

## **INGENIERÍA DE SOFTWARE**



### **BASES DE DATOS I**

#### **Modelo Conceptual y Modelo Lógico del Proyecto Final**

Docentes:

- Prof. Leydi Beatriz Manrique Tejada
- Prof. Juan Carlos Gomez Boza

Estudiantes:

- Edwin Eduardo Ccama Pari
- Diego Fabrizio Leon Araujo

**AREQUIPA, 25 DE SEPTIEMBRE DEL 2024**

## INDICE

1.	RESEÑA.....	3
2.	PROBLEMÁTICA .....	3
3.	MODELO CONCEPTUAL.....	4
4.	RELACIONES DE CARDINALIDAD .....	8
	Relación "Estudiantes" y "Cursos" (Solicita): .....	8
	Relación "Cursos" y "Profesores" (Imparte): .....	8
	Relación "Cursos" y "DisponibilidadCurso" (Verifica): .....	8
	Relación "Cursos" y "Orden" (Pagar):.....	8
	Relación "Salones" y "Orden" (Requiere): .....	8
	Relación "Salones" y "DisponibilidadSalon" (Verifica): .....	8
	Relación "Orden" y "Horario" (Fijar):.....	8
	Relación "Horario" y "Día": .....	8
	Relación ""Horario" y DisponibilidadHorario": .....	8
5.	NORMALIZACION.....	9
5.1.	Primera Forma Normal (1FN):.....	9
5.2.	Segunda Forma Normal (2FN): .....	9
5.3.	Tercera Forma Normal (3FN): .....	10
6.	CREACION DEL MODELO FISICO .....	11
6.1.	Creación de las bases de datos .....	11
6.2.	Insertar Datos .....	14
7.	CONSULTAS A LAS BASES DE DATOS .....	18
7.1.	Querys de consultas.....	18
7.2.	Resultados de Querys.....	19
8.	CREACION DE VISTAS.....	20
8.1.	Creacion de Vistas.....	20
8.2.	Ejecucion de Vistas .....	21
9.	STORE PROCEDURE.....	22
9.1.	Creacion de Store Procedure.....	22
9.2.	Ejecucion de Store Procedure .....	23
10.	TRIGGERS.....	24
10.1.	Creación de Trigger .....	24
10.2.	Probar de Trigger.....	25
11.	USUARIOS Y ROLES.....	25
11.1.	Creación de usuarios.....	25
11.2.	Asignación de roles.....	25
12.	CONEXION CON PYTHON.....	26
11.1.	Creación de la conexión.....	26
11.2.	Query de Consulta hacia la Base de Datos .....	27

# MATHMATIC

## 1. RESEÑA

Mathmatic es una prestigiosa academia de refuerzo académico ubicada en Arequipa, Perú. Desde su fundación en 2015 por Milenia, la academia se ha dedicado a ofrecer servicios de reforzamiento en diversas disciplinas para estudiantes de escuelas y colegios de la región.

Especializada en cursos como Matemáticas, Biología, Química, Termodinámica, Física, Estadística, entre otros, Mathmatic se ha consolidado como una referencia en el ámbito educativo local. Su equipo de profesores, que incluye a destacados profesionales como Angel, Raul, Pedro, Marvin, Miguel, Luis, Jorge, Anabel y Patricio, garantiza una enseñanza de calidad personalizada y adaptada a las necesidades de cada estudiante.

Con el objetivo de fortalecer los conocimientos de sus alumnos y prepararlos para futuros desafíos académicos, Mathmatic se enfoca en proporcionar un aprendizaje efectivo, orientado a la mejora del rendimiento escolar y a la superación personal en materias clave.

## 2. PROBLEMÁTICA

**Pérdida de información importante:** La información sobre los clientes, profesores, horarios y pagos es propensa a extraviarse al ser registrada en formatos físicos.

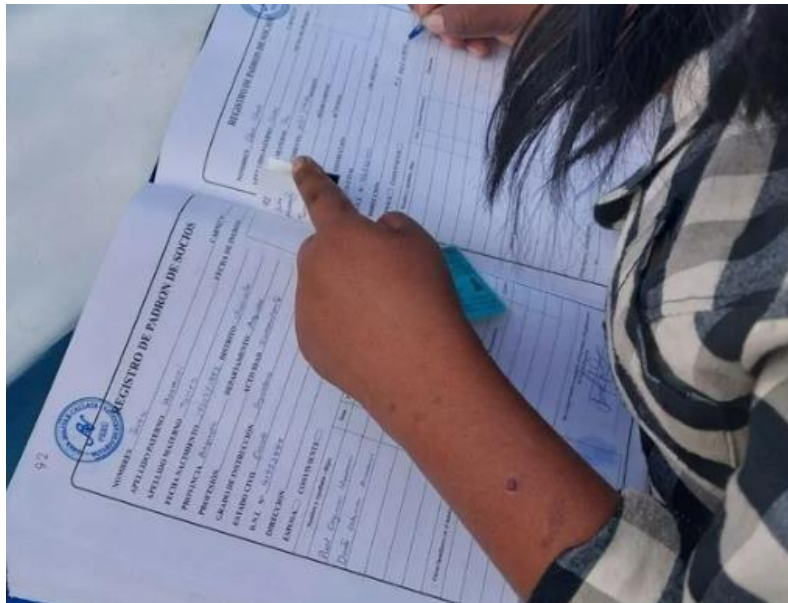
**Desorden administrativo:** La falta de un registro centralizado y actualizado conlleva a errores frecuentes en la asignación de clases, seguimiento de pagos, y control de disponibilidad de los profesores.

**Falta de control y eficiencia:** Los procesos, como la programación de clases y la generación de órdenes de servicio, se realizan de manera manual, lo que retrasa el funcionamiento y afecta la calidad del servicio ofrecido.

Estos problemas afectan la capacidad de la academia para crecer y manejar eficientemente su operación. Un sistema de base de datos estructurado permitiría a Mathmatic solucionar estos inconvenientes, automatizando procesos claves y mejorando la calidad de su servicio.

A continuación, los errores concurrentes al llevar el negocio con un formato manual y un simple cuadernillo de notas y la programación de los cursos:

