



## **Base de datos con SQL SERVER**

### **Proyecto Final:**

### **Gestión de Matrícula Universitaria**

#### **Autor:**

Prado Acuña, Robert Anthony

#### **Docente:**

Ing. Rivera Vergaray, Kevin

Lima, Perú

2024

## **I. Iniciación: Definición de los objetivos y el alcance inicial de la base de datos.**

### **Título del Problema: Gestión de Matrícula Universitaria**

#### **Descripción del Problema:**

La Universidad necesita un sistema para gestionar la matrícula de sus estudiantes en diversas carreras y materias. Este sistema debe registrar información sobre estudiantes, profesores, materias, programas y períodos académicos. Además, se espera que el sistema facilite la asignación de calificaciones y permita generar reportes de desempeño por estudiante y curso.

#### **Requerimientos Funcionales:**

1. Registrar información básica de los estudiantes (nombre completo, fecha de nacimiento, email, teléfono, dirección y carrera inscrita).
2. Gestionar el proceso de matrícula:
  - ❖ Fecha de inicio y fin de matrícula.
  - ❖ Tener referencia de la cantidad de estudiantes que se van a matricular
  - ❖ Saber que tipo de matricula se esta efectuando
3. Verificar la cantidad de materias y/o cursos que se habilitaran por cada periodo académico.
4. Gestionar la plantilla de profesores que dictaran cada materia
5. Habilitar la curricula para cada estudiante
6. Registrar a que período académico se está matriculando los estudiantes.
7. Generar reportes hacia cada escuela profesional sobre la cantidad de alumnado se ha matriculado respecto a la curricula enviada.

#### **Entidades Identificadas:**

1. **Estudiante:** Información básica del estudiante.
2. **Matrícula:** Información sobre las fechas de inicio y fin.
3. **Profesor:** Información básica del profesor.
4. **Materia:** Información básica de la materia con una pequeña descripción.
5. **Periodo Académico:** Información sobre cada periodo académico.

#### **Tablas y Campos Relevantes:**

**1. Tabla Estudiantes:** Contiene la información básica de todos los estudiantes.

- Nombre completo
- Código de estudiante
- Fecha de nacimiento
- Email
- Teléfono
- Dirección
- Carrera Inscrita

**2. Tabla Profesores:** Contiene la información básica de los profesores.

- Nombre completo
- Código de profesor
- Email
- Teléfono
- Especialidad

**3. Tabla Curricula:** Contiene la información básica de cada curricula.

- Nombre
- Descripción

**4. Tabla Escuela Profesional:** Contiene la información básica de cada escuela profesional.

- Código de Escuela Profesional
- Nombre
- Dirección

**5. Tabla Matrícula:** Contiene la información básica de cada matrícula

- Fecha de inicio
- Fecha fin
- Created\_at
- Updated\_at
- Deleted\_at
- Created\_by
- Updated\_by
- Deleted\_by

**6. Tabla de Materia:** Contiene información básica de cada materia.

- Código de materia
- Nombre
- Número de créditos

**7. Tabla de Periodo Académico:** Contiene información básica de cada periodo académico.

- Número de periodo
- Fecha de inicio
- Fecha fin

**8. Tabla de Tipo de Matrícula:** Contiene información básica de cada tipo de matrícula.

- Nombre
- Descripción

#### **Desafíos:**

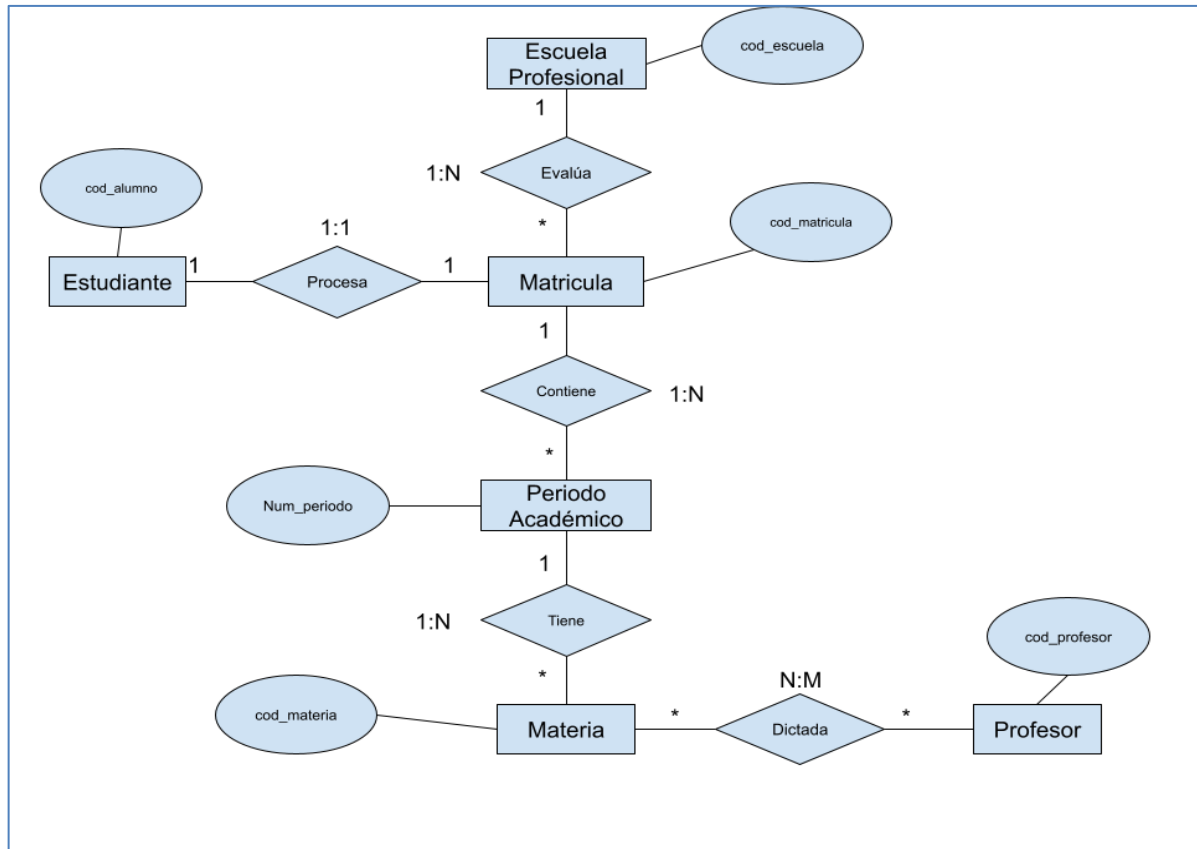
- **Volumen de Estudiantes:** Durante los periodos de matrícula, el alto volumen de estudiantes puede sobrecargar los sistemas y recursos disponibles, lo que puede llevar a fallos técnicos o largos tiempos de espera.
- **Personalización y flexibilidad:** Los estudiantes tienen necesidades diversas, como horarios personalizados, materias optativas y requisitos de programas específicos.
- **Gestión de cupos y disponibilidad:** Asegurar que los cupos en materias o secciones no se llenen de manera desigual o antes de que todos los estudiantes tengan la oportunidad de inscribirse.
- **Integración con otros sistemas:** La matrícula no es un proceso aislado; debe estar conectada con sistemas de pagos, registros académicos y calendarios institucionales.

