

**Centro CTC  
CEI**

**Obligatorio  
Base de datos 2  
Primera entrega**

Alan Restelli,  
Rodrigo Ciosek

Profesor: Pablo Silva

**2025**

# Índice

Introduccion .....	3
Análisis del Problema .....	4
Diccionario de Datos .....	7
Diagrama Relacional .....	17

## Introducción

*Este proyecto tiene como objetivo crear una base de datos relacional para una plataforma de e-learning. La base de datos debe almacenar información sobre docentes, cursos, módulos, clases y estudiantes, estableciendo las relaciones necesarias entre estos elementos. Cada curso estará vinculado a un docente, contará con varios módulos y clases, y cada estudiante podrá inscribirse en los cursos disponibles.*

*Además, la base de datos incluirá una funcionalidad clave: una bitácora de auditoría. Esta bitácora registrará todas las operaciones relevantes, como las inserciones, actualizaciones y eliminaciones de registros, para asegurar que se pueda hacer un seguimiento de las acciones realizadas en el sistema*

## **Análisis**

### **METODO DE EVALUACION**

*El método de evaluación se basa en una estructura progresiva que garantiza que los estudiantes completen todas las etapas de aprendizaje antes de recibir una calificación final.*

*En primera instancia, cada módulo de un curso se compone de varias clases que son obligatorias, pero no serán calificadas.*

*Una vez completadas todas las clases de un módulo, el estudiante podrá acceder al examen que determinará su calificación en esa unidad específica. La nota obtenida en el examen será el único criterio para la evaluación del módulo, estableciendo así un sistema en el que el desempeño de la prueba refleja directamente el nivel de comprensión del contenido brindado en el módulo, necesario para continuar con el resto de las unidades. De esta manera, se garantiza que la evaluación sea objetiva y basada en la capacidad del estudiante para aplicar los conocimientos adquiridos.*

*Finalmente, la nota del curso se calcula como el promedio de las calificaciones obtenidas en los módulos que lo componen. Asegurando que el resultado final represente su progreso acumulado a lo largo de todas las unidades del curso. Para que el estudiante apruebe el curso, deberá alcanzar un promedio mínimo establecido.*

## REGISTRO DE EVENTOS

*Una bitácora es una tabla especial dentro de una base de datos que se utiliza para registrar y llevar el control de las acciones que se realizan sobre otras tablas del sistema. Su función principal es proporcionar un historial detallado de operaciones como inserciones, actualizaciones, eliminaciones o accesos (por ejemplo, inicio de sesión).*

*Por lo tanto para este proyecto, se implementara una tabla de bitácora por cada entidad principal, donde cada una de ellas contendra las siguientes características:*

- Un identificador único.*
- La fecha y hora de la acción.*
- El usuario que realizo la sentencia.*
- El tipo de acción realizada (LOGIN, INSERT, UPDATE, DELETE).*
- La sentencia SQL ejecutada.*

*Lo que buscamos con esta bitácora es llevar un control preciso y confiable de las operaciones realizadas sobre el sistema, permitiendo conocer en todo momento qué cambios se hicieron, quién los hizo, cuándo y cómo.*

*Esto no solo facilita el mantenimiento y la seguridad, sino que también aporta una capa de resolución de errores, ya que nos permite identificar rápidamente las acciones que causaron problemas o inconsistencias, facilitando su corrección de manera más eficiente.*

## LIMPIEZA DE DATOS

*Para garantizar un funcionamiento organizado del sistema y sin inconsistencias, se debe contemplar las opciones de delete para los datos que ya no se encuentran en uso. La idea es no eliminar todo, sino conservar la información clave y eliminar los datos innecesarios.*

*Los estudiantes y cursos se mantendrán de forma permanente en el sistema, pero se les asignará un campo de "estado" que indicará si están activos o no. Esto permitirá conocer qué estudiantes han participado en qué cursos, incluso si el curso ya no está disponible o si el estudiante ha finalizado su formación. De esta manera, el historial académico estará siempre disponible y no se perderá información fundamental, como las certificaciones.*

*Por otro lado, elementos como clases, módulos y docentes podrán ser eliminados, ya que, al mantenerse inactivos, estaríamos acumulando datos que ya no aportan valor al sistema. Por esta razón, solo los estudiantes y cursos se conservarán como registros permanentes.*

## Diccionario de Datos

Tabla: Docentes

Atributo	Tipo de dato	Restricciones	Descripcion
DocenteID	INT	PRIMARY KEY	Identificador único del docente
Nombre	VARCHAR(50)	NOT NULL	Nombre del docente
Apellido	VARCHAR(50)	NOT NULL	Apellido del docente
Correo	VARCHAR(50)	UNIQUE, NOT NULL	Correo electrónico del docente
FechaRegistro	DATE	NOT NULL	Fecha de registro del docente