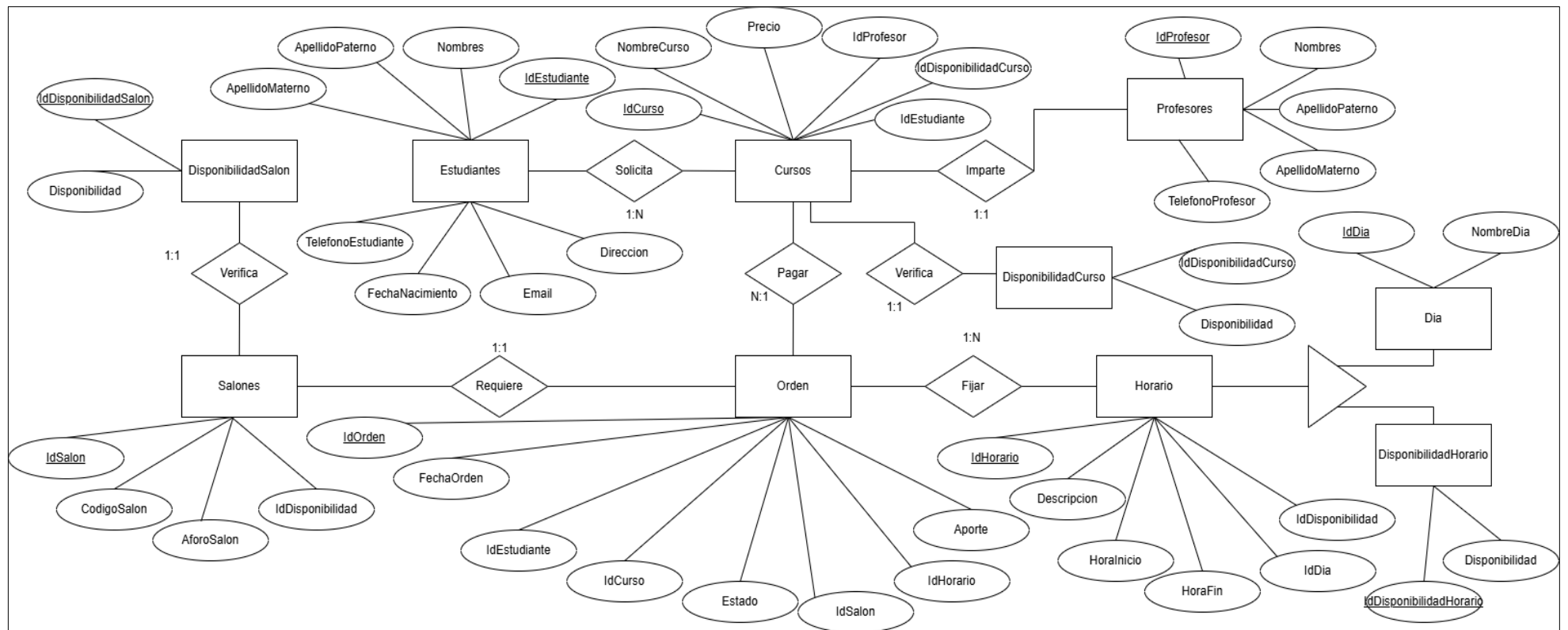
- ✓ **Duplicación de órdenes:** Debido a la falta de control digitalizado, un cliente podría recibir clases duplicadas, lo que causa insatisfacción y requiere un ajuste manual.
- ✓ **Errores en la asignación de profesores:** Dado que los horarios no están controlados en un sistema digital, un profesor puede ser asignado a dos clases a la misma hora, generando conflictos y retrasos en las lecciones.
- ✓ **Falta de seguimiento a pagos:** Al no contar con un registro actualizado, algunos pagos podrían no ser registrados correctamente, lo que podría generar desconfianza en los clientes.



*Registro de Estudiantes Manualmente*

### 3. MODELO CONCEPTUAL

Mathmatic es una academia que ofrece servicios a los Clientes, donde los Profesores imparten Clases de diferentes Cursos como física, matemáticas, biología, etc. Las clases se pactan mediante una Orden, que registra la fecha, hora y duración. Los Profesores (ej: Angel, Raul, Anabel) tienen atributos como IdProfesor, curso dictado, edad y disponibilidad, mientras que los Clientes tienen IdEstudiante, nombres, apellidos, email, fecha de nacimiento, y dirección. Cada Orden tiene IdOrden, IdEstudiante, IdProfesor, curso, salón, y estado de la orden.



**Diagrama: Entidad Relación Mathmatic**

## **Entidades:**

### **1. Estudiantes**

- IdEstudiante(PK)
- Nombres
- ApellidoPaterno
- ApellidoMaterno
- TelefonoEstudiante
- FechaNacimiento
- Email
- Dirección

### **2. Cursos**

- IdCurso(PK)
- NombreCurso
- Precio
- IdDisponibilidadCurso(FK)
- IdProfesor(FK)
- IdEstudiantes(FK)

### **3. Profesores**

- IdProfesor(PK)
- Nombres
- ApellidoPaterno
- ApellidoMaterno
- TelefonoProfesor

### **4. Salones**

- IdSalon(PK)
- CodigoSalon
- AforoSalon
- IdDisponibilidad(FK)

### **5. DisponibilidadSalon**

- IdDisponibilidadSalon(PK)
- Disponibilidad

### **6. Orden**

- IdOrden(PK)
- FechaOrden
- IdEstudiante(FK)
- IdCurso(FK)
- IdSalon(FK)
- IdHorario(FK)

- Estado

## **7. Horario**

- IdHorario(PK)
- Descripcion
- HoraInicio
- HoraFin
- IdDia(FK)
- IdDisponibilidad(FK)

## **8. DisponibilidadCurso**

- IdDisponibilidadCurso(PK)
- Disponibilidad

## **9. Dia**

- IdDia(PK)
- NombreDia

## **10. DisponibilidadHorario**

- IdDisponibilidadHorario(PK)
- Disponibilidad

## **11. Solicita**

- IdEstudiante (FK)
- IdCurso (FK)

## **12. Imparte**

- IdProfesor (FK)
- IdCurso (FK)

## 4. RELACIONES DE CARDINALIDAD

### Relación "Estudiantes" y "Cursos" (Solicita):

- Un Estudiante puede solicitar muchos Cursos (1:N).
- Muchos Cursos puede ser solicitado por un Estudiantes (N:1).

### Relación "Cursos" y "Profesores" (Imparte):

- Un Curso es impartido por un único Profesor (1:1).
- Un Profesor puede impartir un único Curso (1:1).

### Relación "Cursos" y "DisponibilidadCurso" (Verifica):

- Un Curso verifica una única DisponibilidadCurso (1:1).
- Una DisponibilidadCurso está asociada a un único Curso (1:1).

### Relación "Cursos" y "Orden" (Pagar):

- Una Orden está asociada a muchos Cursos (1:N).
- Muchos cursos pueden estar asociado a una unica Órdene (N:1).

### Relación "Salones" y "Orden" (Requiere):

- Un Salón puede estar asociado a una Órdene (1:1).
- Una Orden requiere un único Salón (1:1).

### Relación "Salones" y "DisponibilidadSalon" (Verifica):

- Un Salón verifica una única DisponibilidadSalon (1:1).
- Una DisponibilidadSalon está asociada a un único Salón (1:1).

### Relación "Orden" y "Horario" (Fijar):

- Una Orden puede fijar muchos Horarios (1:N).
- Varios Horarios está asociado a una única Orden (N:1).

### Relación "Horario" y "Día":

- Un Horario puede estar relacionada con varios Días (1:N).
- Varios Días puede tener un Horario asociado (N:1).

### Relación ""Horario" y DisponibilidadHorario":

- Un Horario puede estar asociada a una sola DiponibilidadHorario (1:1).
- Una sola DisponibilidadHorario está asociado a un solo Horario(1:1).



## 5. NORMALIZACION

Para realizar la normalización de los atributos de las entidades, obtenemos todos los componentes existentes identificados:

Componentes (IdEstudiante, Nombres, Apellido Paterno, ApellidoMaterno, TeléfonoAlumno, FechaNacimiento, Email, Direccion, IdCurso, NombreCurso, Precio, IdDisponibilidadCurso, IdProfesor, IdProfesor, Nombres, ApellidoPaterno, ApellidoMaterno, TeléfonoProfesor, IdSalon, CodigoSalon, AforoSalon, IdDisponibilidad, IdDisponibilidadSalon, Disponibilidad, IdDisponibilidadCurso, IdDisponibilidadHorario, IdOrden, FechaOrden, IdEstudiante, IdCurso, IdSalon, IdHorario, Estado, IdHorario, Descripcion, Aporte, HoraInicio, HoraFin, IdDia, NombreDia)

### 5.1. *Primera Forma Normal (1FN):*

Asegura que todos los atributos sean atómicos (sin valores repetidos, atómicos).

