

PREGRADO



UNIDAD 2 | SEMANA 5

PROGRAMACIÓN ORIENTADA A OBJETOS

CONCEPTOS BÁSICOS Y DIAGRAMAS DE CLASE

1ACC0201 | Programación Orientada a Objetos





Objetivo de la sesión

Al concluir la sesión, el estudiante reconoce los conceptos de clases y objetos en la elaboración de programas.





Clases y Objetos

Contenido:

- Programación Estructurada
- Programación Orientada a Objetos
 - Conceptos
 - Ventajas y Desventajas
 - Principios Fundamentales
 - Clases y Objetos.
 - Atributos, Métodos, Mensajes.





Programación Estructurada

- La programación estructurada define un orden específico.
- Separa el código de las estructuras de datos.
- En este enfoque de programación, se prioriza el código sobre las estructuras de datos.



Algoritmos + Estructuras de Datos = Programas



Programación Orientada a Objetos

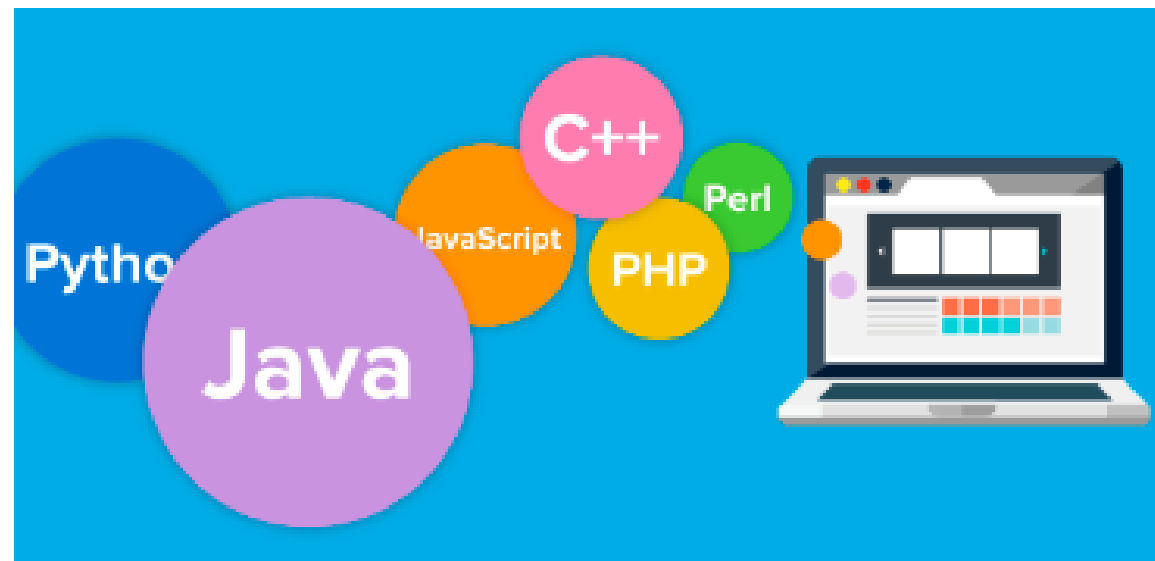
- La Programación Orientada a Objetos (POO) fue desarrollada por aquellos que entendían un entorno lleno de objetos que se relacionan entre sí según su propia esencia.
- La POO es un método para organizar el código que resulta particularmente eficaz en la estructuración y reutilización de este.