#### **Beneficios Esperados:**

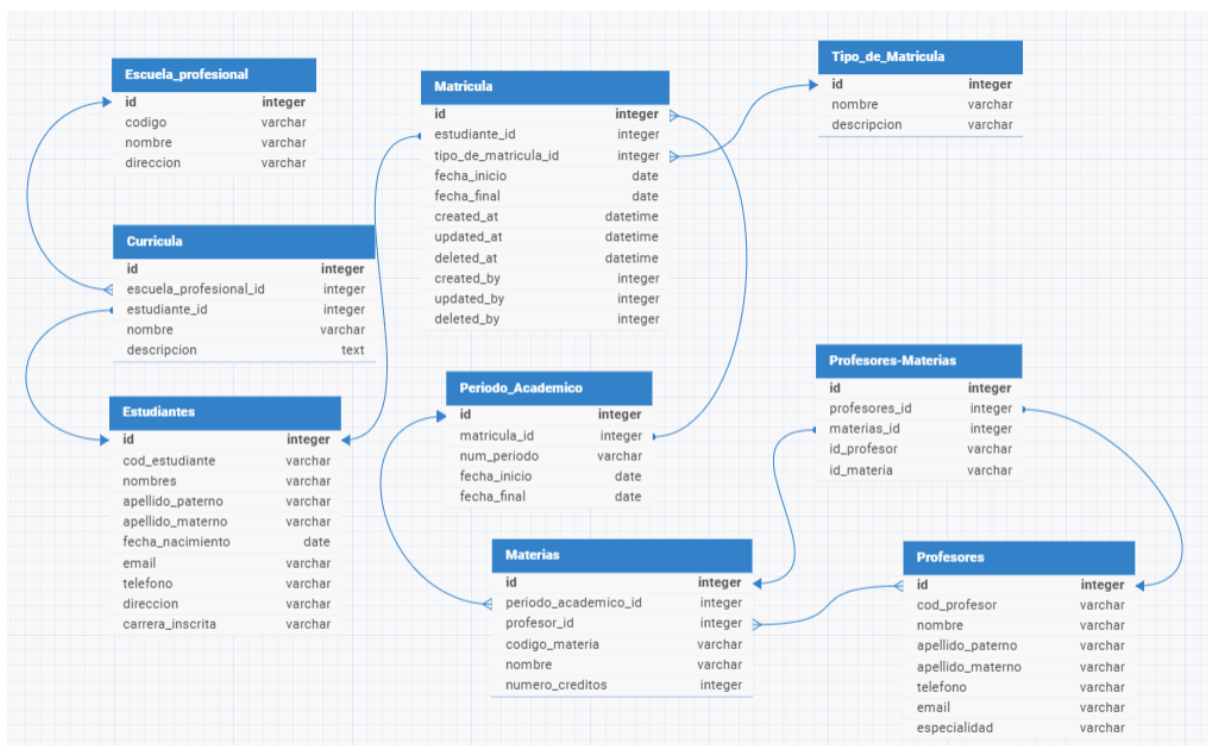
- Implementar sistemas de gestión escalables, como plataformas en la nube, y establecer horarios escalonados para la matrícula.
- Utilizar software de gestión que permita personalización y escenarios dinámicos según el perfil del estudiante.
- Implementar estrategias de asignación de cupos, como matrículas por prioridad, y utilizar sistemas de actualización en tiempo real.
- Usar plataformas integradas que faciliten la comunicación entre diferentes módulos (ERP académico).

## **II. Análisis: Diseño Conceptual de la base de datos.**

## GESTIÓN DE MATRICULA UNIVERSITARIA



### III. Diseño: Definición del Modelo Lógico de la Base de Datos.



#### IV. Construcción: Desarrollo de las sentencias SQL que permiten la construcción de las tablas y demás estructuras de la Base de Datos en SQL SERVER.

