

DIAGRAMA DE CLASES

William Steven Rodríguez Rodríguez
Ángel Eduardo Medina Rojas
Julian Alcides Montenegro
Eduardo Jose Gil

Servicio Nacional de Aprendizaje -SENA, Centro de Electricidad Electrónica y Telecomunicaciones
Técnico en Programación de Software - TPS, IV Trimestre
Bogotá, 21 de Julio de 2023

DIAGRAMA DE CLASES

- Que es una clase?
- Elementos
- POO
- Tipos de relaciones
- Dc UML
- Cardinalidad
- Ejemplo diagrama de clases.

QUE ES UNA CLASE?

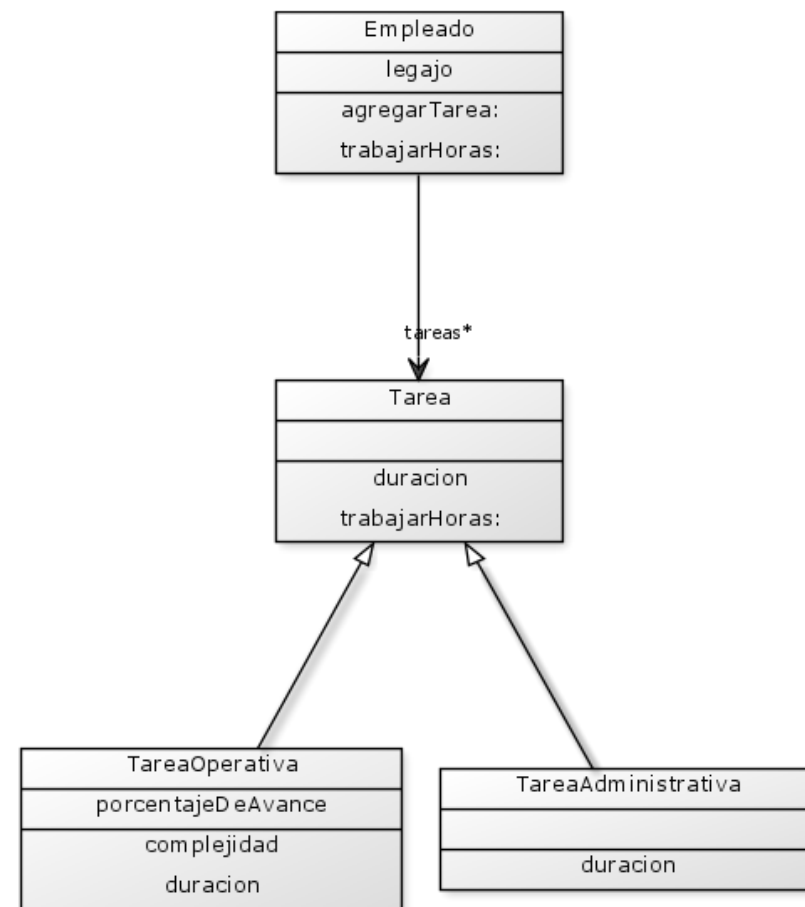


Es una plantilla para crear objetos en programación que define un conjunto de atributos (variables) y métodos (funciones) que pretenden ser utilizados, por los objetos creados a partir de ella.

Elementos hacen parte de la clase

En programación una clase está compuesta por atributos, métodos y constructores. Los atributos representan las características o propiedades del objeto, los métodos son acciones que puede realizar el objeto y los constructores son utilizados para crear instancias de la clase.

Una clase también puede incluir otros elementos como, herencias, interfaces, propiedades y eventos, dependiendo del lenguaje de programación utilizado. Estos elementos ayudan a definir el comportamiento y la estructura de los objetos creados a partir de la clase.



