**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](#_Toc197973045)

[Diccionario de Datos 7](#_Toc197973047)

[Diagrama Relacional 1](#_Toc197973046)7

# Introducción

Este proyecto tiene como objetivo crear una base de datos relacional para una plataforma de e-learning. La base de datos debe almacenara 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 correspondiente, el cual 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 en la prueba refleja directamente el nivel de comprensión del contenido brindado.

Finalmente, la **calificación final del curso** se calcula como el **promedio de los porcentajes obtenidos en cada módulo**. Para aprobar el curso, el estudiante deberá alcanzar un **porcentaje mínimo de aprobación**, el cual se define al momento de la creación del curso.

Este enfoque permite que la evaluación se mantenga coherente incluso si se agregan nuevos módulos al curso, ya que el promedio final se ajusta automáticamente sin necesidad de modificar el criterio mínimo de aprobación. De esta manera, se garantiza un sistema flexible, objetivo y escalable que refleja el progreso real del estudiante a lo largo de todo el proceso formativo.

**LIMPIEZA DE DATOS**

Para garantizar un funcionamiento organizado y sin inconsistencias dentro del sistema, se adoptará una lógica basada en la persistencia de los datos. En lugar de eliminar registros de forma definitiva, se utilizará un **campo de "estado"** para indicar si un registro se encuentra **activo** o **inactivo**.

De esta manera, **todos los elementos del sistema**, incluyendo estudiantes, cursos, docentes, módulos y clases, **se conservarán en la base de datos**, permitiendo mantener un historial completo y trazable de la actividad académica. Esto resulta fundamental para conservar información clave como certificaciones, historial de participación, asignaciones y otras métricas relevantes, incluso cuando los elementos ya no estén en uso activo.

Al marcar registros como inactivos en lugar de eliminarlos, el sistema evitará la pérdida de datos importantes, permitirá auditorías históricas y facilitará la trazabilidad de la información, asegurando al mismo tiempo un funcionamiento más robusto, organizado y seguro.

**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).

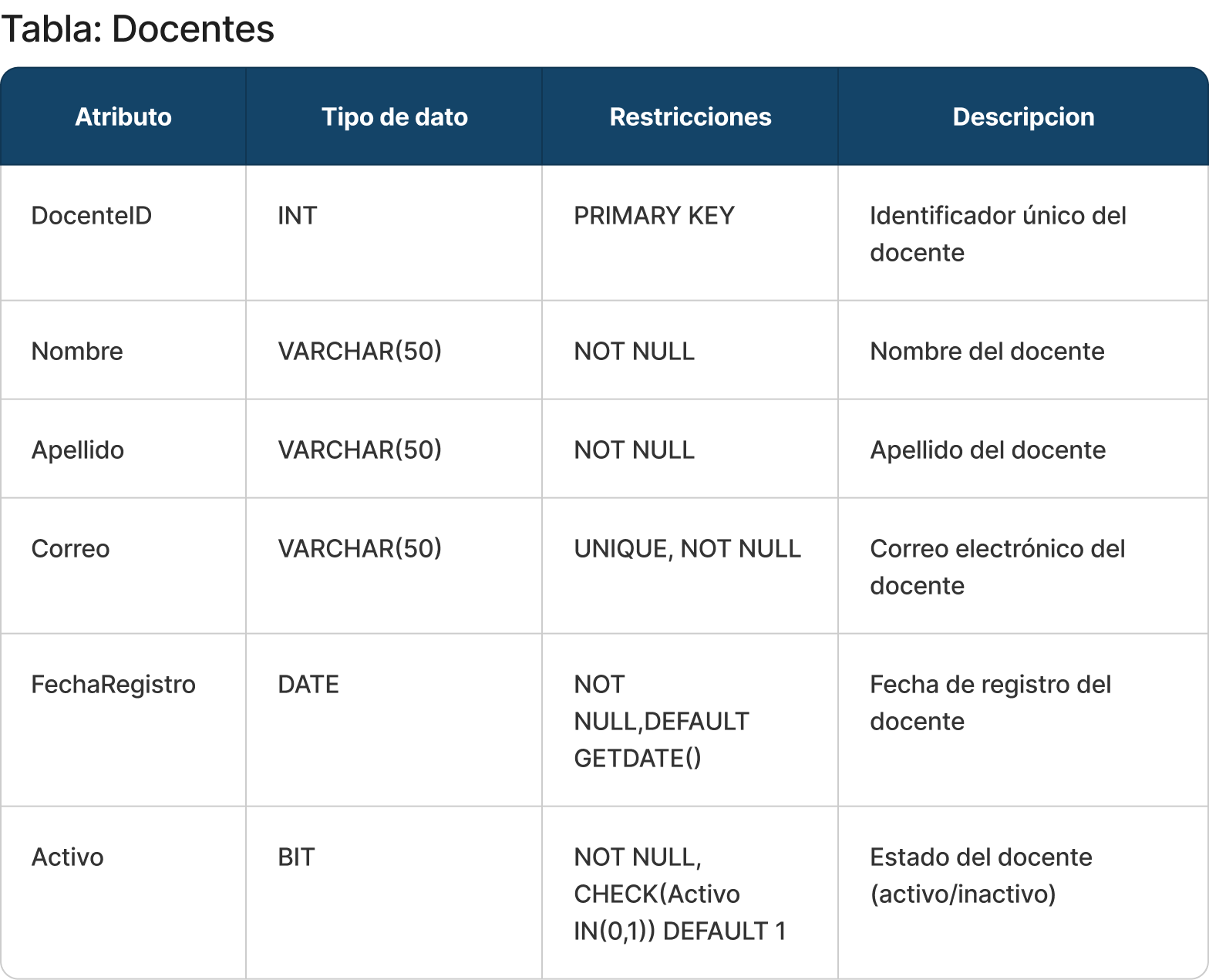
Por lo tanto para este proyecto, se implementara una tabla de bitácora por cada entidad principal, donde cada una de ellas contendra las siguientes caracteristicas:

