**Descripción del problema**

En el contexto de la administración académica de una institución educativa, es necesario gestionar múltiples procesos y entidades asociadas con la enseñanza, incluyendo la **gestión de estudiantes, docentes, asignaturas, cursos, matrículas, calificaciones** y los **usuarios del sistema**. El sistema debe ser capaz de:

* **Estudiantes**: Gestionar los datos de los estudiantes matriculados en la institución, incluyendo un identificador único, nombre completo, correo electrónico, número de teléfono y dirección domiciliaria.
* **Docentes**: Almacenar los datos de los docentes que imparten clases, incluyendo un identificador único, nombre completo, correo electrónico y especialidad.
* **Asignaturas**: Registrar las asignaturas ofertadas por la institución, incluyendo su identificador único, nombre y una descripción.
* **Cursos**: Representan la asignación de una asignatura a un docente en un período académico. Incluye el identificador del curso, el docente asignado, la asignatura correspondiente y el período en el que se dicta.
* **Matrículas**: Gestionar las inscripciones de los estudiantes a los cursos disponibles. Incluye el identificador de la matrícula, el estudiante, el curso y la fecha de inscripción.
* **Calificaciones**: Registrar las notas obtenidas por los estudiantes en cada curso. Incluye el identificador de la calificación, el curso matriculado, la nota obtenida y observaciones.
* **Usuarios**: Gestionar los usuarios del sistema con distintos roles de acceso (Administrador, Desarrollador, Supervisor), cada uno con diferentes permisos.

**Usuarios y Roles**

El sistema debe gestionar tres tipos de usuarios, cada uno con diferentes niveles de acceso y permisos sobre la base de datos:

* **Administrador**: tiene acceso completo y sin restricciones a todas las tablas y objetos de la base de datos. Puede realizar cualquier operación, incluyendo insertar, modificar, eliminar y consultar registros, así como alterar la estructura de la base de datos (crear, modificar o eliminar tablas, índices, vistas, entre otros).
* **Desarrollador**: puede utilizar el sistema para pruebas y desarrollo de consultas y algoritmos. Sin embargo, no tiene permiso para acceder, modificar o visualizar información sensible, como datos de clientes o pagos. Su acceso está limitado a evitar comprometer la confidencialidad y seguridad de la información crítica
* **Supervisor**: cuenta únicamente con permisos de lectura sobre la base de datos. Puede ejecutar consultas SELECT y crear vistas para auditorías o reportes operativos. No puede modificar datos ni alterar la estructura de la base de datos, garantizando así la integridad y estabilidad del sistema.

**Creación de la Base de Datos**

CREATE DATABASE GestionAcademica;

USE GestionAcademica;

**Script SQL de Tablas y Datos:**

**-- Estudiantes**

CREATE TABLE Estudiantes (

estudiante\_id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

correo VARCHAR(100),

telefono VARCHAR(20),

direccion VARCHAR(255)

);

**-- Docentes**

CREATE TABLE Docentes (

docente\_id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

especialidad VARCHAR(100),

correo VARCHAR(100)

);

**-- Asignaturas**

CREATE TABLE Asignaturas (

asignatura\_id INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_asignatura VARCHAR(100),

descripcion TEXT

);

**-- Cursos**

CREATE TABLE Cursos (

curso\_id INT AUTO\_INCREMENT PRIMARY KEY,

asignatura\_id INT,

docente\_id INT,

periodo VARCHAR(20),

FOREIGN KEY (asignatura\_id) REFERENCES Asignaturas(asignatura\_id),

FOREIGN KEY (docente\_id) REFERENCES Docentes(docente\_id)

);

-- Matrículas

CREATE TABLE Matriculas (

matricula\_id INT AUTO\_INCREMENT PRIMARY KEY,

estudiante\_id INT,

curso\_id INT,

fecha\_matricula DATE,

FOREIGN KEY (estudiante\_id) REFERENCES Estudiantes(estudiante\_id),

FOREIGN KEY (curso\_id) REFERENCES Cursos(curso\_id)

);

-- Calificaciones

CREATE TABLE Calificaciones (

calificacion\_id INT AUTO\_INCREMENT PRIMARY KEY,

estudiante\_id INT,

asignatura\_id INT,

nota DECIMAL(4,2),

observaciones TEXT,

FOREIGN KEY (estudiante\_id) REFERENCES Estudiantes(estudiante\_id),

FOREIGN KEY (asignatura\_id) REFERENCES Asignaturas(asignatura\_id)

);

-- Usuarios

CREATE TABLE Usuarios (

Id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

rol ENUM('Administrador', 'Desarrollador', 'Supervisor') NOT NULL

);