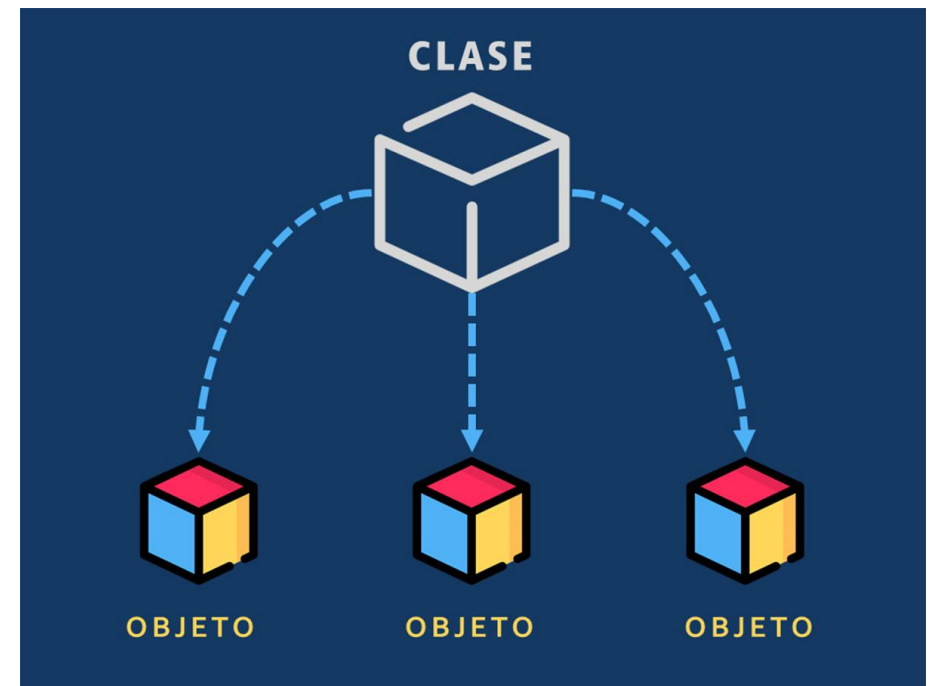
P00



es un paradigma de programación que se basa en la creación de objetos que interactúan entre sí para realizar tareas. En programación orientada a objetos P00, los objetos son instancias de una clase y tienen atributos y métodos que define su comportamiento y características. La P00 se centra en la organización y estructuración de código de una manera más modular y reusable, lo que facilita el mantenimiento y la escalabilidad del software.

Diferencia entre programación estructurada y programación orientada a objetos.

La programación estructurada se basa en la ejecución secuencial de instrucciones, utilizando estructuras como condicionales y bucles para controlar el flujo del programa. En cambio, la programación orientada a objetos se centra en la creación de objetos que interactúan entre sí, utilizando clases, atributos y métodos. La P00 permite una mayor modularidad, reutilización de código y modelado más cercano a la realidad.



¿Qué es una clase y cuáles son los elementos que la componen?

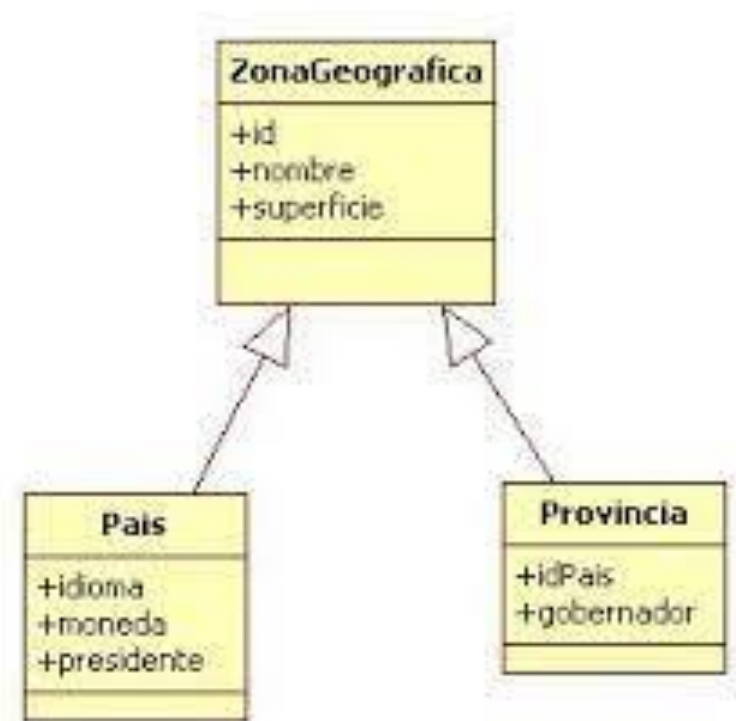
Una clase es una plantilla o estructura en programación que define las características y comportamiento de un objeto. Los elementos que componen una clase son los atributos (variables que almacenan datos), los métodos (funciones que definen el comportamiento) y los constructores (métodos especiales para inicializar objetos). También puede incluir otros elementos como herencia, interfaces, propiedades, eventos, entre otros, dependiendo del lenguaje de programación utilizado.

