menu a:hover {

background-color: #f7f7f7;

}

.dashboard-container {

margin-left: 250px;

margin-top: 60px;

padding: 20px;

width: calc(100% - 250px);

}

.footer {

background-color: #4b6584;

color: #fff;

text-align: center;

padding: 10px;

position: fixed;

bottom: 0;

width: calc(100% - 250px);

margin-left: 250px;

}

</style>

</head>

<body>

<nav class="sidebar">

<ul class="nav-links">

<li><a href="/dashboard"><i class="fas fa-tachometer-alt"></i> Dashboard</a></li>

<li class="dropdown">

<a href="#"><i class="fas fa-school"></i> Primaria</a>

<ul class="dropdown-content">

<li><a href="/primaria/grado1"><i class="fas fa-child"></i> 1er Grado</a></li>

<li><a href="/primaria/grado2"><i class="fas fa-child"></i> 2do Grado</a></li>

<li><a href="/primaria/grado3"><i class="fas fa-child"></i> 3er Grado</a></li>

</ul>

</li>

<li class="dropdown">

<a href="#"><i class="fas fa-chalkboard"></i> Básico</a>

<ul class="dropdown-content">

<li><a href="/basico/grado1"><i class="fas fa-user-graduate"></i> 1er Básico</a></li>

<li><a href="/basico/grado2"><i class="fas fa-user-graduate"></i> 2do Básico</a></li>

<li><a href="/basico/grado3"><i class="fas fa-user-graduate"></i> 3er Básico</a></li>

</ul>

</li>

<li class="dropdown">

<a href="#"><i class="fas fa-university"></i> Diversificado</a>

<ul class="dropdown-content">

<li><a href="/diversificado/grado1"><i class="fas fa-graduation-cap"></i> 1er Diversificado</a></li>

<li><a href="/diversificado/grado2"><i class="fas fa-graduation-cap"></i> 2do Diversificado</a></li>

</ul>

</li>

<li><a href="/usuarios"><i class="fas fa-users"></i> Usuarios</a></li>

</ul>

</nav>

<header class="header">

<h1>CENTRO ESCOLAR CECEC</h1>

</header>

<div class="dashboard-container">

<h2>Contenido del Dashboard</h2>

</div>

<div class="footer">

© 2024 ING. Frida Berlinda Garcia

</div>

</body>

</html>

<% if (typeof grados !== 'undefined' && grados.length > 0) { %>

                <% grados.forEach(function(grado) { %>

**Table:calificaciones**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_calificacion** | int(11) AI PK |
| **id\_estudiante** | int(11) |
| **id\_curso** | int(11) |
| calificacion | decimal(5,2) |

**Table:cursos**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_curso** | int(11) AI PK |
| nombre\_curso | varchar(100) |
| descripcion | text |
| **id\_grado** | int(11) |

**Table:estudiantes**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_estudiante** | int(11) AI PK |
| nombre | varchar(100) |
| email | varchar(100) |
| fecha\_nacimiento | date |
| direccion | text |
| telefono | varchar(20) |

**Table:estudiantes\_grados**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_estudiante\_grado** | int(11) AI PK |
| **id\_estudiante** | int(11) |
| **id\_grado** | int(11) |
| **id\_seccion** | int(11) |
| anio\_escolar | int(4) |

**Table:grados**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_grado** | int(11) AI PK |
| nombre\_grado | varchar(100) |
| nivel\_academico | varchar(50) |

**Table:horarios**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_horario** | int(11) AI PK |
| **id\_curso** | int(11) |
| dia\_semana | varchar(10) |
| hora\_inicio | time |
| hora\_fin | time |

**Table:inscripciones**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_inscripcion** | int(11) AI PK |
| **id\_estudiante** | int(11) |
| **id\_curso** | int(11) |

**Table:pagos**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_pago** | int(11) AI PK |
| fecha\_pago | date |
| monto | decimal(10,2) |
| **id\_estudiante** | int(11) |
| **id\_usuario** | int(11) |

**Table:profesores**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_profesor** | int(11) AI PK |
| nombre | varchar(100) |
| email | varchar(100) |
| especialidad | varchar(100) |
| experiencia\_years | int(11) |
| fecha\_ingreso | date |

**Table:profesores\_grados**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_profesor\_grado** | int(11) AI PK |
| **id\_usuario** | int(11) |
| **id\_grado** | int(11) |
| **id\_seccion** | int(11) |

**Table:recibos**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_recibo** | int(11) AI PK |
| fecha\_emision | date |
| monto | decimal(10,2) |
| **id\_estudiante** | int(11) |
| **id\_usuario** | int(11) |

**Table:reportes**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_reporte** | int(11) AI PK |
| tipo\_reporte | varchar(100) |
| fecha\_generacion | date |
| contenido | text |
| **id\_usuario** | int(11) |

**Table:roles**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_rol** | int(11) AI PK |
| **nombre\_rol** | enum('administrador','profesor') |

**Table:secciones**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_seccion** | int(11) AI PK |
| **id\_grado** | int(11) |
| nombre\_seccion | varchar(50) |

**Table:usuarios**  
  
**Columns:**

|  |  |
| --- | --- |
| **id\_usuario** | int(11) AI PK |
| nombre\_usuario | varchar(100) |
| email | varchar(100) |
| contraseÃ±a | varchar(255) |
| estado | enum('activo','inactivo') |
| **id\_rol** | int(11) |
| telefono | varchar(15) |
| direccion | varchar(255) |
| fecha\_nacimiento | date |
| genero | enum('M','F') |
| foto | varchar(255) |

$('#gradeForm').on('submit', function(e) {

            e.preventDefault(); // Previene el envío por defecto del formulario

            const data = $(this).serialize(); // Serializa los datos del formulario

            const id\_grado = $('#gradeId').val();

            const url = id\_grado ? `/api/grados/update` : `/api/grados/create`; // Determina la URL dependiendo si es creación o actualización

            $.ajax({

                url: url,

                method: 'POST',

                data: data,

                success: function(result) {

                    if (result.success) {

                        loadGrades(); // Recarga los datos en la tabla

                        $('#gradeModal').modal('hide'); // Cierra el modal

                        $('.modal-backdrop').remove(); // Elimina el backdrop

                    } else {

                        alert('Error al guardar el grado.');

                    }

                },

                error: function(jqXHR, textStatus, errorThrown) {

                    console.error('Error en la solicitud:', textStatus, errorThrown);

                    alert('Error al guardar el grado.');

                }

            });

        });

// Función para abrir el modal de eliminar

function deleteGrade(id) {

    $('#deleteGradeId').val(id);

    $('#deleteGradeModal').modal('show');

}

$('#confirmDeleteBtn').on('click', function() {

    const id = $('#deleteGradeId').val();

    $.ajax({

        url: '/api/grados/delete',

        method: 'POST',

        data: { id\_grado: id },

        success: function(result) {

            if (result.success) {

                loadGrades(); // Recarga los datos en la tabla

                $('#deleteGradeModal').modal('hide');

            } else {

                alert('Error al eliminar el grado.');

            }

        },

        error: function(jqXHR, textStatus, errorThrown) {

            console.error('Error en la solicitud:', textStatus, errorThrown);

            alert('Error al eliminar el grado.');

        }

    });

});

        // Función para cargar datos en el modal de editar

        function editGrade(id) {

            console.log('Editando grado con ID:', id);

            $.get(`/api/grados/${id}`, function(grado) {

                console.log('Datos del grado:', grado);

                $('#editGradeId').val(grado.id\_grado);

                $('#editNombreGrado').val(grado.nombre\_grado);

                $('#editNivelAcademico').val(grado.nivel\_academico);

                $('#editSecciones').val(grado.secciones || '');

                $('#editGradeModal').modal('show');

            }).fail(function(jqXHR, textStatus, errorThrown) {

                console.error('Error al cargar los datos del grado:', textStatus, errorThrown);

                alert('Error al cargar los datos del grado.');

            });

        }

        $(document).ready(function() {

            function editGrade(id) {

                console.log('Editando grado con ID:', id);

                $.get(`/api/grados/${id}`, function(grado) {

                    console.log('Datos del grado:', grado);

                    $('#editGradeId').val(grado.id\_grado);

