



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

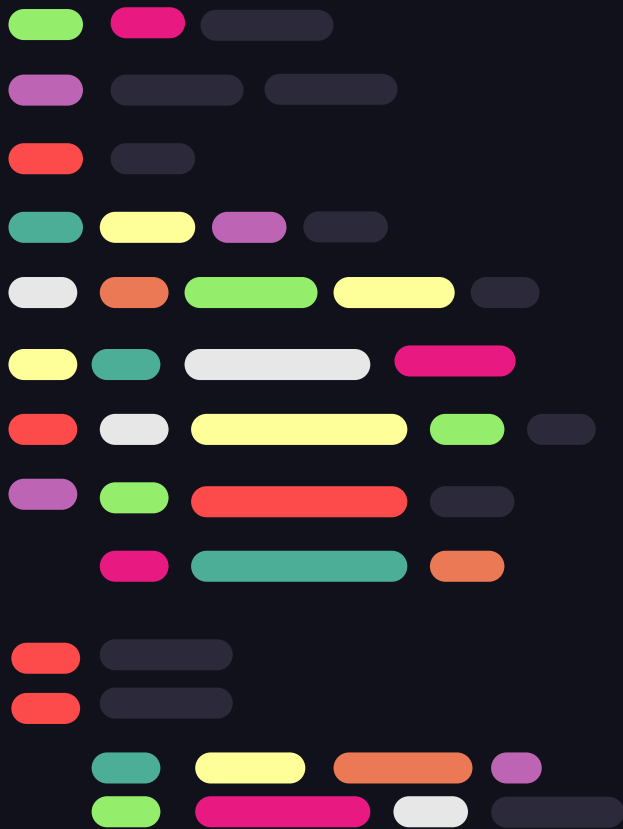
Viernes, 30/08/2024

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [B]

Josué Rodolfo Morales Castillo



{ ..



Clase 6- Agenda

- Foro No. 6
- Preguntas proyecto 1
- Recordatorio de 2 Commits Semanales
- Modalidad Calificación
- Principios básicos de UML (Diagrama de clases)
- Ejemplo práctico (JFreeChart)

} ..

{ Principios básicos de UML

(Diagrama de clases)



...

