

DIAGRAMA DE CLASES DE PROYECTO DEL SOFTWARE

GA4-220501095-AA2-EV04

ANDRÉS ALBERTO BUILES MUÑOZ

INSTRUCTOR

JHON ALEJANDRO NIÑO TAMBO

CENTRO METALMÉCANICO

SERVICIO NACIONAL DE APRENDIZAJE

SENA

MARZO 7, 2025

TABLA DE CONTENIDO

1. Introducción
2. Tipos de Diagrama de Clases
3. Identificación de Clases Principales
4. Relaciones entre clases
5. Diagrama de Clases
6. Descripción de Interacciones
7. Notas Adicionales
8. Conclusión

INTRODUCCIÓN

Los diagramas de clases son una parte fundamental de la modelación en UML (Lenguaje de Modelado Unificado), que se utilizan para representar las estructuras de los sistemas mediante la visualización de sus clases, atributos, métodos y las relaciones entre ellas

TIPOS DE DIAGRAMA DE CLASES

En este diagrama de clases encontraremos los principales tipos con sus variaciones que son de las siguientes:

- **Diagrama de Clases Básico:** Es la que representa las clases del sistema, sus atributos y métodos, así como las relaciones entre ellas
- **Diagrama de Clases de Dominio:** Se centra en las clases del dominio del problema, es decir, las entidades que representan los conceptos del mundo real que el sistema debe manejar
- **Diagrama de Clases de Diseño:** Es la que incluye los detalles más específicos sobre la implementación, como la visibilidad de atributos y métodos, tipos de datos y además pueden incluir patrones de diseño
- **Diagrama de Clases de Interacción:** Es aquella que muestra cómo las clases interactúan entre sí en un contexto específico, a menudo en la relación de un caso de uso particular
- **Diagrama de Clases Persistencia:** Se enfoca en las clases que representan la persistencia de datos, como entidades de base datos y sus relaciones
- **Diagrama de Clases de Componentes:** Es la encargada de representar las clases que forman parte de un componente de Software, mostrando cómo se agrupan y su relación
- **Diagrama de Clases de Sistema:** Se encarga de proporcionar una vista general de las clases que componen un sistema completo, incluyendo sus interacciones y dependencias
- **Diagrama de Clases de Subclases:** En esta clase de diagrama se utiliza para mostrar la jerarquía de clases y subclases, destacando la herencia y las relaciones especializadas

IDENTIFICACIÓN DE CLASES PRINCIPALES

- **Estudiante:** Representa a los alumnos
 - **Atributos:** ID, Nombres, Apellidos, Fecha de Nacimiento, Tarjeta de Identidad o Cédula, Ciudad, País, Celular, Dirección, E-mail
- **Curso:** Asignatura que se imparte
 - **Atributos:** ID, Nombre, Descripción, Año
- **Profesor:** Persona encargada de enseñar
 - **Atributos:** ID, Nombres, Apellidos, Fecha de Nacimiento, Cédula, Ciudad, País, Celular, Dirección, E-mail
- **Asignación:** Relación entre estudiantes, cursos y notas
 - **Atributos:** ID, Clase, Estudiante, Nota
- **Nota:** Calificación que recibe un estudiante o estudiantes en un curso
 - **Atributos:** ID, Estudiante, Curso, Nota, Comentarios
- **Matricula**
 - **Atributos:** ID, Estudiante, Curso, Año, Estado (Activo/Inactivo)

RELACIONES ENTRE CLASES

1. Estudiante – Matricula

- Un estudiante puede estar matriculados en varios cursos
- Una matrícula pertenece a un solo estudiante

2. Curso – Matricula

- Un curso puede tener muchos estudiantes matriculados
- Una matrícula corresponde a un solo curso

3. Profesor – Clase

- Un profesor puede enseñar varias clases
- Una clase es impartida por un solo profesor