                    $('#editNombreGrado').val(grado.nombre\_grado);

                    $('#editNivelAcademico').val(grado.nivel\_academico);

                    $('#editSecciones').val(grado.secciones || '');

                    $('#editGradeModal').modal('show');

                }).fail(function(jqXHR, textStatus, errorThrown) {

                    console.error('Error al cargar los datos del grado:', textStatus, errorThrown);

                    alert('Error al cargar los datos del grado.');

                });

            }

            $('#editGradeForm').on('submit', function(e) {

                e.preventDefault();

                const data = $(this).serialize();

                $.ajax({

                    url: '/api/grados/update',

                    method: 'POST',

                    data: data,

                    success: function(result) {

                        if (result.success) {

                            loadGrades(); // Recarga los datos en la tabla

                            $('#editGradeModal').modal('hide');

                        } else {

                            alert('Error al guardar los cambios.');

                        }

                    },

                    error: function(jqXHR, textStatus, errorThrown) {

                        console.error('Error en la solicitud:', textStatus, errorThrown);

                        alert('Error al guardar los cambios.');

                    }

                });

            });

        });

function loadGrades() {

    $.get('/api/grados', function(grados) {

        console.log(grados); // Verifica la salida en la consola del navegador

        const tbody = $('#gradeTableBody');

        tbody.empty();

        grados.forEach(grado => {

            const secciones = grado.secciones ? grado.secciones.replace(/,/g, ', ') : 'No hay secciones';

            const row = `

                <tr>

                    <td>${grado.id\_grado}</td>

                    <td>${grado.nombre\_grado}</td>

                    <td>${grado.nivel\_academico}</td>

                    <td>${secciones}</td>

                    <td>

                        <button class="btn btn-warning btn-sm" onclick="editGrade(${grado.id\_grado})">Editar</button>

                        <button class="btn btn-danger btn-sm" onclick="deleteGrade(${grado.id\_grado})">Eliminar</button>

                    </td>

                </tr>

            `;

            tbody.append(row);

        });

    }).fail(function(jqXHR, textStatus, errorThrown) {

        console.error('Error en la solicitud:', textStatus, errorThrown);

        alert('Error al cargar los grados.');

    });

}

        $(document).ready(function() {

            loadGrades(); // Llamamos a loadGrades() cuando el documento está listo

        });

Modulo de alumnos

<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

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

  <title>Gestión de Estudiantes</title>

  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

  <style>

    /\* Estilos de los mensajes emergentes \*/

#messageBox {

  position: fixed;

  top: 20px;

  right: 20px;

  z-index: 1050;

  width: 300px;

}

    .pagination {

      display: flex;

      list-style: none;

    }

    .pagination li {

      margin: 0 5px;

    }

    .pagination button {

      padding: 8px 12px;

      border: 1px solid #ddd;

      background-color: white;

      cursor: pointer;

      border-radius: 5px;

    }

    .pagination button:hover {

      background-color: #f0f0f0;

    }

    .pagination button.active {

      background-color: #007bff;

      color: white;

      border-color: #007bff;

    }

    .pagination button.disabled {

      cursor: not-allowed;

      opacity: 0.5;

    }

  </style>

</head>

<body>

  <div class="container mt-4">

    <h1>Gestión de Estudiantes</h1>

    <button class="btn btn-primary mb-3" data-toggle="modal" data-target="#modalAgregarEstudiante">Agregar Estudiante</button>

    <!-- Tabla para listar estudiantes -->

    <table class="table table-bordered">

      <thead>

        <tr>

          <th>ID</th>

          <th>Nombre</th>

          <th>Email</th>

          <th>Fecha de Nacimiento</th>

          <th>Dirección</th>

          <th>Teléfono</th>

          <th>Grado</th>

          <th>Sección</th>

          <th>Año Escolar</th>

          <th>Acciones</th>

        </tr>

      </thead>

      <tbody id="tablaEstudiantes">

        <!-- Las filas se llenarán dinámicamente aquí -->

      </tbody>

    </table>

    <!-- Paginación -->

    <nav aria-label="Page navigation">

      <ul class="pagination justify-content-end" id="pagination">

          <!-- Los botones de paginación se cargarán aquí -->

      </ul>

  </nav>

  </div>

 <!-- Modal de Agregar Estudiante -->

<div class="modal fade" id="modalAgregarEstudiante" tabindex="-1" role="dialog" aria-labelledby="modalAgregarEstudianteLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="modalAgregarEstudianteLabel">Agregar Estudiante</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <form id="formAgregarEstudiante">

          <!-- Datos del Alumno -->

          <div class="form-group">

            <label for="nombre">Nombre</label>

            <input type="text" class="form-control" id="nombre" name="nombre" required>

          </div>

          <div class="form-group">

            <label for="email">Email</label>

            <input type="email" class="form-control" id="email" name="email" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento" name="fecha\_nacimiento" required>

          </div>

          <div class="form-group">

            <label for="direccion">Dirección</label>

            <textarea class="form-control" id="direccion" name="direccion" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono">Teléfono</label>

            <input type="text" class="form-control" id="telefono" name="telefono" required>

          </div>

          <div class="form-group">

            <label for="grado">Grado</label>

            <select class="form-control" id="grado" name="grado" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_modal">Sección</label>

            <select class="form-control" id="seccion\_modal" name="seccion\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar" name="anio\_escolar" required>

          </div>

          <!-- Línea de Separación y Título -->

          <hr>

          <!-- Datos de los Padres -->

          <h6>Datos del Padre</h6>

          <div class="form-row">

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_padre\_edit">Nombre del Padre</label>

                <input type="text" class="form-control" id="nombre\_padre\_edit" name="nombre\_padre\_edit">

              </div>

              <div class="form-group">

                <label for="telefono\_padre\_edit">Teléfono del Padre</label>

                <input type="text" class="form-control" id="telefono\_padre\_edit" name="telefono\_padre\_edit">

              </div>

            </div>

            <div class="col-md-6">

              <div class="form-group">

                <label for="email\_padre\_edit">Email del Padre</label>

                <input type="email" class="form-control" id="email\_padre\_edit" name="email\_padre\_edit">

              </div>

            </div>

          </div>

          <h6>Datos de la Madre</h6>

          <div class="form-row">

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_madre\_edit">Nombre de la Madre</label>

                <input type="text" class="form-control" id="nombre\_madre\_edit" name="nombre\_madre\_edit">

              </div>

              <div class="form-group">

                <label for="telefono\_madre\_edit">Teléfono de la Madre</label>

                <input type="text" class="form-control" id="telefono\_madre\_edit" name="telefono\_madre\_edit">

              </div>

            </div>

            <div class="col-md-6">

              <div class="form-group">

                <label for="email\_madre\_edit">Email de la Madre</label>

                <input type="email" class="form-control" id="email\_madre\_edit" name="email\_madre\_edit">

              </div>

            </div>

          </div>

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>

          <button type="submit" class="btn btn-primary">Agregar Estudiante</button>

        </form>

      </div>

    </div>

  </div>

</div>

 <!-- Modal de Editar Estudiante -->

<!-- Modal de Editar Estudiante -->

<div class="modal fade" id="modalEditarEstudiante" tabindex="-1" role="dialog" aria-labelledby="modalEditarEstudianteLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="modalEditarEstudianteLabel">Editar Estudiante</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <form id="formEditarEstudiante">

          <!-- Datos del Estudiante -->

          <div class="form-group">

            <label for="nombre\_edit">Nombre</label>

            <input type="text" class="form-control" id="nombre\_edit" name="nombre\_edit" required>

          </div>

          <div class="form-group">

            <label for="email\_edit">Email</label>

            <input type="email" class="form-control" id="email\_edit" name="email\_edit" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento\_edit">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento\_edit" name="fecha\_nacimiento\_edit" required>

          </div>

          <div class="form-group">

            <label for="direccion\_edit">Dirección</label>

            <textarea class="form-control" id="direccion\_edit" name="direccion\_edit" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono\_edit">Teléfono</label>

            <input type="text" class="form-control" id="telefono\_edit" name="telefono\_edit" required>

