# Base de Datos (ITIZ2200)

**PROYECTO INTEGRADOR**

**INTEGRANTES**: Daniel Sierra, Pablo Vargas y Ricardo Herrera

**Fecha**: 04/05/2004

**DOCENTE**: Ana María Pazmiño Delgado

# Introducción

El presente proyecto integrador tiene como propósito principal el detallar el proceso de diseño, implementación y evaluación de la base de datos desarrollada para la administración de los cursos extracurriculares de la academia de Talentos 360, problemática ya establecida.

# Objetivos del manual técnico

* Diseñar el modelamiento de la base de datos, a partir del análisis de los requerimientos de la actividad planteada identificando el contexto problema y las restricciones propias para el problema propuesto.
* Implementar la solución técnica seleccionada, utilizando las herramientas y metodologías que den solución al problema.
* Evaluar de manera sistemática el rendimiento de la implementación realizada, utilizando pruebas de estrés que permitan verificar su estabilidad y desempeño bajo diferentes condiciones

# Alcance del diseño de la base de datos

* La creación de un modelo conceptual que integra todas las entidades principales involucrada en el programa académico: Cursos, ediciones, estudiantes, instructores, evaluaciones y encuestas.
* La definición de relaciones y atributos que reflejan las operaciones internas del programa como las inscripciones, asignaciones de instructores, gestión de evaluación, calificaciones y retroalimentación con las encuestas.
* La preparación del modelo lógico para implementación técnica, garantizando integridad referencial, dominios y estructura escalables al futuro.

# Definición de términos y acrónimos

* Data base: Base de datos; sistema estructurado para almacenar, gestionar y recuperar datos de manera eficiente.
* Entidad: Elemento del modelo conceptual que representa objetos reales que tienen atributos.
* Atributo: Característica específica de una entidad.
* Relación: Asociación entre dos o más entidades que refleja cómo interactúan entre sí en el sistema.
* Dominio: Conjunto de valores permitidos para un atributo específico.
* Abstracción: Proceso en el cual se identifican y seleccionan únicamente los elementos esenciales de un sistema para representar de forma simplificada la estructura del sistema.
* Cardinalidad: Restricción que indica el número mínimo y máximo de ocurrencias que una entidad puede tener en relación con otra.
* Modelo conceptual: Representación abstracta de la base de datos que describe entidades, relaciones, atributos y dominios, sin detallar aspectos técnicos ni de implementación.
* Modelo lógico: Representación técnica y detallada del modelo conceptual, adaptando a un sistema gestor de bases de datos específico, incluyendo tipos de datos y restricciones.
* Modelo físico: Implementación concreta del modelo lógico en un sistema gestor de base de datos, considerando aspectos como índice, optimización, almacenamiento y acceso.

# Consigna y problemática del sistema

# Leyenda

* Entidades
* Atributos
* Relaciones
* Dominio

# Problemática

La academia "Talentos 360", especializada en cursos extracurriculares para jóvenes y adultos (como idiomas, programación, arte y desarrollo personal), ha crecido notablemente en los últimos años. Actualmente, enfrenta grandes desafíos para administrar adecuadamente toda la información generada por sus actividades, por lo que requiere desarrollar un sistema de información robusto que le permita gestionar de forma eficiente todos los procesos internos.

Uno de los ejes principales del sistema será la gestión de cursos, cada uno con un código único, nombre, descripción, duración (en horas), nivel (básico, intermedio, avanzado), y el área temática a la que pertenece. Cada curso puede dictarse varias veces al año, por lo que se crean ediciones específicas, cada una con su código, fechas de inicio y fin, y modalidad (presencial, virtual o híbrida).

Por otro lado, los estudiantes deben registrarse para poder tomar cursos. La academia requiere almacenar su nombre completo, número de identificación, fecha de nacimiento, datos de contacto y una ). Un estudiante puede inscribirse a múltiples ediciones de distintos cursos, y cada inscripción debe incluir la fecha y el estado (activo, retirado, finalizado).

La gestión de instructores también es esencial. De cada uno se requiere guardar su número de identificación, nombre completo, especialidad, años de experiencia, y tipo de contrato (tiempo completo, medio tiempo, por horas). Un instructor puede dictar varios cursos y una edición específica de un curso puede tener uno o más instructores asignados.

Para asegurar la calidad educativa, cada edición de curso tiene asociadas evaluaciones como exámenes o proyectos. Cada evaluación tiene una fecha, tipo, descripción y puntaje máximo. Los estudiantes que participan en una edición deben presentar dichas evaluaciones y se requiere registrar su calificación final en cada una.

Finalmente, la academia también realiza encuestas de satisfacción para estudiantes, vinculadas a una edición específica del curso. Estas encuestas deben almacenar el nombre del estudiante, la fecha, los comentarios y una calificación general (de 1 a 5). Esta información permitirá analizar la percepción del servicio y mejorar continuamente la calidad de la enseñanza.

El nuevo sistema de base de datos debe organizar toda esta información, reflejar adecuadamente las relaciones entre entidades, garantizar la integridad de los datos y servir de base para futuros análisis que permitan optimizar la gestión académica, mejorar la toma de decisiones y ofrecer mejores experiencias formativas.

# Diseño conceptual

# Primera propuesta

Para la primera propuesta de solución para la base de datos del programa de cursos extracurriculares de Talentos 360 se ha verificado la coherencia estructural, las relaciones, las cardinalidades y los atributos definidos.

El modelo se basa en identificar los componentes principales del sistema y sus interacciones, empezando por sus entidades clave: Curso, Edición, Estudiante, Instructor, Evaluación y Encuesta.

En cuanto a las relaciones entre entidades, se propone lo siguiente:

* La relación Curso – Edición se establece como 1:M, dado que cada curso puede tener múltiples ediciones, pero cada edición corresponde a un único curso. Además, se ha definido que el