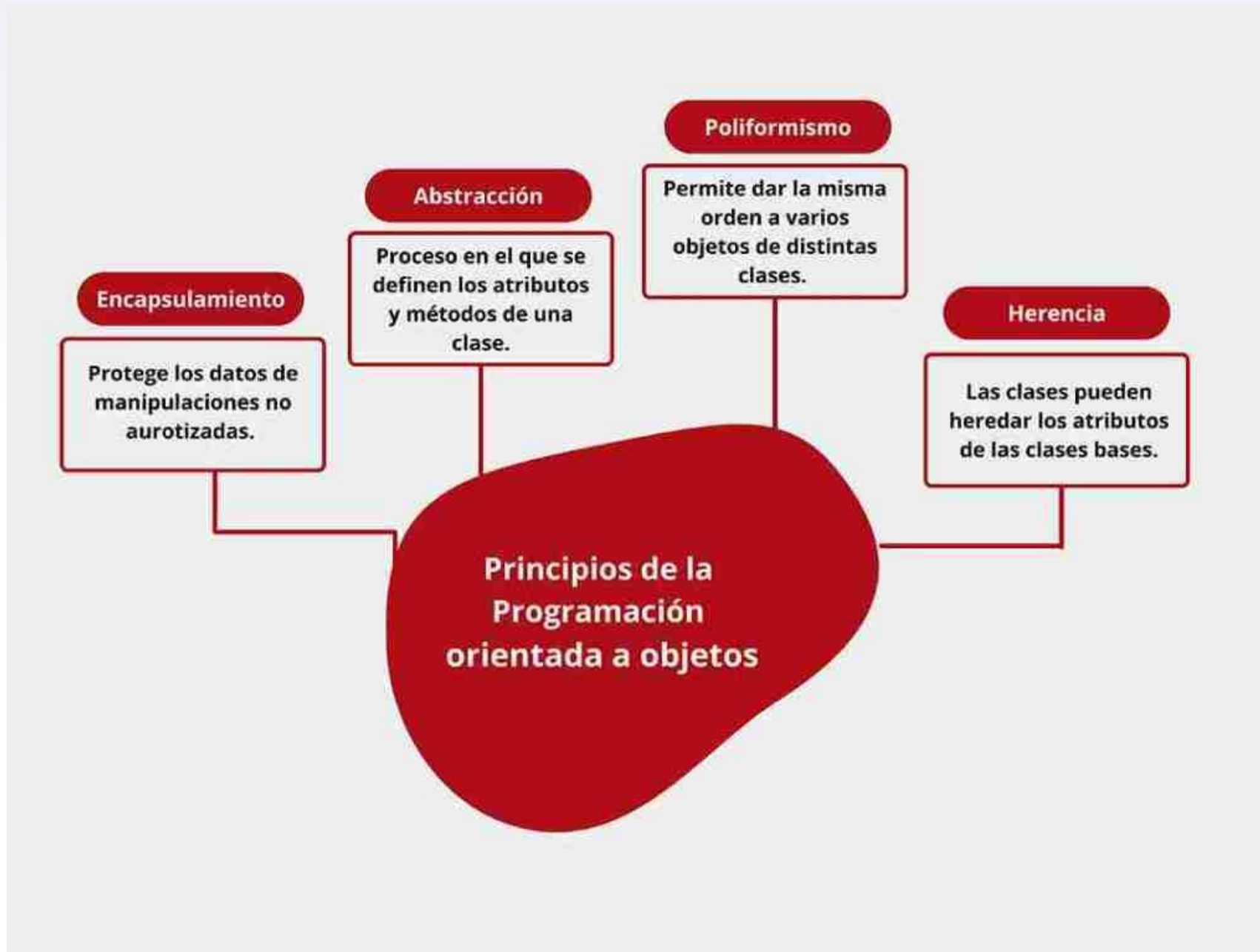
Distinción entre Programación Orientada a Objetos y Programación Estructurada

- La programación estructurada se centra en el uso de procedimientos o funciones, priorizando este enfoque sobre el de estructuras, utilizando principalmente funciones que procesan datos.
- Por otro lado, en la programación orientada a objetos, se comienza definiendo los objetos o estructuras, y luego se procede a invocar sus métodos.





Principios Fundamentales de la POO





Principios Fundamentales de la POO definición

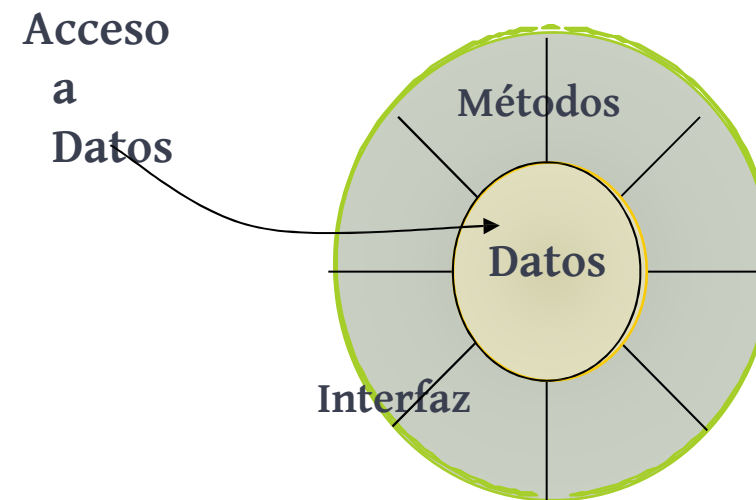
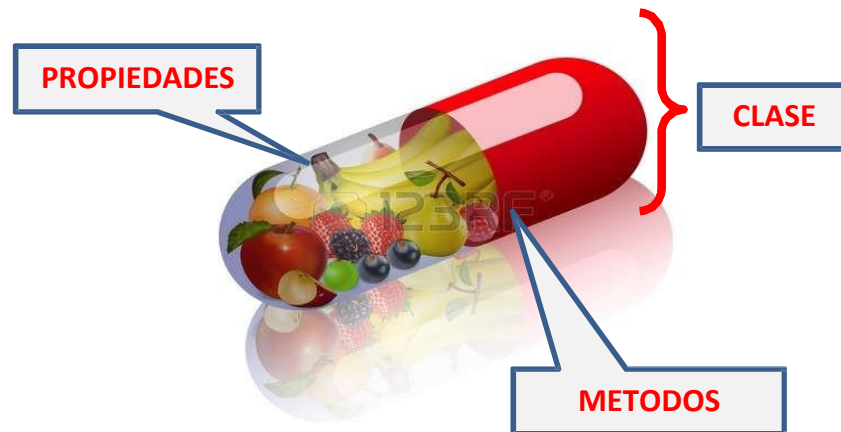
Principio de Abstracción. Cuando se desarrolla con OOP, es necesario entender el problema y centrarse en identificar los objetos relevantes, las propiedades y métodos que necesitaremos para solucionarlo, esto ayuda a reducir la complejidad.