¿Qué es un objeto?

- Un objeto es una instancia específica de una clase en programación. Representa una entidad con características (atributos) y comportamiento (métodos) definidos por la clase a la que pertenece. Los objetos son utilizados para interactuar y realizar tareas en un programa.

¿Qué es herencia, de un ejemplo?

- La herencia en programación es un concepto que permite que una clase herede características y comportamientos de otra clase. La clase que hereda se llama clase derivada o subclase, y la clase de la cual se heredan las características se llama clase base o superclase. Por ejemplo, si tienes una clase "Animal" como superclase y una clase "Perro" como subclase, la clase "Perro" heredaría los atributos y métodos de la clase "Animal".



¿Qué se entiende por encapsulación?

- La encapsulación es un principio de la programación orientada a objetos que consiste en ocultar los detalles internos de una clase y proporcionar una interfaz pública para interactuar con ella. Permite controlar el acceso a los atributos y métodos de una clase, evitando modificaciones no deseadas y promoviendo la modularidad y seguridad del código.

¿Qué es polimorfismo?

- es un concepto de la programación orientada a objetos que permite que un objeto pueda tomar varias formas o comportarse de diferentes maneras según el contexto en el que se utilice. En otras palabras, diferentes objetos pueden responder a un mismo mensaje de diferentes maneras. Esto se logra mediante el uso de clases derivadas o subclasses que sobrescriben los métodos de la clase base o superclase. El polimorfismo permite escribir código más modular, flexible y fácilmente escalable.