Creación de la tabla **Escuela Profesional**:

```
USE rp_Gestion_Matricula_rp;
GO

-- Tabla : Escuela Profesional
CREATE TABLE EscuelaProfesional (
    escuelaprofesional_id INT PRIMARY KEY IDENTITY (1,1),
    codigoesuelaprofesional VARCHAR (50) UNIQUE NOT NULL,
    nombre VARCHAR (100) NOT NULL,
    direccion VARCHAR (255) NOT NULL,
);
```

Creación de la tabla **Curricula**:

```
-- Tabla : Curricula
CREATE TABLE Curricula (
    curricula_id INT PRIMARY KEY IDENTITY (1,1),
    nombre VARCHAR (100) NOT NULL,
    descripcion TEXT NOT NULL,
    escuelaprofesional_id INT NOT NULL FOREIGN KEY (escuelaprofesional_id) REFERENCES EscuelaProfesional(escuelaprofesional_id),
);
```

Creación de la tabla **Tipo de Matrícula**:

```
-- Tabla : Tipo de Matricula
CREATE TABLE TipodeMatricula (
    tipodematricula_id INT PRIMARY KEY IDENTITY (1,1),
    nombre VARCHAR (100) NOT NULL,
    descripcion TEXT NOT NULL,
);
```

Creación de la tabla **Estudiantes**:

```
-- Tabla : Estudiantes
CREATE TABLE Estudiantes (
  estudiante_id INT PRIMARY KEY IDENTITY (1,1),
  codigo_estudiante VARCHAR (30) UNIQUE NOT NULL,
  nombres VARCHAR (50) NOT NULL,
  apellido_paterno VARCHAR (255) NOT NULL,
  apellido_materno VARCHAR (255) NOT NULL,
  fecha_nacimiento DATE NOT NULL,
  email VARCHAR (100) UNIQUE NOT NULL,
  telefono VARCHAR (20) NOT NULL,
  direccion VARCHAR (255) NOT NULL,
  carrera_inscrita VARCHAR (255) NOT NULL,
  curricula_id INT NOT NULL FOREIGN KEY (curricula_id) REFERENCES Curricula (curricula_id),
);
```

Creación de la tabla **Profesores**:

```
-- Tabla : Profesores
CREATE TABLE Profesores (
  profesor_id INT PRIMARY KEY IDENTITY (1,1),
  codigo_profesor VARCHAR (30) UNIQUE NOT NULL,
  nombre VARCHAR (50) NOT NULL,
  apellido_paterno VARCHAR (255) NOT NULL,
  apellido_materno VARCHAR (255) NOT NULL,
  email VARCHAR (100) UNIQUE NOT NULL,
  telefono VARCHAR (20) NOT NULL,
  especialidad VARCHAR(100) NOT NULL,
);
```

Creación de la tabla **Materias**:

```
-- Tabla : Materias
CREATE TABLE Materias (
  materia_id INT PRIMARY KEY IDENTITY (1,1),
  codigo_materia VARCHAR (30) UNIQUE NOT NULL,
  nombre VARCHAR (100) NOT NULL,
  creditos INT NOT NULL,
  descripcion TEXT,
  profesor_id INT NOT NULL FOREIGN KEY (profesor_id) REFERENCES Profesores(profesor_id),
);
```

### Creación de la tabla **Periodo Académico**:

```
-- Tabla : Periodo Academico
CREATE TABLE PeriodosAcademicos (
    periodoacademico_id INT PRIMARY KEY IDENTITY (1,1),
    nombre VARCHAR (100) NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_fin DATE NOT NULL,
    materia_id INT NOT NULL FOREIGN KEY (materia_id) REFERENCES Materias (materia_id),
);
```

### Creación de la tabla **Matrícula**:

```
-- Tabla : Matricula
CREATE TABLE Matricula (
    matricula_id INT PRIMARY KEY IDENTITY (1,1),
    fecha_inicio DATE NOT NULL,
    fecha_fin DATE NOT NULL,
    created_at DATETIME,
    update_at DATETIME,
    deleted_at DATETIME,
    created_by INTEGER,
    update_by INTEGER,
    deleted_by INTEGER,
    estudiante_id INT NOT NULL FOREIGN KEY (estudiante_id) REFERENCES Estudiantes (estudiante_id),
    periodoacademico_id INT NOT NULL FOREIGN KEY (periodoacademico_id) REFERENCES PeriodosAcademicos (periodoacademico_id),
    tipodematricula_id INT NOT NULL FOREIGN KEY (tipodematricula_id) REFERENCES TipodeMatricula (tipodematricula_id),
);
```