Principios Fundamentales de la POO definición

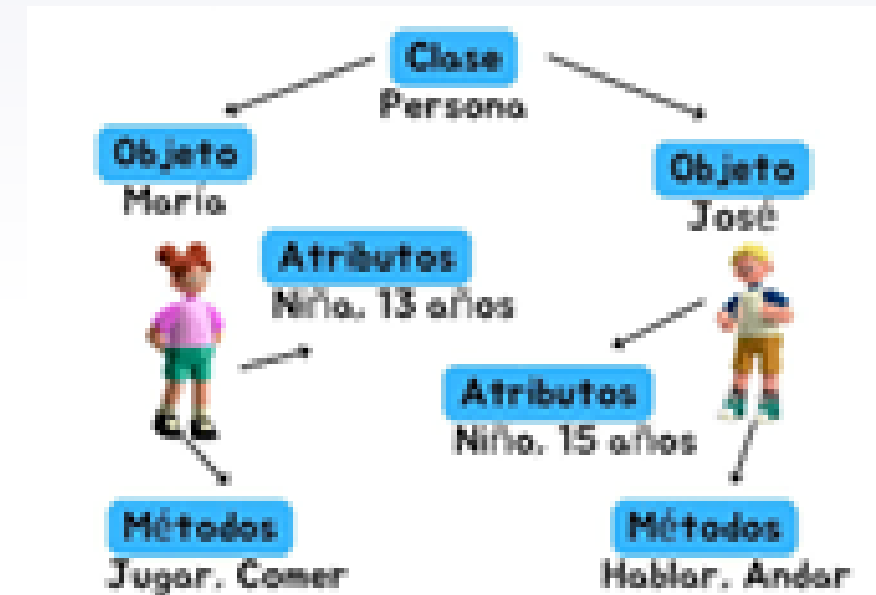
Principio de Encapsulación. Este principio hace referencia a la información almacenada en un objeto y la parte de esta información que puede ser visible para el exterior.





Principios Fundamentales de la POO definición

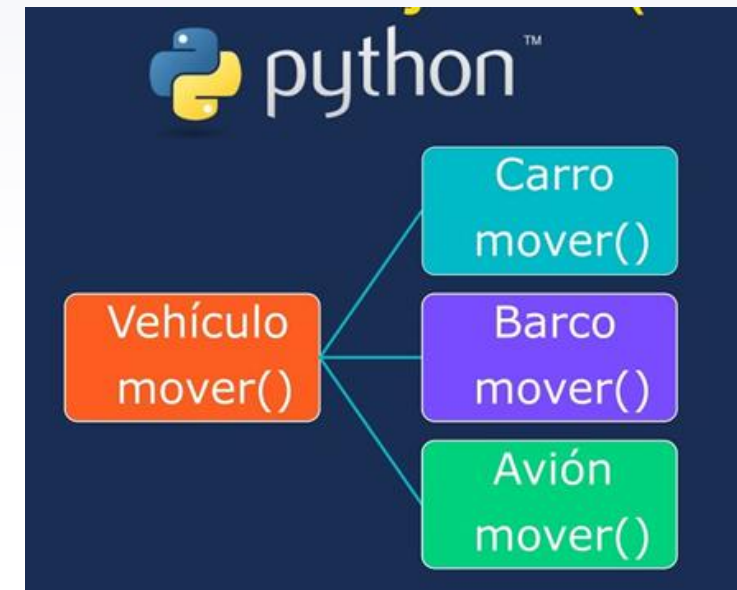
Principio de Herencia. Este principio se refiere a la situación en la que una clase adquiere características de otra. La clase de la que se heredan los atributos se denomina superclase o clase padre, mientras que la clase que recibe estas características se conoce como subclase.