ENCAPSULAMIENTO



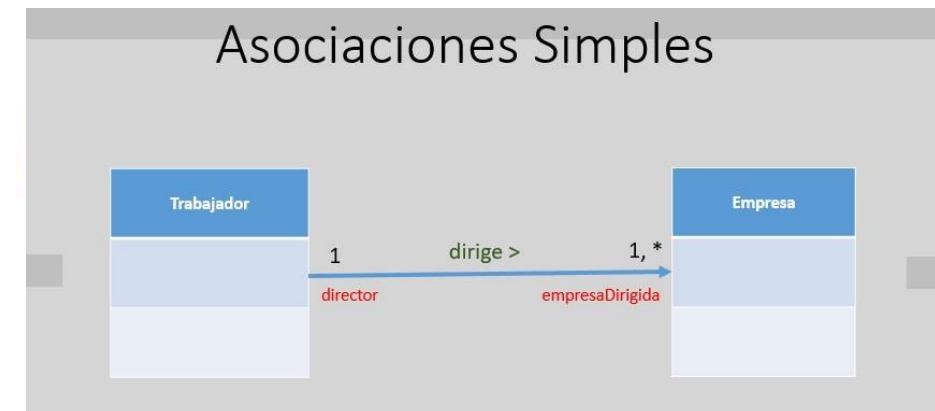
{
variable
variable
variable



Tipos de relaciones de un diagrama de clases UML



- En UML, los diferentes tipos de relaciones entre clases se representan mediante líneas con diferentes formas y direcciones. A continuación, se describen los tipos de relaciones más comunes:
- **Asociación:** Representa una relación entre dos clases donde una clase usa o interactúa con la otra. Se representa con una línea sólida que conecta las dos clases.
- **Agregación:** Representa una relación de "todo-parte" entre dos clases, donde una clase es parte de la otra clase. Se representa con una línea sólida que tiene un rombo en el extremo que apunta a la clase que contiene a la otra.
- **Composición:** Es similar a la agregación, pero en este caso la parte no puede existir sin el todo. Se representa con una línea sólida que tiene un rombo lleno en el extremo que apunta a la clase que contiene a la otra.
- **Dependencia:** Representa una relación donde una clase depende de otra clase para realizar alguna tarea o función. Se representa con una línea punteada que apunta desde la clase dependiente hacia la clase de la cual depende.
- **Herencia:** Representa una relación entre dos clases donde una clase (subclase) hereda los atributos y métodos de otra clase (superclase). Se representa con una línea sólida con una flecha que apunta desde la subclase hacia la superclase.



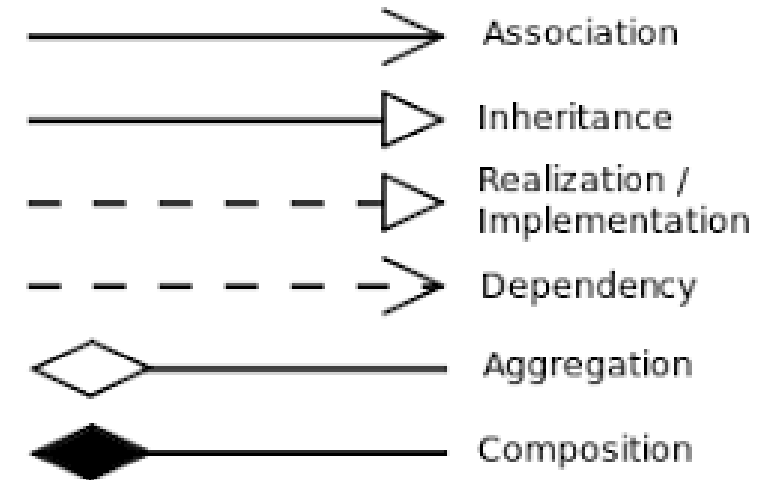
Tipos de relaciones de un diagrama de clases UML



•**Realización:** La implementación (realización) se utiliza principalmente para especificar la relación entre las interfaces y las clases de implementación .

Agregación: Es La relación entre el todo y la parte, y el todo y la parte se pueden separar. Las relaciones agregadas también representan la relación entre el todo y una parte de la clase, los objetos miembros son parte del objeto general, pero el objeto miembro puede existir independientemente del objeto general .

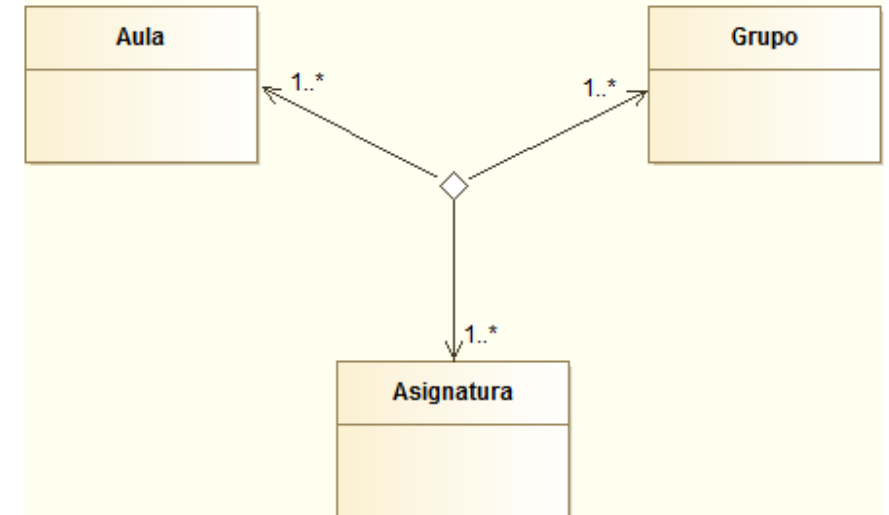
• En los seis tipos de relaciones, la estructura de código de los tres tipos de relaciones, como composición, agregación y asociación, es la misma que usar atributos para almacenar las referencias de otra clase. Por lo tanto, deben distinguirse por la relación entre los contenidos.