          </div>

          <div class="form-group">

            <label for="grado\_edit">Grado</label>

            <select class="form-control" id="grado\_edit" name="grado\_edit" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_edit\_modal">Sección</label>

            <select class="form-control" id="seccion\_edit\_modal" name="seccion\_edit\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar\_edit">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar\_edit" name="anio\_escolar\_edit" required>

          </div>

          <!-- Línea de separación -->

          <hr>

          <!-- Datos de los Padres -->

          <h6>Datos del Padre</h6>

          <div class="form-row">

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_padre\_edit">Nombre del Padre</label>

                <input type="text" class="form-control" id="nombre\_padre\_edit" name="nombre\_padre\_edit">

              </div>

              <div class="form-group">

                <label for="telefono\_padre\_edit">Teléfono del Padre</label>

                <input type="text" class="form-control" id="telefono\_padre\_edit" name="telefono\_padre\_edit">

              </div>

            </div>

            <div class="col-md-6">

              <div class="form-group">

                <label for="email\_padre\_edit">Email del Padre</label>

                <input type="email" class="form-control" id="email\_padre\_edit" name="email\_padre\_edit">

              </div>

            </div>

          </div>

          <h6>Datos de la Madre</h6>

          <div class="form-row">

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_madre\_edit">Nombre de la Madre</label>

                <input type="text" class="form-control" id="nombre\_madre\_edit" name="nombre\_madre\_edit">

              </div>

              <div class="form-group">

                <label for="telefono\_madre\_edit">Teléfono de la Madre</label>

                <input type="text" class="form-control" id="telefono\_madre\_edit" name="telefono\_madre\_edit">

              </div>

            </div>

            <div class="col-md-6">

              <div class="form-group">

                <label for="email\_madre\_edit">Email de la Madre</label>

                <input type="email" class="form-control" id="email\_madre\_edit" name="email\_madre\_edit">

              </div>

            </div>

          </div>

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>

          <button type="submit" class="btn btn-primary">Guardar Cambios</button>

        </form>

      </div>

    </div>

  </div>

</div>

<!-- Añade esto para mostrar mensajes -->

<div id="messageBox" class="alert" style="display:none;"></div>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <script>

    function showMessage(type, message) {

      const messageBox = document.getElementById('messageBox');

      messageBox.className = `alert alert-${type}`;

      messageBox.innerText = message;

      messageBox.style.display = 'block'; // Muestra el mensaje

      setTimeout(() => messageBox.style.display = 'none', 3000); // Oculta el mensaje después de 3 segundos

  }

    document.addEventListener('DOMContentLoaded', () => {

      const estudiantesPorPagina = 10;

  let estudiantes = [];

  let paginaActual = 1;

  async function cargarEstudiantes() {

    try {

      const response = await fetch('/api/estudiantes');

      if (!response.ok) throw new Error('Error en la respuesta de la API');

      estudiantes = await response.json();

      console.log('Estudiantes cargados:', estudiantes);

      mostrarPagina(paginaActual);

      crearBotonesDePaginacion();

    } catch (error) {

      console.error('Error al cargar estudiantes:', error);

    }

  }

  function mostrarPagina(pagina) {

    const inicio = (pagina - 1) \* estudiantesPorPagina;

    const fin = inicio + estudiantesPorPagina;

    const estudiantesPagina = estudiantes.slice(inicio, fin);

    const tbody = document.getElementById('tablaEstudiantes');

    tbody.innerHTML = '';

    estudiantesPagina.forEach(estudiante => {

      const tr = document.createElement('tr');

      tr.innerHTML = `

        <td>${estudiante.id\_estudiante}</td>

        <td>${estudiante.nombre}</td>

        <td>${estudiante.email}</td>

        <td>${formatearFecha(estudiante.fecha\_nacimiento)}</td>

        <td>${estudiante.direccion}</td>

        <td>${estudiante.telefono}</td>

        <td>${estudiante.nombre\_grado}</td>

        <td>${estudiante.nombre\_seccion}</td>

        <td>${estudiante.anio\_escolar}</td>

        <td>

          <button type="button" class="btn btn-warning btn-sm" data-toggle="modal" data-target="#modalEditarEstudiante">Editar</button>

          <button type="button" data-view-ficha data-id="${estudiante.id\_estudiante}" class="btn btn-info btn-sm">Ver Ficha Completa</button>

          <button type="submit" class="btn btn-danger btn-sm">Eliminar</button>

        </td>

      `;

      tbody.appendChild(tr);

    });

  }

  function crearBotonesDePaginacion() {

    const totalPaginas = Math.ceil(estudiantes.length / estudiantesPorPagina);

    const paginacionDiv = document.getElementById('pagination');

    paginacionDiv.innerHTML = '';

    // Botón Anterior

    const anteriorBtn = document.createElement('button');

    anteriorBtn.textContent = 'Anterior';

    anteriorBtn.classList.add('page-link');

    if (paginaActual === 1) {

      anteriorBtn.classList.add('disabled');

      anteriorBtn.setAttribute('disabled', true);

    }

    anteriorBtn.addEventListener('click', () => {

      if (paginaActual > 1) {

        paginaActual--;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(anteriorBtn);

    // Botones numéricos

    for (let i = 1; i <= totalPaginas; i++) {

      const boton = document.createElement('button');

      boton.textContent = i;

      boton.classList.add('page-link');

      if (i === paginaActual) {

        boton.classList.add('active');

      }

      boton.addEventListener('click', () => {

        paginaActual = i;

        mostrarPagina(i);

        crearBotonesDePaginacion();

      });

      paginacionDiv.appendChild(boton);

    }

    // Botón Siguiente

    const siguienteBtn = document.createElement('button');

    siguienteBtn.textContent = 'Siguiente';

    siguienteBtn.classList.add('page-link');

    if (paginaActual === totalPaginas) {

      siguienteBtn.classList.add('disabled');

      siguienteBtn.setAttribute('disabled', true);

    }

    siguienteBtn.addEventListener('click', () => {

      if (paginaActual < totalPaginas) {

        paginaActual++;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(siguienteBtn);

  }

      async function cargarGrados() {

        try {

          const response = await fetch('/api/grados');

          if (!response.ok) throw new Error('Error al cargar grados');

          const grados = await response.json();

          const gradoSelects = [document.getElementById('grado'), document.getElementById('grado\_edit')];

          gradoSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione un grado</option>'; // Limpiar opciones

            grados.forEach(grado => {

              const option = document.createElement('option');

              option.value = grado.id\_grado;

              option.textContent = grado.nombre\_grado; // Asegúrate de que 'nombre\_grado' sea el nombre correcto en la BD

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar grados:', error);

        }

      }

      async function cargarSecciones(gradoId) {

        try {

          // Muestra el ID del grado en la consola para depuración

          console.log('Cargando secciones para el grado ID:', gradoId);

          // Verifica que el ID del grado no esté vacío

          if (!gradoId) {

            console.warn('No se seleccionó un grado. Limpiando secciones.');

            const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

            seccionSelects.forEach(select => {

              select.innerHTML = '<option value="">Seleccione una sección</option>';

            });

            return;

          }

          // Realiza la solicitud para obtener las secciones

          const response = await fetch(`/api/secciones?grado=${gradoId}`);

          // Verifica si la respuesta es satisfactoria

          if (!response.ok) {

            throw new Error('Error al cargar secciones');

          }

          // Procesa la respuesta

          const secciones = await response.json();

          console.log('Secciones cargadas:', secciones);

          // Actualiza los selectores de secciones

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>'; // Limpiar opciones

            secciones.forEach(seccion => {

              const option = document.createElement('option');

              option.value = seccion.id\_seccion;

              option.textContent = seccion.nombre\_seccion;

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar secciones:', error);

        }

      }

      // Event listener para el select de grados

      document.getElementById('grado').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        console.log('Grado seleccionado:', gradoId); // Mensaje de depuración

        cargarSecciones(gradoId);

      });

      function formatearFecha(fechaISO) {

        const fecha = new Date(fechaISO);

        const dia = fecha.getDate().toString().padStart(2, '0');

        const mesesAbreviados = ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"];

        const mes = mesesAbreviados[fecha.getMonth()];

        const anio = fecha.getFullYear();

        return `${dia} ${mes} ${anio}`;