Principios Fundamentales de la POO definición

Definición de Polimorfismo. Este principio permite que un método muestre distintos comportamientos según la clase que lo esté utilizando. Sin el uso del polimorfismo mediante la herencia, sería necesario desarrollar una función independiente para cada clase.





Principio número 1: Abstracción



Principio #1: Abstracción

- Este es un concepto fundamental que se refiere a la **capacidad de simplificar la complejidad** del sistema al ocultar los detalles internos de implementación y exponer solo la funcionalidad esencial al usuario.
- Consiste en **representar las características esenciales** de un objeto, dejando de lado otras no esenciales.
- Definir una abstracción significa describir una entidad del mundo real, no importa lo compleja que sea y luego utilizarla en la descripción de un programa.
- En otras palabras, permite que los programadores trabajen con **conceptos y operaciones de alto nivel** sin preocuparse por los detalles internos.
- Una **Clase es una abstracción** de un grupo de objetos, que tienen la posibilidad de realizar una serie de operaciones.





Principio de Abstracción: Aplicación

CARACTERIZANDO EL OBJETO COCHE

- ▶ ¿Qué conozco del coche?, ¿Qué caracteriza un coche?

- Kilometraje → **Atributos**

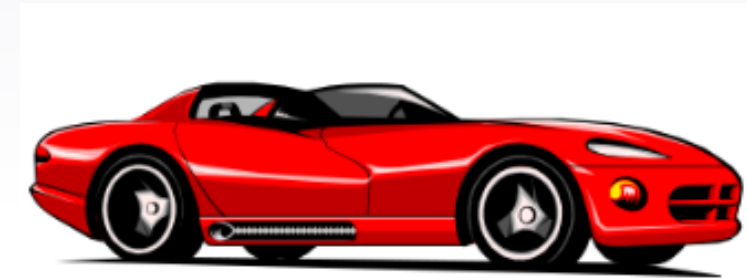
- ▶ ¿Qué cálculos se puede hacer al coche?

- Calcular la velocidad
 - Calcular el kilometraje

- ▶ ¿Qué otros cálculos u operaciones pueden hacerse?