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

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.

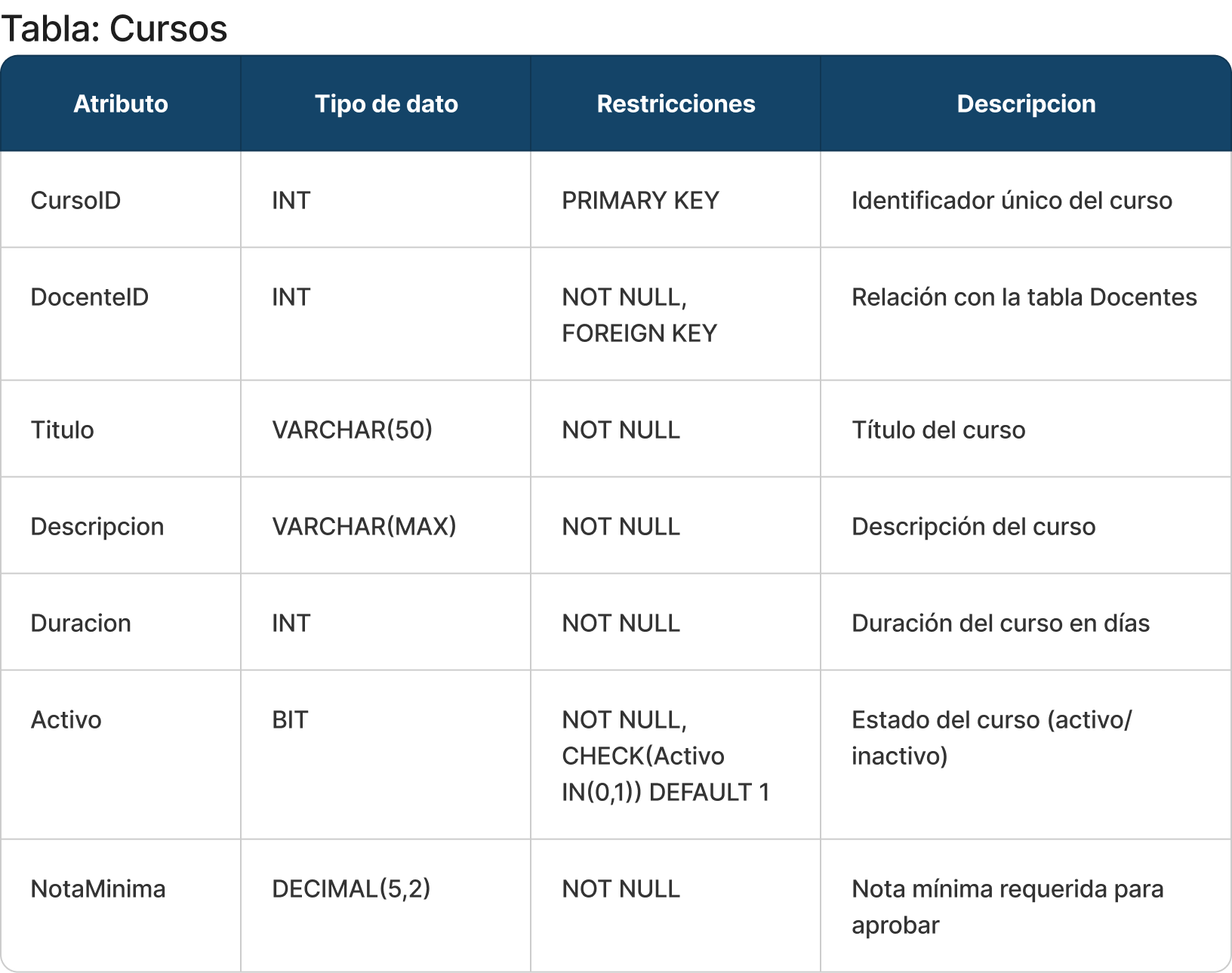
Esto no solo facilita el mantenimiento y la seguridad, sino que también aporta una capa de resolución de errores, ya que nos permite identificar rápidamente las acciones que causaron problemas o inconsistencias, facilitando su corrección de manera más eficiente.