      }

      function abrirModalEditar(idEstudiante) {

        fetch(`/api/estudiantes/${idEstudiante}`)

          .then(response => response.json())

          .then(estudiante => {

            // Rellena los campos del formulario con los datos del estudiante

            document.getElementById('nombre\_edit').value = estudiante.nombre;

            document.getElementById('email\_edit').value = estudiante.email;

            document.getElementById('fecha\_nacimiento\_edit').value = estudiante.fecha\_nacimiento;

            document.getElementById('direccion\_edit').value = estudiante.direccion;

            document.getElementById('telefono\_edit').value = estudiante.telefono;

            document.getElementById('anio\_escolar\_edit').value = estudiante.anio\_escolar;

            // Rellena el campo del grado

            const gradoSelect = document.getElementById('grado\_edit');

            gradoSelect.value = estudiante.grado;

            // Carga y actualiza las secciones cuando el grado se haya establecido

      cargarSecciones(estudiante.grado).then(() => {

        const seccionSelect = document.getElementById('seccion\_edit\_modal');

        seccionSelect.value = estudiante.seccion; // Establece la sección seleccionada

      });

            // Muestra el modal

            $('#modalEditarEstudiante').modal('show');

          })

          .catch(error => {

            console.error('Error al cargar los datos del estudiante:', error);

            showMessage('danger', 'Hubo un error al cargar los datos del estudiante.');

          });

      }

      // Event listener para el formulario de editar estudiante

document.getElementById('formEditarEstudiante').addEventListener('submit', async (event) => {

  event.preventDefault(); // Previene el envío por defecto del formulario

  const formData = new FormData(event.target);

  const data = {};

  // Recolecta los datos del formulario en un objeto

  formData.forEach((value, key) => {

    data[key] = value;

  });

  try {

    // Envía la solicitud PUT al servidor para actualizar el estudiante

    const response = await fetch('/api/estudiantes/editar', {

      method: 'PUT',

      headers: {

        'Content-Type': 'application/json',

      },

      body: JSON.stringify(data),

    });

    // Verifica si la respuesta es satisfactoria

    if (!response.ok) {

      const errorData = await response.json();

      throw new Error(errorData.message || 'Error al actualizar el estudiante');

    }

    // Procesa la respuesta del servidor

    const result = await response.json();

    console.log('Éxito:', result);

    // Muestra mensaje de éxito

    showMessage('success', 'Estudiante actualizado correctamente.');

    // Cierra el modal

    $('#modalEditarEstudiante').modal('hide');

    // Actualiza la lista de estudiantes

    await cargarEstudiantes(); // Espera a que la lista se actualice

  } catch (error) {

    console.error('Error:', error);

    // Muestra mensaje de error con información detallada

    showMessage('danger', `Hubo un error al actualizar el estudiante: ${error.message}`);

  }

});

      // Cargar datos iniciales

      cargarEstudiantes();

      cargarGrados();

      // Añadir el evento para cargar secciones al cambiar el grado

      document.getElementById('grado').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        if (gradoId) {

          cargarSecciones(gradoId);

        } else {

          // Si no hay grado seleccionado, limpiar las opciones de sección

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>';

          });

        }

      });

      // Event listener para el formulario de agregar estudiante

      document.getElementById('formAgregarEstudiante').addEventListener('submit', async (event) => {

        event.preventDefault(); // Previene el envío por defecto del formulario

        const formData = new FormData(event.target);

        const data = {};

        // Solo incluimos los datos que queremos enviar al servidor

        const camposRelevantes = ['nombre', 'email', 'fecha\_nacimiento', 'direccion', 'telefono', 'grado', 'seccion\_modal', 'anio\_escolar'];

        formData.forEach((value, key) => {

          if (camposRelevantes.includes(key)) {

            data[key] = value;

          }

        });

        try {

          const response = await fetch('/estudiantes', {

            method: 'POST',

            headers: {

              'Content-Type': 'application/json',

            },

            body: JSON.stringify(data),

          });

          if (!response.ok) {

            throw new Error('Error al guardar el estudiante');

          }

          const result = await response.json();

          console.log('Éxito:', result);

          // Mostrar mensaje de éxito

          showMessage('success', 'Estudiante agregado correctamente.');

          // Cierra el modal

          $('#modalAgregarEstudiante').modal('hide');

          // Actualiza la lista de estudiantes

          await cargarEstudiantes(); // Espera a que la lista se actualice

        } catch (error) {

          console.error('Error:', error);

          // Mostrar mensaje de error

          showMessage('danger', 'Hubo un error al agregar el estudiante.');

        }

      });

      // Manejo de botones para abrir el modal

  document.querySelectorAll('[data-open-modal]').forEach(button => {

    button.addEventListener('click', function() {

      const idEstudiante = this.getAttribute('data-id');

      abrirModalEditar(idEstudiante);

    });

  });

});

  </script>

</body>

</html>

async function cargarSecciones(gradoId) {

try {

console.log('Cargando secciones para el grado ID:', gradoId);

if (!gradoId) {

console.warn('No se seleccionó un grado. Limpiando secciones.');

const seccionSelect = document.getElementById('seccion\_edit\_modal');

seccionSelect.innerHTML = '<option value="">Seleccione una sección</option>';

return;

}

const response = await fetch(`/api/secciones?grado=${gradoId}`);

if (!response.ok) {

throw new Error('Error al cargar secciones');

}

const secciones = await response.json();

console.log('Secciones cargadas:', secciones);

const seccionSelect = document.getElementById('seccion\_edit\_modal');

seccionSelect.innerHTML = '<option value="">Seleccione una sección</option>'; // Limpiar opciones

secciones.forEach(seccion => {

const option = document.createElement('option');

option.value = seccion.id\_seccion;

option.textContent = seccion.nombre\_seccion;

seccionSelect.appendChild(option);

});

} catch (error) {

console.error('Error al cargar secciones:', error);

}

}

// Event listener para el select de grados

document.getElementById('grado\_edit').addEventListener('change', (event) => {

const gradoId = event.target.value;

console.log('Grado seleccionado:', gradoId);

cargarSecciones(gradoId);

});

Modulo alumnos, es el bueno

<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

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

  <title>Gestión de Estudiantes</title>

  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

  <style>

    /\* Estilos de los mensajes emergentes \*/

#messageBox {

  position: fixed;

  top: 20px;

  right: 20px;

  z-index: 1050;

  width: 300px;

}

    .pagination {

      display: flex;

      list-style: none;

    }

    .pagination li {

      margin: 0 5px;

    }

    .pagination button {

      padding: 8px 12px;

      border: 1px solid #ddd;

      background-color: white;

      cursor: pointer;

      border-radius: 5px;

    }

    .pagination button:hover {

      background-color: #f0f0f0;

    }

    .pagination button.active {

      background-color: #007bff;

      color: white;

      border-color: #007bff;

    }

    .pagination button.disabled {

      cursor: not-allowed;

      opacity: 0.5;

    }

  </style>

</head>

<body>

  <div class="container mt-4">

    <h1>Gestión de Estudiantes</h1>

    <button class="btn btn-primary mb-3" data-toggle="modal" data-target="#modalAgregarEstudiante">Agregar Estudiante</button>

    <!-- Tabla para listar estudiantes -->

    <table class="table table-bordered">

      <thead>

        <tr>

          <th>ID</th>

          <th>Nombre</th>

          <th>Email</th>

          <th>Fecha de Nacimiento</th>

          <th>Dirección</th>

          <th>Teléfono</th>

          <th>Grado</th>

          <th>Sección</th>

          <th>Año Escolar</th>

          <th>Acciones</th>

        </tr>

      </thead>

      <tbody id="tablaEstudiantes">

        <!-- Las filas se llenarán dinámicamente aquí -->

      </tbody>

    </table>

    <!-- Paginación -->

    <nav aria-label="Page navigation">

      <ul class="pagination justify-content-end" id="pagination">

          <!-- Los botones de paginación se cargarán aquí -->

      </ul>

  </nav>

  </div>

 <!-- Modal de Agregar Estudiante -->

<div class="modal fade" id="modalAgregarEstudiante" tabindex="-1" role="dialog" aria-labelledby="modalAgregarEstudianteLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="modalAgregarEstudianteLabel">Agregar Estudiante</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <form id="formAgregarEstudiante">