- ▶ Reconocer el valor de la velocidad
 - ▶ Modificar el valor del kilometraje

} → **Métodos**



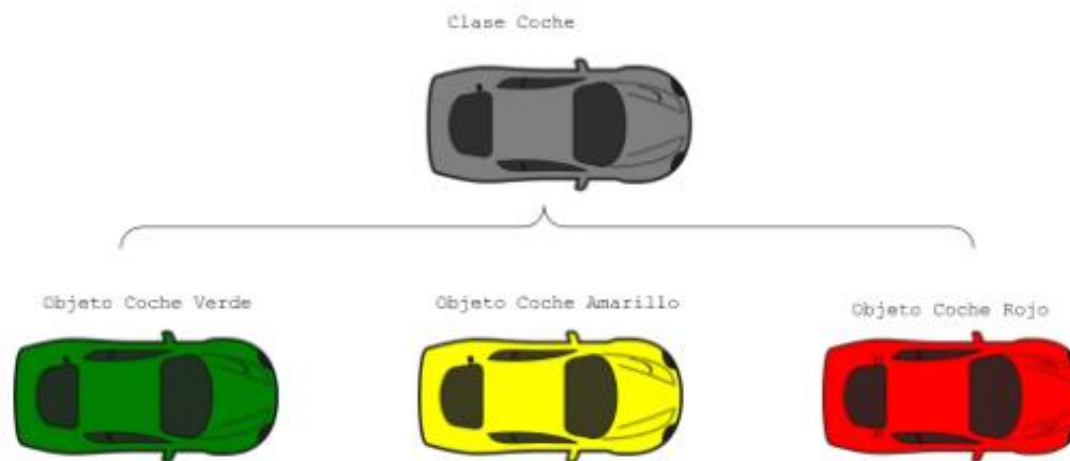


Clases y Objetos



Clase y Objetos

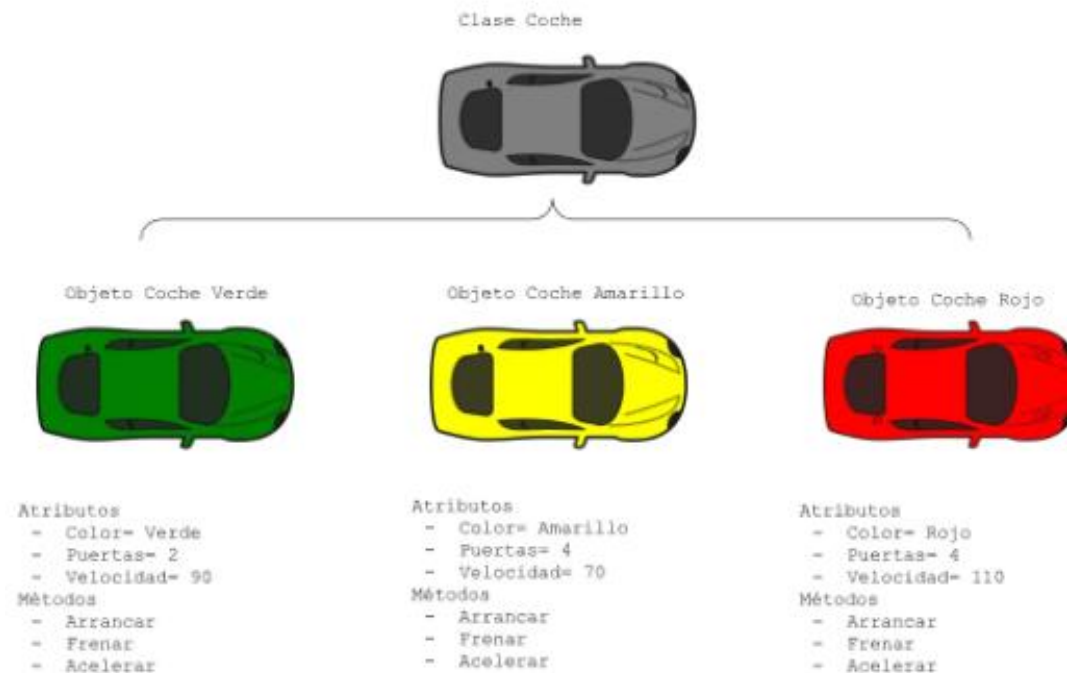
- Una **CLASE** representa un nivel superior de abstracción que se relaciona con un **grupo de objetos** que comparten las mismas características y comportamientos.





Clases y Objetos

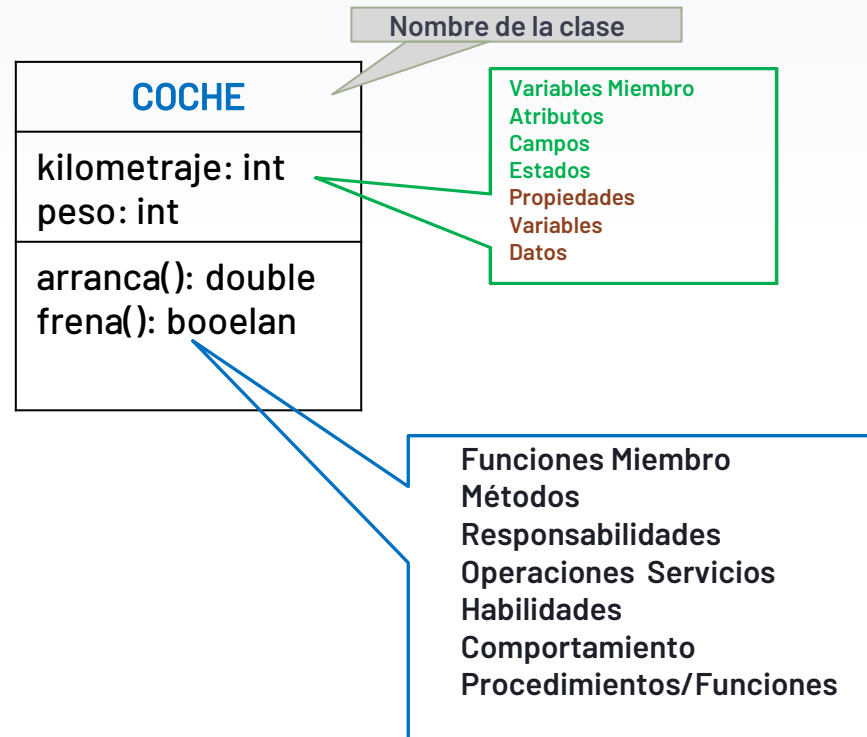
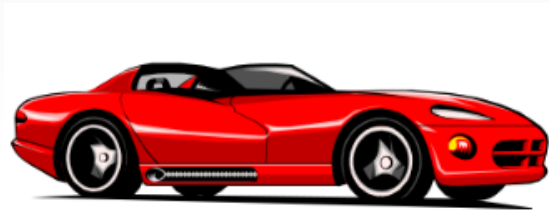
- La **CLASE** cuenta con atributos y métodos.
- Los atributos son variables que están vinculadas a un tipo de dato específico.
- Una **CLASE** se *representa/visualiza* gráficamente a través de un *Diagrama* que ilustra la clase.