# Diccionario de Datos



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.

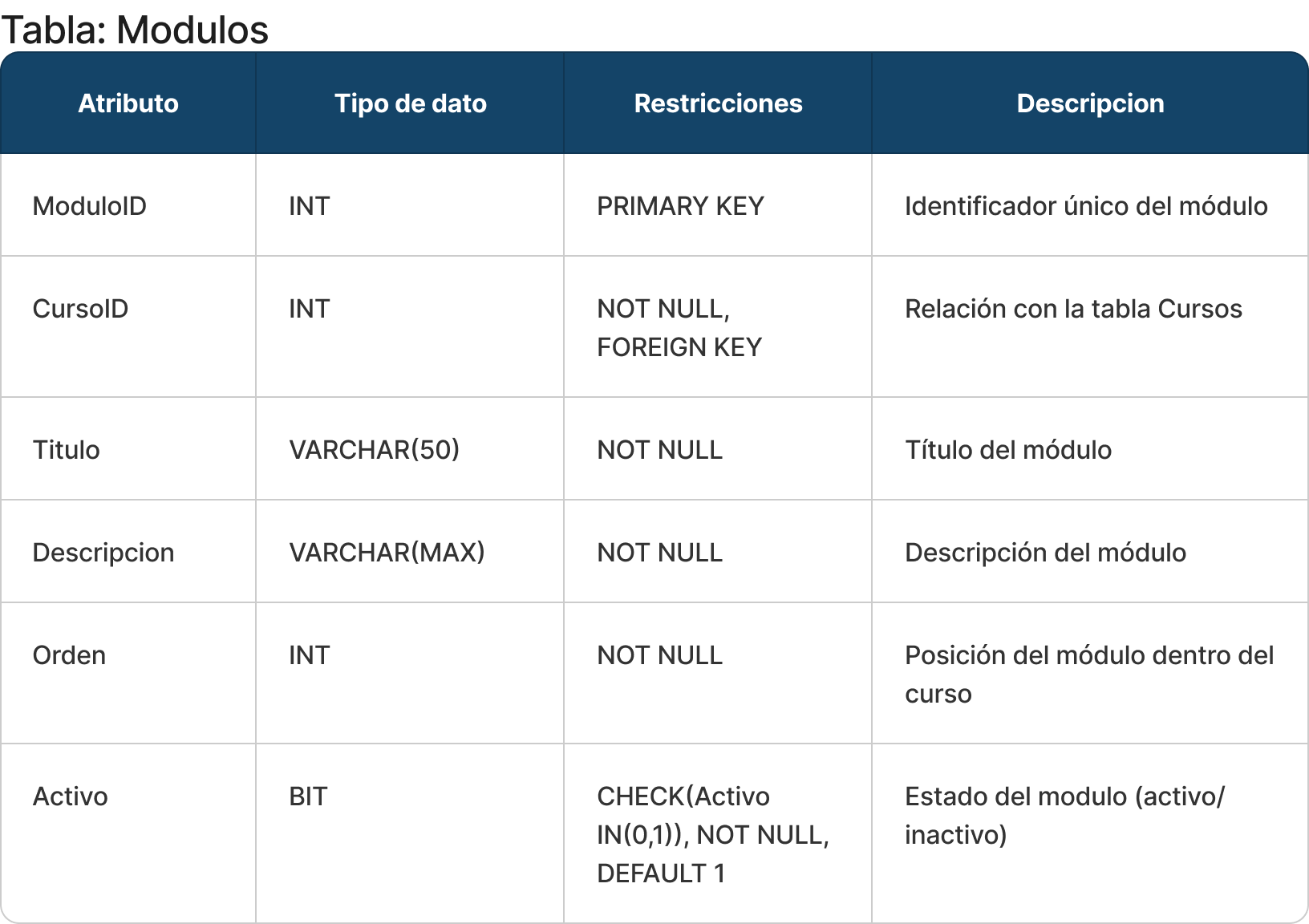
Además, un docente puede estar asociado a muchos cursos dentro de la plataforma. Esta relación uno a muchos permite que un mismo docente sea responsable de varios cursos.