### Creación de la tabla **Profesores-Materias**:

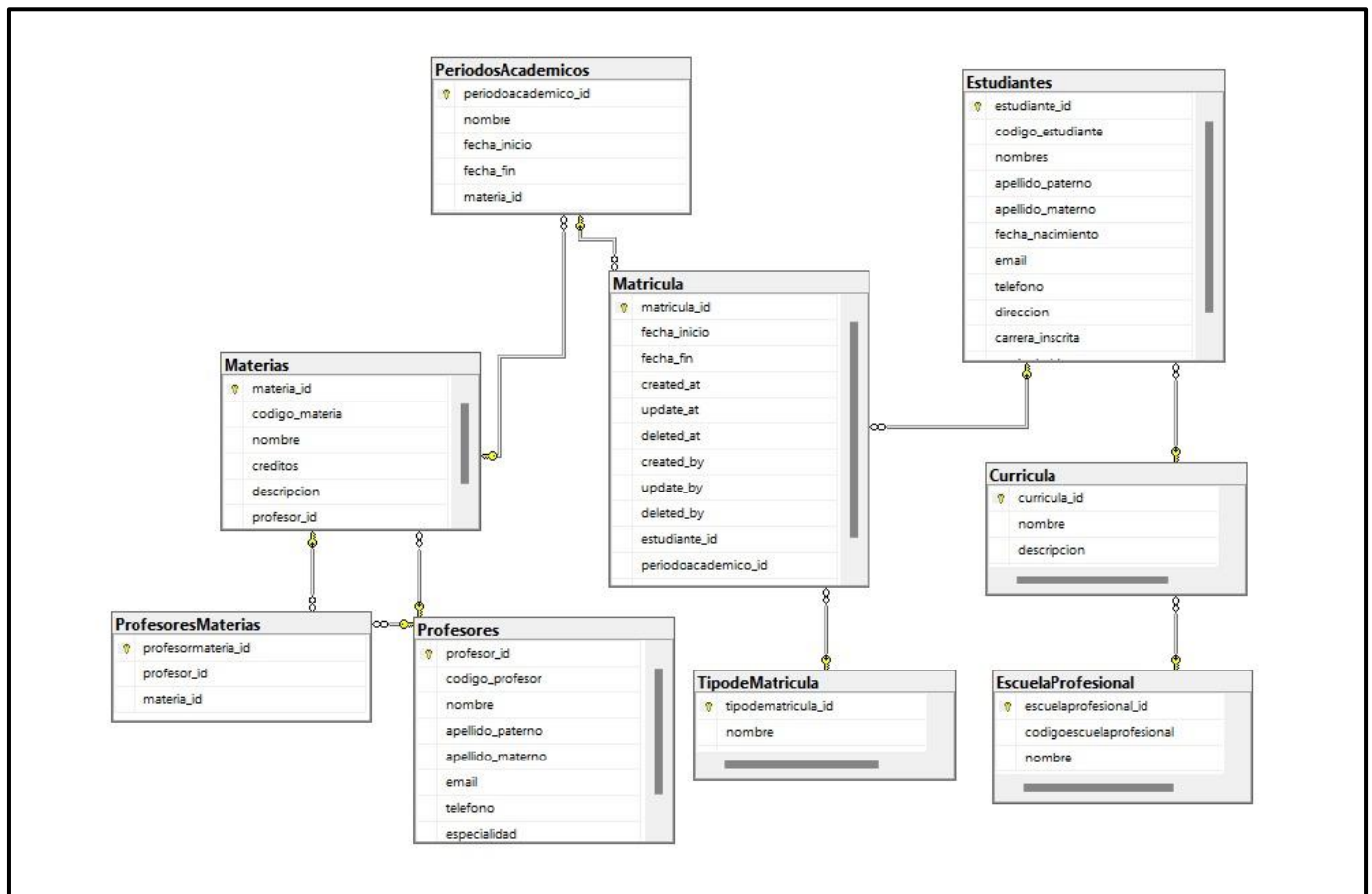
```
-- Tabla : Profesores-Materias
CREATE TABLE ProfesoresMaterias (
    profesormateria_id INT PRIMARY KEY IDENTITY (1,1),
    profesor_id INT NOT NULL FOREIGN KEY (profesor_id) REFERENCES Profesores (profesor_id),
    materia_id INT NOT NULL FOREIGN KEY (materia_id) REFERENCES Materias (materia_id),
);

GO
```



**V. Producto: Diseño Físico, Script de la Base de datos con inserción de datos (Mínimo 5 registros por tablas independientes o no transaccionales y mínimo 15 registros por cada tabla transaccional).**

**DISEÑO FÍSICO**



**Link del SCRIPT**

Se adjunta link de mi repositorio en **GITHUB** donde está incluido el Script:

- [https://github.com/RobertPrado01/rp\\_Gestion\\_Matricula\\_Universitaria\\_bda\\_6](https://github.com/RobertPrado01/rp_Gestion_Matricula_Universitaria_bda_6)

## INSERCIÓN DE DATOS

### Tabla Escuela Profesional

```
-- Escuela Profesional
SELECT * FROM EscuelaProfesional;

INSERT INTO EscuelaProfesional (codigoescuelaprofesional, nombre, direccion)
VALUES
('EP001', 'Escuela de Ingeniería Civil', 'Av. Principal 123, Ciudad Universitaria'),
('EP002', 'Escuela de Ingeniería de Sistemas', 'Av. Innovación 456, Edificio B'),
('EP003', 'Escuela de Arquitectura', 'Calle Diseño 789, Edificio C'),
('EP004', 'Escuela de Ingeniería Ambiental', 'Av. Verde 101, Bloque D'),
('EP005', 'Escuela de Medicina', 'Av. Salud 234, Complejo Médico'),
('EP006', 'Escuela de Ingeniería Industrial', 'Av. Proceso 567, Edificio E'),
('EP007', 'Escuela de Ingeniería Química', 'Av. Química 890, Edificio F'),
('EP008', 'Escuela de Psicología', 'Calle Humanidades 111, Edificio G'),
('EP009', 'Escuela de Ingeniería Electrónica', 'Av. Circuito 222, Edificio H'),
('EP010', 'Escuela de Ingeniería Mecánica', 'Av. Máquina 333, Edificio I');
```