Clases y Objetos

- Un diagrama de CLASE se compone de tres secciones que representan la clase, sus propiedades y sus métodos.



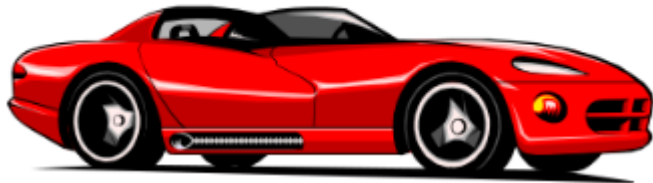
Clase
Propiedades
Métodos

Otros términos utilizados
para referirse a cada
sección del diagrama



Clases y Objetos

- Una vez que una CLASE ha sido formalizada, puede comenzar su proceso de codificación. Hay una relación directa entre el diagrama y el código que implementa la clase.



```
class Coche:
    def __init__(self, color, modelo, ano):
        self.color = color
        self.modelo = modelo
        self.ano = ano

    def mostrar_informacion(self):
        print(f"Coche: {self.modelo}, Color: {self.color}, Año: {self.ano}")

    def conducir(self):
        print(f"El coche {self.modelo} está en movimiento.")

# Crear una instancia de la clase Coche
mi_coche = Coche(color="Rojo", modelo="Toyota Corolla", ano=2022)

# Llamar a los métodos de la instancia
mi_coche.mostrar_informacion()
mi_coche.conducir()
```



Clases y Objetos

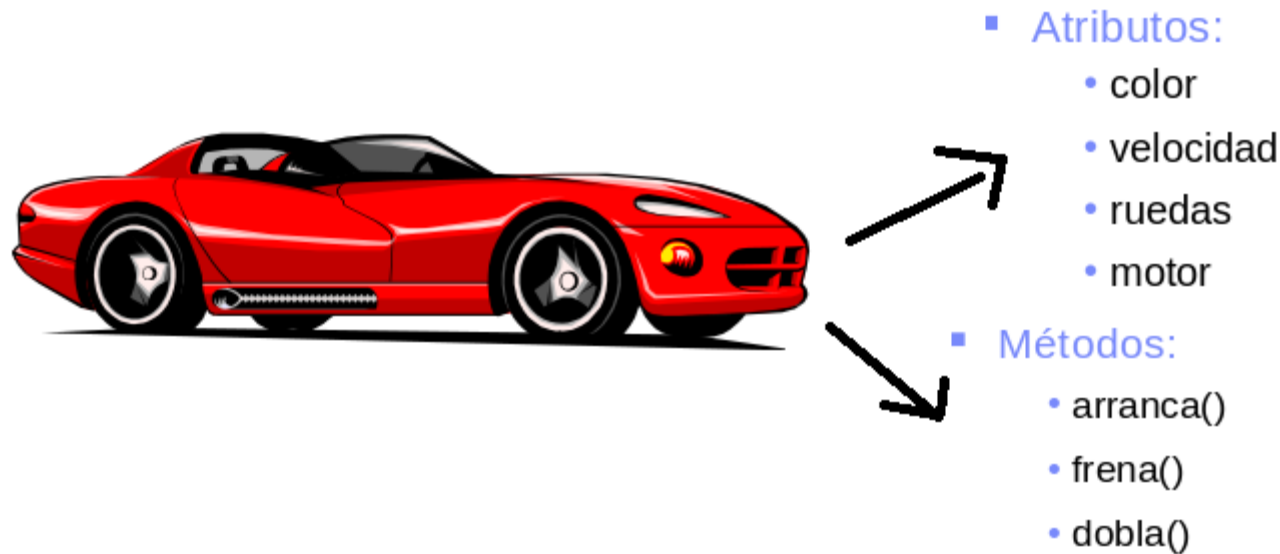
- Un **OBJETO** representa una instancia de un concepto del mundo real que puede ser representado a través de una clase.
- Un **OBJETO** tiene las siguientes características:
 - **Tiempo de Vida:** Se refiere a la duración de un objeto dentro de un programa.
 - **Estado:** Se determina por sus atributos (el valor de cada uno de ellos).
 - **Comportamiento:** Cada objeto debe ofrecer una interfaz que esté definida por sus métodos.





Clases y Objetos

- ▶ Un **OBJETO** posee un identificador inmutable, así como comportamientos y atributos que son característicos de su clase. Sin embargo, cada objeto tiene valores únicos para cada uno de sus atributos.