Estudiante (IdEstudiante, IdOrden, Nombres, ApellidosPaterno, ApellidoMaterno, FechaNacimiento, Direccion, TelefonoEstudiante, Email).

Cursos (IdCurso, IdOrden, NombreCurso, Precio, DisponibilidadCurso).

Profesor (IdProfesor, IdOrden, Nombres, ApellidoPaterno, ApellidoMaterno, TelefonoProfesor).

Orden (IdOrden, FechaOrden, Horario, IdCurso, IdEstudiante, IdProfesorEstado, HorasDictadas, Salon, Importe).

### 5.2. *Segunda Forma Normal (2FN):*

Todas las tablas deben depender completamente de la clave primaria, no debe haber dependencias parciales.

Estudiante (IdEstudiante, Nombres, ApellidosPaterno, ApellidoMaterno, FechaNacimiento, Direccion, TelefonoEstudiante, Email).

Cursos (IdCurso, NombreCurso, Precio, DisponibilidadCurso, IdProfesor).

Profesor (IdProfesor, Nombres, ApellidoPaterno, ApellidoMaterno, TelefonoProfesor).

Horario (IdHorario, Descripcion, HoraInicio, HoraFin, Dia, Disponibilidad)

Salones (IdSalon, CodigoSalon, AforoSalon, Disponibilidad).

Orden (IdOrden, FechaOrden, IdCurso, IdEstudiante, IdProfesorEstado, HorasDictadas, Importe)

### **5.3. Tercera Forma Normal (3FN):**

Asegura que no existan dependencias transitivas.

Estudiante (IdEstudiante, Nombres, ApellidosPaterno, ApellidoMaterno, FechaNacimiento, Direccion, TelefonoEstudiante, Email).

Cursos (IdCurso, NombreCurso, Precio, DisponibilidadCurso, IdProfesor).

Profesor (IdProfesor, Nombres, ApellidoPaterno, ApellidoMaterno, TelefonoProfesor)

Horario (IdHorario, Descripcion, HoraInicio, HoraFin)

DisponibilidadHorario (IdDisponibilidadHorario, Disponibilidad)

Dia (IdDia, NombreDia).

DisponibilidadDia (IdDisponibilidadDia, Disponibilidad).

Salones (IdSalon, CodigoSalon, AforoSalon).

DisponibilidadSalon (IdDisponibilidadesSalon, Disponibilidad).

Orden (IdOrden, FechaOrden, IdCurso, IdEstudiante, IdProfesorEstado, HorasDictadas, Importe).

Solicita (IdEstudiante, IdCurso).

Imparte (IdCurso, IdProfesor).

## 6. CREACION DEL MODELO FISICO

### 6.1. Creación de las bases de datos

Después de realizar el modelo conceptual y el modelo lógico, se comienza a estructurar el modelo físico, comenzando por la base de datos, las tablas, sus atributos y llaves respectivas:

```
-- Crear la base de datos
CREATE DATABASE BD_Mathmatic_01;
GO

-- Usar la base de datos creada
USE BD_Mathmatic_01;
GO

-- Crear la tabla Estudiantes
CREATE TABLE Estudiantes (
    IdEstudiante INT PRIMARY KEY IDENTITY(1,1),
    Nombres NVARCHAR(100) NOT NULL,
    ApellidoPaterno NVARCHAR(100) NOT NULL,
    ApellidoMaterno NVARCHAR(100) NOT NULL,
    TelefonoEstudiante NVARCHAR(15) NOT NULL,
    FechaNacimiento DATE NOT NULL,
    Email NVARCHAR(150),
    Direccion NVARCHAR(200)
);

-- Crear la tabla Profesores
CREATE TABLE Profesores (
    IdProfesor INT PRIMARY KEY IDENTITY(1,1),
    Nombres NVARCHAR(100) NOT NULL,
    ApellidoPaterno NVARCHAR(100) NOT NULL,
    ApellidoMaterno NVARCHAR(100) NOT NULL,
    TelefonoProfesor NVARCHAR(15) NOT NULL
);

-- Crear la tabla DisponibilidadSalon
CREATE TABLE DisponibilidadSalon (
    IdDisponibilidadSalon INT PRIMARY KEY IDENTITY(1,1),
    Disponibilidad BIT NOT NULL
);

-- Crear la tabla Salones
CREATE TABLE Salones (
    IdSalon INT PRIMARY KEY IDENTITY(1,1),
    CodigoSalon NVARCHAR(50) NOT NULL,
    AforoSalon INT NOT NULL,
    IdDisponibilidad INT NOT NULL,
    FOREIGN KEY (IdDisponibilidad) REFERENCES
DisponibilidadSalon(IdDisponibilidadSalon)
);

-- Crear la tabla DisponibilidadCurso
CREATE TABLE DisponibilidadCurso (
    IdDisponibilidadCurso INT PRIMARY KEY IDENTITY(1,1),
    Disponibilidad BIT NOT NULL
);

-- Crear la tabla Cursos
CREATE TABLE Cursos (
    IdCurso INT PRIMARY KEY IDENTITY(1,1),
    NombreCurso NVARCHAR(100) NOT NULL,
```

```

        Precio DECIMAL(10,2) NOT NULL,
        IdDisponibilidadCurso INT NOT NULL,
        IdProfesor INT NOT NULL,
        FOREIGN KEY (IdDisponibilidadCurso) REFERENCES
DisponibilidadCurso(IdDisponibilidadCurso),
        FOREIGN KEY (IdProfesor) REFERENCES Profesores(IdProfesor)
    );

-- Crear la tabla Dia
CREATE TABLE Dia (
    IdDia INT PRIMARY KEY IDENTITY(1,1),
    NombreDia NVARCHAR(50) NOT NULL
);

-- Crear la tabla DisponibilidadHorario
CREATE TABLE DisponibilidadHorario (
    IdDisponibilidadHorario INT PRIMARY KEY IDENTITY(1,1),
    Disponibilidad BIT NOT NULL
);

-- Crear la tabla Horario
CREATE TABLE Horario (
    IdHorario INT PRIMARY KEY IDENTITY(1,1),
    Descripcion NVARCHAR(200),
    HoraInicio TIME NOT NULL,
    HoraFin TIME NOT NULL,
    IdDia INT NOT NULL,
    IdDisponibilidad INT NOT NULL,
    FOREIGN KEY (IdDia) REFERENCES Dia(IdDia),
    FOREIGN KEY (IdDisponibilidad) REFERENCES
DisponibilidadHorario(IdDisponibilidadHorario)
);

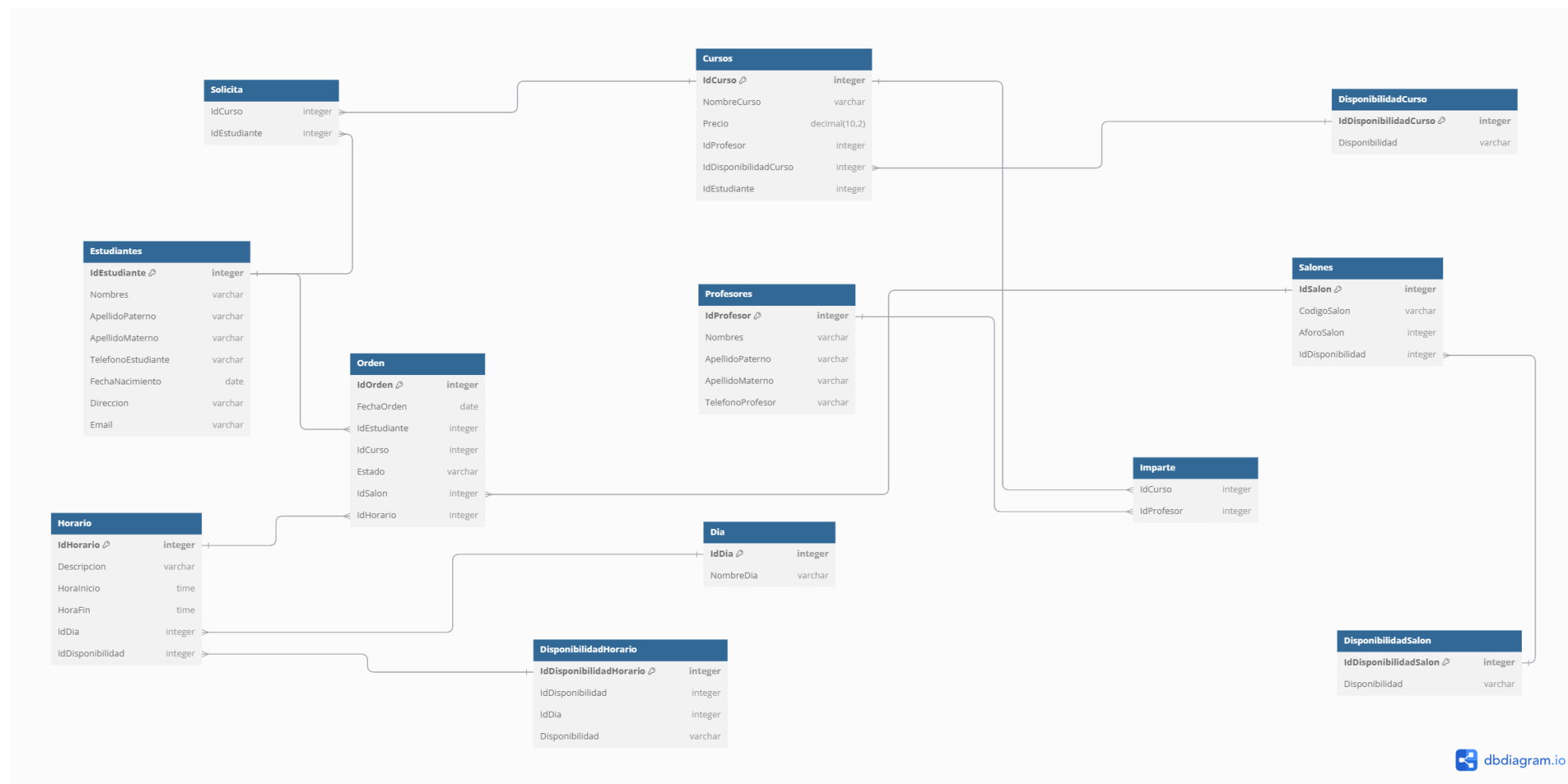
-- Crear la tabla Orden
CREATE TABLE Orden (
    IdOrden INT PRIMARY KEY IDENTITY(1,1),
    FechaOrden DATE NOT NULL,
    IdEstudiante INT NOT NULL,
    IdCurso INT NOT NULL,
    IdSalon INT NOT NULL,
    IdHorario INT NOT NULL,
    Estado BIT NOT NULL,
    FOREIGN KEY (IdEstudiante) REFERENCES Estudiantes(IdEstudiante),
    FOREIGN KEY (IdCurso) REFERENCES Cursos(IdCurso),
    FOREIGN KEY (IdSalon) REFERENCES Salones(IdSalon),
    FOREIGN KEY (IdHorario) REFERENCES Horario(IdHorario)
);

-- Crear la tabla Solicita (relación entre Estudiantes y Cursos)
CREATE TABLE Solicita (
    IdEstudiante INT NOT NULL,
    IdCurso INT NOT NULL,
    PRIMARY KEY (IdEstudiante, IdCurso),
    FOREIGN KEY (IdEstudiante) REFERENCES Estudiantes(IdEstudiante),
    FOREIGN KEY (IdCurso) REFERENCES Cursos(IdCurso)
);

-- Crear la tabla Imparte (relación entre Profesores y Cursos)
CREATE TABLE Imparte (
    IdProfesor INT NOT NULL,
    IdCurso INT NOT NULL,
    PRIMARY KEY (IdProfesor, IdCurso),
    FOREIGN KEY (IdProfesor) REFERENCES Profesores(IdProfesor),
    FOREIGN KEY (IdCurso) REFERENCES Cursos(IdCurso)
);
GO