**Insertar datos de ejemplo**

USE GestionAcademica;

**-- Estudiantes**

INSERT INTO Estudiantes (nombre, correo, telefono, direccion) VALUES

('Ana Pérez', 'ana.perez@edu.ec', '0999999991', 'Av. Siempre Viva 123'),

('Luis Torres', 'luis.torres@edu.ec', '0988888882', 'Calle Luna 456');

**-- Docentes**

INSERT INTO Docentes (nombre, especialidad, correo) VALUES

('Dr. Julio Herrera', 'Matemáticas', 'jherrera@edu.ec'),

('Mg. Carmen Delgado', 'Lengua y Literatura', 'cdelgado@edu.ec');

**-- Asignaturas**

INSERT INTO Asignaturas (nombre\_asignatura, descripcion) VALUES

('Álgebra Lineal', 'Estudio de vectores, matrices y sistemas de ecuaciones.'),

('Literatura Ecuatoriana', 'Análisis de obras y autores nacionales.');

**-- Cursos**

INSERT INTO Cursos (asignatura\_id, docente\_id, periodo) VALUES

(1, 1, '2025-A'),

(2, 2, '2025-A');

**-- Matrículas**

INSERT INTO Matriculas (estudiante\_id, curso\_id, fecha\_matricula) VALUES

(1, 1, '2025-03-15'),

(2, 2, '2025-03-15');

**-- Calificaciones**

INSERT INTO Calificaciones (matricula\_id, nota, observaciones) VALUES

(1, 9.5, 'Excelente rendimiento'),

(2, 8.2, 'Buen desempeño');

**-- Usuarios**

INSERT INTO usuario (nombre\_usuario, contrasena\_encriptada, rol, estado) VALUES

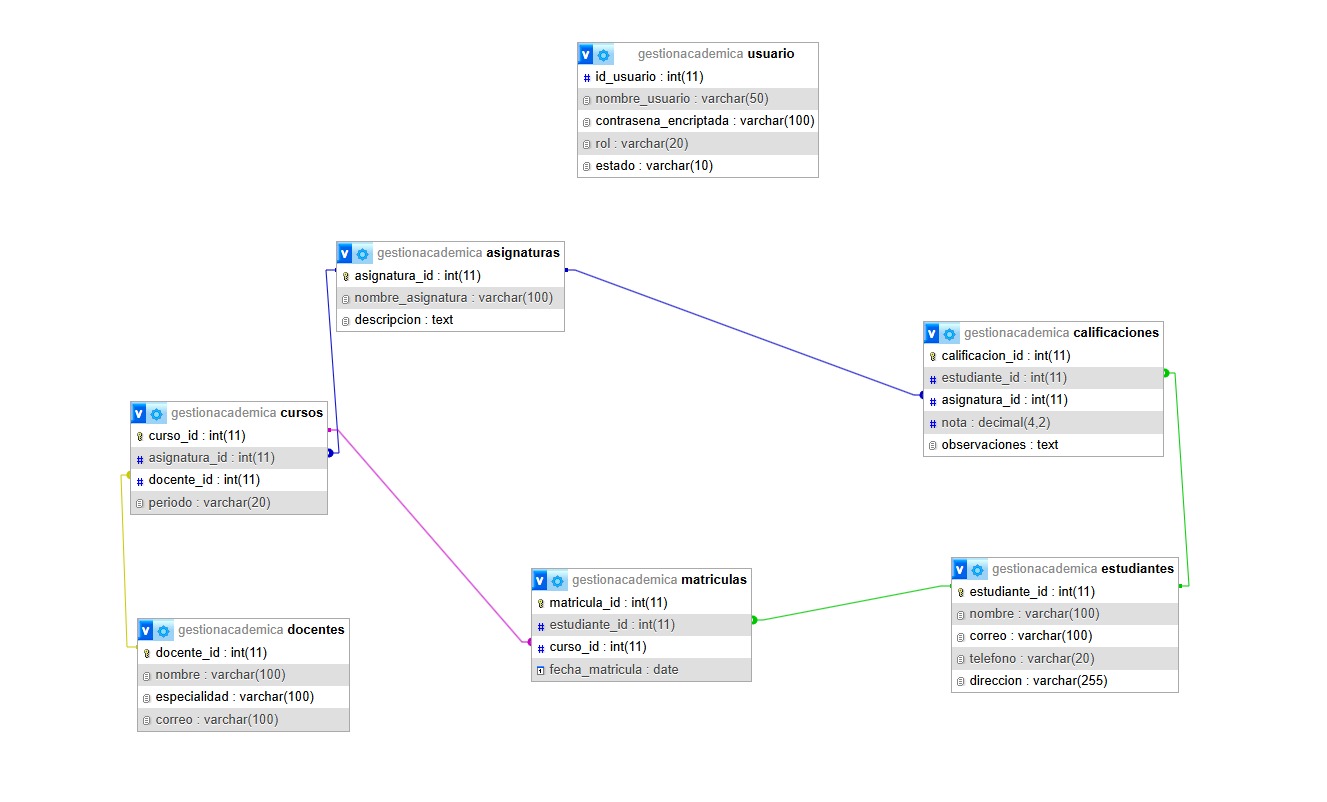
('admin', '$2y$10$abc123EjemploHashAdmin...', 'Administrador', 'activo'),