Atributos y Estado

- Los **ATRIBUTOS** representan las características o propiedades que exhiben todos los objetos de una clase. Se emplean para describir, identificar o comunicar el estado.
- El **ESTADO** de un objeto o clase se define por los valores de sus **ATRIBUTOS** en un momento específico.



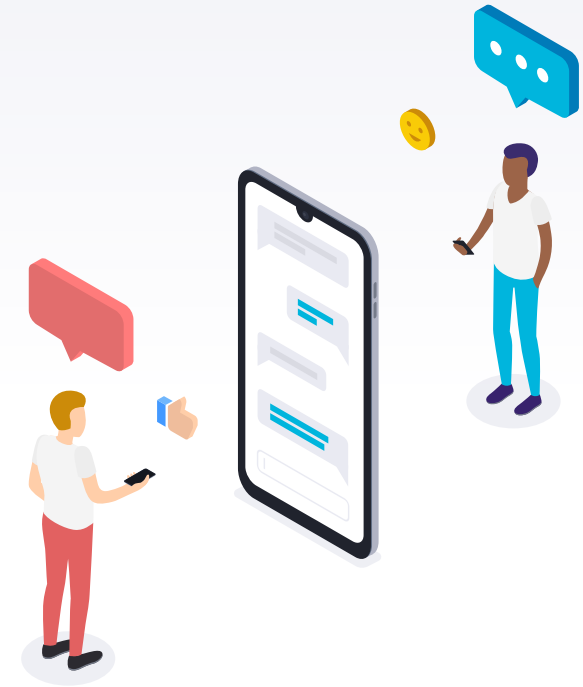
Métodos

- Son procedimientos o funciones.
- Representan las acciones que un objeto de una clase debe llevar a cabo.
- Estas son las responsabilidades del objeto, que abarcan tanto su comportamiento como el acceso a sus atributos y los de las clases.
- Se trata de los algoritmos que manipulan los datos o atributos de un objeto y se encuentran dentro de un bloque de código.

Mensajes



- ▶ En el ámbito de la programación orientada a objetos en Python, **los mensajes** son la forma en que los objetos se comunican entre sí mediante métodos. Es decir, cuando un objeto (o instancia de una clase) requiere interactuar con otro objeto, lo logra enviándole un mensaje, que normalmente implica invocar un método del objeto que recibe el mensaje.





Mensajes

- En la programación orientada a objetos, un **mensaje** se define como una solicitud que un objeto envía a otro para que ejecute una acción o proporcione información. Esta solicitud se lleva a cabo al invocar un método en el objeto que recibe el mensaje.
- En este caso, al invocar `persona.saludar()`, estamos enviando un mensaje al objeto `persona` para que lleve a cabo el método `saludar`.

```
class Persona:
    def saludar(self):
        print("¡Hola!")

persona = Persona()
persona.saludar() # Aquí 'persona' envía un mensaje 'saludar' al objeto Persona.
```



UML y diagramas de clases





Conceptos básicos de UML



UML

- El Lenguaje de Modelado Unificado (UML – Unified Modeling Language) es un lenguaje gráfico de uso general.
- Se considera un estándar en la industria para visualizar, especificar y documentar todos los componentes del Desarrollo de Sistemas mediante diagramas y textos de apoyo.
- Facilita, entre otras funciones, la especificación de la estructura y/o el comportamiento de un producto de software.