          <!-- Datos del Alumno -->

          <div class="form-group">

            <label for="nombre">Nombre</label>

            <input type="text" class="form-control" id="nombre" name="nombre" required>

          </div>

          <div class="form-group">

            <label for="email">Email</label>

            <input type="email" class="form-control" id="email" name="email" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento" name="fecha\_nacimiento" required>

          </div>

          <div class="form-group">

            <label for="direccion">Dirección</label>

            <textarea class="form-control" id="direccion" name="direccion" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono">Teléfono</label>

            <input type="text" class="form-control" id="telefono" name="telefono" required>

          </div>

          <div class="form-group">

            <label for="grado">Grado</label>

            <select class="form-control" id="grado" name="grado" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_modal">Sección</label>

            <select class="form-control" id="seccion\_modal" name="seccion\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar" name="anio\_escolar" required>

          </div>

          <!-- Línea de Separación y Título -->

          <hr class="my-4">

          <h6>Datos de los Padres</h6>

          <!-- Datos de los Padres -->

          <div class="row">

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_padre">Nombre del Padre</label>

                <input type="text" class="form-control" id="nombre\_padre" name="nombre\_padre">

              </div>

              <div class="form-group">

                <label for="telefono\_padre">Teléfono del Padre</label>

                <input type="text" class="form-control" id="telefono\_padre" name="telefono\_padre">

              </div>

              <div class="form-group">

                <label for="email\_padre">Email del Padre</label>

                <input type="email" class="form-control" id="email\_padre" name="email\_padre">

              </div>

            </div>

            <div class="col-md-6">

              <div class="form-group">

                <label for="nombre\_madre">Nombre de la Madre</label>

                <input type="text" class="form-control" id="nombre\_madre" name="nombre\_madre">

              </div>

              <div class="form-group">

                <label for="telefono\_madre">Teléfono de la Madre</label>

                <input type="text" class="form-control" id="telefono\_madre" name="telefono\_madre">

              </div>

              <div class="form-group">

                <label for="email\_madre">Email de la Madre</label>

                <input type="email" class="form-control" id="email\_madre" name="email\_madre">

              </div>

            </div>

          </div>

          <button type="button" class="btn btn-secondary" data-dismiss="modal">Cerrar</button>

          <button type="submit" class="btn btn-primary">Agregar Estudiante</button>

        </form>

      </div>

    </div>

  </div>

</div>

 <!-- Modal de Editar Estudiante -->

<!-- Modal de Editar Estudiante -->

<div class="modal fade" id="modalEditarEstudiante" tabindex="-1" role="dialog" aria-labelledby="modalEditarEstudianteLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="modalEditarEstudianteLabel">Editar Estudiante</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <form id="formEditarEstudiante">

          <!-- Datos del Estudiante -->

          <div class="form-group">

            <label for="nombre\_edit">Nombre</label>

            <input type="text" class="form-control" id="nombre\_edit" name="nombre\_edit" required>

          </div>

          <div class="form-group">

            <label for="email\_edit">Email</label>

            <input type="email" class="form-control" id="email\_edit" name="email\_edit" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento\_edit">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento\_edit" name="fecha\_nacimiento\_edit" required>

          </div>

          <div class="form-group">

            <label for="direccion\_edit">Dirección</label>

            <textarea class="form-control" id="direccion\_edit" name="direccion\_edit" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono\_edit">Teléfono</label>

            <input type="text" class="form-control" id="telefono\_edit" name="telefono\_edit" required>

          </div>

          <div class="form-group">

            <label for="grado\_edit">Grado</label>

            <select class="form-control" id="grado\_edit" name="grado\_edit" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_edit\_modal">Sección</label>

            <select class="form-control" id="seccion\_edit\_modal" name="seccion\_edit\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar\_edit">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar\_edit" name="anio\_escolar\_edit" required>

          </div>

          <!-- Línea de separación -->

          <hr>

          <!-- Datos de los Padres -->

          <div class="form-group">

            <h6>Datos del Padre</h6>

            <label for="nombre\_padre\_edit">Nombre del Padre</label>

            <input type="text" class="form-control" id="nombre\_padre\_edit" name="nombre\_padre\_edit">

          </div>

          <div class="form-group">

            <label for="telefono\_padre\_edit">Teléfono del Padre</label>

            <input type="text" class="form-control" id="telefono\_padre\_edit" name="telefono\_padre\_edit">

          </div>

          <div class="form-group">

            <label for="email\_padre\_edit">Email del Padre</label>

            <input type="email" class="form-control" id="email\_padre\_edit" name="email\_padre\_edit">

          </div>

          <div class="form-group">

            <h6>Datos de la Madre</h6>

            <label for="nombre\_madre\_edit">Nombre de la Madre</label>

            <input type="text" class="form-control" id="nombre\_madre\_edit" name="nombre\_madre\_edit">

          </div>

          <div class="form-group">

            <label for="telefono\_madre\_edit">Teléfono de la Madre</label>

            <input type="text" class="form-control" id="telefono\_madre\_edit" name="telefono\_madre\_edit">

          </div>

          <div class="form-group">

            <label for="email\_madre\_edit">Email de la Madre</label>

            <input type="email" class="form-control" id="email\_madre\_edit" name="email\_madre\_edit">

          </div>

          <button type="submit" class="btn btn-primary">Guardar Cambios</button>

        </form>

      </div>

    </div>

  </div>

</div>

<!-- Añade esto para mostrar mensajes -->

<div id="messageBox" class="alert" style="display:none;"></div>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <script>

    function showMessage(type, message) {

      const messageBox = document.getElementById('messageBox');

      messageBox.className = `alert alert-${type}`;

      messageBox.innerText = message;

      messageBox.style.display = 'block'; // Muestra el mensaje

      setTimeout(() => messageBox.style.display = 'none', 3000); // Oculta el mensaje después de 3 segundos

  }

    document.addEventListener('DOMContentLoaded', () => {

      const estudiantesPorPagina = 10;

  let estudiantes = [];

  let paginaActual = 1;

  async function cargarEstudiantes() {

    try {

      const response = await fetch('/api/estudiantes');

      if (!response.ok) throw new Error('Error en la respuesta de la API');

      estudiantes = await response.json();

      console.log('Estudiantes cargados:', estudiantes);

      mostrarPagina(paginaActual);

      crearBotonesDePaginacion();

    } catch (error) {

      console.error('Error al cargar estudiantes:', error);

    }

  }

  function mostrarPagina(pagina) {

    const inicio = (pagina - 1) \* estudiantesPorPagina;

    const fin = inicio + estudiantesPorPagina;

    const estudiantesPagina = estudiantes.slice(inicio, fin);

    const tbody = document.getElementById('tablaEstudiantes');

    tbody.innerHTML = '';

    estudiantesPagina.forEach(estudiante => {

      const tr = document.createElement('tr');

      tr.innerHTML = `

        <td>${estudiante.id\_estudiante}</td>

        <td>${estudiante.nombre}</td>

        <td>${estudiante.email}</td>

        <td>${formatearFecha(estudiante.fecha\_nacimiento)}</td>

        <td>${estudiante.direccion}</td>

        <td>${estudiante.telefono}</td>

        <td>${estudiante.nombre\_grado}</td>

        <td>${estudiante.nombre\_seccion}</td>

        <td>${estudiante.anio\_escolar}</td>

        <td>

          <button type="button" class="btn btn-warning btn-sm" data-toggle="modal" data-target="#modalEditarEstudiante">Editar</button>

          <button type="button" data-view-ficha data-id="${estudiante.id\_estudiante}" class="btn btn-info btn-sm">Ver Ficha Completa</button>

          <button type="submit" class="btn btn-danger btn-sm">Eliminar</button>

        </td>

      `;

      tbody.appendChild(tr);

    });

  }

  function crearBotonesDePaginacion() {

    const totalPaginas = Math.ceil(estudiantes.length / estudiantesPorPagina);

    const paginacionDiv = document.getElementById('pagination');

    paginacionDiv.innerHTML = '';