}

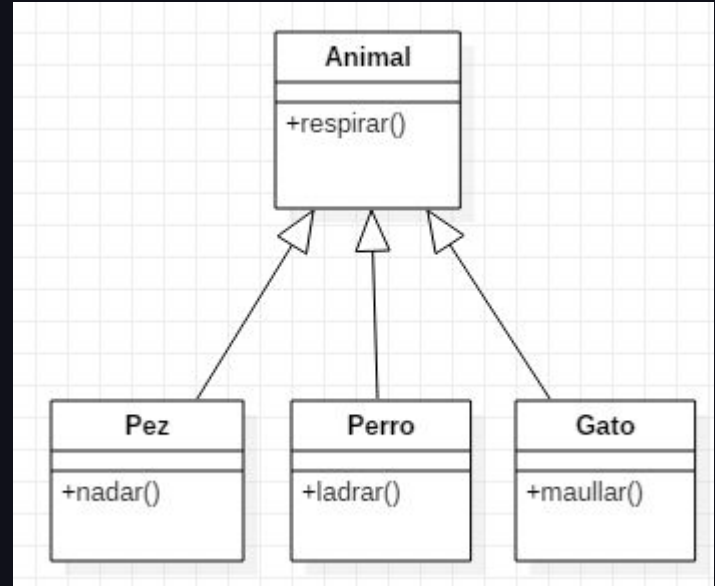


<https://youtu.be/Z0yLerU0g-Q?si=to4mg0Op7Bn5JYCM>

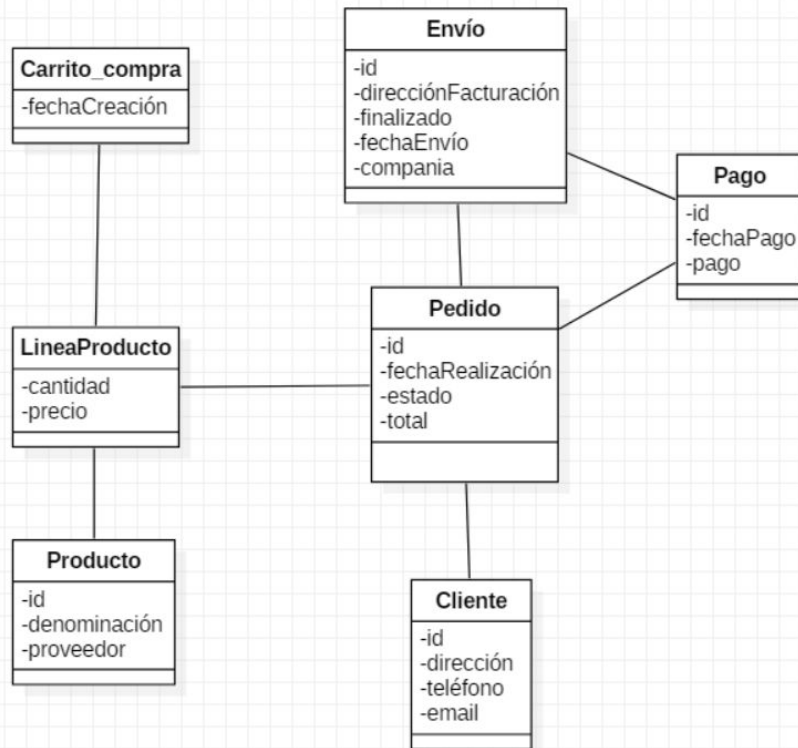


¿Qué es un diagrama de clases?

Es una representación gráfica que describe la estructura de un sistema orientado a objetos, delineando las clases, sus atributos y métodos, así como las relaciones entre ellas, como asociaciones, herencias y composiciones.



Utilidad



Se utiliza para el diseño de programas orientados a objetos. Permite visualizar la estructura de un sistema, sus componentes y cómo interactúan entre sí.

Componentes de Clases

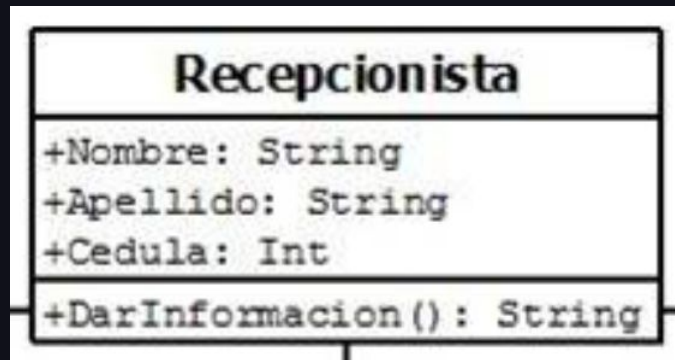
Clase: Es un conjunto de objetos con características y comportamientos comunes, representada como un rectángulo dividido en tres secciones: nombre, atributos y métodos.

Atributos: Características de la clase. Pueden tener diferentes niveles de visibilidad (públicos, privados, protegidos). Ejemplo:

`+nombre: String` indica un atributo público llamado nombre de tipo String.

Métodos: Comportamientos que la clase puede realizar, también tienen niveles de visibilidad. Ejemplo:

`+calcularPrecio(): double` método público llamado calcularPrecio que devuelve un valor de tipo double.



Ámbito de las propiedades

Público (+)

Accesible desde cualquier parte del sistema.

Privado (-)

Solo accesible dentro de la propia clase

Protegido (#)

Accesible dentro de la propia clase y sus subclases.

Paquete (~)

Accesible dentro del paquete que contiene la clase

Room
- x : int
- y : int
- height : double
- width : double
+ remove()
+ clone()

Window
- opacity : double
- isOpen : boolean
+ close() : void
+ open() : void
+ isOpen() : boolean

Relaciones

Asociación

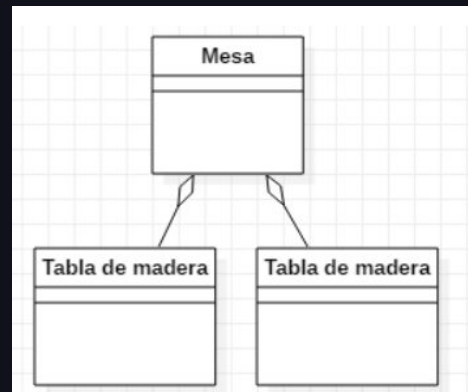
Indica que una instancia de una clase está relacionada con una instancia de otra clase.



