

## **Manual de Usuario**

### **Sistema de Gestión Escolar**

**Fecha:** Julio 01 2025

**Autor:** Angelica Maria Marcillo Alba

## **Introducción**

Estimado usuario, esta aplicación ha sido diseñada para facilitar la administración integral de instituciones educativas, permitiendo gestionar de manera eficiente estudiantes, profesores, administrativos, cursos e inscripciones.

## **Funcionalidades Principales**

- Registro y administración de personas
- Control de estudiantes y matrículas
- Gestión del cuerpo docente
- Administración de personal administrativo
- Creación y manejo de cursos
- Control de inscripciones
- Sistema de autenticación seguro

## **Requisitos del Sistema**

Para utilizar correctamente el sistema, asegúrese de contar con:

- **Java:** Versión 17 o superior
- **MySQL:** Versión 8.0 o superior
- **Navegador Web:** Chrome, Firefox o Edge actualizados
- **Conexión a Internet:** Para acceso a la documentación

## **Instalación**

### **Preparación de la Base de Datos**

1. Instale MySQL en su sistema
2. Cree una nueva base de datos con el nombre gestion\_escolar
3. Ejecute el archivo gestion\_escolar.sql incluido en el proyecto

### **Configuración de la Aplicación**

1. Descargue el proyecto desde el repositorio
2. Configure las credenciales de base de datos en el archivo application.properties
3. Ejecute el comando: mvn spring-boot:run
4. La aplicación estará disponible en: <http://localhost:8080>

## **Primer Acceso al Sistema**

### **Credenciales de Administrador**

Para su primer acceso, puede utilizar cualquiera de las siguientes credenciales:

**Usuario:** patricia.salazar

**Contraseña:** patricia123

**Usuario:** miguel.torres

**Contraseña:** miguel123

**Usuario:** gloria.ramos

**Contraseña:** gloria123

## Proceso de Autenticación

1. Realice una petición POST a /api/auth/login
2. Incluya sus credenciales en formato JSON
3. El sistema le proporcionará un token de acceso
4. Incluya este token en todas sus peticiones posteriores

## Ejemplo de petición de login:

```
{  
  "username": "patricia.salazar",  
  "password": "patricia123"  
}
```

## Módulos del Sistema

### Gestión de Personas

Este módulo permite administrar la información personal básica de todos los usuarios del sistema.

**Endpoint principal:** /api/personas

### Campos requeridos:

- Nombre y apellido
- Fecha de nacimiento
- Email (debe ser único)
- Teléfono (10 dígitos)

## **Gestión de Estudiantes**

Administre la información académica de los estudiantes de su institución.

**Endpoint principal:** /api/estudiantes

### **Información específica:**

- Número de matrícula (formato: EST + 7 dígitos)
- Grado académico
- Datos personales completos

## **Gestión de Profesores**

Controle la información del cuerpo docente y sus especialidades.

**Endpoint principal:** /api/profesores

### **Datos específicos:**

- Especialidad académica
- Fecha de contratación
- Información personal

## **Gestión de Administrativos**

Administre el personal administrativo y sus roles dentro de la institución.

**Endpoint principal:** /api/administrativos

### **Información requerida:**

- Cargo desempeñado
- Departamento de trabajo
- Datos personales

## **Gestión de Cursos**

Cree y administre las materias disponibles en su institución.

**Endpoint principal:** /api/cursos

### **Características:**

- Nombre del curso
- Descripción detallada
- Número de créditos (1-10)
- Profesor asignado (opcional)

## **Gestión de Inscripciones**

Controle las matriculaciones de estudiantes en los diferentes cursos.

**Endpoint principal:** /api/inscripciones

### **Validaciones:**

- El estudiante debe existir en el sistema
- El curso debe estar disponible
- No se permiten inscripciones duplicadas

## **Operaciones Básicas**

### **Crear un Nuevo Registro**

Para crear cualquier registro nuevo:

1. Realice una petición POST al endpoint correspondiente
2. Incluya todos los campos requeridos
3. Verifique que el formato de los datos sea correcto
4. El sistema le confirmará la creación exitosa

### **Consultar Información**

Para consultar registros existentes:

1. Utilice peticiones GET al endpoint deseado

2. Para consultas específicas, incluya el ID del registro
3. Para listados completos, use el endpoint sin parámetros

### **Actualizar Registros**

Para modificar información existente:

1. Realice una petición PUT al endpoint con el ID específico
2. Incluya todos los campos que desea modificar
3. Mantenga los campos que no requieren cambios

### **Eliminar Registros**

Para eliminar información:

1. Utilice peticiones DELETE con el ID específico
2. Confirme que el registro no tenga dependencias
3. La eliminación será permanente

### **Interfaz Swagger**

Para facilitar el uso del sistema, se ha incluido una interfaz gráfica Swagger.

**Acceso:** <http://localhost:8080/swagger-ui/index.html>

### **Autenticación en Swagger**

1. Realice login usando el endpoint correspondiente
2. Copie el token recibido
3. Haga clic en "Authorize" en la parte superior
4. Ingrese "Bearer " seguido de su token
5. Confirme la autorización

## Paginación de Resultados

Para mejorar el rendimiento con grandes volúmenes de datos, varios endpoints ofrecen paginación:

- /api/personas/paged
- /api/estudiantes/paged
- /api/profesores/paged
- /api/administrativos/paged
- /api/cursos/paged
- /api/inscripciones/paged

### Parámetros disponibles:

- **page:** Número de página (inicia en 0)
- **size:** Elementos por página (predeterminado: 10)
- **sortBy:** Campo para ordenar
- **sortDir:** Dirección de ordenamiento (asc/desc)

## Solución de Problemas Comunes

### Error de Autenticación (401)

Si recibe este error, verifique:

- Que el token esté incluido correctamente
- Que el token no haya expirado (24 horas)
- Que el formato sea: "Bearer [token]"

**Solución:** Realice un nuevo login para obtener un token actualizado.

### Error de Recurso No Encontrado (404)

Este error indica que el recurso solicitado no existe.

**Solución:** Verifique que el ID sea correcto y que el recurso no haya sido eliminado.

### Error de Datos Duplicados (409)

Ocurre cuando intenta crear registros con información que ya existe.

**Casos comunes:**

- Email duplicado
- Número de matrícula repetido
- Inscripción duplicada

**Solución:** Modifique los datos conflictivos antes de intentar nuevamente.

**Error de Validación (400)**

Indica que los datos enviados no cumplen con los requisitos.

**Solución:** Revise que todos los campos requeridos estén completos y con el formato correcto.

**Casos de Uso Prácticos**

**Inscribir un Nuevo Estudiante**

1. Cree el registro del estudiante en `/api/estudiantes`
2. Verifique los cursos disponibles en `/api/cursos`
3. Cree las inscripciones necesarias en `/api/inscripciones`

**Crear un Nuevo Curso**

1. Verifique los profesores disponibles en `/api/profesores`
2. Cree el curso en `/api/cursos`
3. Asigne el profesor correspondiente

**Actualizar Información Personal**

1. Consulte el registro actual
2. Modifique únicamente los campos necesarios
3. Envíe la actualización completa



### **Datos de Prueba Incluidos**

Para facilitar sus primeras pruebas, el sistema incluye:

- 5 estudiantes con matrículas de ejemplo
- 4 profesores con diferentes especialidades
- 3 administrativos en diversos departamentos
- 7 cursos con variados créditos
- 15 inscripciones de muestra

**Desarrolladora:** Angelica Maria Marcillo Alba

**Repositorio del Proyecto:** <https://github.com/ange2310/prueba-backend>