### Tabla Curricula

```
-- Curricula
SELECT * FROM Curricula;

INSERT INTO Curricula (nombre, descripcion, escuela_profesional_id)
VALUES
('Curricula 2024 Ingeniería Civil', 'Plan curricular actualizado para Ingeniería Civil en 2024.', 1),
('Curricula 2023 Ingeniería de Sistemas', 'Estructura curricular de 2023 para la Escuela de Ingeniería de Sistemas.', 2),
('Curricula 2025 Arquitectura', 'Plan enfocado en diseño y urbanismo, versión 2025.', 3),
('Curricula 2023 Ingeniería Ambiental', 'Planificación de cursos para sostenibilidad y medio ambiente.', 4),
('Curricula 2024 Medicina', 'Diseñado para cumplir estándares internacionales en Medicina.', 5),
('Curricula 2022 Ingeniería Industrial', 'Plan enfocado en optimización de procesos industriales.', 6),
('Curricula 2025 Ingeniería Química', 'Estructura académica diseñada para química aplicada e industrial.', 7),
('Curricula 2023 Psicología', 'Plan curricular centrado en salud mental y psicología aplicada.', 8),
('Curricula 2024 Ingeniería Electrónica', 'Plan enfocado en circuitos y sistemas electrónicos avanzados.', 9),
('Curricula 2025 Ingeniería Mecánica', 'Currículo diseñada para técnicas de mecánica avanzada.', 10);
```



## Tabla Estudiantes

```
-- Estudiantes
```

```
SELECT * FROM Estudiantes;
```

```
INSERT INTO Estudiantes (codigo_estudiante, nombres, apellido_paterno, apellido_materno, fecha_nacimiento, email, telefono, direccion, carrera_inscrita, curricula_id)
VALUES
```

```
('E001', 'Luis', 'Pérez', 'Gómez', '2000-05-15', 'luis.perez@gmail.com', '987654321', 'Av. Principal 123', 'Ingeniería Civil', 3),
('E002', 'María', 'López', 'Torres', '1999-11-20', 'maria.lopez@gmail.com', '987654322', 'Calle Secundaria 45', 'Ingeniería de Sistemas', 4),
('E003', 'Carlos', 'Ramírez', 'Díaz', '2001-02-28', 'carlos.ramirez@gmail.com', '987654323', 'Jr. Las Flores 56', 'Arquitectura', 5),
('E004', 'Ana', 'Hernández', 'Cruz', '2002-07-10', 'ana.hernandez@gmail.com', '987654324', 'Av. Los Rosales 789', 'Ingeniería Ambiental', 6),
('E005', 'Jorge', 'García', 'Martínez', '1998-09-12', 'jorge.garcia@gmail.com', '987654325', 'Calle Nueva 67', 'Medicina', 7),
('E006', 'Laura', 'Martínez', 'Sánchez', '2000-12-30', 'laura.martinez@gmail.com', '987654326', 'Jr. Independencia 89', 'Ingeniería Industrial', 8),
('E007', 'David', 'Castro', 'Luna', '1999-04-22', 'david.castro@gmail.com', '987654327', 'Av. Universitaria 23', 'Ingeniería Química', 9),
('E008', 'Sofía', 'Rojas', 'Morales', '2001-01-14', 'sofia.rojas@gmail.com', '987654328', 'Calle Primavera 56', 'Psicología', 10),
('E009', 'Daniel', 'Chávez', 'Ortega', '1997-03-08', 'daniel.chavez@gmail.com', '987654329', 'Jr. Del Sol 14', 'Ingeniería Electrónica', 11),
('E010', 'Lucía', 'Guzmán', 'Peña', '2000-08-19', 'lucia.guzman@gmail.com', '987654330', 'Av. Colonial 456', 'Ingeniería Mecánica', 12),
('E011', 'Pedro', 'Flores', 'Vargas', '1998-06-21', 'pedro.flores@gmail.com', '987654331', 'Calle Central 234', 'Ingeniería Civil', 3),
('E012', 'Mónica', 'Salazar', 'Velázquez', '1999-10-15', 'monica.salazar@gmail.com', '987654332', 'Jr. América 67', 'Ingeniería de Sistemas', 4),
('E013', 'Hugo', 'Navarro', 'Jiménez', '2001-09-30', 'hugo.navarro@gmail.com', '987654333', 'Av. La Paz 98', 'Arquitectura', 5),
('E014', 'Paula', 'Cáceres', 'Ortiz', '2002-11-22', 'paula.caceres@gmail.com', '987654334', 'Calle Maravilla 43', 'Ingeniería Ambiental', 6),
('E015', 'Roberto', 'Zúñiga', 'Espinoza', '1997-05-05', 'roberto.zuniga@gmail.com', '987654335', 'Av. Progreso 76', 'Medicina', 7),
('E016', 'Carla', 'Aguilar', 'Carrasco', '2000-07-18', 'carla.aguilar@gmail.com', '987654336', 'Jr. Los Andes 12', 'Ingeniería Industrial', 8),
('E017', 'Juan', 'Villanueva', 'Mendoza', '1999-01-10', 'juan.villanueva@gmail.com', '987654337', 'Calle Azul 89', 'Ingeniería Química', 9),
('E018', 'Andrea', 'Reyes', 'Palacios', '2001-02-26', 'andrea.reyes@gmail.com', '987654338', 'Av. Del Sol 45', 'Psicología', 10),
('E019', 'Diego', 'Lara', 'Trujillo', '1998-10-11', 'diego.lara@gmail.com', '987654339', 'Calle Real 90', 'Ingeniería Electrónica', 11),
('E020', 'Camila', 'Fuentes', 'Estrada', '2000-05-03', 'camila.fuentes@gmail.com', '987654340', 'Jr. Comercio 56', 'Ingeniería Mecánica', 12);
```