    // Botón Anterior

    const anteriorBtn = document.createElement('button');

    anteriorBtn.textContent = 'Anterior';

    anteriorBtn.classList.add('page-link');

    if (paginaActual === 1) {

      anteriorBtn.classList.add('disabled');

      anteriorBtn.setAttribute('disabled', true);

    }

    anteriorBtn.addEventListener('click', () => {

      if (paginaActual > 1) {

        paginaActual--;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(anteriorBtn);

    // Botones numéricos

    for (let i = 1; i <= totalPaginas; i++) {

      const boton = document.createElement('button');

      boton.textContent = i;

      boton.classList.add('page-link');

      if (i === paginaActual) {

        boton.classList.add('active');

      }

      boton.addEventListener('click', () => {

        paginaActual = i;

        mostrarPagina(i);

        crearBotonesDePaginacion();

      });

      paginacionDiv.appendChild(boton);

    }

    // Botón Siguiente

    const siguienteBtn = document.createElement('button');

    siguienteBtn.textContent = 'Siguiente';

    siguienteBtn.classList.add('page-link');

    if (paginaActual === totalPaginas) {

      siguienteBtn.classList.add('disabled');

      siguienteBtn.setAttribute('disabled', true);

    }

    siguienteBtn.addEventListener('click', () => {

      if (paginaActual < totalPaginas) {

        paginaActual++;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(siguienteBtn);

  }

      async function cargarGrados() {

        try {

          const response = await fetch('/api/grados');

          if (!response.ok) throw new Error('Error al cargar grados');

          const grados = await response.json();

          const gradoSelects = [document.getElementById('grado'), document.getElementById('grado\_edit')];

          gradoSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione un grado</option>'; // Limpiar opciones

            grados.forEach(grado => {

              const option = document.createElement('option');

              option.value = grado.id\_grado;

              option.textContent = grado.nombre\_grado; // Asegúrate de que 'nombre\_grado' sea el nombre correcto en la BD

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar grados:', error);

        }

      }

      async function cargarSecciones(gradoId) {

        try {

          const response = await fetch(`/api/secciones?grado=${gradoId}`);

          if (!response.ok) throw new Error('Error al cargar secciones');

          const secciones = await response.json();

          console.log('Secciones cargadas:', secciones); // Agrega esto para depurar

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>'; // Limpiar opciones

            secciones.forEach(seccion => {

              const option = document.createElement('option');

              option.value = seccion.id\_seccion;

              option.textContent = seccion.nombre\_seccion;

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar secciones:', error);

        }

      }

      document.getElementById('grado\_edit').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        if (gradoId) {

          cargarSecciones(gradoId, true); // Usamos `true` para indicar que es en el formulario de edición

        } else {

          // Si no hay grado seleccionado, limpiar las opciones de sección

          document.getElementById('seccion\_edit\_modal').innerHTML = '<option value="">Seleccione una sección</option>';

        }

      });

      function formatearFecha(fechaISO) {

        const fecha = new Date(fechaISO);

        const dia = fecha.getDate().toString().padStart(2, '0');

        const mesesAbreviados = ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"];

        const mes = mesesAbreviados[fecha.getMonth()];

        const anio = fecha.getFullYear();

        return `${dia} ${mes} ${anio}`;

      }

      function abrirModalEditar(idEstudiante) {

        fetch(`/api/estudiantes/${idEstudiante}`)

            .then(response => response.json())

            .then(estudiante => {

                // Rellena los campos del formulario con los datos del estudiante

                document.getElementById('nombre\_edit').value = estudiante.nombre;

                document.getElementById('email\_edit').value = estudiante.email;

                document.getElementById('fecha\_nacimiento\_edit').value = estudiante.fecha\_nacimiento;

                document.getElementById('direccion\_edit').value = estudiante.direccion;

                document.getElementById('telefono\_edit').value = estudiante.telefono;

                document.getElementById('grado\_edit').value = estudiante.grado;

                document.getElementById('seccion\_edit\_modal').value = estudiante.seccion;

                document.getElementById('anio\_escolar\_edit').value = estudiante.anio\_escolar;

                // Datos de los padres (si es necesario)

                document.getElementById('nombre\_padre\_edit').value = estudiante.nombre\_padre || '';

                document.getElementById('telefono\_padre\_edit').value = estudiante.telefono\_padre || '';

                document.getElementById('email\_padre\_edit').value = estudiante.email\_padre || '';

                document.getElementById('nombre\_madre\_edit').value = estudiante.nombre\_madre || '';

                document.getElementById('telefono\_madre\_edit').value = estudiante.telefono\_madre || '';

                document.getElementById('email\_madre\_edit').value = estudiante.email\_madre || '';

                // Muestra el modal

                $('#modalEditarEstudiante').modal('show');

            })

            .catch(error => {

                console.error('Error al cargar los datos del estudiante:', error);

                showMessage('danger', 'Hubo un error al cargar los datos del estudiante.');

            });

    }

      // Event listener para el formulario de editar estudiante

      document.getElementById('formEditarEstudiante').addEventListener('submit', async (event) => {

        event.preventDefault(); // Previene el envío por defecto del formulario

        const formData = new FormData(event.target);

        const data = {};

        formData.forEach((value, key) => {

          data[key] = value;

        });

        try {

          const response = await fetch('/api/estudiantes/editar', {

            method: 'PUT',

            headers: {

              'Content-Type': 'application/json',

            },

            body: JSON.stringify(data),

          });

          if (!response.ok) {

            throw new Error('Error al actualizar el estudiante');

          }

          const result = await response.json();

          console.log('Éxito:', result);

          // Mostrar mensaje de éxito

          showMessage('success', 'Estudiante actualizado correctamente.');

          // Cierra el modal

          $('#modalEditarEstudiante').modal('hide');

          // Actualiza la lista de estudiantes

          await cargarEstudiantes(); // Espera a que la lista se actualice

        } catch (error) {

          console.error('Error:', error);

          // Mostrar mensaje de error

          showMessage('danger', 'Hubo un error al actualizar el estudiante.');

        }

      });

      // Cargar datos iniciales

      cargarEstudiantes();

      cargarGrados();

      // Añadir el evento para cargar secciones al cambiar el grado

      document.getElementById('grado').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        if (gradoId) {

          cargarSecciones(gradoId);

        } else {

          // Si no hay grado seleccionado, limpiar las opciones de sección

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>';

          });

        }

      });

      // Event listener para el formulario de agregar estudiante

      document.getElementById('formAgregarEstudiante').addEventListener('submit', async (event) => {

        event.preventDefault(); // Previene el envío por defecto del formulario

        const formData = new FormData(event.target);

        const data = {};

        // Solo incluimos los datos que queremos enviar al servidor

        const camposRelevantes = ['nombre', 'email', 'fecha\_nacimiento', 'direccion', 'telefono', 'grado', 'seccion\_modal', 'anio\_escolar'];

        formData.forEach((value, key) => {

          if (camposRelevantes.includes(key)) {

            data[key] = value;

          }

        });

        try {

          const response = await fetch('/estudiantes', {

            method: 'POST',

            headers: {

              'Content-Type': 'application/json',

            },

            body: JSON.stringify(data),

          });

          if (!response.ok) {

            throw new Error('Error al guardar el estudiante');

          }

          const result = await response.json();

          console.log('Éxito:', result);

          // Mostrar mensaje de éxito

          showMessage('success', 'Estudiante agregado correctamente.');

          // Cierra el modal

          $('#modalAgregarEstudiante').modal('hide');

          // Actualiza la lista de estudiantes

          await cargarEstudiantes(); // Espera a que la lista se actualice

        } catch (error) {

          console.error('Error:', error);

          // Mostrar mensaje de error

          showMessage('danger', 'Hubo un error al agregar el estudiante.');

        }

      });

      // Manejo de botones para abrir el modal

  document.querySelectorAll('[data-open-modal]').forEach(button => {

    button.addEventListener('click', function() {

      const idEstudiante = this.getAttribute('data-id');

      abrirModalEditar(idEstudiante);

    });

  });

});

  </script>

</body>

</html>

Modulo de alumnos (segunda versión)