———— Asociación

Agregación

Es una relación "todo-parte" entre clases. Indica que una clase (el todo) está compuesta por otras clases (las partes).

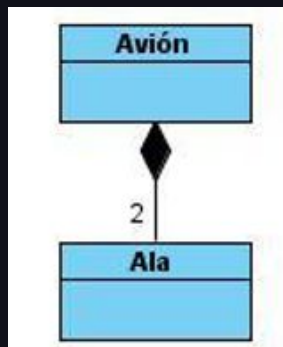


◊ ————— Agregación

Relaciones

Composicion

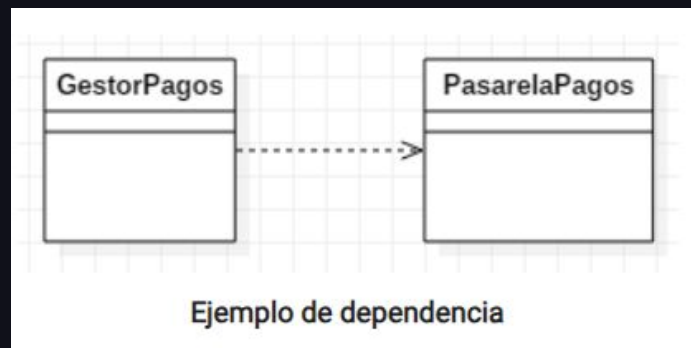
Es una forma más fuerte de agregación, indicando que una clase es parte de otra y no puede existir independientemente.



→ Composición

Dependencia

Indica que un cambio en una clase puede afectar a otra. Representar que una clase requiere de otra para ofrecer sus funcionalidades



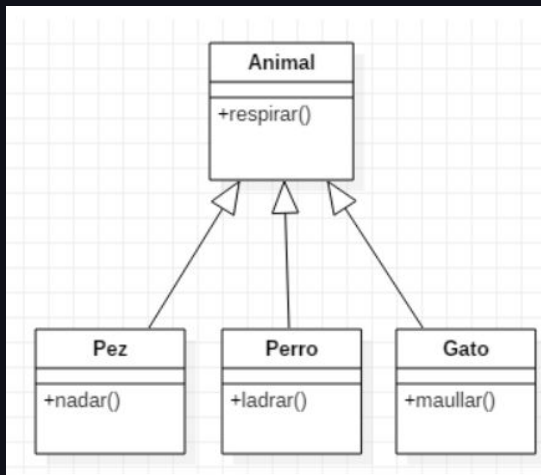
-----> Dependencia

Relaciones

Herencia

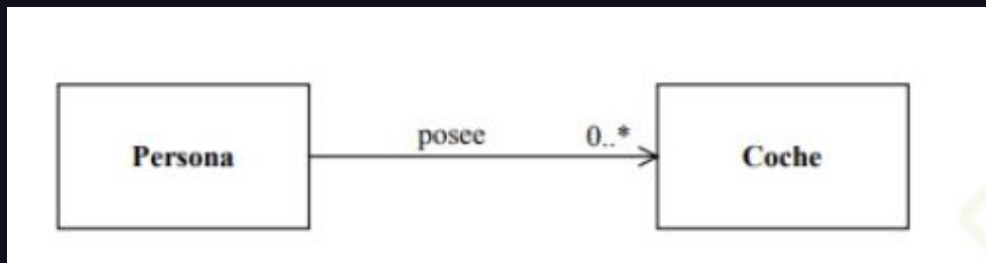
Representa una relación "es-un" entre clases. La subclase hereda atributos y métodos de la superclase.