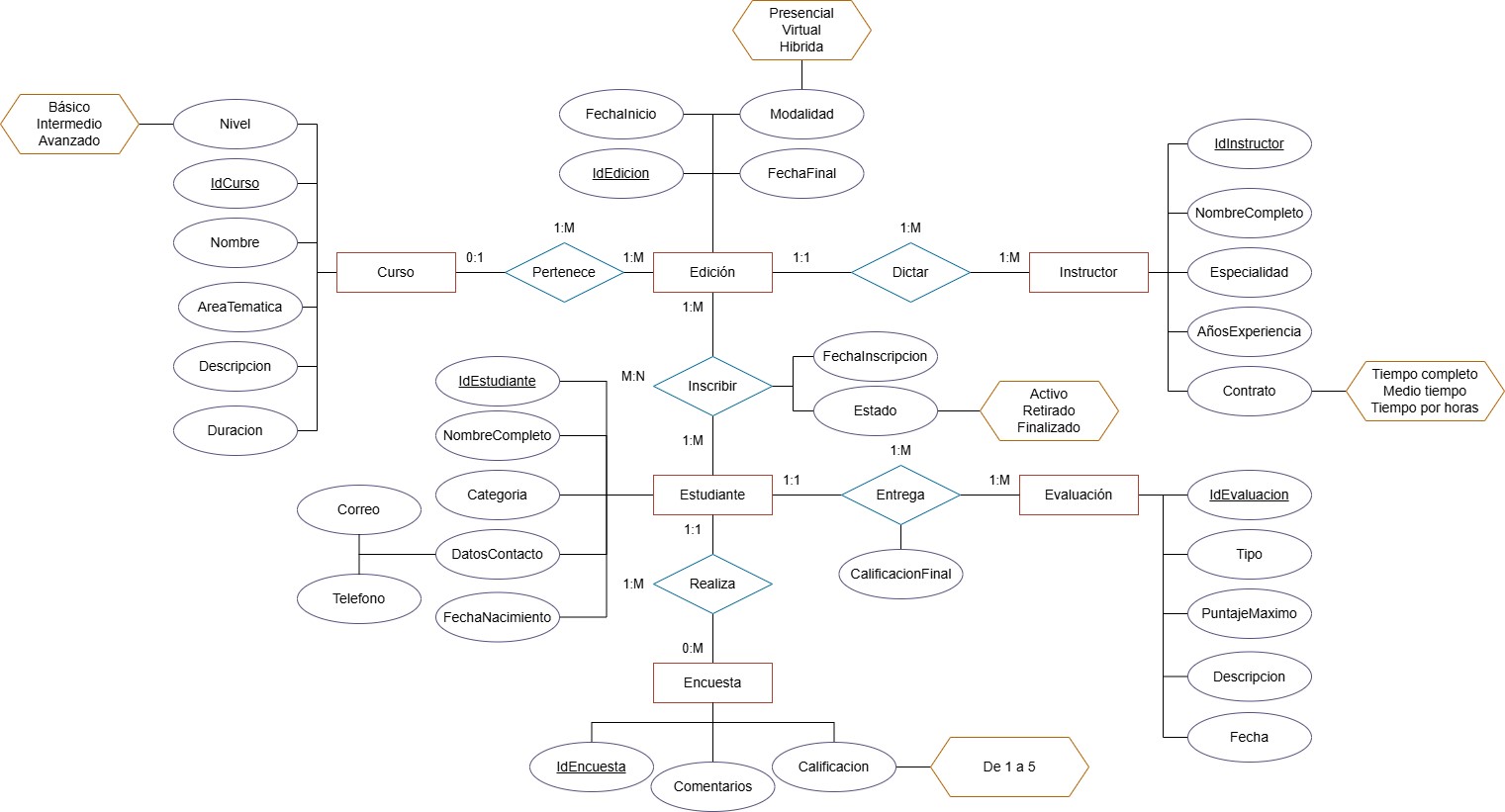
mínimo de participación del lado de **Curso** es 0, lo que significa que puede existir un curso registrado en el sistema sin ninguna edición asociada aún. Este planteamiento responde a escenarios donde se diseñan o programan cursos nuevos que todavía no tienen fechas activas ni ediciones planificadas

* La relación Edición – Estudiante se modela como M:N, ya que un estudiante puede inscribirse a múltiples ediciones y cada edición puede tener múltiples estudiantes inscritos; además, la entidad intermedia Inscribir guarda atributos propios como la fecha de inscripción y el estado (activo, retirado, finalizado).
* La relación Edición – Instructor se plantea inicialmente como 1:M (una edición con varios instructores), aunque se señala como posible punto de ajuste a futuro dependiendo de las necesidades de escalabilidad del negocio.
* Para la relación Estudiante – Evaluación, se incorpora la entidad intermedia Entrega, con el atributo CalificaciónFinal, para resolver la relación M:N, ya que cada estudiante puede participar en múltiples evaluaciones y cada evaluación involucra múltiples estudiantes.
* Finalmente, en la relación Estudiante – Encuesta, se establece que cada estudiante puede realizar varias encuestas, pero cada encuesta está asociada únicamente a un estudiante.

En cuanto a los dominios y atributos agregados, se decidió crear un atributo compuesto en la entidad Estudiante denominado DatosContacto, que agrupa los atributos de correo y teléfono. Asimismo, se definieron dominios restringidos para asegurar la integridad de los datos, incluyendo:

* Nivel del curso: Básico, Intermedio, Avanzado.
* Modalidad: Presencial, Virtual, Híbrida.
* Tipo de contrato del instructor: Tiempo completo, Medio tiempo, Por horas.
* Estado de inscripción: Activo, Retirado, Finalizado.
* Calificación general de encuesta: valores enteros del 1 al 5.

**Diseño conceptual aplicado:**



**Segunda propuesta:**

Para la segunda propuesta de solución para la base de datos del programa de cursos extracurriculares de Talentos 360 se ha revisado y optimizado el modelo conceptual asegurando coherencia estructural, relaciones múltiples, cardinalidades, mínimos y máximos, así como la integración de atributos compuestos y dominios restringidos.

El modelo parte de la identificación de las entidades principales del sistema: Curso, Edición, Estudiante, Instructor, Encuesta y Evaluación. Estas entidades representan los pilares del manejo académico, operativo y de calidad del programa.

En cuanto a las relaciones entre entidades, se propone lo siguiente:

* **Curso - Edición:** Se establece como relación **1:M**, dado que cada curso puede tener múltiples ediciones (mínimo cero para cursos nuevos), pero cada edición corresponde a un único curso.
* **Edición - Instructor (Dicta):** Se modela como **1:M**, ya que cada edición tiene al menos un instructor asignado, mientras que un instructor puede dictar varias ediciones a lo largo del tiempo.
* **Edición - Estudiante (Inscribe):** Esta es una relación **M:N**, porque muchos estudiantes pueden inscribirse en muchas ediciones distintas. Además, se define la entidad intermedia **Inscribe** con atributos propios como FechaInscripción y Estado (activo, retirado, finalizado), alineada a los procesos administrativos reales.
* **Edición - Encuesta (Contiene):** Se establece como **1:M**, dado que cada edición contiene múltiples encuestas de satisfacción, pero cada encuesta pertenece únicamente a una edición.
* **Edición - Evaluación (Contiene):** Se mantiene como **1:M**, reflejando que cada edición posee múltiples evaluaciones (exámenes, proyectos), pero cada evaluación está vinculada solo a una edición específica.
* **Estudiante - Encuesta (Responde):** Se introduce una relación adicional **1:M**, ya que un estudiante puede responder varias encuestas (mínimo cero), y cada encuesta está asociada solo a un estudiante.
* **Estudiante - Evaluación (Realiza):** Se modela como **M:N** usando la relación **Realiza** con el atributo CalificaciónFinal, porque cada estudiante puede presentar múltiples evaluaciones y cada evaluación involucra a muchos estudiantes.