## Tabla Profesores

```
-- Profesores
```

```
SELECT * FROM Profesores;
```

```
INSERT INTO Profesores (codigo_profesor, nombre, apellido_paterno, apellido_materno, email, telefono, especialidad)
VALUES
```

```
('PROF001', 'Juan', 'Pérez', 'González', 'juan.perez@univ.edu', '987654321', 'Matemáticas'),
('PROF002', 'María', 'López', 'Torres', 'maria.lopez@univ.edu', '987654322', 'Física'),
('PROF003', 'Carlos', 'Ramírez', 'Díaz', 'carlos.ramirez@univ.edu', '987654323', 'Química'),
('PROF004', 'Ana', 'Hernández', 'Cruz', 'ana.hernandez@univ.edu', '987654324', 'Biología'),
('PROF005', 'Jorge', 'García', 'Martínez', 'jorge.garcia@univ.edu', '987654325', 'Historia'),
('PROF006', 'Laura', 'Martínez', 'Sánchez', 'laura.martinez@univ.edu', '987654326', 'Literatura'),
('PROF007', 'David', 'Castro', 'Luna', 'david.castro@univ.edu', '987654327', 'Filosofía'),
('PROF008', 'Sofía', 'Rojas', 'Morales', 'sofia.rojas@univ.edu', '987654328', 'Economía'),
('PROF009', 'Daniel', 'Chávez', 'Ortega', 'daniel.chavez@univ.edu', '987654329', 'Psicología'),
('PROF010', 'Lucía', 'Guzmán', 'Peña', 'lucia.guzman@univ.edu', '987654330', 'Ingeniería Mecánica'),
('PROF011', 'Pedro', 'Flores', 'Vargas', 'pedro.flores@univ.edu', '987654331', 'Ingeniería Civil'),
('PROF012', 'Mónica', 'Salazar', 'Velázquez', 'monica.salazar@univ.edu', '987654332', 'Ingeniería Química'),
('PROF013', 'Hugo', 'Navarro', 'Jiménez', 'hugo.navarro@univ.edu', '987654333', 'Arquitectura'),
('PROF014', 'Paula', 'Cáceres', 'Ortiz', 'paula.caceres@univ.edu', '987654334', 'Ingeniería Ambiental'),
('PROF015', 'Roberto', 'Zúñiga', 'Espinoza', 'roberto.zuniga@univ.edu', '987654335', 'Ingeniería Industrial'),
('PROF016', 'Carla', 'Aguilar', 'Carrasco', 'carla.aguilar@univ.edu', '987654336', 'Ingeniería Electrónica'),
('PROF017', 'Juan', 'Villanueva', 'Mendoza', 'juan.villanueva@univ.edu', '987654337', 'Medicina'),
('PROF018', 'Andrea', 'Reyes', 'Palacios', 'andrea.reyes@univ.edu', '987654338', 'Odontología'),
('PROF019', 'Diego', 'Lara', 'Trujillo', 'diego.lara@univ.edu', '987654339', 'Física'),
('PROF020', 'Camila', 'Fuentes', 'Estrada', 'camila.fuentes@univ.edu', '987654340', 'Química');
```



Tabla **Materias**

```
-- Materias
SELECT * FROM Materias

INSERT INTO Materias (codigo_materia, nombre, credits, profesor_id)
VALUES
('MAT101', 'Cálculo Diferencial', 4, 1),
('FIS102', 'Física General', 4, 2),
('QUI103', 'Química General', 4, 3),
('COM104', 'Comunicación Oral y Escrita', 3, 4),
('ING105', 'Introducción a la Ingeniería', 2, 5),
('MAT201', 'Cálculo Integral', 4, 6),
('MAT202', 'Álgebra Lineal', 3, 7),
('FIS203', 'Física II', 4, 8),
('PRO204', 'Programación Básica', 3, 9),
('EST205', 'Estadística y Probabilidad', 3, 10),
('MAT301', 'Cálculo Vectorial', 4, 11),
('MAT302', 'Ecuaciones Diferenciales', 4, 12),
('MEC303', 'Mecánica de Fluidos', 4, 13),
('ELE304', 'Electromagnetismo', 4, 14),
('DIS305', 'Diseño Gráfico por Computadora', 3, 15),
('MAT401', 'Matemáticas Discretas', 3, 16),
('FIS402', 'Física Moderna', 4, 17),
('CIR403', 'Circuitos Eléctricos', 4, 18),
('QUI404', 'Química Orgánica', 4, 19),
('DIB405', 'Dibujo Técnico', 3, 20),
('PRO501', 'Procesos de Manufactura', 4, 1),
('ANA502', 'Análisis Numérico', 4, 2),
('TER503', 'Termodinámica', 4, 3),
('GES504', 'Gestión de Proyectos', 3, 4),
('IA505', 'Inteligencia Artificial', 3, 5),
('DIS601', 'Diseño de Sistemas', 4, 6).
```

Tabla **Periodo Académico**

```
-- Periodo Academico
SELECT * FROM PeriodosAcademicos;

INSERT INTO PeriodosAcademicos (nombre, fecha_inicio, fecha_fin, materia_id)
VALUES
('Primer Semestre 2020', '2020-01-01', '2020-06-30', 1),
('Segundo Semestre 2020', '2020-07-01', '2020-12-31', 2),
('Primer Semestre 2021', '2021-01-01', '2021-06-30', 3),
('Segundo Semestre 2021', '2021-07-01', '2021-12-31', 4),
('Primer Semestre 2022', '2022-01-01', '2022-06-30', 5),
('Segundo Semestre 2022', '2022-07-01', '2022-12-31', 6),
('Primer Semestre 2023', '2023-01-01', '2023-06-30', 7),
('Segundo Semestre 2023', '2023-07-01', '2023-12-31', 8),
('Primer Semestre 2024', '2024-01-01', '2024-06-30', 9),
('Segundo Semestre 2024', '2024-07-01', '2024-12-31', 10),
('Primer Semestre 2025', '2025-01-01', '2025-06-30', 11),
('Segundo Semestre 2025', '2025-07-01', '2025-12-31', 12),
('Primer Semestre 2026', '2026-01-01', '2026-06-30', 13),
('Segundo Semestre 2026', '2026-07-01', '2026-12-31', 14),
('Primer Semestre 2027', '2027-01-01', '2027-06-30', 15),
('Segundo Semestre 2027', '2027-07-01', '2027-12-31', 16),
('Primer Semestre 2028', '2028-01-01', '2028-06-30', 17),
('Segundo Semestre 2028', '2028-07-01', '2028-12-31', 18),
('Primer Semestre 2029', '2029-01-01', '2029-06-30', 19),
('Segundo Semestre 2029', '2029-07-01', '2029-12-31', 20);
```