<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

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

  <title>Gestión de Estudiantes</title>

  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

  <style>

    /\* Estilos de los mensajes emergentes \*/

#messageBox {

  position: fixed;

  top: 20px;

  right: 20px;

  z-index: 1050;

  width: 300px;

}

    .pagination {

      display: flex;

      list-style: none;

    }

    .pagination li {

      margin: 0 5px;

    }

    .pagination button {

      padding: 8px 12px;

      border: 1px solid #ddd;

      background-color: white;

      cursor: pointer;

      border-radius: 5px;

    }

    .pagination button:hover {

      background-color: #f0f0f0;

    }

    .pagination button.active {

      background-color: #007bff;

      color: white;

      border-color: #007bff;

    }

    .pagination button.disabled {

      cursor: not-allowed;

      opacity: 0.5;

    }

  </style>

</head>

<body>

  <div class="container mt-4">

    <h1>Gestión de Estudiantes</h1>

    <button class="btn btn-primary mb-3" data-toggle="modal" data-target="#modalAgregarEstudiante">Agregar Estudiante</button>

    <!-- Tabla para listar estudiantes -->

    <table class="table table-bordered">

      <thead>

        <tr>

          <th>ID</th>

          <th>Nombre</th>

          <th>Email</th>

          <th>Fecha de Nacimiento</th>

          <th>Dirección</th>

          <th>Teléfono</th>

          <th>Grado</th>

          <th>Sección</th>

          <th>Año Escolar</th>

          <th>Acciones</th>

        </tr>

      </thead>

      <tbody id="tablaEstudiantes">

        <!-- Las filas se llenarán dinámicamente aquí -->

      </tbody>

    </table>

    <!-- Paginación -->

    <nav aria-label="Page navigation">

      <ul class="pagination justify-content-end" id="pagination">

          <!-- Los botones de paginación se cargarán aquí -->

      </ul>

  </nav>

  </div>

 <!-- Modal de Agregar Estudiante -->

<div class="modal fade" id="modalAgregarEstudiante" tabindex="-1" role="dialog" aria-labelledby="modalAgregarEstudianteLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="modalAgregarEstudianteLabel">Agregar Estudiante</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        <form id="formAgregarEstudiante">

          <!-- Datos del Alumno -->

          <div class="form-group">

            <label for="nombre">Nombre</label>

            <input type="text" class="form-control" id="nombre" name="nombre" required>

          </div>

          <div class="form-group">

            <label for="email">Email</label>

            <input type="email" class="form-control" id="email" name="email" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento" name="fecha\_nacimiento" required>

          </div>

          <div class="form-group">

            <label for="direccion">Dirección</label>

            <textarea class="form-control" id="direccion" name="direccion" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono">Teléfono</label>

            <input type="text" class="form-control" id="telefono" name="telefono" required>

          </div>

          <div class="form-group">

            <label for="grado">Grado</label>

            <select class="form-control" id="grado" name="grado" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_modal">Sección</label>

            <select class="form-control" id="seccion\_modal" name="seccion\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar" name="anio\_escolar" required>

          </div>

 <!-- Modal de Editar Estudiante -->

<!-- Modal para editar estudiante -->

<div class="modal fade" id="editStudentModal" tabindex="-1" aria-labelledby="editStudentModalLabel" aria-hidden="true">

  <div class="modal-dialog">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="editStudentModalLabel">Editar Estudiante</h5>

        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Cerrar"></button>

      </div>

      <div class="modal-body">

        <form id="editStudentForm">

          <input type="hidden" id="id\_estudiante\_edit">

          <!-- Otros campos -->

          <div class="mb-3">

            <label for="nombre\_edit" class="form-label">Nombre</label>

            <input type="text" class="form-control" id="nombre\_edit">

          </div>

          <div class="form-group">

            <label for="email\_edit">Email</label>

            <input type="email" class="form-control" id="email\_edit" name="email\_edit" required>

          </div>

          <div class="form-group">

            <label for="fecha\_nacimiento\_edit">Fecha de Nacimiento</label>

            <input type="date" class="form-control" id="fecha\_nacimiento\_edit" name="fecha\_nacimiento\_edit" required>

          </div>

          <div class="form-group">

            <label for="direccion\_edit">Dirección</label>

            <textarea class="form-control" id="direccion\_edit" name="direccion\_edit" rows="3" required></textarea>

          </div>

          <div class="form-group">

            <label for="telefono\_edit">Teléfono</label>

            <input type="text" class="form-control" id="telefono\_edit" name="telefono\_edit" required>

          </div>

          <div class="form-group">

            <label for="grado\_edit">Grado</label>

            <select class="form-control" id="grado\_edit" name="grado\_edit" required>

              <option value="">Seleccione un grado</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="seccion\_edit\_modal">Sección</label>

            <select class="form-control" id="seccion\_edit\_modal" name="seccion\_edit\_modal" required>

              <option value="">Seleccione una sección</option>

              <!-- Opciones se llenarán dinámicamente -->

            </select>

          </div>

          <div class="form-group">

            <label for="anio\_escolar\_edit">Año Escolar</label>

            <input type="number" class="form-control" id="anio\_escolar\_edit" name="anio\_escolar\_edit" required>

          </div>

        </form>

      </div>

      <div class="modal-footer">

        <button type="button" class="btn btn-primary" id="updateStudentButton">Actualizar</button>

        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>

      </div>

    </div>

  </div>

</div>

<!-- Añade esto para mostrar mensajes -->

<div id="messageBox" class="alert" style="display:none;"></div>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <script>

    function showMessage(type, message) {

      const messageBox = document.getElementById('messageBox');

      messageBox.className = `alert alert-${type}`;

      messageBox.innerText = message;

      messageBox.style.display = 'block'; // Muestra el mensaje

      setTimeout(() => messageBox.style.display = 'none', 3000); // Oculta el mensaje después de 3 segundos

  }

    document.addEventListener('DOMContentLoaded', () => {

      const estudiantesPorPagina = 10;

  let estudiantes = [];

  let paginaActual = 1;

  async function cargarEstudiantes() {

    try {

      const response = await fetch('/api/estudiantes');

      if (!response.ok) throw new Error('Error en la respuesta de la API');

      estudiantes = await response.json();

      console.log('Estudiantes cargados:', estudiantes);

      mostrarPagina(paginaActual);

      crearBotonesDePaginacion();

    } catch (error) {

      console.error('Error al cargar estudiantes:', error);

    }

  }

  function mostrarPagina(pagina) {

    const inicio = (pagina - 1) \* estudiantesPorPagina;

    const fin = inicio + estudiantesPorPagina;

    const estudiantesPagina = estudiantes.slice(inicio, fin);

    const tbody = document.getElementById('tablaEstudiantes');

    tbody.innerHTML = '';

    estudiantesPagina.forEach(estudiante => {

      const tr = document.createElement('tr');

      tr.innerHTML = `

        <td>${estudiante.id\_estudiante}</td>

        <td>${estudiante.nombre}</td>

        <td>${estudiante.email}</td>

        <td>${formatearFecha(estudiante.fecha\_nacimiento)}</td>

        <td>${estudiante.direccion}</td>

        <td>${estudiante.telefono}</td>

        <td>${estudiante.nombre\_grado}</td>

        <td>${estudiante.nombre\_seccion}</td>

        <td>${estudiante.anio\_escolar}</td>

        <td>

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#editStudentModal" onclick="prepararEdicionEstudiante(1)">Editar Estudiante</button>          <button type="button" data-view-ficha data-id="${estudiante.id\_estudiante}" class="btn btn-info btn-sm">Ver Ficha Completa</button>

          <button type="submit" class="btn btn-danger btn-sm">Eliminar</button>

        </td>

      `;

      tbody.appendChild(tr);

    });

  }

  function crearBotonesDePaginacion() {

    const totalPaginas = Math.ceil(estudiantes.length / estudiantesPorPagina);

    const paginacionDiv = document.getElementById('pagination');

    paginacionDiv.innerHTML = '';

    // Botón Anterior

    const anteriorBtn = document.createElement('button');

    anteriorBtn.textContent = 'Anterior';

    anteriorBtn.classList.add('page-link');

    if (paginaActual === 1) {

      anteriorBtn.classList.add('disabled');

      anteriorBtn.setAttribute('disabled', true);