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

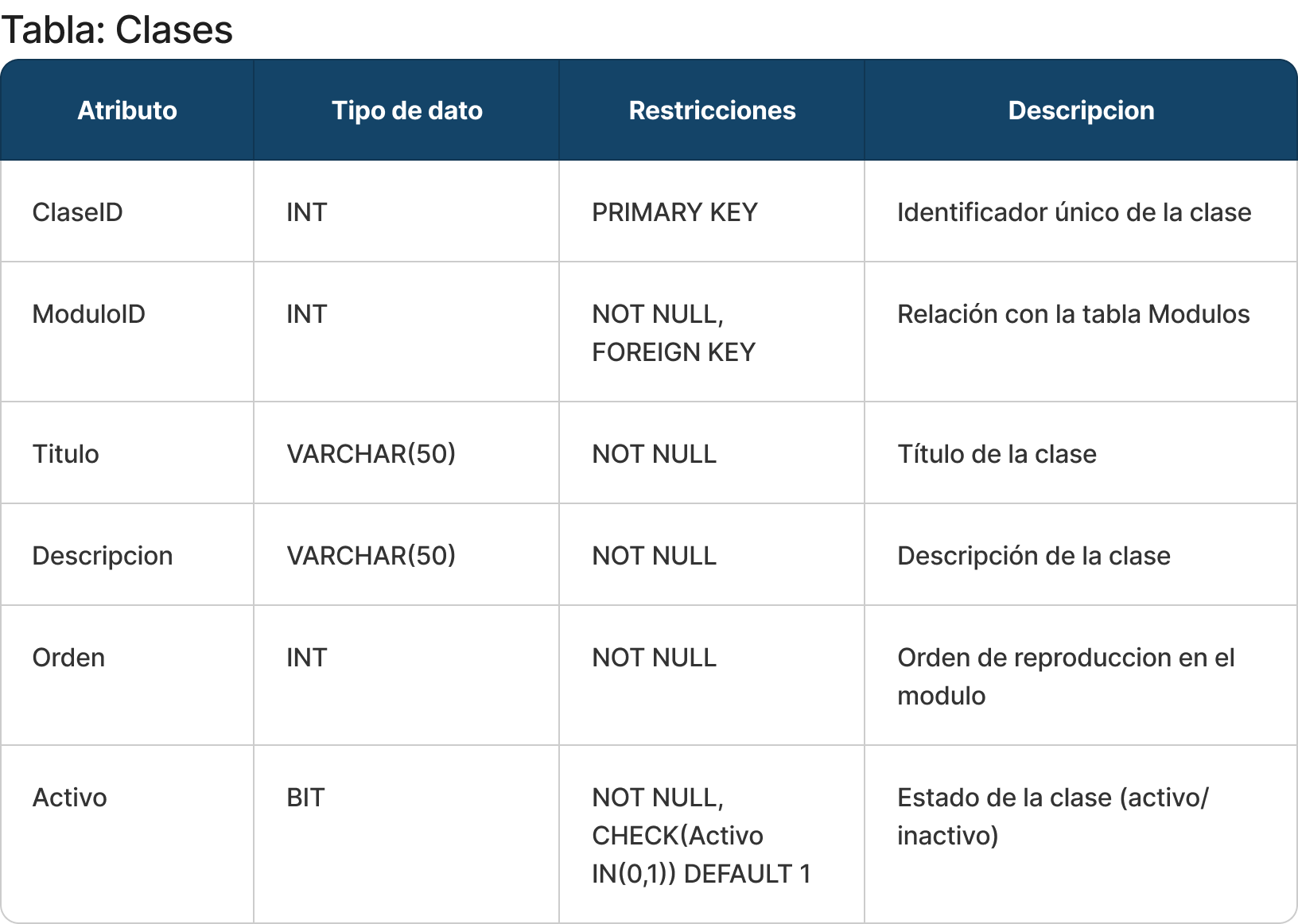
* **Docentes- Cursos**: Un curso esta dictado por un único profesor, lo que establece una relación uno a muchos entre Docentes y Cursos. El docenteid en la tabla Cursos es una clave foránea que hace referencia al docenteid en la tabla Docentes.



