

ITAM

DATA SCIENCE AND MACHINE LEARNING APPLIED TO FINANCIAL MARKETS¹

Modulo III

Tarea: Creación Modelos SQLAlchemy

Objetivo:

Que el estudiante modele un **esquema de base de datos realista** usando **SQLAlchemy**, incluyendo:

- Relaciones 1–N y N–M.
- Restricciones de dominio y unicidad.
- Buenas prácticas de integridad referencial.
- Atributos adicionales para enriquecer el modelo

Instrucciones

1. Define un archivo Python con las clases que representen las tablas:

- Estudiante
- Curso
- Profesor
- inscripción (tabla puente N–M entre Estudiante y Curso).

2. Requisitos de la tabla Estudiante:

- id: entero, clave primaria, autoincremental.
- nombre: obligatorio, máximo 100 caracteres, no vacío.
- email: obligatorio, único y con formato válido (ejemplo@dominio.com).
- matricula: obligatorio, único, 8 caracteres alfanuméricos.
- fecha_inscripcion: fecha obligatoria.

- Restricciones: nombre no vacío, email con formato válido, matrícula exactamente de 8 caracteres.

3. Requisitos de la tabla Profesor:

- id: entero, clave primaria.
- nombre: obligatorio, máximo 100 caracteres, no vacío.
- email: obligatorio, único y con formato válido.
- departamento: obligatorio, máximo 50 caracteres.
- Relación: un profesor puede impartir varios cursos.

4. Requisitos de la tabla Curso:

- id: entero, clave primaria.
- nombre: obligatorio, máximo 80 caracteres, no vacío.
- credits: entero, entre 1 y 10.
- nivel: texto, valores válidos: 'Licenciatura', 'Maestría', 'Doctorado'.
- profesor_id: clave foránea → Profesor.
- Relación inversa: un curso pertenece a un profesor.

5. Requisitos de la tabla Inscripcion (relación N-M):

- id: clave primaria.
- estudiante_id: FK → Estudiante.
- curso_id: FK → Curso.
- fecha_inscripcion: fecha obligatoria.
- calificacion: número entre 0 y 100 (puede ser NULL si no hay aún).
- **Restricciones:**
 1. Un estudiante no puede inscribirse dos veces al mismo curso (constraint de unicidad).
 2. calificacion entre 0 y 100 si no es nula.

6. Al final del archivo, asegúrate de que se pueda crear la base de datos con:

```
Base.metadata.create_all(engine)
```

Conceptos para practicar:

- PK, FK, Unique, CheckConstraints.
- Relaciones 1–N (Profesor–Curso) y N–M (Estudiante–Curso vía Inscripción).
- Validaciones reales: emails, matrícula fija, créditos con rango, calificación con rango.

Formato de entrega

- **Archivo:** un script en Python (.py) con la definición completa.
- **PDF adjunto:** que contenga la explicación breve (1–2 páginas) de tu diseño, describiendo las restricciones y justificación de las mismas.
- **Nomenclatura:**
Tarea_SQLAlchemy_NombreApellidoAA.py
Tarea_SQLAlchemy_NombreApellidoAA.pdf
- **Entrega:** por correo en una carpeta comprimida .zip con ambos archivos, a más tardar el **13/10/2025 12:00 a.m.**