## Tabla Tipo de Matricula

```
-- Tipo de Matricula
SELECT * FROM TipodeMatricula;

INSERT INTO TipoDeMatricula (nombre, descripcion)
VALUES
('Ordinaria', 'Matrícula regular dentro de los plazos establecidos'),
('Extraordinaria', 'Matrícula fuera del plazo ordinario con recargo adicional'),
('Especial', 'Matrícula para estudiantes con casos específicos o excepcionales');
```

## Tabla Matricula

```
-- Matricula
SELECT * FROM Matricula;

INSERT INTO Matricula (fecha_inicio, fecha_fin, created_at, update_at, deleted_at, created_by, update_by, deleted_by, estudiante_id, periodoacademico_id, tipodematricula)
VALUES
('2024-01-02', '2024-06-30', GETDATE(), NULL, NULL, 1, NULL, NULL, 7, 1, 1),
('2024-01-02', '2024-06-30', GETDATE(), NULL, NULL, 1, NULL, NULL, 8, 1, 1),
('2024-07-01', '2024-12-31', GETDATE(), NULL, NULL, 2, NULL, NULL, 9, 2, 2),
('2024-07-01', '2024-12-31', GETDATE(), NULL, NULL, 2, NULL, NULL, 10, 2, 2),
('2025-01-01', '2025-06-30', GETDATE(), NULL, NULL, 3, NULL, NULL, 11, 3, 1),
('2025-01-01', '2025-06-30', GETDATE(), NULL, NULL, 3, NULL, NULL, 12, 3, 1),
('2025-07-01', '2025-12-31', GETDATE(), NULL, NULL, 4, NULL, NULL, 13, 4, 3),
('2025-07-01', '2025-12-31', GETDATE(), NULL, NULL, 4, NULL, NULL, 14, 4, 3),
('2026-01-01', '2026-06-30', GETDATE(), NULL, NULL, 5, NULL, NULL, 15, 5, 1),
('2026-01-01', '2026-06-30', GETDATE(), NULL, NULL, 5, NULL, NULL, 16, 5, 1),
('2026-07-01', '2026-12-31', GETDATE(), NULL, NULL, 6, NULL, NULL, 17, 6, 2),
('2026-07-01', '2026-12-31', GETDATE(), NULL, NULL, 6, NULL, NULL, 18, 6, 2),
('2027-01-01', '2027-06-30', GETDATE(), NULL, NULL, 7, NULL, NULL, 19, 7, 3),
('2027-01-01', '2027-06-30', GETDATE(), NULL, NULL, 7, NULL, NULL, 20, 7, 3),
('2027-07-01', '2027-12-31', GETDATE(), NULL, NULL, 8, NULL, NULL, 21, 8, 1),
('2027-07-01', '2027-12-31', GETDATE(), NULL, NULL, 8, NULL, NULL, 22, 8, 1),
('2028-01-01', '2028-06-30', GETDATE(), NULL, NULL, 9, NULL, NULL, 23, 9, 2),
('2028-01-01', '2028-06-30', GETDATE(), NULL, NULL, 9, NULL, NULL, 24, 9, 2),
('2028-07-01', '2028-12-31', GETDATE(), NULL, NULL, 10, NULL, NULL, 25, 10, 3),
('2028-07-01', '2028-12-31', GETDATE(), NULL, NULL, 10, NULL, NULL, 26, 10, 3),
('2029-01-01', '2029-06-30', GETDATE(), NULL, NULL, 11, NULL, NULL, 7, 11, 1),
('2029-01-01', '2029-06-30', GETDATE(), NULL, NULL, 11, NULL, NULL, 8, 11, 1),
('2029-07-01', '2029-12-31', GETDATE(), NULL, NULL, 12, NULL, NULL, 9, 12, 2),
('2029-07-01', '2029-12-31', GETDATE(), NULL, NULL, 12, NULL, NULL, 10, 12, 2),
('2030-01-01', '2030-06-30', GETDATE(), NULL, NULL, 13, NULL, NULL, 11, 13, 3),
('2030-01-01', '2030-06-30', GETDATE(), NULL, NULL, 13, NULL, NULL, 12, 13, 3),
```

- V. Anexos: Evidencias de Consultas, funciones, procedimientos almacenados, vistas, etc. Realizados sobre la base de datos implementada. (5 ejercicios por integrante del grupo, Integrantes individuales (7 ejercicios))**

## **-- CONSULTAS SQL**

### **-- 1° CONSULTA, TODAS LAS ESCUELAS PROFESIONALES CON SUS NOMBRES Y DIRECCIONES ORDENADOS ALFABETICAMENTE.**

```
SELECT
    codigoescuelaprofesional, nombre, direccion
FROM EscuelaProfesional
ORDER BY nombre;
```