Cardinalidad de las relaciones y como se clasifican



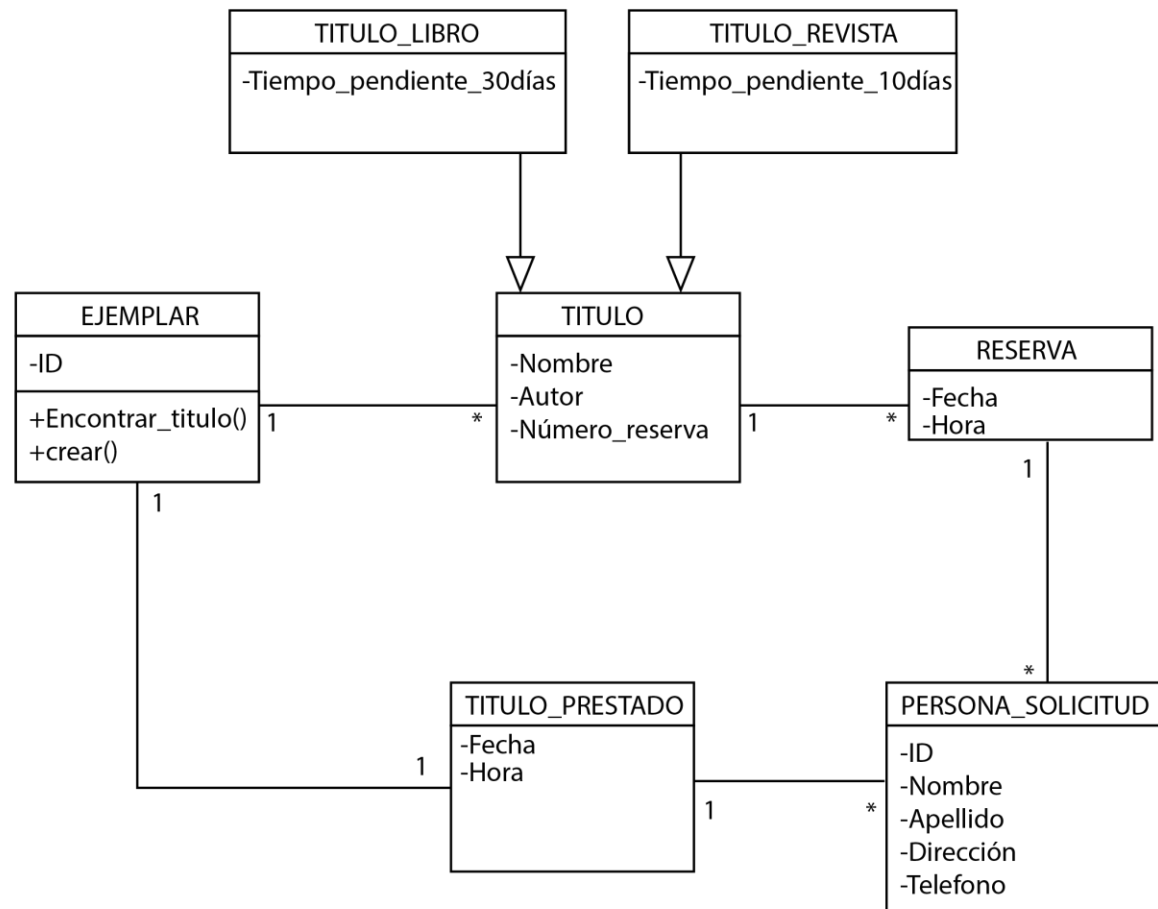
- La cardinalidad en las relaciones de un diagrama de clases UML indica la cantidad de instancias que pueden estar involucradas en una relación entre dos clases. Se clasifica en cuatro tipos:
- 1. Uno a uno (1:1): Indica que una instancia de una clase se relaciona con exactamente una instancia de otra clase, y viceversa.
- 2. Uno a muchos (1: N): Indica que una instancia de una clase se relaciona con varias instancias de otra clase, pero cada instancia de la segunda clase solo se relaciona con una instancia de la primera clase.
- Muchos a uno (N:1): Indica que varias instancias de una clase se relacionan con una única instancia de otra clase.
- Muchos a muchos (N: N): Indica que varias instancias de una clase se relacionan con varias instancias de otra clase.
- La cardinalidad se representa en un diagrama de clases UML mediante números o símbolos cerca de las líneas que conectan las clases, indicando la cantidad de instancias permitidas en cada lado de la relación.