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.

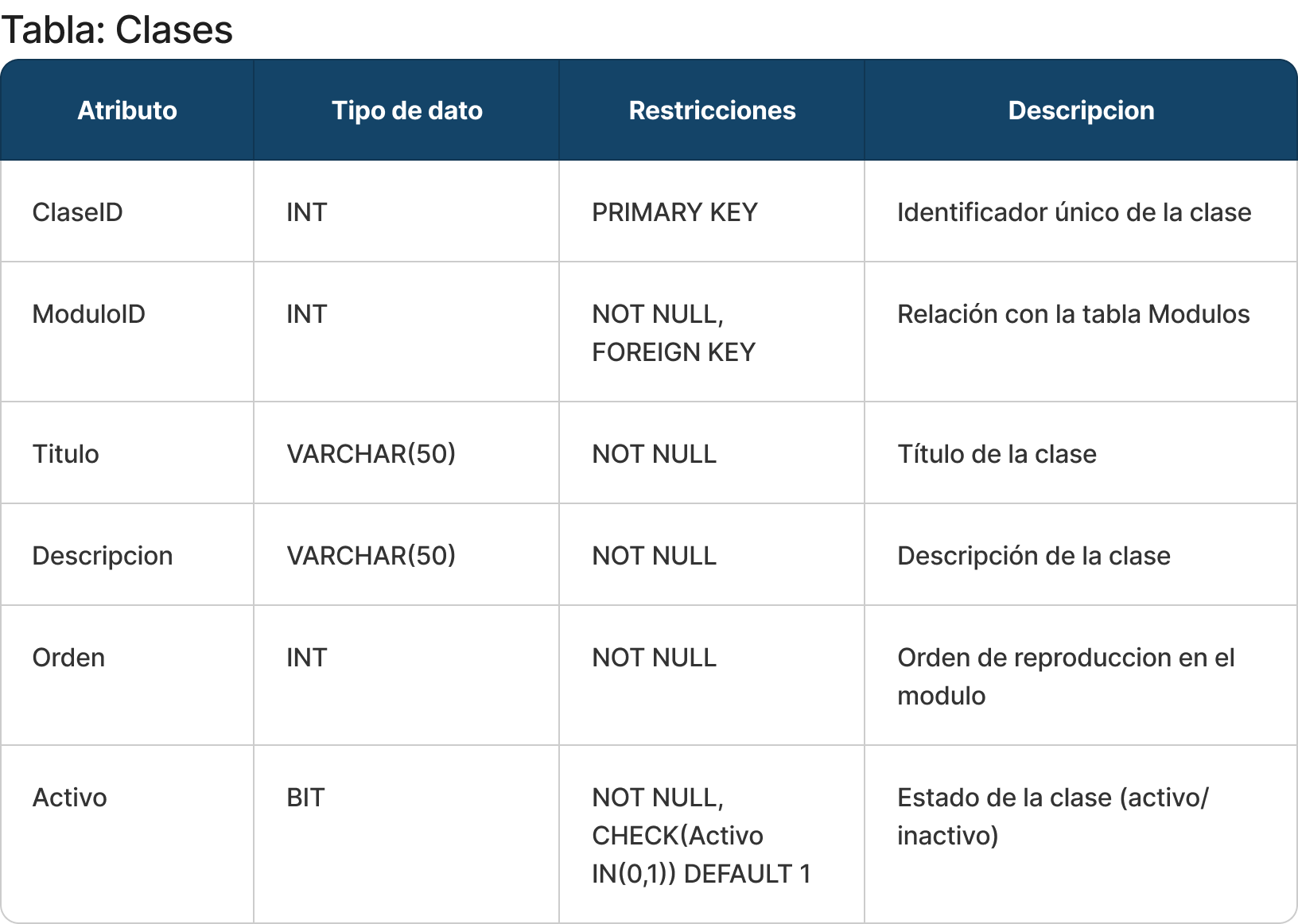
Cada módulo se compone de una o varias clases.