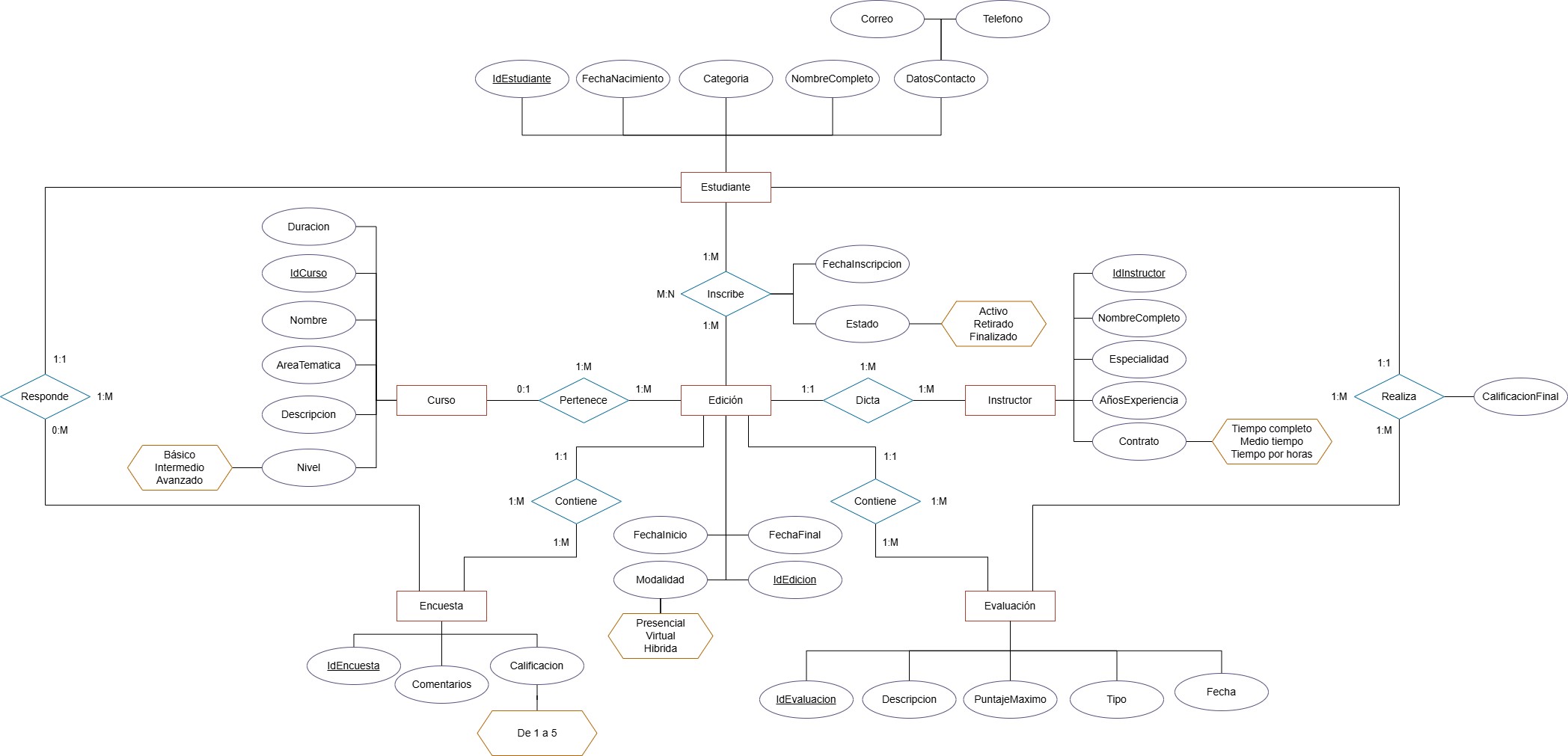
Se ha incorporado un atributo compuesto en Estudiante llamado DatosContacto, que incluye Correo y Teléfono.

Se han definido dominios restringidos:

* Nivel del curso: Básico, intermedio, avanzado.
* Modalidad de edición: Presencial, virtual, híbrida.
* Tipo de contrato del instructor: Tiempo completo, medio tiempo, por horas.
* Estado de inscripción: Activo, retirado, finalizado.
* Calificación general de encuesta: Valores entre 1 a 5.

Además, se incorporó una redistribución en el modelo para ser mucho más preciso con las interacciones entre los estudiantes y los procesos del sistema, creando una relación directa doble tanto en la entidad Encuesta y Evaluación, ya que las dos están directamente relaciones tanto como con la Edición como los Estudiantes. Esto se hizo porque, además de la inscripción a ediciones de cursos, los estudiantes interactúan de forma directa tanto con las evaluaciones como con las encuestas. Esta redistribución también contribuye a mejorar la comprensión de la entidad principal Edición, asegurando que el modelo se mantenga limpio, claro, escalable y alineado a las prácticas de la institución.

**Diseño conceptual aplicado:**



# Elección de la mejor propuesta

Tras comparar ambas propuestas de diseño conceptual para la base de datos del programa de cursos extracurriculares de Talentos 360, se concluye que la segunda propuesta es la más adecuada y completa para ser implementada.

La elección se justifica porque esta última propuesta:

* Incorpora relaciones adicionales necesarias para reflejar con precisión la interacción entre estudiantes, encuestas y evaluaciones, evitando omisiones funcionales que la primera propuesta no contemplaba.
* Define claramente las cardinalidades mínimas y máximas en cada relación, alineadas estrictamente al contexto del problema.
* Introduce atributos compuestos y dominios restringidos de manera organizada, mejorando la normalización y facilitando la integridad de los datos.
* Resuelve de forma eficiente la sobrecarga de relaciones en entidades centrales, como Edición, usando conexiones claras, sin necesidad de duplicar entidades ni generar ambigüedades.

En resumen, la segunda propuesta representa una mejora sustancial en términos de coherencia estructural, alineación con las buenas prácticas de diseño de bases de datos, y capacidad para satisfacer las necesidades específicas del proyecto, garantizando integridad, escalabilidad y eficiencia operativa.

# Diseño lógico

Para la resolución de la segunda fase de este proyecto se transformo del modelo conceptual al modelo relacional, aplicando las reglase y normas básicas del diseño lógico para garantizar la integridad de las relaciones y la normalización de la base de datos. Algunas de las transformaciones que se realizo fueron las relaciones de cardinalidad M:N, así como la incorporación de atributos claves dentro de las relaciones.