4. Curso -Clase

- Un curso puede tener varias clases
- Una clase corresponde a un solo curso

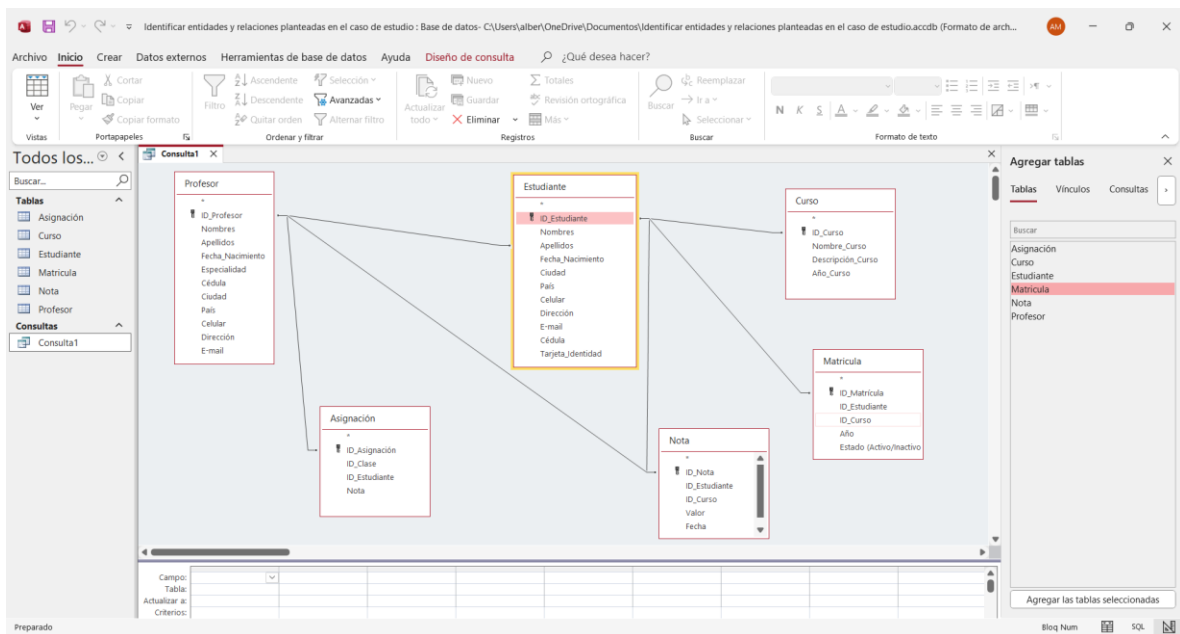
5. Estudiante – Asignación

- Un estudiante puede tener varias asignaciones en diferentes clases
- Una asignación corresponde a un solo estudiante

DIAGRAMA DE CLASES

El diagrama de clases representaría las clases identificadas y sus relaciones

Aquí tienes un esquema básico del diagrama de clases:



DESCRIPCIÓN DE INTERACCIONES

Es útil añadir descripciones de cómo interactúan las clases. Por ejemplo:

- Un estudiante puede ser matriculado en varios cursos o técnicos, lo que implica que sus notas se registran por curso
- Los profesores pueden acceder a las notas de los estudiantes en los cursos o técnicos que imparten

NOTAS ADICIONALES

- Puedes usar herramientas como UML (Lenguaje Modelado Unificado) para así crear los diagramas más visuales
- Considerar la posibilidad de añadir clases auxiliares como Administrativo para gestionar los cursos y las matrículas

CONCLUSIÓN

La conclusión de los diagramas de clases en un proyecto de software es fundamental para entender la estructura y el diseño del sistema. A continuación, se presentan algunos puntos clave que pueden incluirse en la conclusión:

- 1. Estructura del Sistema:** Los diagramas de clases proporcionan una representación visual de las clases, sus atributos y métodos, así como las relaciones entre ellas. Esto ayuda a comprender cómo se organiza el sistema y cómo interactúan los diferentes componentes
- 2. Identificación de Clases Clave:** A través del análisis de los diagramas, se pueden identificar las clases más importantes y sus responsabilidades. Esto es crucial para asegurar que el diseño sea coherente y que cada clase tenga un propósito claro
- 3. Relaciones y Dependencias:** Los diagramas de clases muestran las relaciones entre las diferentes clases, como la herencia, la asociación y la composición. Esto permite identificar dependencias y posibles áreas de mejora en el diseño, así como facilitar la implementación y el mantenimiento del software
- 4. Facilitación de la Comunicación:** Los diagramas de clases sirven como una herramienta de comunicación efectiva entre los miembros del equipo de desarrollo, así como con los interesados. Proporcionan un lenguaje común que ayuda a alinear la visión del proyecto
- 5. Base para la Implementación:** Los diagramas de clases actúan como una guía para la implementación del código. Al tener una representación clara de las clases y sus interacciones, los desarrolladores pueden escribir código de manera más eficiente y con menos errores
- 6. Posibilidad de Reutilización:** Un buen diseño de clases puede facilitar la reutilización de componentes en futuros proyectos. Al

identificar clases que son genéricas o que pueden ser utilizadas en diferentes contextos, se puede ahorrar tiempo y esfuerzo en el desarrollo

7. **Mejora Continua:** Finalmente, los diagramas de clases no son estáticos. A medida que el proyecto avanza y se reciben comentarios, es importante revisarlos y actualizarlos para reflejar los cambios en los requisitos o en la arquitectura del sistema