Ejemplos diagrama de clases



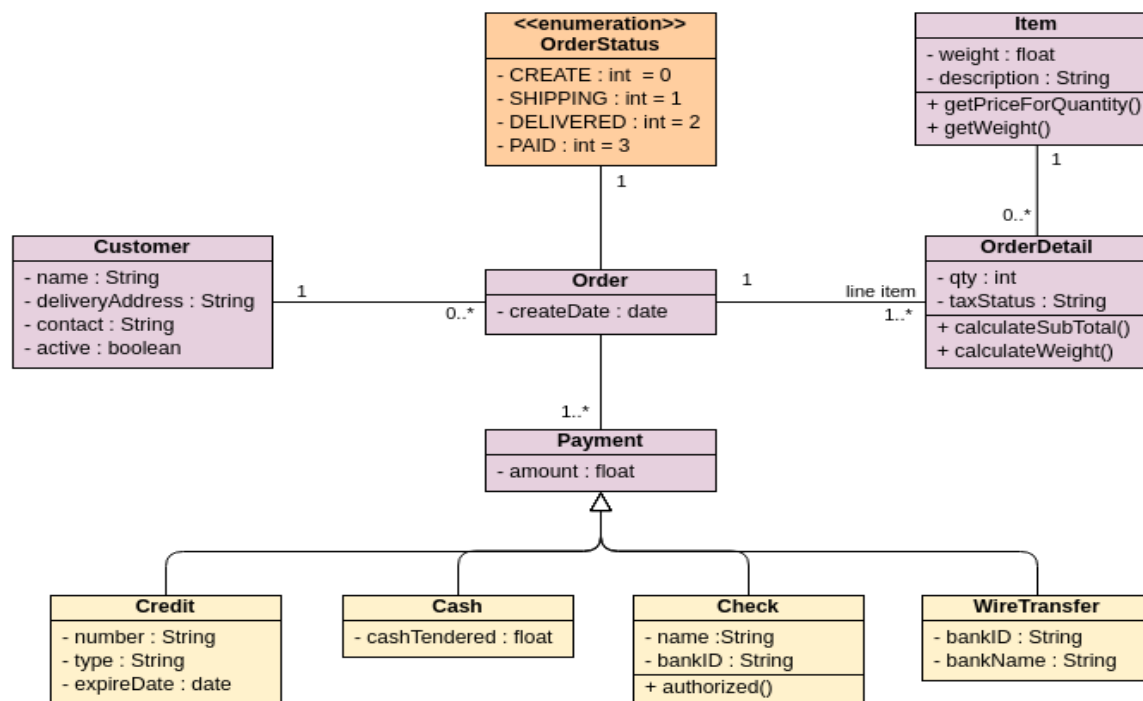
DIAGRAMA DE CLASES GESTIÓN DE BIBLIOTECA



Ejemplos diagrama de clases



El siguiente diagrama de clases modela un pedido de cliente de un catálogo minorista. La clase central es la **Orden**. Asociados a ella están el **Cliente** que realiza la compra y el **Pago**. Un **pago** es uno de cuatro tipos: **efectivo**, **cheque**, **crédito** o **transferencia bancaria**. El pedido contiene **OrderDetails** (artículos de línea), cada uno con su **artículo** asociado.





GRACIAS

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co