# Transformación de relaciones M:N

En el modelo conceptual se identificaron dos relaciones M:N:

1. Estudiante – Edición (Inscribir)

En el diseño relaciones, esta relación se transformó en la tabla intermedia:

InscricipciónEstudianteEdicion, permitiendo representar múltiples estudiantes inscritos en múltiples ediciones, manteniendo la información transaccional de cada inscripción

1. Estudiantes – Evaluación (Realiza)

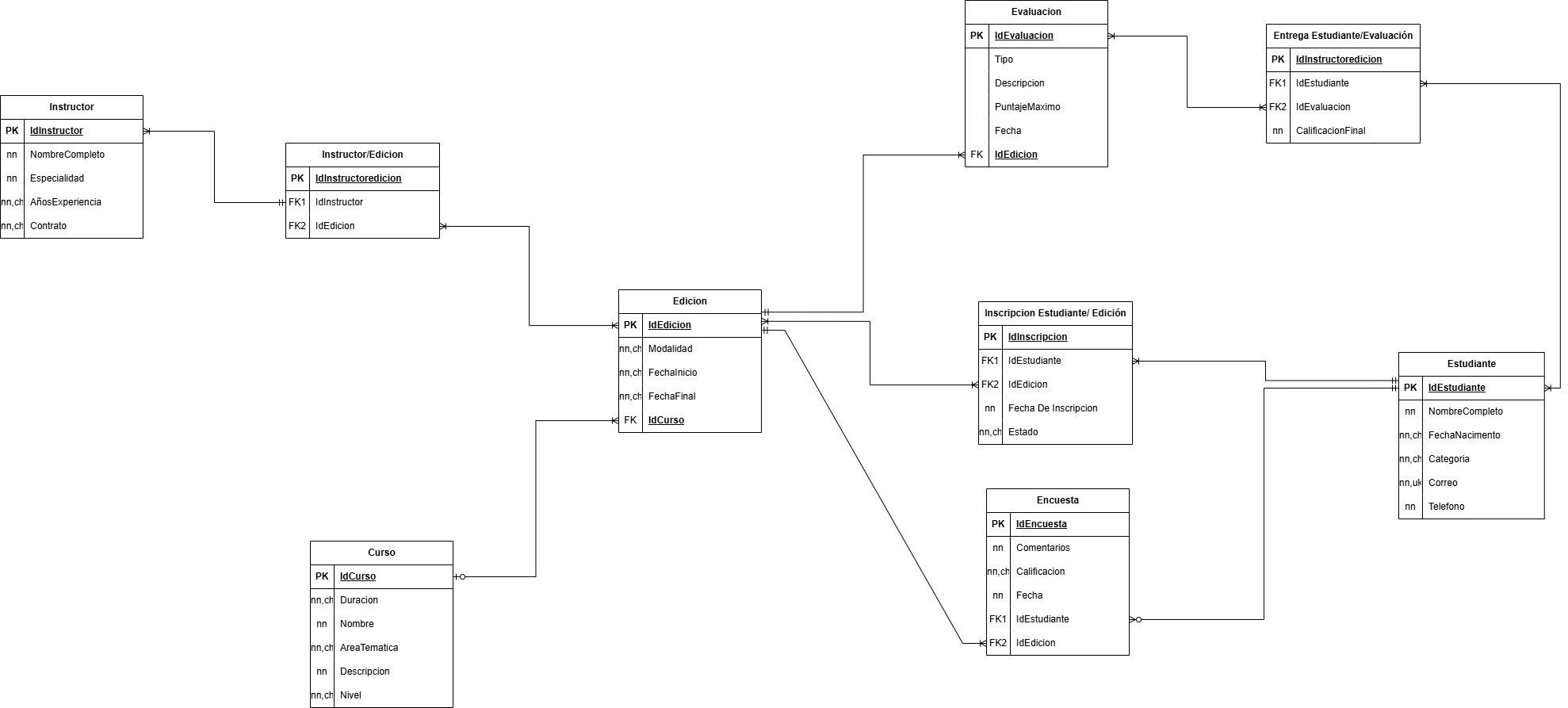
Esta relación fue modelada mediante la tabla: EntregaEstudianteEvaluación, permitiendo registrar el resultado específico de cada estudiante en cada evaluación, resolviendo correctamente la multiplicidad M:N.

# Transformación de relaciones 1:N y 1:1

Las relaciones 1:N, como: Curso – Edición, Edición – Evaluación, Edición – Instructor, Edición – Encuesta fueron representadas incorporando claves foráneas (FK) en las tablas, respetando la integridad referencial.

Aunque no existía una relación recursiva ni una relación estrictamente 1:1 entre dos entidades independientes, sí se cuidó las relaciones 1:1 con cardinalidades obligatorias fueran representadas correctamente. Por ejemplo, una Encuesta solo pertenece a una edición y un estudiante, lo cual se refleja con la combinación de ambas FKs.