    }

    anteriorBtn.addEventListener('click', () => {

      if (paginaActual > 1) {

        paginaActual--;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(anteriorBtn);

    // Botones numéricos

    for (let i = 1; i <= totalPaginas; i++) {

      const boton = document.createElement('button');

      boton.textContent = i;

      boton.classList.add('page-link');

      if (i === paginaActual) {

        boton.classList.add('active');

      }

      boton.addEventListener('click', () => {

        paginaActual = i;

        mostrarPagina(i);

        crearBotonesDePaginacion();

      });

      paginacionDiv.appendChild(boton);

    }

    // Botón Siguiente

    const siguienteBtn = document.createElement('button');

    siguienteBtn.textContent = 'Siguiente';

    siguienteBtn.classList.add('page-link');

    if (paginaActual === totalPaginas) {

      siguienteBtn.classList.add('disabled');

      siguienteBtn.setAttribute('disabled', true);

    }

    siguienteBtn.addEventListener('click', () => {

      if (paginaActual < totalPaginas) {

        paginaActual++;

        mostrarPagina(paginaActual);

        crearBotonesDePaginacion();

      }

    });

    paginacionDiv.appendChild(siguienteBtn);

  }

      async function cargarGrados() {

        try {

          const response = await fetch('/api/grados');

          if (!response.ok) throw new Error('Error al cargar grados');

          const grados = await response.json();

          const gradoSelects = [document.getElementById('grado'), document.getElementById('grado\_edit')];

          gradoSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione un grado</option>'; // Limpiar opciones

            grados.forEach(grado => {

              const option = document.createElement('option');

              option.value = grado.id\_grado;

              option.textContent = grado.nombre\_grado; // Asegúrate de que 'nombre\_grado' sea el nombre correcto en la BD

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar grados:', error);

        }

      }

      async function cargarSecciones(gradoId) {

        try {

          const response = await fetch(`/api/secciones?grado=${gradoId}`);

          if (!response.ok) throw new Error('Error al cargar secciones');

          const secciones = await response.json();

          console.log('Secciones cargadas:', secciones); // Agrega esto para depurar

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>'; // Limpiar opciones

            secciones.forEach(seccion => {

              const option = document.createElement('option');

              option.value = seccion.id\_seccion;

              option.textContent = seccion.nombre\_seccion;

              select.appendChild(option);

            });

          });

        } catch (error) {

          console.error('Error al cargar secciones:', error);

        }

      }

      document.getElementById('grado\_edit').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        if (gradoId) {

          cargarSecciones(gradoId, true); // Usamos `true` para indicar que es en el formulario de edición

        } else {

          // Si no hay grado seleccionado, limpiar las opciones de sección

          document.getElementById('seccion\_edit\_modal').innerHTML = '<option value="">Seleccione una sección</option>';

        }

      });

      function formatearFecha(fechaISO) {

        const fecha = new Date(fechaISO);

        const dia = fecha.getDate().toString().padStart(2, '0');

        const mesesAbreviados = ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"];

        const mes = mesesAbreviados[fecha.getMonth()];

        const anio = fecha.getFullYear();

        return `${dia} ${mes} ${anio}`;

      }

      // Función para abrir el modal y cargar los datos del estudiante

      window.editStudent = function(button) {

        const id = button.getAttribute('data-id');

        fetch(`/api/estudiantes/${id}`)

            .then(response => response.json())

            .then(student => {

                document.getElementById('id\_estudiante\_edit').value = student.id\_estudiante;

                document.getElementById('nombre\_edit').value = student.nombre;

                document.getElementById('email\_edit').value = student.email;

                document.getElementById('fecha\_nacimiento\_edit').value = formatISOToDate(student.fecha\_nacimiento);

                document.getElementById('direccion\_edit').value = student.direccion;

                document.getElementById('telefono\_edit').value = student.telefono;

                document.getElementById('grado\_edit').value = student.grado;

                document.getElementById('seccion\_edit\_modal').value = student.seccion;

                document.getElementById('anio\_escolar\_edit').value = student.anio\_escolar;

                // Inicializa y muestra el modal

                const editStudentModal = new bootstrap.Modal(document.getElementById('editStudentModal'));

                editStudentModal.show();

            })

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

    };

    document.getElementById('updateStudentButton').addEventListener('click', () => {

      const id\_estudiante\_edit = document.getElementById('id\_estudiante\_edit').value.trim();

      const nombre = document.getElementById('nombre\_edit').value.trim();

      const email = document.getElementById('email\_edit').value.trim();

      const fecha\_nacimiento = document.getElementById('fecha\_nacimiento\_edit').value.trim();

      const direccion = document.getElementById('direccion\_edit').value.trim();

      const telefono = document.getElementById('telefono\_edit').value.trim();

      const grado = document.getElementById('grado\_edit').value.trim();

      const seccion = document.getElementById('seccion\_edit\_modal').value.trim();

      const anio\_escolar = document.getElementById('anio\_escolar\_edit').value.trim();

      // Verifica si todos los campos están presentes

      if (!id\_estudiante\_edit || !nombre || !email || !fecha\_nacimiento || !direccion || !telefono || !grado || !seccion || !anio\_escolar) {

        alert('Todos los campos son requeridos.');

        return;

      }

      fetch('/api/estudiantes/update', {

        method: 'POST',

        headers: {

          'Content-Type': 'application/json'

        },

        body: JSON.stringify({

          id\_estudiante\_edit,

          nombre,

          email,

          fecha\_nacimiento,

          direccion,

          telefono,

          grado,

          seccion,

          anio\_escolar

        })

      })

      .then(response => response.json())

      .then(data => {

        if (data.success) {

          alert(data.message);

          $('#editStudentModal').modal('hide');

          // Actualiza la lista de estudiantes o redibujar la tabla

        } else {

          alert(data.message);

        }

      })

      .catch(error => {

        console.error('Error:', error);

      });

    });

      // Cargar datos iniciales

      cargarEstudiantes();

      cargarGrados();

      // Añadir el evento para cargar secciones al cambiar el grado

      document.getElementById('grado').addEventListener('change', (event) => {

        const gradoId = event.target.value;

        if (gradoId) {

          cargarSecciones(gradoId);

        } else {

          // Si no hay grado seleccionado, limpiar las opciones de sección

          const seccionSelects = [document.getElementById('seccion\_modal'), document.getElementById('seccion\_edit\_modal')];

          seccionSelects.forEach(select => {

            select.innerHTML = '<option value="">Seleccione una sección</option>';

          });

        }

      });

      // Event listener para el formulario de agregar estudiante

      document.getElementById('formAgregarEstudiante').addEventListener('submit', async (event) => {

        event.preventDefault(); // Previene el envío por defecto del formulario

        const formData = new FormData(event.target);

        const data = {};

        // Solo incluimos los datos que queremos enviar al servidor

        const camposRelevantes = ['nombre', 'email', 'fecha\_nacimiento', 'direccion', 'telefono', 'grado', 'seccion\_modal', 'anio\_escolar'];

        formData.forEach((value, key) => {

          if (camposRelevantes.includes(key)) {

            data[key] = value;

          }

        });

        try {

          const response = await fetch('/estudiantes', {

            method: 'POST',

            headers: {

              'Content-Type': 'application/json',

            },

            body: JSON.stringify(data),

          });

          if (!response.ok) {

            throw new Error('Error al guardar el estudiante');

          }

          const result = await response.json();

          console.log('Éxito:', result);

          // Mostrar mensaje de éxito

          showMessage('success', 'Estudiante agregado correctamente.');

          // Cierra el modal

          $('#modalAgregarEstudiante').modal('hide');

          // Actualiza la lista de estudiantes

          await cargarEstudiantes(); // Espera a que la lista se actualice

        } catch (error) {

          console.error('Error:', error);

          // Mostrar mensaje de error

          showMessage('danger', 'Hubo un error al agregar el estudiante.');

        }

      });

      // Manejo de botones para abrir el modal

  document.querySelectorAll('[data-open-modal]').forEach(button => {

    button.addEventListener('click', function() {

      const idEstudiante = this.getAttribute('data-id');

      abrirModalEditar(idEstudiante);

    });

  });

});

  </script>

</body>

</html>