→ Herencia



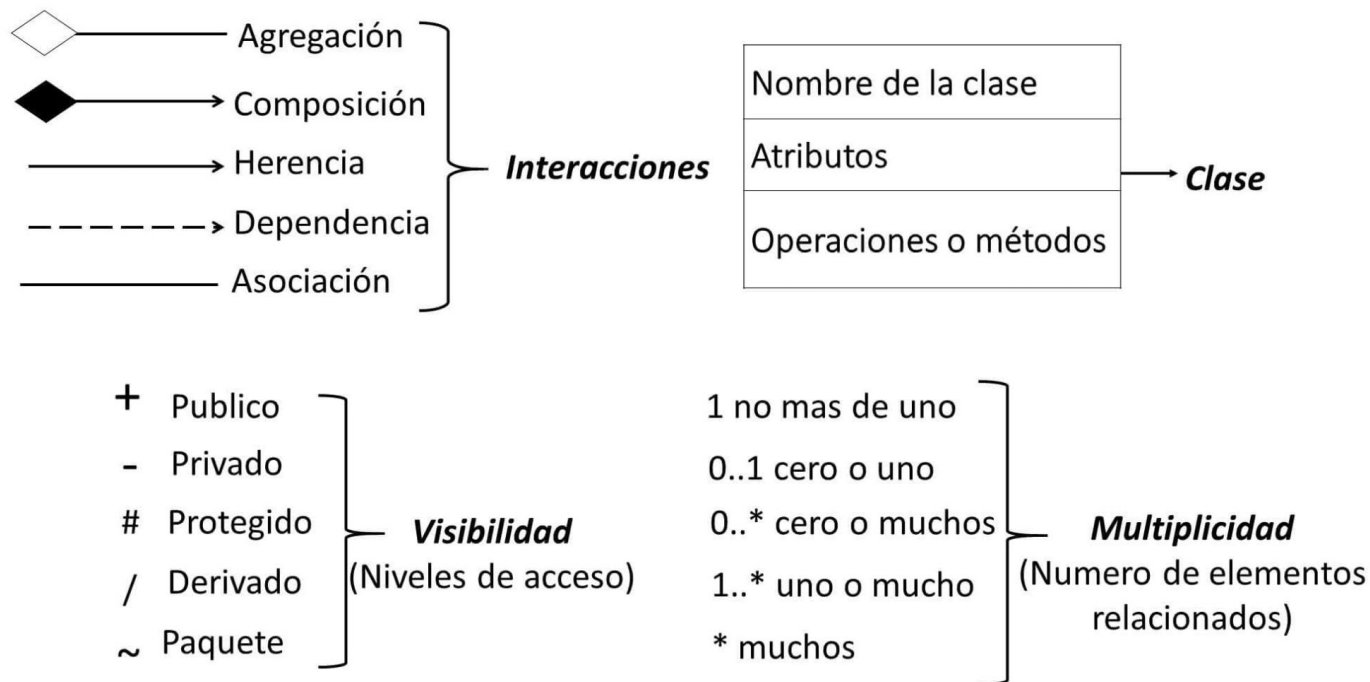
Multiplicidad

Especifica cuántas instancias de una clase pueden estar asociadas con una instancia de la otra clase.



Por ejemplo una multiplicidad de "0..*" para la relación entre "Persona" y "Coche" indica que un cliente puede tener ninguna o varias Coches.