```

## Esquema Físico de las Tablas



**Diagrama: Esquema Físico de la Base de Datos Mathmatic**

## 6.2. Insertar Datos

Es sumamente importante insertar los datos considerando las restricciones de las llaves primarias y foráneas, en esta oportunidad hemos insertado los datos manuales de la siguiente manera:

```
use BD_Mathmatic_01
go
--DisponibilidadCurso
INSERT INTO DisponibilidadCurso (Disponibilidad)
VALUES (1),(0); -- 1 indica que los cursos están disponibles (true)
-- Insertar datos en la tabla Profesores
INSERT INTO Profesores (Nombres, ApellidoPaterno, ApellidoMaterno, TelefonoProfesor)
VALUES
('Juan', 'Perez', 'Lopez', '951234567'),
('Maria', 'Gonzalez', 'Diaz', '952345678'),
('Carlos', 'Ramirez', 'Torres', '953456789'),
('Ana', 'Martinez', 'Hernandez', '954567890'),
('Luis', 'Garcia', 'Castro', '955678901'),
('Sofia', 'Lopez', 'Morales', '956789012'),
('Miguel', 'Hernandez', 'Vargas', '957890123'),
('Laura', 'Torres', 'Cruz', '958901234'),
('Daniel', 'Sanchez', 'Gutierrez', '959012345'),
('Lucia', 'Cruz', 'Rojas', '951123456'),
('Fernando', 'Castillo', 'Martinez', '952234567'),
('Gabriela', 'Vargas', 'Soto', '953345678'),
('Jose', 'Flores', 'Perez', '954456789'),
('Paola', 'Mendoza', 'Ramirez', '955567890'),
('Antonio', 'Ortega', 'Ruiz', '956678901'),
('Veronica', 'Silva', 'Jimenez', '957789012'),
('Jorge', 'Morales', 'Fernandez', '958890123'),
('Patricia', 'Gutierrez', 'Vega', '959901234'),
('Diego', 'Rojas', 'Campos', '951234567'),
('Carolina', 'Reyes', 'Villanueva', '952345678');

-- Insertar datos en la tabla Cursos
INSERT INTO Cursos (NombreCurso, Precio, IdDisponibilidadCurso, IdProfesor)
VALUES
('Matematica Basica', 20.00, 1, 1),
('Biologia', 25.00, 1, 2),
('Quimica', 30.00, 1, 3),
('Fisica', 35.00, 1, 4),
('Calculo', 20.00, 1, 5),
('Termodinamica', 25.00, 1, 6),
('Estatica', 30.00, 1, 7),
('Algebra', 35.00, 1, 8),
('Razonamiento Matematico', 20.00, 1, 9),
('Analisis de Logaritmos', 25.00, 1, 10),
('Python', 30.00, 1, 11),
('C++', 35.00, 1, 12),
('JavaScript', 20.00, 1, 13),
('Trigonometria', 25.00, 1, 14),
('Programacion Orientada a Objetos', 30.00, 1, 15),
('Bases de Datos', 35.00, 1, 16),
('Redes de Computadoras', 20.00, 1, 17),
('Ingenieria de Software', 25.00, 1, 18),
('Geometria Analitica', 30.00, 1, 19),
('Estadistica y Probabilidades', 35.00, 1, 20);
GO
```

```

INSERT INTO DisponibilidadSalon (Disponibilidad)
VALUES (1),(0); -- 1 indica que los cursos están disponibles (true)
INSERT INTO DisponibilidadHorario (Disponibilidad)
VALUES (1),(0); -- 1 indica que los cursos están disponibles (true)
INSERT INTO Dia (NombreDia)
VALUES
('Lunes'),('Martes'),('Miercoles'),('Jueves'),('Viernes'),('Sabado'),('Domingo'); -- 1
indica que los cursos están disponibles (true)

--Insertar Horario
INSERT INTO Horario (Descripcion, HoraInicio, HoraFin, IdDia, IdDisponibilidadHorario)
VALUES
('Primera hora: de 8 a 9', '08:00:00', '09:00:00', 2, 1),
('Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 2, 1),
('Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 2, 1),
('Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 2, 1),
('Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 2, 1),
('Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 2, 1),
('Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 2, 1),
('Octava hora: de 15 a 16', '15:00:00', '16:00:00', 2, 1),
('Novena hora: de 16 a 17', '16:00:00', '17:00:00', 2, 1),
('Décima hora: de 17 a 18', '17:00:00', '18:00:00', 2, 1),
('Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 2, 1),
('Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 2, 1),
('Primera hora: de 8 a 9', '08:00:00', '09:00:00', 3, 1),
('Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 3, 1),
('Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 3, 1),
('Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 3, 1),
('Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 3, 1),
('Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 3, 1),
('Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 3, 1),
('Octava hora: de 15 a 16', '15:00:00', '16:00:00', 3, 1),
('Novena hora: de 16 a 17', '16:00:00', '17:00:00', 3, 1),
('Décima hora: de 17 a 18', '17:00:00', '18:00:00', 3, 1),
('Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 3, 1),
('Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 3, 1),
('Primera hora: de 8 a 9', '08:00:00', '09:00:00', 4, 1),
('Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 4, 1),
('Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 4, 1),
('Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 4, 1),
('Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 4, 1),
('Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 4, 1),
('Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 4, 1),
('Octava hora: de 15 a 16', '15:00:00', '16:00:00', 4, 1),
('Novena hora: de 16 a 17', '16:00:00', '17:00:00', 4, 1),
('Décima hora: de 17 a 18', '17:00:00', '18:00:00', 4, 1),
('Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 4, 1),
('Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 4, 1),
('Primera hora: de 8 a 9', '08:00:00', '09:00:00', 5, 1),
('Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 5, 1),
('Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 5, 1),
('Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 5, 1),
('Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 5, 1),
('Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 5, 1),
('Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 5, 1),
('Octava hora: de 15 a 16', '15:00:00', '16:00:00', 5, 1),
('Novena hora: de 16 a 17', '16:00:00', '17:00:00', 5, 1),
('Décima hora: de 17 a 18', '17:00:00', '18:00:00', 5, 1),
('Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 5, 1),
('Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 5, 1),
('Primera hora: de 8 a 9', '08:00:00', '09:00:00', 6, 1),
('Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 6, 1),
('Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 6, 1),
('Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 6, 1),

```

```
( 'Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 6, 1),
( 'Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 6, 1),
( 'Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 6, 1),
( 'Octava hora: de 15 a 16', '15:00:00', '16:00:00', 6, 1),
( 'Novena hora: de 16 a 17', '16:00:00', '17:00:00', 6, 1),
( 'Décima hora: de 17 a 18', '17:00:00', '18:00:00', 6, 1),
( 'Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 6, 1),
( 'Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 6, 1),
( 'Primera hora: de 8 a 9', '08:00:00', '09:00:00', 7, 1),
( 'Segunda hora: de 9 a 10', '09:00:00', '10:00:00', 7, 1),
( 'Tercera hora: de 10 a 11', '10:00:00', '11:00:00', 7, 1),
( 'Cuarta hora: de 11 a 12', '11:00:00', '12:00:00', 7, 1),
( 'Quinta hora: de 12 a 13', '12:00:00', '13:00:00', 7, 1),
( 'Sexta hora: de 13 a 14', '13:00:00', '14:00:00', 7, 1),
( 'Séptima hora: de 14 a 15', '14:00:00', '15:00:00', 7, 1),
( 'Octava hora: de 15 a 16', '15:00:00', '16:00:00', 7, 1),
( 'Novena hora: de 16 a 17', '16:00:00', '17:00:00', 7, 1),
( 'Décima hora: de 17 a 18', '17:00:00', '18:00:00', 7, 1),
( 'Undécima hora: de 18 a 19', '18:00:00', '19:00:00', 7, 1),
( 'Duodécima hora: de 19 a 20', '19:00:00', '20:00:00', 7, 1);
```

```
INSERT INTO Salones (CodigoSalon, AforoSalon, IdDisponibilidad)
VALUES
```

```
( 'A101', 5, 1),
( 'A102', 5, 1),
( 'A103', 5, 1),
( 'A104', 5, 1),
( 'A105', 5, 2),
( 'A106', 5, 2),
( 'A107', 5, 2),
( 'A108', 5, 1),
( 'A109', 5, 1),
( 'A110', 10, 1),
( 'A111', 10, 1),
( 'A112', 10, 1),
( 'A113', 10, 1),
( 'A114', 10, 1),
( 'A115', 10, 1),
( 'A116', 15, 2),
( 'A117', 15, 2),
( 'A118', 15, 2),
( 'A119', 15, 2),
( 'A120', 15, 2);
```

```
INSERT INTO Estudiantes (Nombres, ApellidoPaterno, ApellidoMaterno,
TelefonoEstudiante, FechaNacimiento, Email, Direccion)
VALUES
```

```
( 'Luis', 'Gómez', 'Rojas', '959876543', '2001-05-14', 'luis.gomez@gmail.com', 'Calle
Santa Catalina 123, Arequipa'),
( 'María', 'Pérez', 'Quispe', '951234567', '2002-03-22', 'maria.perez@gmail.com',
'Calle San Lázaro 45, Arequipa'),
( 'José', 'Fernández', 'Chávez', '952345678', '2003-11-10', 'jose.fernandez@gmail.com',
'Av. Independencia 567, Arequipa'),
( 'Ana', 'Lopez', 'Mamani', '953456789', '2004-01-08', 'ana.lopez@gmail.com', 'Calle
Mercaderes 789, Arequipa'),
( 'Carlos', 'Vega', 'Huamán', '954567890', '2005-06-17', 'carlos.vega@gmail.com', 'Av.
Ejército 321, Arequipa'),
( 'Sofía', 'Salas', 'Vargas', '955678901', '2001-09-23', 'sofia.salas@gmail.com',
'Calle Puente Grau 654, Arequipa'),
( 'Jorge', 'Álvarez', 'Huarachi', '956789012', '2002-04-15', 'jorge.alvarez@gmail.com',
'Av. Mariscal Castilla 456, Arequipa'),
( 'Valeria', 'Medina', 'Paredes', '957890123', '2003-08-30',
'valeria.medina@gmail.com', 'Calle Bolívar 789, Arequipa'),
( 'Ricardo', 'Zúñiga', 'Ramos', '958901234', '2000-12-01', 'ricardo.zuniga@gmail.com',
'Av. Dolores 234, Arequipa'),
```



```

('Lucía', 'Castro', 'Flores', '959012345', '2001-02-14', 'lucia.castro@gmail.com',
'Calle Ejército 876, Arequipa'),
('Pedro', 'Sánchez', 'Torres', '951345678', '2003-05-12', 'pedro.sanchez@gmail.com',
'Calle Colón 987, Arequipa'),
('Rosa', 'Ramos', 'Apaza', '952456789', '2004-07-20', 'rosa.ramos@gmail.com', 'Calle
Villalba 345, Arequipa'),
('David', 'Quispe', 'Quispe', '953567890', '2002-11-11', 'david.quispe@gmail.com',
'Av. Parra 123, Arequipa'),
('Marta', 'Huaman', 'Cruz', '954678901', '2001-03-09', 'marta.huaman@gmail.com',
'Calle Misti 543, Arequipa'),
('Fernando', 'García', 'Lopez', '955789012', '2005-10-05',
'fernando.garcia@gmail.com', 'Av. Kennedy 987, Arequipa'),
('Isabel', 'Vargas', 'Montoya', '956890123', '2004-12-22', 'isabel.vargas@gmail.com',
'Calle Jerusalén 234, Arequipa'),
('Gabriel', 'Mamani', 'Cárdenas', '957901234', '2000-08-14',
'gabriel.mamani@gmail.com', 'Av. Socabaya 456, Arequipa'),
('Sandra', 'Apaza', 'Huayhua', '958012345', '2003-02-18', 'sandra.apaza@gmail.com',
'Calle Siglo XX 678, Arequipa'),
('Felipe', 'Chávez', 'Arias', '959123456', '2001-06-21', 'felipe.chavez@gmail.com',
'Calle Paucarpata 789, Arequipa'),
('Laura', 'Zeballos', 'Ortega', '951234578', '2002-09-10', 'laura.zeballos@gmail.com',
'Av. Grau 654, Arequipa'),
('Roberto', 'Delgado', 'Quispe', '952345789', '2003-11-30',
'roberto.delgado@gmail.com', 'Calle Álvarez Thomas 321, Arequipa'),
('Karla', 'Nina', 'Fernández', '953456780', '2005-01-01', 'karla.nina@gmail.com',
'Calle Melgar 345, Arequipa'),
('Manuel', 'Choque', 'Salas', '954567891', '2004-07-07', 'manuel.choque@gmail.com',
'Av. Salaverry 567, Arequipa'),
('Paola', 'Quispe', 'Vega', '955678912', '2001-02-24', 'paola.quispe@gmail.com',
'Calle Arancota 987, Arequipa'),
('Diana', 'Cáceres', 'Mendoza', '956789023', '2002-12-17', 'diana.caceres@gmail.com',
'Av. Bustamante 876, Arequipa'),
('Alfredo', 'Huayhua', 'Chambi', '957890134', '2003-03-03',
'alfredo.huayhua@gmail.com', 'Calle San Agustín 543, Arequipa'),
('Carmen', 'Flores', 'Arce', '958901245', '2000-06-11', 'carmen.flores@gmail.com',
'Av. Hunter 321, Arequipa'),
('Francisco', 'Rojas', 'Tito', '959012356', '2001-09-19', 'francisco.rojas@gmail.com',
'Calle Bolívar 234, Arequipa'),
('Adriana', 'Cruz', 'Vilca', '951345679', '2002-10-28', 'adriana.cruz@gmail.com', 'Av.
Umacollo 789, Arequipa'),
('Esteban', 'Mendoza', 'Acuña', '952456790', '2004-04-15',
'esteban.mendoza@gmail.com', 'Calle Loreto 456, Arequipa');