*El sistema permite registrar y gestionar la información de los docentes de forma individual, el campo docenteid es la clave primaria que garantiza que cada docente esté identificado de forma única, mientras que el correo se utiliza para establecer la identidad del docente en el sistema, la fecha\_registro permite llevar un historial sobre cuándo se registró cada docente.*

*Además, un docente puede estar asociado a muchos cursos dentro de la plataforma. Esta relación uno a muchos permite que un mismo docente sea responsable de varios cursos.*

Tabla: Cursos

Atributo	Tipo de dato	Restricciones	Descripción
CursoID	INT	PRIMARY KEY	Identificador único del curso
DocenteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Docentes
Título	VARCHAR(50)	NOT NULL	Título del curso
Descripción	VARCHAR(50)	NOT NULL	Descripción del curso
Duración	INT	NOT NULL	Duración del curso en días
Estado	BIT	NOT NULL, CHECK(Estado IN(0,1))	Estado del curso (activo/ inactivo)
NotaMinima	DECIMAL(5,2)	NOT NULL	Nota mínima requerida para aprobar

La tabla *Cursos* tiene como objetivo almacenar la información relacionada con los cursos. Esta tabla gestiona los detalles de cada curso, como el docente asignado, el título, la descripción, la duración, el estado y la nota mínima de aprobación.

Además, la tabla permite vincular cada curso con el docente encargado de dictarlo

- **Docentes- Cursos:** Un curso está dictado por un único profesor, lo que establece una relación uno a muchos entre *Docentes* y *Cursos*. El *docenteid* en la tabla *Cursos* es una clave foránea que hace referencia al *docenteid* en la tabla *Docentes*.



Tabla: Modulos

Atributo	Tipo de dato	Restricciones	Descripcion
ModuloID	INT	PRIMARY KEY	Identificador único del módulo
Cursoid	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
Titulo	VARCHAR(50)	NOT NULL	Título del módulo
Descripcion	VARCHAR(50)	NOT NULL	Descripción del módulo
Orden	INT	NOT NULL	Posición del módulo dentro del curso
NotaMaxima	DECIMAL(5,2)	NOT NULL, DEFAULT 0	Nota máxima posible del módulo

La tabla Módulos tiene como objetivo almacenar la información relacionada con los módulos que conforman los cursos . Cada módulo está asociado a un curso específico y contiene detalles como su título, descripción, el orden en que debe ser presentado dentro del curso, y la nota mínima requerida para aprobarlo.

Cada módulo se compone de una o varias clases.

- **Módulos – Cursos:** Un módulo pertenece a un único curso, lo que establece una relación uno a muchos entre Cursos y Módulos. El cursoid en la tabla Módulos es una clave foránea que hace referencia al cursoid en la tabla Cursos.

Tabla: Clases

Atributo	Tipo de dato	Restricciones	Descripcion
ClaseID	INT	PRIMARY KEY	Identificador único de la clase
ModuloID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Modulos
Título	VARCHAR(50)	NOT NULL	Título de la clase
Descripcion	VARCHAR(50)	NOT NULL	Descripción de la clase

// cpz poner orden de clase para saber el orden dentro del modulo

*La tabla Clases almacena información sobre las clases individuales que forman parte de los módulos en los cursos . Cada clase está vinculada a un módulo específico y contiene detalles como el título, la descripción, y el orden en que se presenta dentro del módulo. De esta manera los modulos se organizan de forma clara y con un temario progresivo.*

- **Módulos - Clases:** *Cada clase pertenece a un único módulo. Por lo tanto, la relación entre Módulos y Clases es de uno a muchos. Es decir, un módulo puede tener varias clases, pero cada clase está asociada a un único módulo. Esta relación se establece a través de la clave foránea Moduloid en la tabla Clases, que hace referencia al Moduloid de la tabla Módulos*

Tabla: Estudiantes

Atributo	Tipo de dato	Restricciones	Descripcion
Estudiantes	INT	PRIMARY KEY	Identificador único del estudiante
Nombre	VARCHAR(50)	NOT NULL	Nombre del estudiante
Apellido	VARCHAR(50)	NOT NULL	Apellido del estudiante
Correo	VARCHAR(50)	UNIQUE, NOT NULL	Correo electrónico del estudiante
FechaRegistro	DATE	NOT NULL, DEFAULT GETDATE()	Fecha de registro del estudiante
Activo	BIT	CHECK(Activo IN(0,1)), NOT NULL, DEFAULT 1	Estado del estudiante (activo/ inactivo)

La tabla *Estudiantes* almacena la información personal y de estado de los usuarios que participan en los cursos .