### **-- 2° CONSULTA, OBTEN EL LISTADO DE ESTUDIANTES JUNTO CON EL NOMBRE DE LA CURRICULA A LA QUE ESTAN INSCRITOS.**

```
SELECT
    E.codigo_estudiante,
    CONCAT (E.nombres, ' ', E.apellido_paterno, ' ', E.apellido_materno) AS 'Nombre
completo',
    C.nombre AS curricula_nombre
FROM Estudiantes E
INNER JOIN Curricula C ON E.curricula_id = C.curricula_id;
```

### **-- 3° CONSULTA, MUESTRA EL LISTADO DE PROFESORES Y LAS MATERIAS QUE DICTAN, INCLUYENDO SUS CREDITOS.**

```
SELECT
    CONCAT (P.nombre, ' ', P.apellido_paterno, ' ', P.apellido_materno) AS 'Profesor nombre
completo',
    P.especialidad,
    M.nombre AS 'materia',
    M.creditos
FROM Profesores P
INNER JOIN Materias M ON P.profesor_id = M.profesor_id;
```

### **-- 4° CONSULTA, OBTEN EL NOMBRE COMPLETO DE LOS ESTUDIANTES Y LAS FECHAS DE MATRICULA DEL PERIODO ACADEMICO ACTUAL.**

```
SELECT
    CONCAT (E.nombres, ' ', E.apellido_paterno, ' ', E.apellido_materno) AS 'Nombre
completo',
    M.fecha_inicio,
    M.fecha_fin
FROM Estudiantes E
INNER JOIN Matrícula M ON E.estudiante_id = M.estudiante_id
```

```
INNER JOIN PeriodosAcademicos PA ON M. periodoacademico_id = PA.  
periodoacademico_id;
```

**-- 5° CONSULTA, MUESTRA LAS MATERIAS QUE SE DICTAN EN EL PERIODO ACADEMICO ACTUAL CON SUS FECHAS DE INICIO Y FIN.**

```
SELECT  
    PA. nombre AS 'Periodo Academico',  
    M.nombre AS 'Materia',  
    PA. fecha_inicio,  
    PA. fecha_fin  
FROM PeriodosAcademicos PA  
INNER JOIN Materias M ON PA. materia_id = M. materia_id;
```

**-- 6° CONSULTA, CAMBIA EL NUMERO DE CREDITOS DE LA MATERIA "CALCULO DIFERENCIAL" A 6.**

```
UPDATE Materias  
SET credits = 6  
WHERE nombre = 'Cálculo Diferencial';
```

**-- 7° CONSULTA, OBTENER EL NUMERO TOTAL DE ESTUDIANTES POR CADA CURRICULA.**

```
SELECT  
    C.nombre AS 'Curricula',  
    COUNT (E. estudiante_id) AS 'Total estudiantes'  
FROM Curricula C  
LEFT JOIN Estudiantes E ON C. curricula_id = E. curricula_id  
GROUP BY C. nombre;
```

**-- 8° CONSULTA, OBTEN EL LISTADO DE ESTUDIANTES QUE AUN NO TIENEN MATRICULA REGISTRADA.**

```
SELECT  
    E. estudiante_id,  
    CONCAT (E. nombres, ' ', E. apellido_paterno, ' ', E. apellido_materno) AS 'Nombre  
completo'  
FROM Estudiantes E  
LEFT JOIN Matricula M ON E. estudiante_id = M. estudiante_id  
WHERE M. matricula_id IS NULL;
```

**-- 9° CONSULTA, MUESTRA EL ESTUDIANTE JUNTO CON EL TOTAL DE CREDITOS DE LAS MATERIAS QUE ESTAN MATRICULADOS.**

```
SELECT  
    CONCAT (E. nombres, ' ', E. apellido_paterno, ' ', E. apellido_materno) AS 'Nombre  
completo',
```



```

SUM (M. creditos) AS 'Total creditos'
FROM Estudiantes E
JOIN Matricula MA ON E. estudiante_id = MA. estudiante_id
JOIN PeriodosAcademicos PA ON MA. periodoacademico_id = PA. periodoacademico_id
JOIN Materias M ON PA. materia_id = M. materia_id
GROUP BY E. estudiante_id, E. nombres, E. apellido_paterno, E. apellido_materno;

```

**-- 10° CONSULTA, ENCUENTRA AL PROFESOR CON MAS MATERIAS ASIGNADAS Y MUESTRA SU NOMBRE COMPLETO JUNTO CON EL NUMERO DE MATERIAS.**

```

SELECT TOP 1
    CONCAT (P. nombre, ' ', P.apellido_paterno, ' ', P.apellido_materno) AS 'Profesor nombre completo',
    COUNT (M. materia_id) AS 'Total materias'
FROM Profesores P
INNER JOIN Materias M ON P. profesor_id = M. profesor_id
GROUP BY P. profesor_id, P. nombre, P. apellido_paterno, P. apellido_materno
ORDER BY 'Total materias' DESC;

```

**-- 11° CONSULTA, LISTA EL NOMBRE DE LOS ESTUDIANTES JUNTO CON LAS MATERIAS QUE ESTAN CURSANDO ACTUALMENTE.**

```

SELECT
    CONCAT (E. nombres, ' ', E. apellido_paterno, ' ', E. apellido_materno) AS 'Nombre completo',
    M.nombre AS 'Materia'
FROM Estudiantes E
INNER JOIN Matrícula MA ON E. estudiante_id = MA. estudiante_id
INNER JOIN PeriodosAcademicos PA ON MA. periodoacademico_id = PA. periodoacademico_id
INNER JOIN Materias M ON PA. materia_id = M. materia_id
WHERE GETDATE () BETWEEN PA. fecha_inicio AND PA. fecha_fin;

```