```

```

INSERT INTO Orden (FechaOrden, IdEstudiante, IdCurso, IdSalon, IdHorario, Estado)
VALUES

```

```

('2024-11-01', 1, 10, 1, 1, 1),
('2024-11-02', 2, 12, 2, 2, 1),
('2024-11-03', 3, 3, 3, 3, 0),
('2024-11-04', 4, 4, 4, 13, 1),
('2024-11-05', 5, 5, 5, 10, 1),
('2024-11-06', 6, 6, 6, 6, 0),
('2024-11-07', 7, 7, 7, 20, 1),
('2024-11-08', 8, 8, 8, 8, 1),
('2024-11-09', 9, 9, 9, 9, 0),
('2024-11-10', 10, 10, 10, 10, 1);

```

```

INSERT INTO Solicita (IdEstudiante, IdCurso)
VALUES

```

```

(1, 10),
(2, 12),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),

```

```

(9, 9),
(10, 10);

INSERT INTO Imparte (IdProfesor, IdCurso)
VALUES
(1, 10),
(2, 12),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);

```

## 7. CONSULTAS A LAS BASES DE DATOS

### 7.1. *Querys de consultas*

```

use BD_Mathmatic_01
GO

--1. Hallar el máximo (Máximo ID del Estudiante)
SELECT MAX(IdEstudiante) AS MaximoEstudiante
FROM Estudiantes;

--2. Hallar el promedio (Promedio de IdCurso en la tabla Orden)

SELECT AVG(IdCurso) AS PromedioIdCurso
FROM Orden;

--3.Cursos cuyo nombre contiene la palabra 'Matemática'
SELECT IdCurso, NombreCurso
FROM Cursos
WHERE NombreCurso LIKE '%Matem%';

--4.Estudiantes que solicitan un curso en particular (por ejemplo, IdCurso = 3)
SELECT E.IdEstudiante, E.Nombres, S.IdCurso
FROM Solicita S
INNER JOIN Estudiantes E ON S.IdEstudiante = E.IdEstudiante
WHERE S.IdCurso = 3;

--5.Profesores que dictan más de un curso
SELECT I.IdProfesor, P.Nombres AS NombreProfesor, COUNT(I.IdCurso) AS CursosImpartidos
FROM Imparte I
INNER JOIN Profesores P ON I.IdProfesor = P.IdProfesor
GROUP BY I.IdProfesor, P.Nombres
HAVING COUNT(I.IdCurso) > 5;

--6. Consulta con INNER JOIN (Estudiantes y los cursos que solicitan)
SELECT
    E.IdEstudiante,
    E.Nombres AS NombreEstudiante,
    C.IdCurso,
    C.NombreCurso
FROM Solicita S
INNER JOIN Estudiantes E ON S.IdEstudiante = E.IdEstudiante
INNER JOIN Cursos C ON S.IdCurso = C.IdCurso;

--7. Consulta con BETWEEN (Órdenes realizadas en un rango de fechas
SELECT *

```

```

FROM Orden
WHERE FechaOrden BETWEEN '2024-11-01' AND '2024-11-05';

--8. Consulta con WHERE (Cursos impartidos por un profesor específico)
SELECT
    P.Nombres AS NombreProfesor,
    C.NombreCurso
FROM Imparte I
INNER JOIN Profesores P ON I.IdProfesor = P.IdProfesor
INNER JOIN Cursos C ON I.IdCurso = C.IdCurso
WHERE P.Nombres = 'Juan';

--9. Consulta con GROUP BY (Cantidad de órdenes por estado)
SELECT
    Estado,
    COUNT(*) AS CantidadOrdenes
FROM Orden
GROUP BY Estado;

--10. Consulta con LIKE (Buscar estudiantes con nombres que comiencen con 'A')
SELECT *
FROM Estudiantes
WHERE Nombres LIKE 'A%';