# Estructura del diagrama relacional

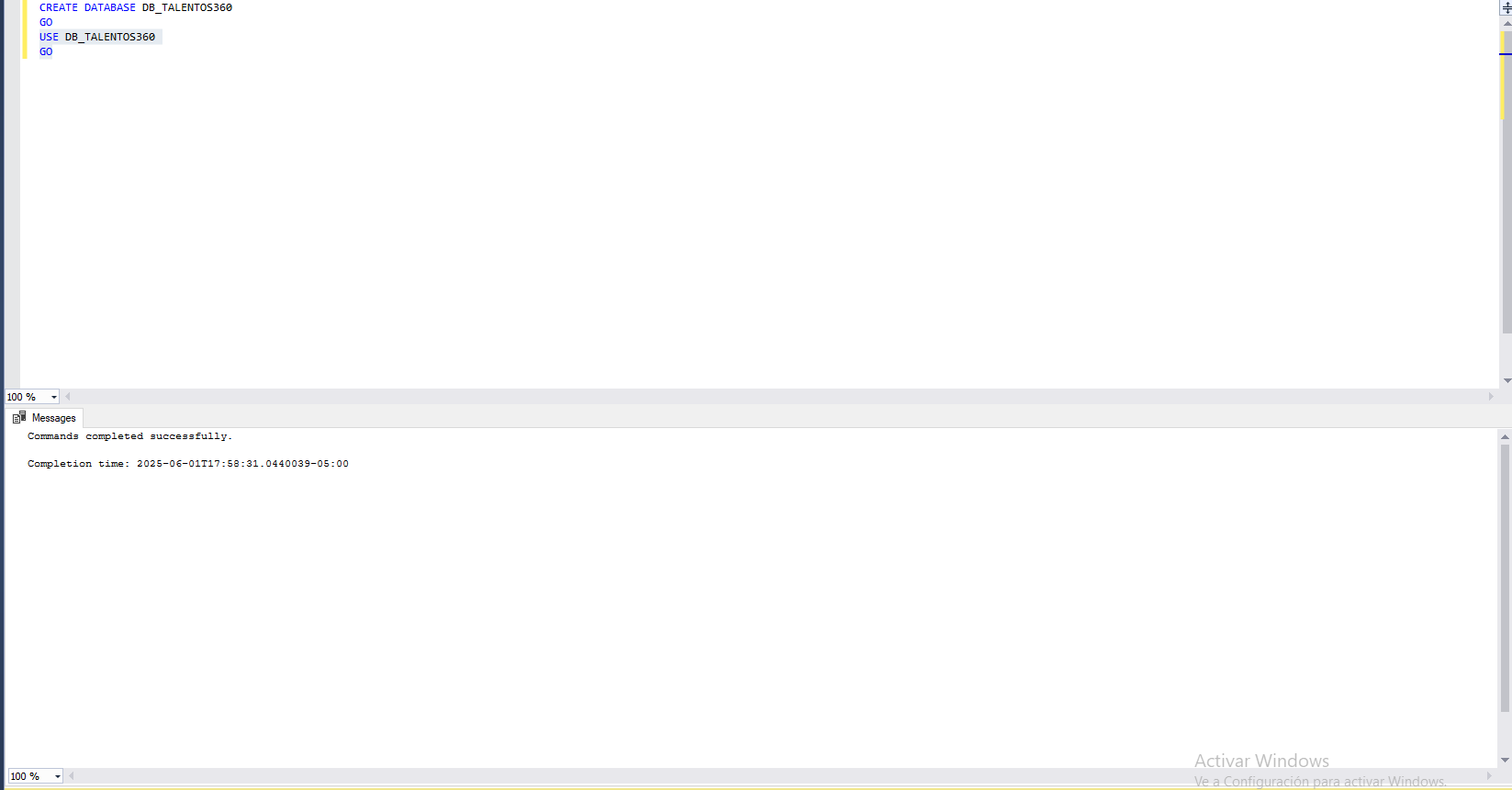


# Diseño físico

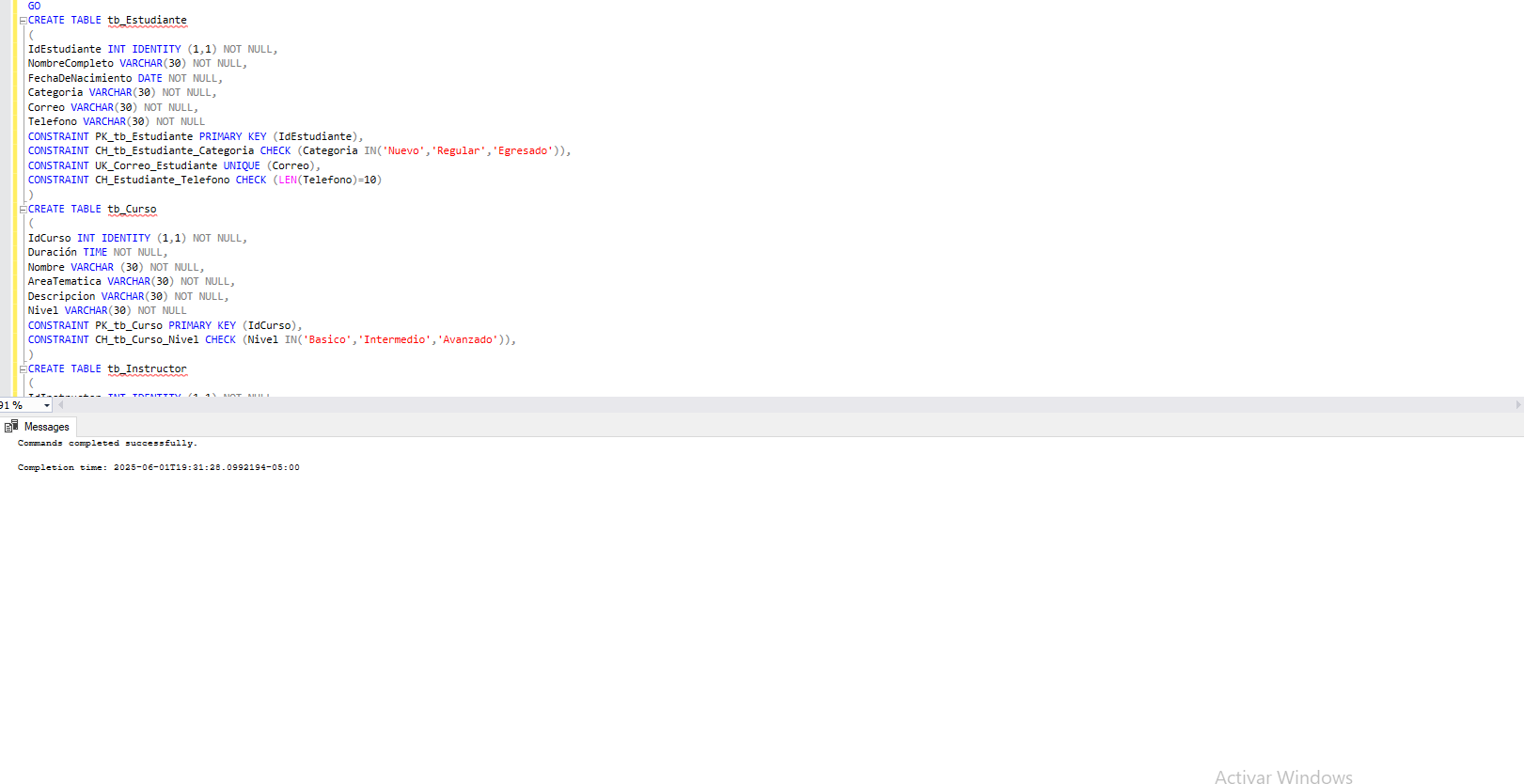
1. Selección de la plataforma de base de datos (SQL Server)