- ***Estudiantes\_Cursos (Inscripciones):***

*Cada estudiante puede inscribirse en varios cursos, y cada curso puede tener múltiples estudiantes. Por lo tanto, la relación entre Estudiantes y Cursos es de muchos a muchos. Esta relación se implementa mediante una tabla intermedia que contiene las claves foráneas estudianteid y cursoid, las cuales hacen referencia a las tablas Estudiantes y Cursos respectivamente. También contará con un campo tipo DATE el cual almacenará la fecha en la cual se haya realizado la inscripción.*

- ***Estudiantes\_Módulos :***

*Cada estudiante puede realizar varios módulos, y cada módulo puede ser completado por muchos estudiantes. Por lo tanto, la relación entre*

*Estudiantes y Módulos también es de muchos a muchos. Esta relación se gestiona a través de una tabla intermedia que contiene moduloid, cursoid, estudianteid y notaexamen. Las claves foráneas establecen las conexiones necesarias con las tablas Módulos, Cursos y Estudiantes. Esto permite almacenar la calificación obtenida por cada estudiante en cada módulo específico.*

- **Estudiantes\_Clases :**

*Cada estudiante puede ver muchas clases, y cada clase puede ser vista por muchos estudiantes. Por lo tanto, la relación entre Estudiantes y Clases es de muchos a muchos. Se utiliza una tabla intermedia que incluye claseid, cursoid, estudianteid y un campo visto (con valores 0 o 1). Las claves foráneas conectan esta tabla con Estudiantes, Clases y Cursos. Esta relación permite registrar si un estudiante ya visualizó una clase determinada, y así llevar el control del avance dentro del curso*

Tabla: EstudiantesModulos

Atributo	Tipo de dato	Restricciones	Descripcion
Cursoid	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
NotaExamen	DECIMAL(5,2)		Nota del examen del módulo

Tabla: EstudiantesClases

Atributo	Tipo de dato	Restricciones	Descripcion
CursoID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
Visto	BIT	CHECK(Visto IN(0,1)), NOT NULL	Indica si el estudiante vio la clase

Tabla: Inscripcion

Atributo	Tipo de dato	Restricciones	Descripcion
InscripcionID	INT	PRIMARY KEY	Identificador único de la inscripción
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
CursoID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
FechaInscripcion	DATE	NOT NULL, DEFAULT GETDATE()	Fecha de inscripción en el curso

Tabla: Progreso

Atributo	Tipo de dato	Restricciones	Descripcion
ProgresoID	INT	PRIMARY KEY	Identificador único del progreso
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
CursoID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
PorcentajeAvance	DECIMAL(5,2)	NOT NULL, DEFAULT 0	Porcentaje de avance del estudiante

La tabla Progreso almacena información sobre el avance individual de los estudiantes en los cursos disponibles . Cada registro en esta tabla representa el progreso de un estudiante específico en un curso determinado, incluyendo el porcentaje de avance.

- **Estudiantes – Progreso:** Cada progreso está asociado a un único estudiante, lo que significa que un estudiante puede tener múltiples registros de progreso (uno por cada curso), pero cada progreso pertenece solo a un estudiante. Esta relación es de uno a muchos y se establece mediante la clave foránea EstudianteID en la tabla Progreso, que hace referencia al EstudianteID en la tabla Estudiantes.
- **Cursos – Progreso:** Cada progreso también se relaciona con un único curso. Un curso puede tener muchos estudiantes asociados a sus respectivos progresos, pero cada registro de progreso corresponde a un único curso. Esta relación uno a muchos se define mediante la clave foránea CursoID en la tabla Progreso, que apunta al CursoID de la tabla Cursos.

Tabla: Calificacion

Atributo	Tipo de dato	Restricciones	Descripcion
CalificacionID	INT	PRIMARY KEY	Identificador único de la calificación
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
CursoID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
Nota	DECIMAL(5,2)	NOT NULL, DEFAULT 0	Nota obtenida en el curso

La tabla *Calificación* almacena las notas que los estudiantes obtienen en los cursos. Cada calificación pertenece a un único estudiante y a un único curso, lo que permite registrar la nota final de un estudiante en un curso específico.

- **Estudiantes – Calificación:** Un estudiante puede tener muchas calificaciones (una por cada curso). Relación de uno a muchos mediante la clave foránea *EstudianteID*.
- **Cursos – Calificación:** Un curso puede tener muchas calificaciones (una por cada estudiante). Cada calificación está asociada a un único curso mediante la clave foránea *CursoID*.

Tabla: Certificacion

Atributo	Tipo de dato	Restricciones	Descripcion
CertificacionID	INT	PRIMARY KEY	Identificador único de la certificación
EstudianteID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Estudiantes
CursoID	INT	NOT NULL, FOREIGN KEY	Relación con la tabla Cursos
FechaEmision	DATE	NOT NULL, DEFAULT GETDATE()	Fecha de emisión de la certificación

La tabla *Certificación* registra los certificados emitidos a los estudiantes que completan satisfactoriamente un curso. Cada registro incluye al estudiante, el curso y la fecha de emisión del certificado.

- *Estudiantes - Certificación:* Un estudiante puede recibir muchas certificaciones (una por cada curso completado). Relación de uno a muchos mediante la clave foránea *EstudianteID*.
- *Cursos - Certificación:* Un curso puede certificar a muchos estudiantes. Cada certificación está asociada a un único curso, mediante la clave foránea *CursoID*.



# Diagrama Relacional