```

## 7.2. Resultados de Querys

Results Messages

	MaximoEstudiante							
1	30							
	PromedioIdCurso							
1	7							
	IdCurso	NombreCurso						
1	3	Matematica Basica						
2	11	Razonamiento Matematico						
	IdEstudiante	Nombres	IdCurso					
1	3	José	3					
	IdProfesor	NombreProfesor	CursosImpartidos					
	IdEstudiante	NombreEstudiante	IdCurso	NombreCurso				
1	1	Luis	10	Algebra				
2	2	María	12	Analisis de Logaritmos				
3	3	José	3	Matematica Basica				
4	4	Ana	4	Biologia				
5	5	Carlos	5	Quimica				
6	6	Sofía	6	Fisica				
7	7	Jorge	7	Calculo				
8	8	Valeria	8	Termodinamica				
	IdOrden	FechaOrden	IdEstudiante	IdCurso	IdSalon	IdHorario	Estado	
1	2	2024-11-01	1	10	1	1	1	
2	3	2024-11-02	2	12	2	2	1	
3	4	2024-11-03	3	3	3	3	0	
4	5	2024-11-04	4	4	4	13	1	
5	6	2024-11-05	5	5	5	10	1	
	NombreProfesor	NombreCurso						
1	Juan	Algebra						
	Estado	CantidadOrdenes						
1	0	3						
2	1	7						
	IdEstudiante	Nombres	ApellidoPaterno	ApellidoMaterno	TelefonoEstudiante	FechaNacimiento	Email	Direccion
1	4	Ana	Lopez	Mamani	953456789	2004-01-08	ana.lopez@gmail.com	Calle Mercaderes 789, Arequipa
2	26	Alfredo	Huayhua	Chambi	957890134	2003-03-03	alfredo.huayhua@gmail.com	Calle San Agustín 543, Arequipa
3	29	Adriana	Cruz	Vilca	951345679	2002-10-28	adriana.cruz@gmail.com	Av. Umacollo 789, Arequipa

## 8. CREACION DE VISTAS

### 8.1. Creacion de Vistas

```
use BD_Mathmatic_01
GO
--1. Vista de Estudiantes con sus Órdenes Activas
CREATE VIEW Vista_EstudiantesOrdenesActivas AS
SELECT
    E.IdEstudiante,
    E.NombreS AS NombreEstudiante,
    C.NombreCurso,
    O.FechaOrden
FROM Estudiantes E
INNER JOIN Orden O ON E.IdEstudiante = O.IdEstudiante
INNER JOIN Cursos C ON O.IdCurso = C.IdCurso
WHERE O.Estado = 1;

--2. Vista de Profesores con los Cursos que Imparten
CREATE VIEW Vista_ProfesoresCursos AS
SELECT
    P.IdProfesor,
    P.NombreS AS NombreProfesor,
    C.IdCurso,
    C.NombreCurso,
    C.Precio
FROM Profesores P
INNER JOIN Imparte I ON P.IdProfesor = I.IdProfesor
INNER JOIN Cursos C ON I.IdCurso = C.IdCurso;

--3. Vista de Salones Asignados a Órdenes
CREATE VIEW Vista_SalonesAsignados AS
SELECT
    S.IdSalon,
    S.CodigoSalon,
    C.NombreCurso,
    E.Nombres AS NombreEstudiante,
    O.FechaOrden
FROM Salones S
INNER JOIN Orden O ON S.IdSalon = O.IdSalon
INNER JOIN Cursos C ON O.IdCurso = C.IdCurso
INNER JOIN Estudiantes E ON O.IdEstudiante = E.IdEstudiante;

--4. Vista de Cursos Solicitados por Estudiantes
CREATE VIEW Vista_CursosSolicitados AS
SELECT
    E.IdEstudiante,
    E.Nombres AS NombreEstudiante,
    C.IdCurso,
    C.NombreCurso,
    C.Precio
FROM Estudiantes E
INNER JOIN Solicita S ON E.IdEstudiante = S.IdEstudiante
INNER JOIN Cursos C ON S.IdCurso = C.IdCurso;

--5. Vista de Horarios de Cursos con Órdenes Activas
CREATE VIEW Vista_HorariosCursosActivos AS
SELECT
    H.IdHorario,
    H.HoraInicio,
    H.HoraFin,
```

```

        C.NombreCurso,
        E.Nombres AS NombreEstudiante,
        O.FechaOrden
FROM Horario H
INNER JOIN Orden O ON H.IdHorario = O.IdHorario
INNER JOIN Cursos C ON O.IdCurso = C.IdCurso
INNER JOIN Estudiantes E ON O.IdEstudiante = E.IdEstudiante
        WHERE O.Estado = 1;

```

## 8.2. Ejecucion de Vistas

```
SELECT * FROM Vista_EstudiantesOrdenesActivas;
```

	IdEstudiante	NombreEstudiante	NombreCurso	FechaOrden
1	1	Luis	Algebra	2024-11-01
2	2	María	Analisis de Logaritmos	2024-11-02
3	4	Ana	Biología	2024-11-04
4	5	Carlos	Química	2024-11-05
5	7	Jorge	Calculo	2024-11-07
6	8	Valeria	Termodinamica	2024-11-08
7	10	Lucía	Algebra	2024-11-10

```
SELECT * FROM Vista_ProfesoresCursos;
```

	IdProfesor	NombreProfesor	IdCurso	NombreCurso	Precio
1	1	Juan	10	Algebra	35.00
2	2	Maria	12	Analisis de Logaritmos	25.00
3	3	Carlos	3	Matematica Basica	20.00
4	4	Ana	4	Biología	25.00
5	5	Luis	5	Química	30.00
6	6	Sofia	6	Fisica	35.00
7	7	Miguel	7	Calculo	20.00
8	8	Laura	8	Termodinamica	25.00
9	9	Daniel	9	Estatica	30.00
10	10	Lucia	10	Algebra	35.00

```
SELECT * FROM Vista_SalonesAsignados;
```

	IdSalon	CodigoSalon	NombreCurso	NombreEstudiante	FechaOrden
1	1	A101	Algebra	Luis	2024-11-01
2	2	A102	Analisis de Logaritmos	María	2024-11-02
3	3	A103	Matematica Basica	José	2024-11-03
4	4	A104	Biología	Ana	2024-11-04
5	5	A105	Química	Carlos	2024-11-05
6	6	A106	Fisica	Sofía	2024-11-06
7	7	A107	Calculo	Jorge	2024-11-07
8	8	A108	Termodinamica	Valeria	2024-11-08
9	9	A109	Estatica	Ricardo	2024-11-09
10	10	A110	Algebra	Lucía	2024-11-10

```
SELECT * FROM Vista_CursosSolicitados;
```

Results Messages					
	IdEstudiante	NombreEstudiante	IdCurso	NombreCurso	Precio
1	1	Luis	10	Algebra	35.00
2	2	María	12	Análisis de Logaritmos	25.00
3	3	José	3	Matemática Básica	20.00
4	4	Ana	4	Biología	25.00
5	5	Carlos	5	Química	30.00
6	6	Sofía	6	Física	35.00
7	7	Jorge	7	Cálculo	20.00
8	8	Valeria	8	Termodinámica	25.00
9	9	Ricardo	9	Estática	30.00
10	10	Lucía	10	Algebra	35.00

```
SELECT * FROM Vista_HorariosCursosActivos;
```