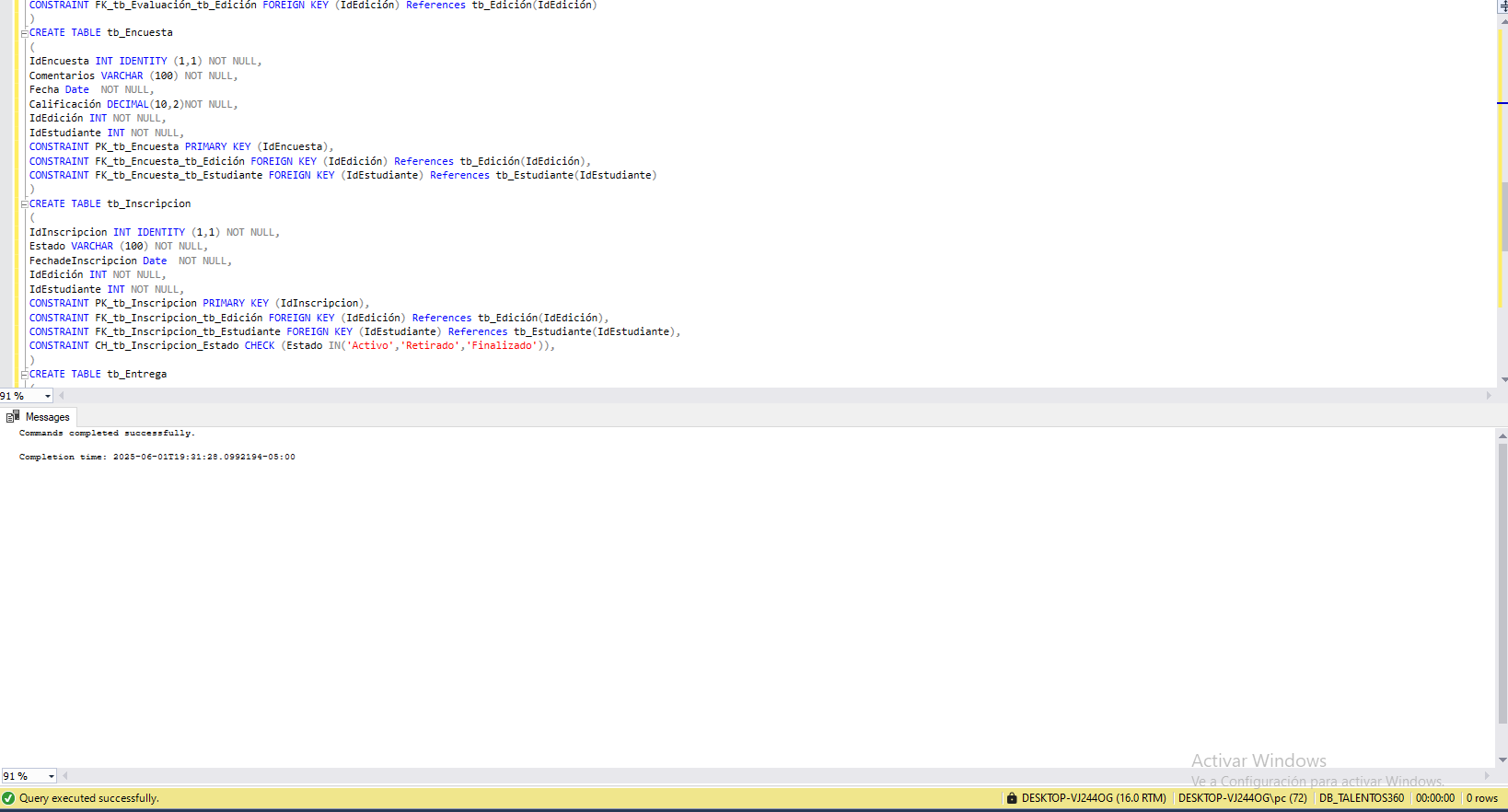
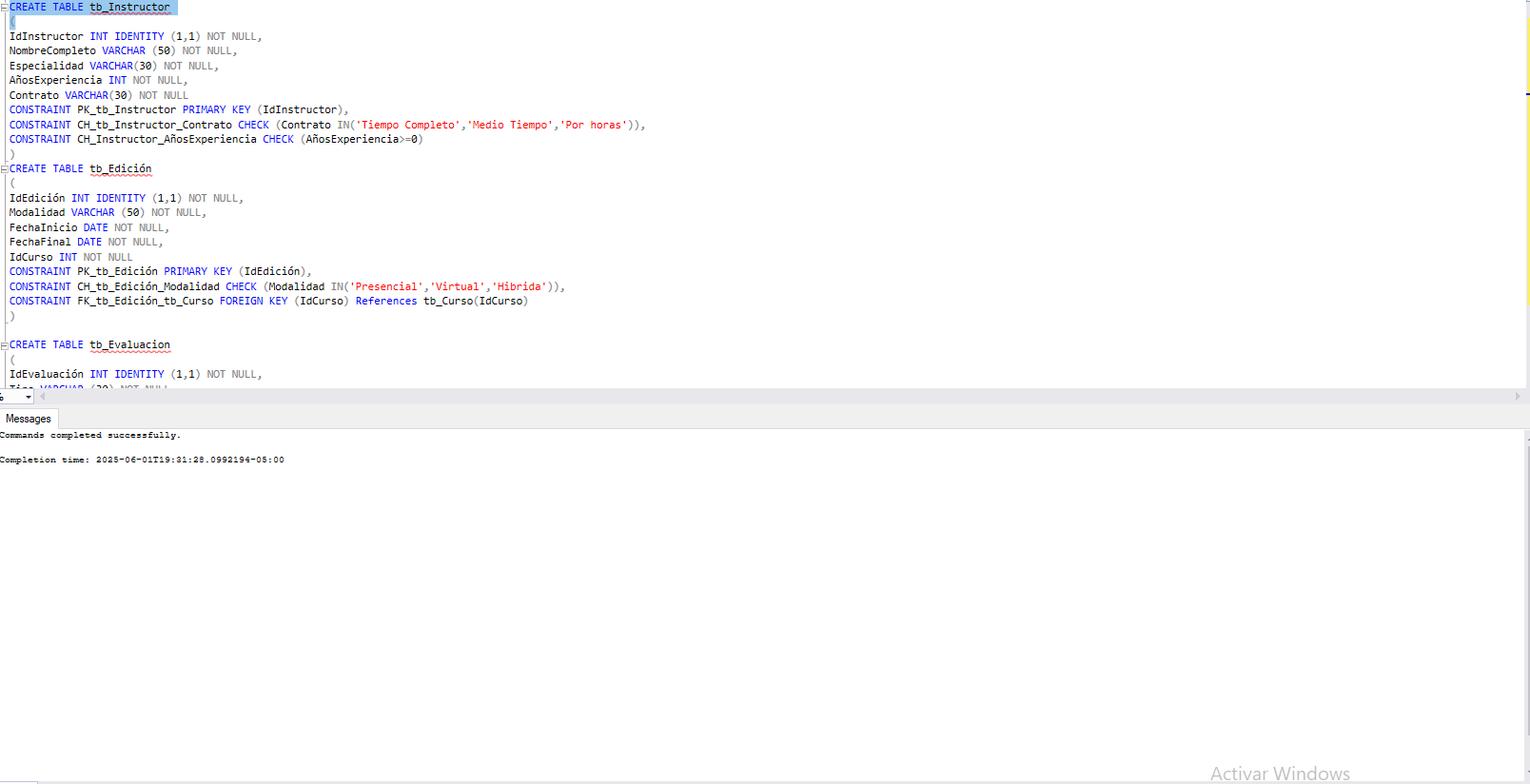
Elegimos Microsoft SQL Server debido a su robustez, facilidad de administración, compatibilidad con el lenguaje T-SQL y su integración con entornos de desarrollo .NET. Se usó la versión SQL Server 2022 Developer para este proyecto.