* **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.



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.

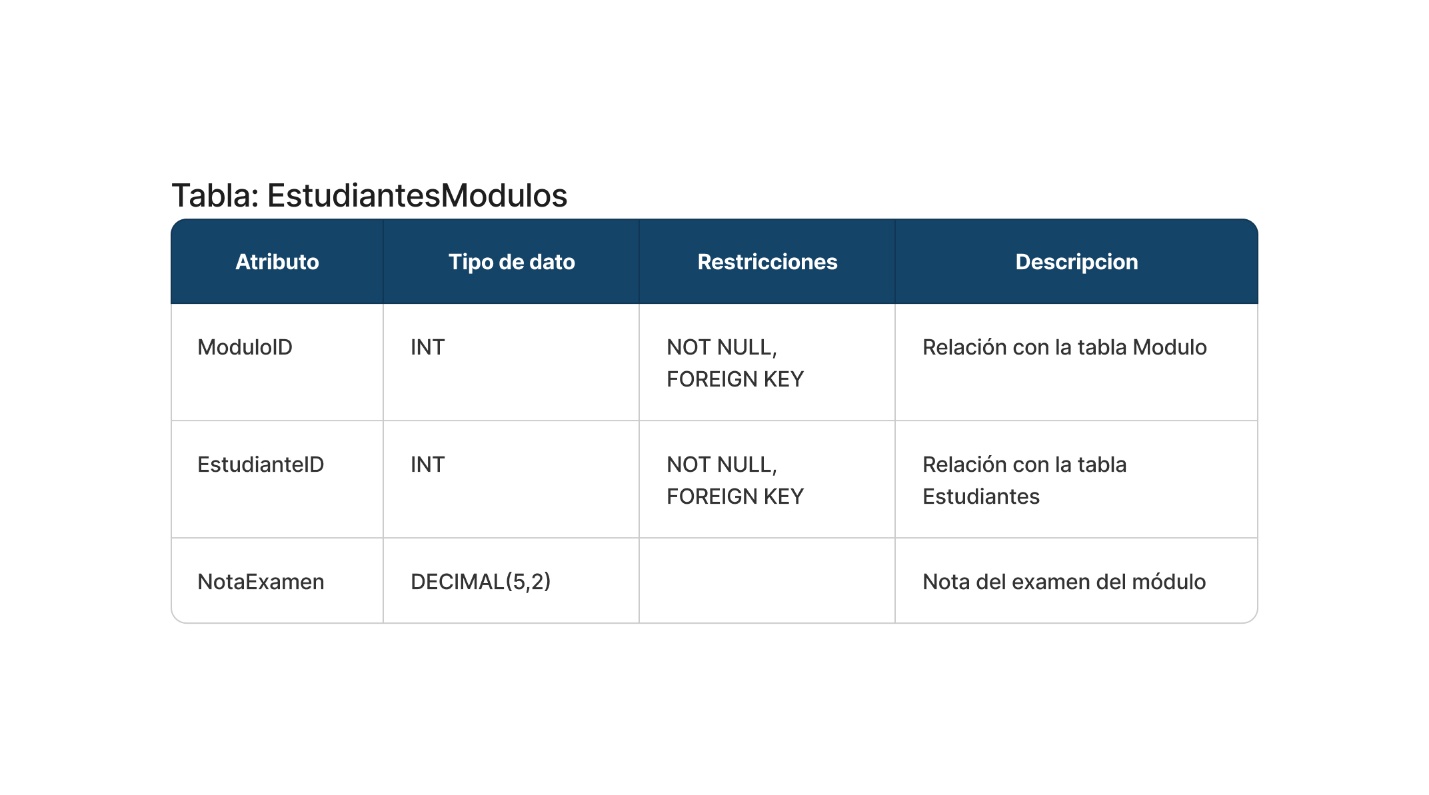
* **Módulos - Clases**: Cada clase pertenece a un único módulo. Por lo tanto, la relación entre Módulos y Clases es de uno a muchos. Es decir, un módulo puede tener varias clases, pero cada clase está asociada a un único módulo. Esta relación se establece a través de la clave foránea Moduloid en la tabla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



