

MANUAL TECNICO SISTEMA REGISTRO UNIVERSITARIO

VERSION: PRACTICA NRO. 2

a) Roles del Sistema

Rol	Permisos
Admin	<ul style="list-style-type: none">"/api/auth/" – Autenticación"/api/admin/" – Administrador"/api/docentes/" – Gestión de Docentes"/api/estudiantes/" – Gestión de Estudiantes"/api/inscripciones/" – Gestión de Inscripciones"/api/materias/" – Gestión de Materias (Asignación de docentes)
Docente	<ul style="list-style-type: none">"/api/auth/" – Autenticación"/api/docentes/" – Gestión de Docentes"/api/estudiantes/" – Gestión de Estudiantes"/api/materias/" – Gestión de Materias (Asignación de docentes)
Estudiante	<ul style="list-style-type: none">"/api/auth/" – Autenticación"/api/estudiantes/" – Gestión de Estudiantes

b) Endpoints Disponibles

Inscripciones		
Método	Endpoint	Descripción
GET	/api/inscripciones	Lista todas las inscripciones.
GET	/api/inscripciones/{id}	Consulta una inscripción por ID.
GET	/api/inscripciones/estudiante/{idEstudiante}	Consulta inscripciones por estudiante.
GET	/api/inscripciones/materia/{idMateria}	Consulta inscripciones por materia.
POST	/api/inscripciones	Crea una nueva inscripción.
PUT	/api/inscripciones/{id}	Actualiza una inscripción.
PUT	/api/inscripciones/{id}/abandonar	Abandona una inscripción.
Docentes		
Método	Endpoint	Descripción
GET	/api/docentes	Lista todos los docentes.
POST	/api/docentes	Crea un nuevo docente.

GET	/api/docentes/empleado/{nroEmpleado}	Consulta docente por número de empleado.
GET	/api/docentes/{nroEmpleado}/materias	Consulta materias asignadas por docente.
PUT	/api/docentes/{id}	Actualiza un docente.
DELETE	/api/docentes/{id}	Elimina a un docente.
Estudiantes		
Método	Endpoint	Descripción
GET	/api/estudiantes	Lista todos los estudiantes.
POST	/api/estudiantes	Crea un nuevo estudiante.
GET	/api/estudiantes/{id}/materias	Consulta las materias por estudiante.
GET	/api/estudiantes/inscripcion/{numeroInscripcion}	Consulta al estudiante por número de inscripción.
GET	/api/estudiantes/activos	Lista todos los estudiantes activos.
PUT	/api/estudiantes/{id}	Actualiza un estudiante.
PUT	/api/estudiantes/{id}/baja	Da de baja a un estudiante.
Materias		
Método	Endpoint	Descripción
GET	/api/materias	Lista todas las materias.
POST	/api/materias	Crea una nueva materia.
GET	/api/materias/{id}	Consulta una materia por ID.
GET	/api/materias/codigo/{codigoUnico}	Consulta una materia por código único.
PUT	/api/materias/{id}	Actualiza una materia.
PUT	/api/materias/{materialId}/asignar-docentes	Asigna una materia uno o varios docentes.
DELETE	/api/materias/{id}	Elimina la materia.
Autenticación		
Método	Endpoint	Descripción
POST	/api/auth/signup	Crea un nuevo usuario.
POST	/api/auth/login	Iniciar sesión.
POST	/api/auth/logout	Cerrar sesión.
GET	/api/auth/session-info	Consulta la información de sesión del usuario que se hizo login.

c) Configuración de la Aplicación

- Backend: Spring Boot 3.x
- Base de Datos: PostgreSQL
- ORM: Hibernate (JPA)
- API Docs: Swagger OpenAPI (<http://localhost:8080/swagger-ui.html>)
- Gestor de Dependencias: Maven
- Puerto por defecto: 8080

application.properties:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/universidad
spring.datasource.username=postgres
spring.datasource.password=123456
spring.jpa.hibernate.ddl-auto=update
springdoc.api-docs.path=/api-docs
springdoc.swagger-ui.path=/swagger-ui.html
```