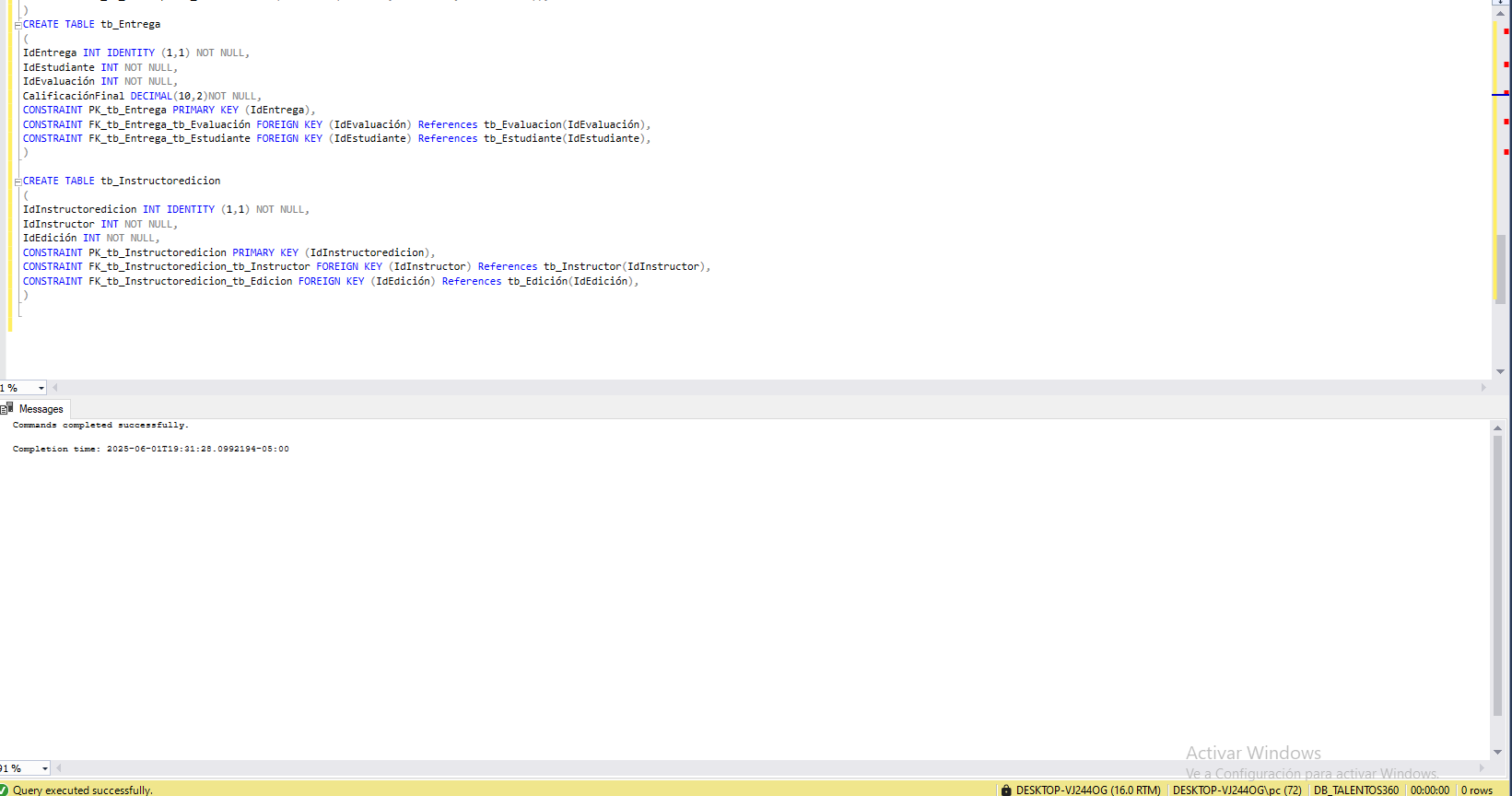
1. Creación de la base de datos



1. Creación de tablas, columnas , restricciones y definición de validaciones de integridad referencial en SQL Server







CREATE DATABASE DB\_TALENTOS360

GO

USE DB\_TALENTOS360

GO

CREATE TABLE tb\_Estudiante

(

IdEstudiante INT IDENTITY (1,1) NOT NULL,

NombreCompleto VARCHAR(30) NOT NULL,

FechaDeNacimiento DATE NOT NULL,

Categoria VARCHAR(30) NOT NULL,

Correo VARCHAR(30) NOT NULL,

Telefono VARCHAR(30) NOT NULL

CONSTRAINT PK\_tb\_Estudiante PRIMARY KEY (IdEstudiante),

CONSTRAINT CH\_tb\_Estudiante\_Categoria CHECK (Categoria IN('Nuevo','Regular','Egresado')),

CONSTRAINT UK\_Correo\_Estudiante UNIQUE (Correo),

CONSTRAINT CH\_Estudiante\_Telefono CHECK (LEN(Telefono)=10)

)

CREATE TABLE tb\_Curso

(

IdCurso INT IDENTITY (1,1) NOT NULL,

Duración TIME NOT NULL,

Nombre VARCHAR (30) NOT NULL,

AreaTematica VARCHAR(30) NOT NULL,

Descripcion VARCHAR(30) NOT NULL,

Nivel VARCHAR(30) NOT NULL

CONSTRAINT PK\_tb\_Curso PRIMARY KEY (IdCurso),

CONSTRAINT CH\_tb\_Curso\_Nivel CHECK (Nivel IN('Basico','Intermedio','Avanzado')),

)

CREATE TABLE tb\_Instructor

(

IdInstructor INT IDENTITY (1,1) NOT NULL,

NombreCompleto VARCHAR (50) NOT NULL,

Especialidad VARCHAR(30) NOT NULL,

AñosExperiencia INT NOT NULL,

Contrato VARCHAR(30) NOT NULL

CONSTRAINT PK\_tb\_Instructor PRIMARY KEY (IdInstructor),

CONSTRAINT CH\_tb\_Instructor\_Contrato CHECK (Contrato IN('Tiempo Completo','Medio Tiempo','Por horas')),

CONSTRAINT CH\_Instructor\_AñosExperiencia CHECK (AñosExperiencia>=0)

)

CREATE TABLE tb\_Edición

(

IdEdición INT IDENTITY (1,1) NOT NULL,

Modalidad VARCHAR (50) NOT NULL,

FechaInicio DATE NOT NULL,

FechaFinal DATE NOT NULL,

IdCurso INT NOT NULL

CONSTRAINT PK\_tb\_Edición PRIMARY KEY (IdEdición),

CONSTRAINT CH\_tb\_Edición\_Modalidad CHECK (Modalidad IN('Presencial','Virtual','Hibrida')),

CONSTRAINT FK\_tb\_Edición\_tb\_Curso FOREIGN KEY (IdCurso) References tb\_Curso(IdCurso)

)

CREATE TABLE tb\_Evaluacion

(

IdEvaluación INT IDENTITY (1,1) NOT NULL,

Tipo VARCHAR (30) NOT NULL,

Descripcion VARCHAR (30) NOT NULL,

PuntajeMaximo DECIMAL(10,2)NOT NULL,

Fecha DATETIME NOT NULL,

IdEdición INT NOT NULL,

CONSTRAINT PK\_tb\_Evaluación PRIMARY KEY (IdEvaluación),

CONSTRAINT FK\_tb\_Evaluación\_tb\_Edición FOREIGN KEY (IdEdición) References tb\_Edición(IdEdición)

)

CREATE TABLE tb\_Encuesta

(

IdEncuesta INT IDENTITY (1,1) NOT NULL,

Comentarios VARCHAR (100) NOT NULL,

Fecha Date NOT NULL,

Calificación DECIMAL(10,2)NOT NULL,

IdEdición INT NOT NULL,

IdEstudiante INT NOT NULL,

CONSTRAINT PK\_tb\_Encuesta PRIMARY KEY (IdEncuesta),

CONSTRAINT FK\_tb\_Encuesta\_tb\_Edición FOREIGN KEY (IdEdición) References tb\_Edición(IdEdición),

