**1. Login**

**Descripción:**

El endpoint de **login** permite autenticar al usuario (profesor). Devuelve un **token JWT** que deberá ser utilizado en las solicitudes a los endpoints protegidos.

**Endpoint:**

**POST** /login

**Cuerpo de la solicitud (JSON):**

{

"correo": "profesor@example.com",

"password": "password123"

}

**Respuesta exitosa (200 OK)**:

{

"nombre": "Juan",

"apellido": "Pérez",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

**2. Profesores**

**Descripción:**

Los endpoints relacionados con **profesores** permiten crear, obtener, actualizar y eliminar profesores. **Todos estos endpoints requieren autenticación con el token JWT.**

**2.1. Obtener todos los profesores**

**Endpoint:**

**GET** /api/profesores

**Headers:**

Authorization: Bearer <token>

**Respuesta exitosa (200 OK)**:

[

{

"id": 1,

"nombre": "Juan",

"apellido": "Pérez",

"dni": "12345678",

"telefono": "123456789",

"correo": "juan.perez@example.com"

},

{

"id": 2,

"nombre": "María",

"apellido": "López",

"dni": "87654321",

"telefono": "987654321",

"correo": "maria.lopez@example.com"

}

]

**2.2. Crear un nuevo profesor**

**Endpoint:**

**POST** /api/profesores

**Headers:**

Authorization: Bearer <token>

{

"nombre": "Pedro",

"apellido": "Gómez",

"dni": "98765432",

"telefono": "1122334455",

"correo": "pedro.gomez@example.com",

"password": "Password123!"

}

**Respuesta exitosa (201 Created)**:

{

"message": "Registro creado correctamente"

}

**2.3. Obtener un profesor por ID**

**Endpoint:**

**GET** /api/profesores/:id

**Headers:**

Authorization: Bearer <token>

**Respuesta exitosa (200 OK)**:

{

"id": 1,

"nombre": "Juan",

"apellido": "Pérez",

"dni": "12345678",

"telefono": "123456789",

"correo": "juan.perez@example.com"

}

**2.4. Actualizar un profesor por ID**

**Endpoint:**

**PUT** /api/profesores/:id

**Headers:**

Authorization: Bearer <token>

**Cuerpo de la solicitud (JSON)**:

{

"nombre": "Juan",

"apellido": "Pérez",

"dni": "12345678",

"telefono": "123456789",

"correo": "juan.perez@example.com",

"password": "Password123!"

}

**2.5. Eliminar un profesor por ID**

**Endpoint:**

**DELETE** /api/profesores/:id

**Headers:**

Authorization: Bearer <token>

**Respuesta exitosa (200 OK)**:

{

"message": "Profesor y todas sus relaciones eliminadas con éxito"

}

**3. Asistencias**

**Descripción:**

Los endpoints de **asistencias** permiten registrar la asistencia de un profesor y contar sus asistencias e inasistencias.