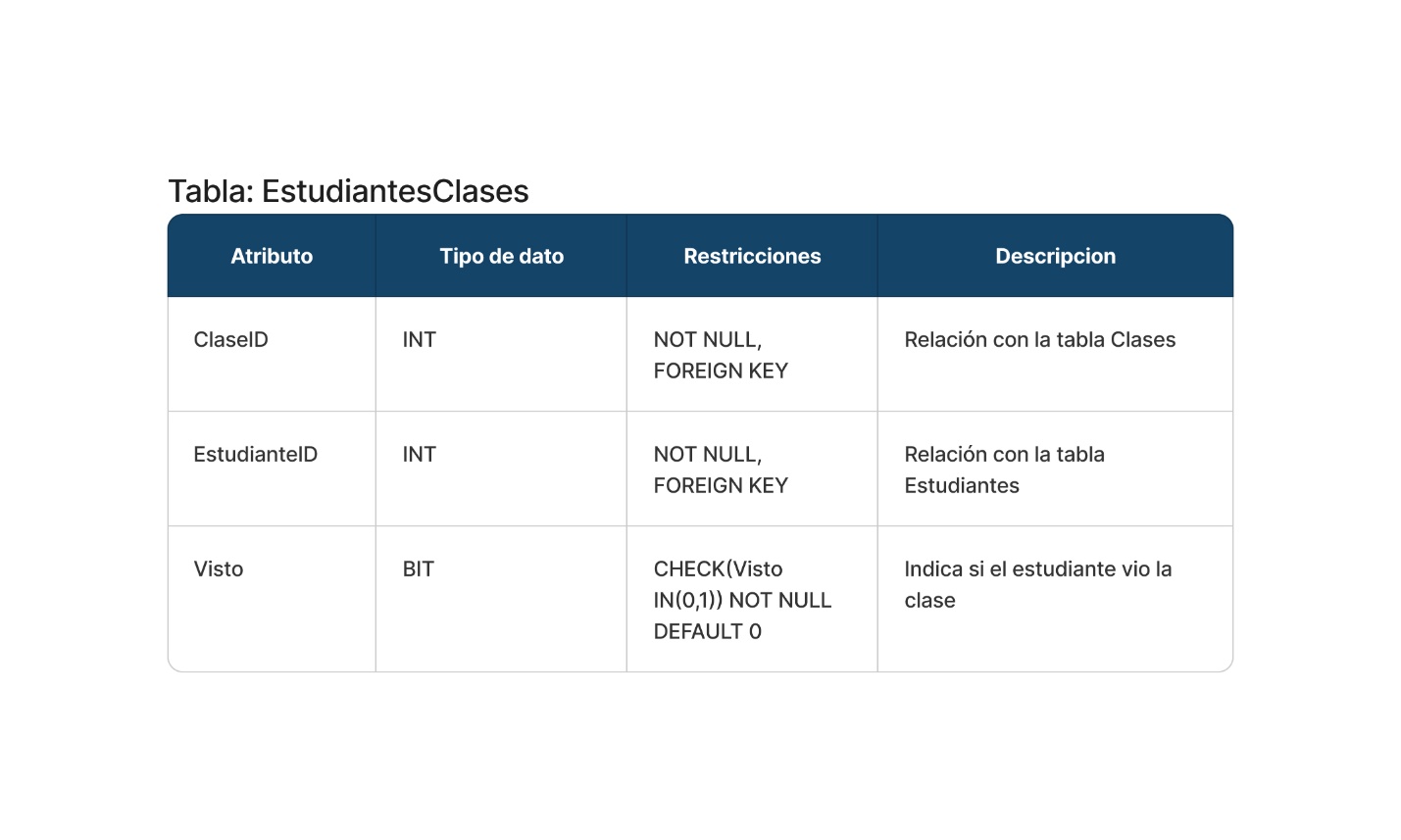
La tabla Estudiantes almacena la información personal y de estado de los usuarios que participan en los cursos .