CONSTRAINT FK\_tb\_Encuesta\_tb\_Estudiante FOREIGN KEY (IdEstudiante) References tb\_Estudiante(IdEstudiante)

)

CREATE TABLE tb\_Inscripcion

(

IdInscripcion INT IDENTITY (1,1) NOT NULL,

Estado VARCHAR (100) NOT NULL,

FechadeInscripcion Date NOT NULL,

IdEdición INT NOT NULL,

IdEstudiante INT NOT NULL,

CONSTRAINT PK\_tb\_Inscripcion PRIMARY KEY (IdInscripcion),

CONSTRAINT FK\_tb\_Inscripcion\_tb\_Edición FOREIGN KEY (IdEdición) References tb\_Edición(IdEdición),

CONSTRAINT FK\_tb\_Inscripcion\_tb\_Estudiante FOREIGN KEY (IdEstudiante) References tb\_Estudiante(IdEstudiante),

CONSTRAINT CH\_tb\_Inscripcion\_Estado CHECK (Estado IN('Activo','Retirado','Finalizado')),

)

CREATE TABLE tb\_Entrega

(

IdEntrega INT IDENTITY (1,1) NOT NULL,

IdEstudiante INT NOT NULL,

IdEvaluación INT NOT NULL,

CalificaciónFinal DECIMAL(10,2)NOT NULL,

CONSTRAINT PK\_tb\_Entrega PRIMARY KEY (IdEntrega),

CONSTRAINT FK\_tb\_Entrega\_tb\_Evaluación FOREIGN KEY (IdEvaluación) References tb\_Evaluacion(IdEvaluación),

CONSTRAINT FK\_tb\_Entrega\_tb\_Estudiante FOREIGN KEY (IdEstudiante) References tb\_Estudiante(IdEstudiante),

)

CREATE TABLE tb\_Instructoredicion

(

IdInstructoredicion INT IDENTITY (1,1) NOT NULL,

IdInstructor INT NOT NULL,

IdEdición INT NOT NULL,

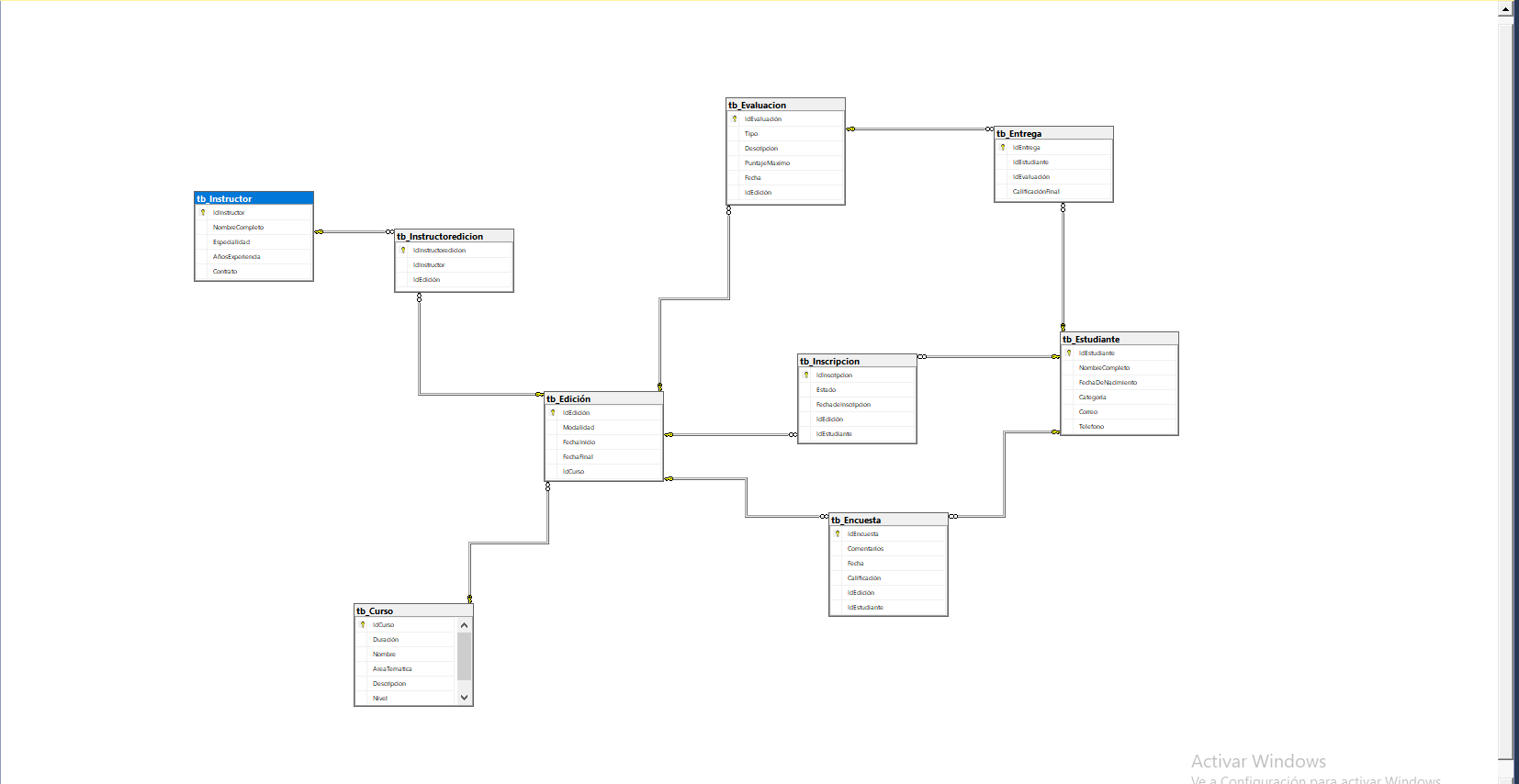
CONSTRAINT PK\_tb\_Instructoredicion PRIMARY KEY (IdInstructoredicion),

CONSTRAINT FK\_tb\_Instructoredicion\_tb\_Instructor FOREIGN KEY (IdInstructor) References tb\_Instructor(IdInstructor),

CONSTRAINT FK\_tb\_Instructoredicion\_tb\_Edicion FOREIGN KEY (IdEdición) References tb\_Edición(IdEdición),

)

1. Diagrama resultante



# Objetos programables

1. Planteamiento de objetivos de la creación de los objetos programables-
2. Diseño, Desarrollo e Implementación de los objetos programables.

# Formatos y buenas prácticas

1. Estándares de nomenclatura y convenciones de codificación
2. Documentación y comentarios del código SQL

# Conclusiones y recomendaciones

1. Resumen de los logros alcanzados en el diseño de la base de datos
2. Recomendaciones para el mantenimiento y futuras mejoras
3. Posibles áreas de desarrollo adicional

# Anexos

1. Diagramas adicionales o detalles técnicos relevantes
2. Scripts SQL de creación de tablas y restricciones
3. Ejemplos de consultas SQL para acceder a los datos