Elementos y Símbolos

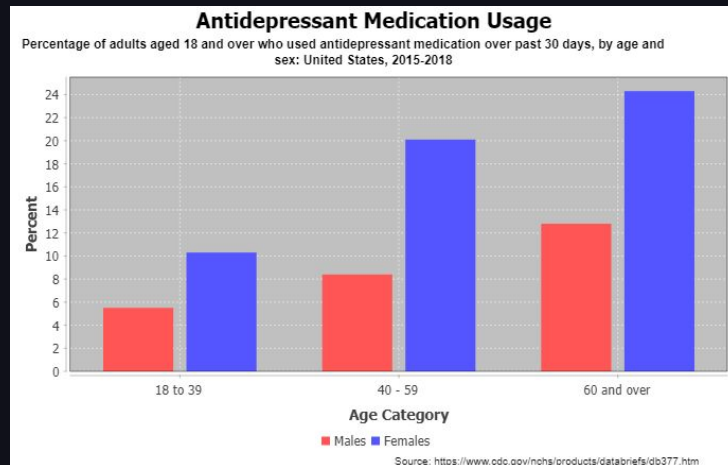


JFreeChart

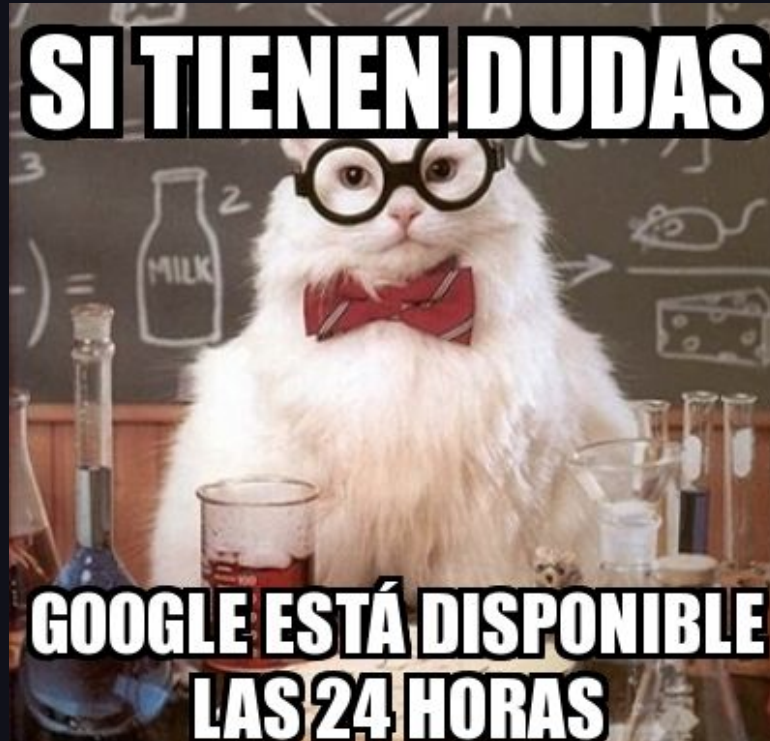
Es una biblioteca de código abierto en Java que permite crear gráficos de manera sencilla, como gráficos de líneas, barras, pasteles, entre otros.

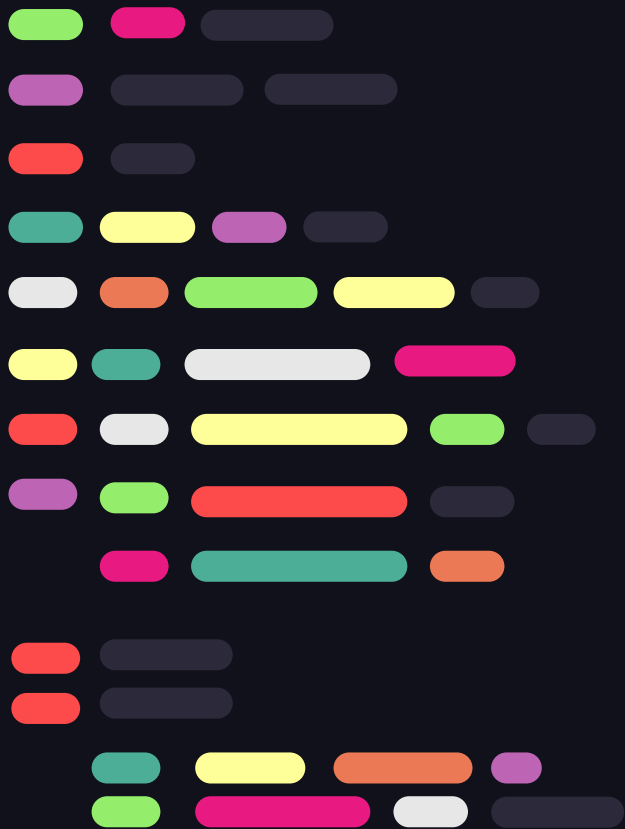
Link de Descarga:

<https://sourceforge.net/projects/jfreechart/>



¿Dudas?





Ejemplo

Descargar JFreeChart:

<https://sourceforge.net/projects/jfreechart/files/>

Look and Feel:

<http://dar10comyr.blogspot.com/2014/11/look-and-feel.html>