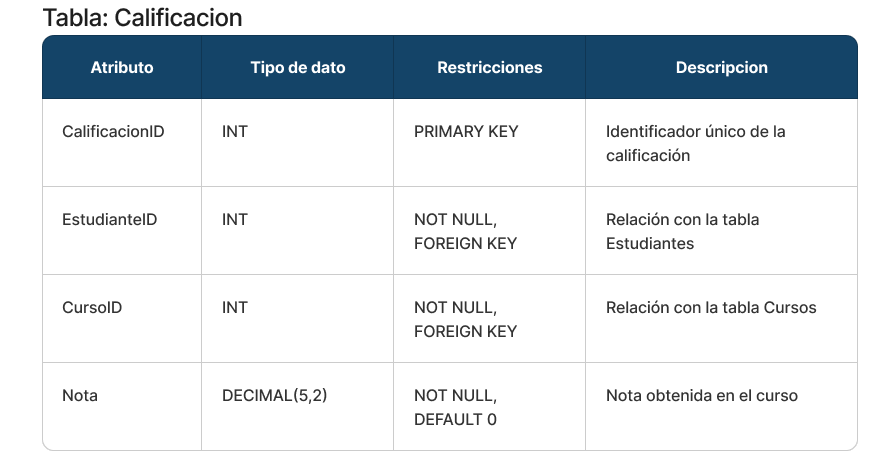
* **Inscripcion:**   
  Cada estudiante puede inscribirse en varios cursos, y cada curso puede tener múltiples estudiantes. Por lo tanto, la relación entre Estudiantes y Cursos es de muchos a muchos. Esta relación se implementa mediante una tabla intermedia que contiene las claves foráneas estudianteid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



* **Estudiantes\_Módulos :**   
  Cada estudiante puede realizar varios módulo, 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, estudianteid y notaexamen. Las claves foráneas establecen las conexiones necesarias con las tablas Módulos y Estudiantes. Esto permite almacenar la calificación obtenida por cada estudiante en cada módulo específico.



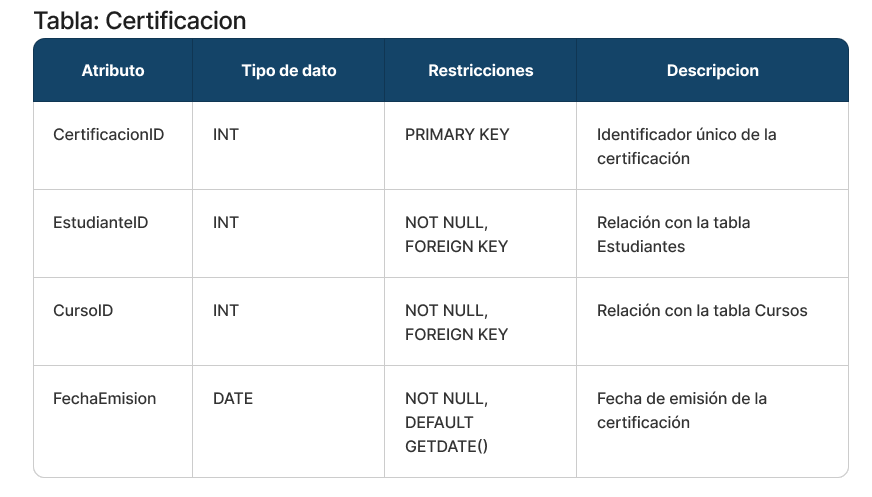
* **Estudiantes\_Clases :**   
  Cada estudiante puede ver muchas clases, y cada clase puede ser vista por muchos estudiantes. Por lo tanto, la relación entre Estudiantes y Clases es de muchos a muchos. Se utiliza una tabla intermedia que incluye claseid, estudianteid y un campo visto (con valores 0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



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**.
* C**ursos - Certificación**: Un curso puede certificar a muchos estudiantes. Cada certificación está asociada a un único curso, mediante la clave foránea **CursoID.**

# Diagrama Relacional

# 