

Sistema de Gestión Académica Universitaria

1. Descripción General

API RESTful para gestionar información académica básica de una institución universitaria, basada en las siguientes entidades:

- Student
- Academic_Record
- Professor
- Department
- Catalog_Course (Subject)
- Enrollment

El sistema permite administrar estudiantes, profesores, departamentos, récords académicos, asignaturas y matrículas.

Alcance:

Sistema académico básico para fines académicos/prácticos.

No es un producto comercial.

2. Objetivo del Sistema

1. Centralizar información estudiantil y académica.
2. Administrar historial académico por estudiante.
3. Controlar matrícula por asignatura.
4. Gestionar profesores y departamentos.
5. Facilitar consultas administrativas.

3. Modelo de Datos Soportado

1. **Student:** (StudentId (PK), FirstName, LastName, BirthDay, Gender, Email, PhoneNumber, Address, Nationality)
2. **Academic_Record:** (RecordId (PK), StudentId (FK), Matricula, CareerId, FacultyId, YearEnrollment, CurrentPeriod, Average, State)
3. **Professor:** (ProfessorId (PK), FirstName, LastName, Email, Phone, DepartmentId (FK), Specialty, Status)
4. **Department:** (DepartmentId (PK), Name, Faculty)
5. **Catalog_Course (Subject):** (SubjectId (PK), RecordId (FK), Name, Code, Score, ProfessorId (FK))
6. **Enrollment:** (EnrollmentId (PK), RecordId (FK), SubjectId (FK), EnrollDate, Status)

4. Módulos del Sistema

1. Gestión de Estudiantes

Nota: Para la primera tarea mínimo se debe completar este módulo, los demás se completarán a medida que practique lo aprendido en clase.

Funciones:

- Registrar estudiantes.
- Actualizar datos personales.
- Consultar información.
- Eliminar estudiantes.
- Consultar récord académico.

Relación:

Un Student tiene un Academic_Record.

2. Gestión de Récord Académico (Es parte de estudiante)

Funciones:

- Crear récord académico. (Se crea automáticamente cuando se registra un estudiante)
- Asociar récord a estudiante. (Sigue el proceso de creación)
- Actualizar período actual. (Se actualiza de forma automática)
- Calcular promedio general.
- Gestionar estado académico.

Relación:

Un Academic_Record pertenece a un Student.

Un Academic_Record tiene múltiples asignaturas.

3. Gestión de Profesores

Funciones:

- Registrar profesores.
- Asignar a departamentos.
- Actualizar especialidad.
- Activar/Inactivar profesores.
- Consultar asignaturas impartidas.

Relación:

Un Professor pertenece a un Department.

Un Professor puede impartir varios cursos.

4. Gestión de Departamentos

Funciones:

- Crear departamentos.
- Actualizar datos.
- Consultar profesores por departamento.
- Eliminar departamentos.

Relación:

Un Department tiene múltiples Professors.

5. Gestión de Asignaturas (Catalog_Course)

Funciones:

- Crear asignaturas.
- Asignar profesor.
- Asociar al récord académico.
- Registrar calificación (Score).
- Consultar historial por asignatura.

Relación:

Una asignatura pertenece a un Academic_Record.

Una asignatura es impartida por un Professor.

6. Gestión de Matrícula (Enrollment)

Funciones:

- Matricular asignaturas.

- Registrar fecha de matrícula.
- Actualizar estado (Activo, Retirado, Aprobado, Reprobado).
- Consultar matrícula por estudiante.
- Cancelar matrícula.

Relación:

Enrollment vincula Academic_Record con Subject.

7. Reglas de Negocio

1. Un estudiante debe existir antes de crear un récord académico.
2. No puede existir matrícula sin récord académico.
3. Un curso debe tener un profesor válido.
4. Un profesor debe pertenecer a un departamento.
5. Una matrícula solo puede referenciar cursos existentes.
6. El promedio del récord se calcula a partir del Score.
7. No se permite duplicar matrícula para el mismo curso y récord

8. Endpoints Principales

Students:

- GET /api/v1/students
- POST /api/v1/students
- GET /api/v1/students/{id}
- PUT /api/v1/students/ Especificar la id en el body
- DELETE /api/v1/students/{matricula}
- GET /api/v1/students/records/{matricula}

8.2 Academic Records:

- GET /api/v1/records
- POST /api/v1/records
- PUT /api/v1/records/{id}

8.3 Professors

- GET /api/v1/professors
- POST /api/v1/professors
- PUT /api/v1/professors/{id}
- DELETE /api/v1/professors/{id}

8.4 Departments

- GET /api/v1/departments
- POST /api/v1/departments
- PUT /api/v1/departments/{id}
- DELETE /api/v1/departments/{id}

8.5 Subjects

- GET /api/v1/subjects
- POST /api/v1/subjects
- GET /api/v1/subjects/{id}
- PUT /api/v1/subjects/{id}

8.6 Enrollments

- POST /api/v1/enrollments
- GET /api/v1/enrollments/record/{recordId}
- DELETE /api/v1/enrollments/{id}

10. Entregables

- Código fuente
- Migraciones de la DB
- Diagrama ER
- Colección Postman