**3.1. Registrar asistencia**

**Endpoint:**

**POST** /api/login/asistencia

**Headers:**

**Authorization: Bearer <token>**

**Cuerpo de la solicitud (JSON):**

**{**

**"qrToken": "<token\_QR>"**

**}**

**Respuesta exitosa (200 OK):**

**{**

**"message": "Asistencia registrada con éxito"**

**}**

**3.2. Contar asistencias e inasistencias**

**Endpoint:**

**GET /api/login/contarasistencias**

**Headers:**

**Authorization: Bearer <token>**

**Respuesta exitosa (200 OK):**

**{**

**"message": "Totales de asistencias e inasistencias del profesor con ID 1",**

**"asistencias": 10,**

**"inasistencias": 2**

**}**

**4. Cursos**

**Descripción:**

**Los endpoints de cursos permiten listar y crear cursos.**

**4.1. Listar cursos**

**Endpoint:**

**GET /api/listado/cursos**

**Headers:**

**Authorization: Bearer <token>**

**Respuesta exitosa (200 OK):**

**[**

**{**

**"CursoID": 1,**

**"Anio": 2023,**

**"Division": "1ra"**

**},**

**{**

**"CursoID": 2,**

**"Anio": 2023,**

**"Division": "2da"**

**}**

**]**

**4.2. Crear un nuevo curso**

**Endpoint:**

**POST /api/crear/curso**

**Headers:**

**Authorization: Bearer <token>**

**Cuerpo de la solicitud (JSON):**

**{**

**"anio": 2023,**

**"division": "1ra"**

**}**

**Respuesta exitosa (201 Created):**

**{**

**"message": "Curso creado exitosamente"**

**}**

**5. Horarios**

**Descripción:**

**Los endpoints de horarios permiten gestionar los horarios de los cursos y profesores.**

**5.1. Insertar un nuevo horario**

**Endpoint:**

**POST /api/insertar/horario**

**Headers:**

**Authorization: Bearer <token>**

**Cuerpo de la solicitud (JSON):**

**{**

**"profesorId": 1,**

**"cursoId": 1,**

**"materiaId": 1,**

**"dia": "Lunes",**

**"fechaInicio": "08:00",**

**"fechaFin": "09:30"**

**}**

**Respuesta exitosa (201 Created):**

**{**

**"message": "Horario insertado exitosamente"**

**}**

**5.2. Mostrar horarios de un profesor**

**Endpoint:**

**GET /api/listado/horario/profesor**

**Headers:**

**Authorization: Bearer <token>**

**Cuerpo de la solicitud (JSON):**

**{**

**"profesorId": 1**

**}**

**Respuesta exitosa (200 OK)**:

[

{

"HorarioID": 1,

"Dia": "Lunes",

"fechaInicio": "08:00",

"fechaFin": "09:30",

"Curso": {

"CursoID": 1,

"Anio": 2023,

"Division": "1ra"

},

"Materia": {

"MateriaID": 1,

"NombreMateria": "Matemáticas"

}

}

]

**6. Materias**

**Descripción:**

Los endpoints de **materias** permiten insertar y listar las materias.

**6.1. Insertar una nueva materia**

**Endpoint:**

**POST** /api/insertar/materias

**Headers:**

**Authorization: Bearer <token>**

**Cuerpo de la solicitud (JSON):**

**{**

**"NombreMateria": "Matemáticas"**

**}**

**Respuesta exitosa (201 Created):**

**{**

**"message": "Materia insertada exitosamente"**

**}**

**6.2. Listar todas las materias**

**Endpoint:**

**GET /api/listado/materias**

**Headers:**

**Authorization: Bearer <token>**

**Respuesta exitosa (200 OK):**

**[**

**{**

**"MateriaID": 1,**

**"NombreMateria": "Matemáticas"**

**},**

**{**

**"MateriaID": 2,**

**"NombreMateria": "Historia"**

**}**

**]**

**Notas importantes:**

* **Autenticación: La mayoría de los endpoints requieren un token JWT que se debe enviar en el encabezado Authorization como Bearer <token>.**
* **Códigos de estado: Los códigos de estado comunes son:**
  + **200 OK: Solicitud exitosa.**
  + **201 Created: Recurso creado exitosamente.**
  + **400 Bad Request: Error en los datos proporcionados.**
  + **401 Unauthorized: Token inválido o no proporcionado.**
  + **403 Forbidden: Acceso denegado por un token inválido o expirado.**
  + **404 Not Found: Recurso no encontrado.**

**Ejemplo de un fetch:  
const token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...";**

**const data = {**

**nombre: "Juan",**

**apellido: "Pérez",**

**correo: "juan.perez@example.com",**

**password: "Password123!"**

**};**

**fetch('https://tu-api.com/api/profesores', {**

**method: 'POST',**

**headers: {**

**'Authorization': `Bearer ${token}`, // Encabezado de autorización con el token**

**'Content-Type': 'application/json' // Tipo de contenido para la solicitud**

**},**

**body: JSON.stringify(data) // Convertir los datos a JSON**

**})**

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

**.then(data => {**

**console.log('Profesor creado:', data); // Manejar la respuesta exitosa**

**})**

**.catch(error => {**

**console.error('Error:', error); // Manejar errores**

**});**