	IdHorario	Horainicio	HoraFin	NombreCurso	NombreEstudiante	FechaOrden
1	1	08:00:00.0000000	09:00:00.0000000	Algebra	Luis	2024-11-01
2	2	09:00:00.0000000	10:00:00.0000000	Análisis de Logaritmos	María	2024-11-02
3	13	08:00:00.0000000	09:00:00.0000000	Biología	Ana	2024-11-04
4	10	17:00:00.0000000	18:00:00.0000000	Química	Carlos	2024-11-05
5	20	15:00:00.0000000	16:00:00.0000000	Cálculo	Jorge	2024-11-07
6	8	15:00:00.0000000	16:00:00.0000000	Termodinámica	Valeria	2024-11-08
7	10	17:00:00.0000000	18:00:00.0000000	Algebra	Lucía	2024-11-10

## 9. STORE PROCEDURE

### 9.1. Creacion de Store Procedure

**Procedimiento almacenado para obtener estudiantes con órdenes activas:**

```
use BD_Mathmatic_01
go

CREATE PROCEDURE ObtenerEstudiantesConOrdenesActivas
    @NombreCurso NVARCHAR(100)
AS
BEGIN
    SELECT E.NombreEstudiante, O.FechaOrden, C.NombreCurso
    FROM Estudiantes E
    INNER JOIN Orden O ON E.IdEstudiante = O.IdEstudiante
    INNER JOIN Cursos C ON O.IdCurso = C.IdCurso
    WHERE C.NombreCurso = @NombreCurso AND O.Estado = 1;
END;
```

**Procedimiento almacenado para copiar a otra base de datos las tablas Estudiantes, Orden, Profesores y Cursos:**

```
CREATE PROCEDURE BD_Mathmatic_01_Replica
AS
BEGIN
    -- Insertar datos en la tabla Estudiantes de respaldo
    INSERT INTO BD_Mathmatic_01_BK.dbo.Estudiantes
    SELECT * FROM BD_Mathmatic_01.dbo.Estudiantes;

    -- Insertar datos en la tabla Profesores de respaldo
    INSERT INTO BD_Mathmatic_01_BK.dbo.Profesores
    SELECT * FROM BD_Mathmatic_01.dbo.Profesores;
```

```

-- Insertar datos en la tabla Cursos de respaldo
INSERT INTO BD_Mathmatic_01_BK.dbo.Cursos
SELECT * FROM BD_Mathmatic_01.dbo.Cursos;

-- Insertar datos en la tabla Orden de respaldo
INSERT INTO BD_Mathmatic_01_BK.dbo.Orden
SELECT * FROM BD_Mathmatic_01.dbo.Orden;

PRINT 'Datos copiados exitosamente a la base de datos de respaldo.';
END;
GO

```

## 9.2. *Ejecucion de Store Procedure*

```

EXEC ObtenerEstudiantesConOrdenesActivas @NombreCurso = 'Matemáticas';

EXEC BD_Mathmatic_01_Replica;
GO

```

## 10. TRIGGERS

### 10.1. Creación de Trigger

```
CREATE TRIGGER trg_ValFecha_Nacimiento
ON Estudiantes
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TRY_CAST(FechaNacimiento AS DATE) IS NULL
    )
    BEGIN
        ROLLBACK TRANSACTION;
        THROW 50002, 'La Fecha de Nacimiento debe ser una fecha válida (formato
AAAA-MM-DD).', 1;
    END
END;
GO

-----

Use BD_Mathmatic_01
go

CREATE TRIGGER trg_Valtlf_Estudiant
ON Estudiantes
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TelefonoEstudiante NOT LIKE '[0-9]%'
    )
    BEGIN
        ROLLBACK TRANSACTION;
        THROW 50001, 'El teléfono del estudiante debe contener solo números.', 1;
    END
END;
GO

-----

CREATE TRIGGER trg_ValidarPrecioCurso
ON Cursos
AFTER INSERT, UPDATE
AS
BEGIN
    -- Verifica si hay algún registro que no cumpla con el formato numérico del
precio
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE Precio IS NULL OR Precio < 0
    )
    BEGIN
        ROLLBACK TRANSACTION;
        THROW 50003, 'El campo Precio en la tabla Cursos debe ser un valor numérico
mayor o igual a 0.', 1;
    END
END;
```



```
END
END;
GO
```

## 10.2. Probar de Trigger

```
INSERT INTO Cursos (NombreCurso, Precio, IdDisponibilidadCurso, IdProfesor)
VALUES ('BD1', 150.50, 1, 1);
GO
```

```
INSERT INTO Cursos (NombreCurso, Precio, IdDisponibilidadCurso, IdProfesor)
VALUES ('Física Avanzada', -50, 1, 1);
GO
```

# 11. USUARIOS Y ROLES

## 11.1. Creación de usuarios

```
USE BD_Mathmatic_01;
GO

-- Crear login para user_administrator con una contraseña
CREATE LOGIN user_administratorLogin
WITH PASSWORD = 'tevogliobene1.';
GO

-- Crear login para user_invitado con una contraseña
CREATE LOGIN user_invitadoLogin
WITH PASSWORD = 'tevoglio1.';
GO
```

## 11.2. Asignación de roles

```
\
USE BD_Mathmatic_01;
GO

-- Crear usuario user_administrator con privilegios de lectura y escritura
CREATE USER user_administrator FOR LOGIN user_administratorLogin;
GO

-- Asignar roles de base de datos (db_datareader y db_datawriter)
ALTER ROLE db_datareader ADD MEMBER user_administrator;
ALTER ROLE db_datawriter ADD MEMBER user_administrator;
GO

-- Crear usuario user_invitado con privilegios solo de lectura
CREATE USER user_invitado FOR LOGIN user_invitadoLogin;
GO

-- Asignar solo el rol de lectura (db_datareader)
ALTER ROLE db_datareader ADD MEMBER user_invitado;
```

## 12. CONEXION CON PYTHON

### 11.1. Creación de la conexión

```
import pyodbc

HOST="LAPTOP-85T0Q0UQ"
USER="user_python"
PASSWORD="user_python"
DATABASE="BD_Mathmatic_01"

def conexionDB():

    try:
        conexion = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL Server};SERVER='
+ HOST + ';DATABASE='+ DATABASE + ';UID=' + USER + ';PWD=' + PASSWORD +
";TrustServerCertificate=YES")
        return conexion
    except Exception as e:
        print("Problema de conexion con la base de datos" + e)
    return None

def ConsultaMathmatic():
    conn = conexionDB()
    cur = conn.cursor()

    sql = "select*from  dbo.Estudiantes"

    cur.execute(sql)

    filas = cur.fetchall()

    cur.close()
    conn.close()

    return filas
```

## 11.2. Query de Consulta hacia la Base de Datos

```
import pyodbcconn as C

Seguir = True

while (Seguir):

    print("\n")
    print("|-----Bienvenido a sistema de Mathematic-----|")
    print("|")
    print("|-----<MENU>-----|")
    print("|")
    print("| [1] Lista de todos los cursos |")
    print("| [2] Finalizar la consulta |")
    print("|")
    print("|-----|")
    print("\n")
    Seguir2 = True
    while(Seguir2):
        try:
            opcion = int(input("Ingrese la opcion a realizar--->"))
            if opcion>3 or opcion <1:
                print("La opcion ingresada es incorrecta")
            else:
                Seguir2 = False
        except:
            print("Ingrese valor numerico")

    if opcion == 1:

        Consultas = C.ConsultaMathmatic()

        print(Consultas)

        for estudiantes in Consultas:
            print(estudiantes)

    elif opcion == 2:
        Seguir = False
```