('devuser', '$2y$10$def456EjemploHashDev...', 'Desarrollador', 'activo'),

('supervisor1', '$2y$10$ghi789EjemploHashSup...', 'Supervisor', 'activo');

**Explicación:**

* **Estudiantes → Matrículas:**  
  Un estudiante puede estar matriculado en varios cursos a lo largo de los períodos académicos. Esta relación se representa con la clave foránea estudiante\_id en la tabla Matriculas, lo que permite llevar un registro completo de los cursos en los que cada estudiante está inscrito, junto con la fecha de matrícula.
* **Docentes → Cursos:**  
  Un docente puede impartir múltiples cursos en distintos períodos académicos.  
  Esta relación se define mediante la clave foránea docente\_id en la tabla Cursos, lo que permite asignar un docente responsable a cada curso específico.
* **Asignaturas → Cursos:**Una asignatura puede estar asociada a varios cursos en diferentes períodos.  
  La tabla Cursos actúa como intermediaria entre Asignaturas y Docentes, permitiendo identificar qué docente imparte una asignatura en un período determinado.
* **Cursos → Matrículas:**  
  Un curso puede tener múltiples estudiantes matriculados. La tabla Matriculas se encarga de enlazar los cursos con los estudiantes mediante las claves foráneas curso\_id y estudiante\_id, manteniendo un historial de inscripciones.
* **Estudiantes → Calificaciones:**  
  Cada estudiante puede tener múltiples calificaciones correspondientes a distintas asignaturas o cursos.
* **Asignaturas → Calificaciones:**  
  Una calificación está vinculada a una asignatura específica. Esto se maneja mediante la clave foránea asignatura\_id en la tabla Calificaciones, permitiendo registrar qué nota corresponde a qué materia.



**Modelo Entidad/Relación (ER)**

1. **Estudiantes**
   * estudiante\_id (PK)
   * nombre
   * correo
   * teléfono
   * dirección
2. **Docentes**
   * docente\_id (PK)
   * nombre
   * especialidad
   * correo
3. **Asignaturas**
   * asignatura\_id (PK)
   * nombre\_asignatura
   * descripción
4. **Cursos**
   * curso\_id (PK)
   * asignatura\_id (FK → Asignaturas)
   * docente\_id (FK → Docentes)
   * periodo
5. **Matriculas**
   * matricula\_id (PK)
   * estudiante\_id (FK → Estudiantes)
   * curso\_id (FK → Cursos)
   * fecha\_matricula
6. **Calificaciones**
   * calificacion\_id (PK)
   * estudiante\_id (FK → Estudiantes)
   * asignatura\_id (FK → Asignaturas)
   * nota
   * observaciones

**Creación de Usuarios y Asignación de Permisos**

**-- Usuarios MySQL**

CREATE USER 'administrador'@'localhost' IDENTIFIED BY 'admin123';

CREATE USER 'desarrollador'@'localhost' IDENTIFIED BY 'dev123';

CREATE USER 'supervisor'@'localhost' IDENTIFIED BY 'super123';

**PERMISOS**

**-- Otorgar todos los privilegios sobre toda la base de datos**

GRANT ALL PRIVILEGES ON gestionacademica.\* TO 'administrador'@'localhost';

**-- Otorgar permisos SELECT, INSERT, UPDATE, DELETE en todas las tablas relevantes**

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.estudiantes TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.docentes TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.asignaturas TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.cursos TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.matriculas TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.calificaciones TO 'desarrollador'@'localhost';

GRANT SELECT, INSERT, UPDATE, DELETE ON gestionacademica.usuarios TO 'desarrollador'@'localhost';

**-- Permisos de solo lectura para todas las tablas del sistema**

GRANT SELECT ON gestionacademica.estudiantes TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.docentes TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.asignaturas TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.cursos TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.matriculas TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.calificaciones TO 'supervisor'@'localhost';

GRANT SELECT ON gestionacademica.usuarios TO 'supervisor'@'localhost';