UML

Elementos y símbolos en los diagramas de clases UML

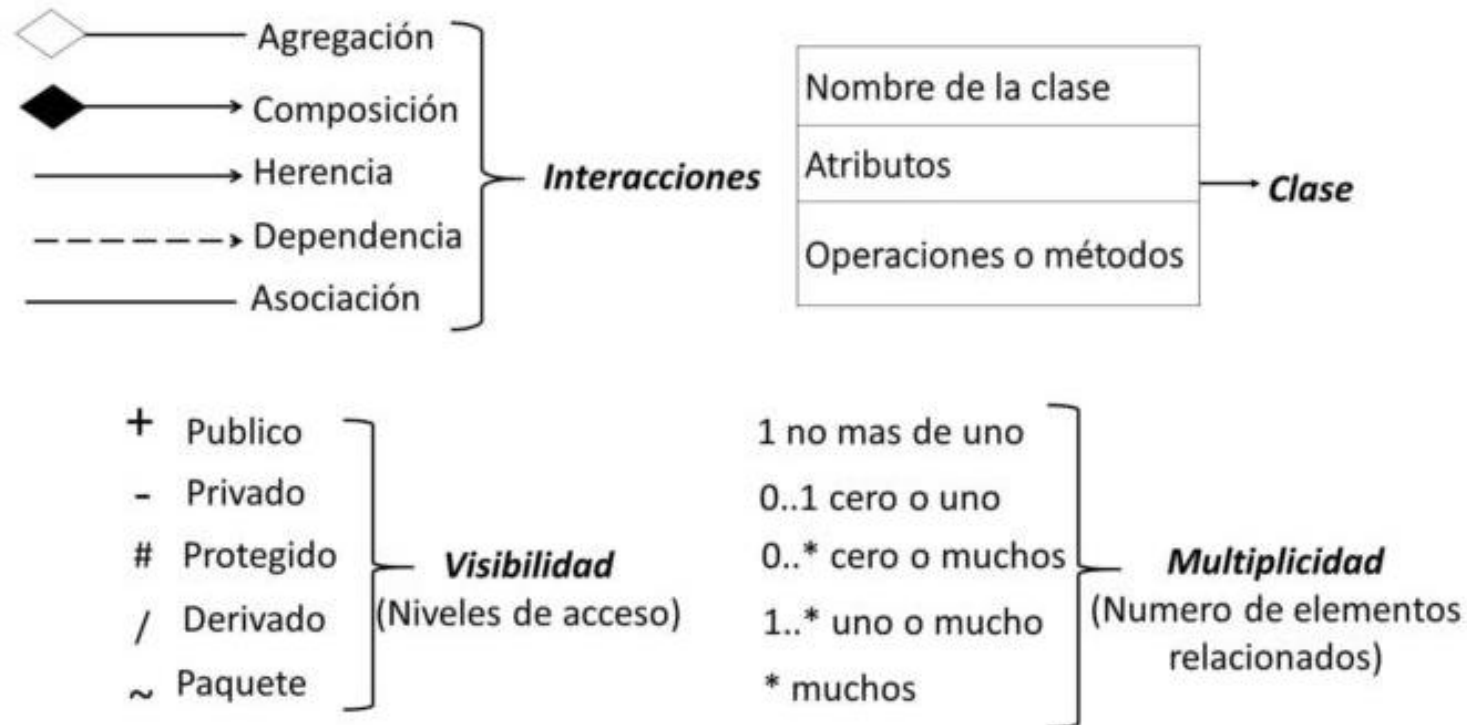




Diagrama de Clases

- Explica los diversos tipos de objetos que forman parte de un sistema y las relaciones estáticas que pueden establecerse entre ellos.
- Muestra, de forma gráfica, las características de cada objeto, destacando el nombre de la clase, los atributos asociados a la clase y los métodos que gestionan sus propiedades.

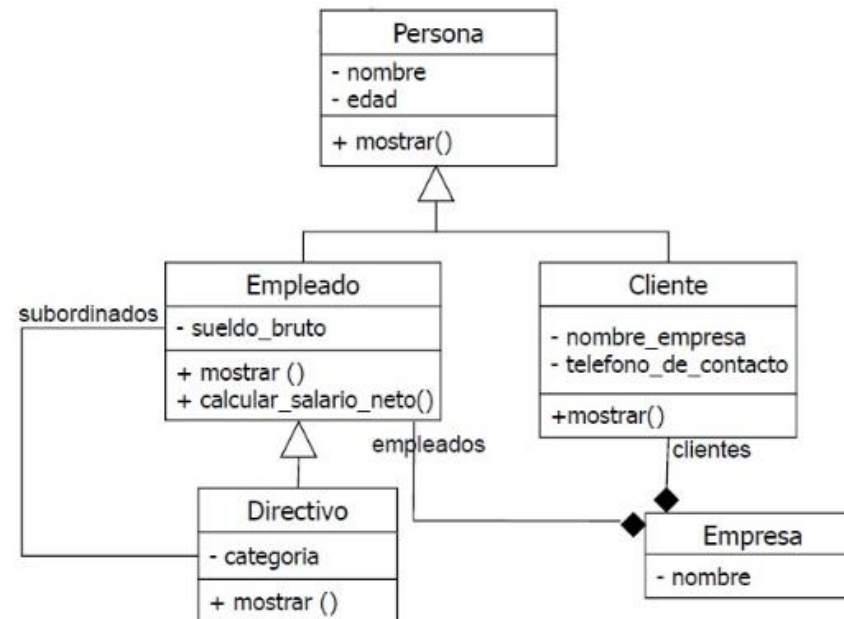




Diagrama de Clases

- Las clases son la manera de agrupar objetos. Se ilustran mediante un rectángulo dividido en tres secciones:
- Nombre de la clase
- Sección para los atributos
- Sección para los métodos

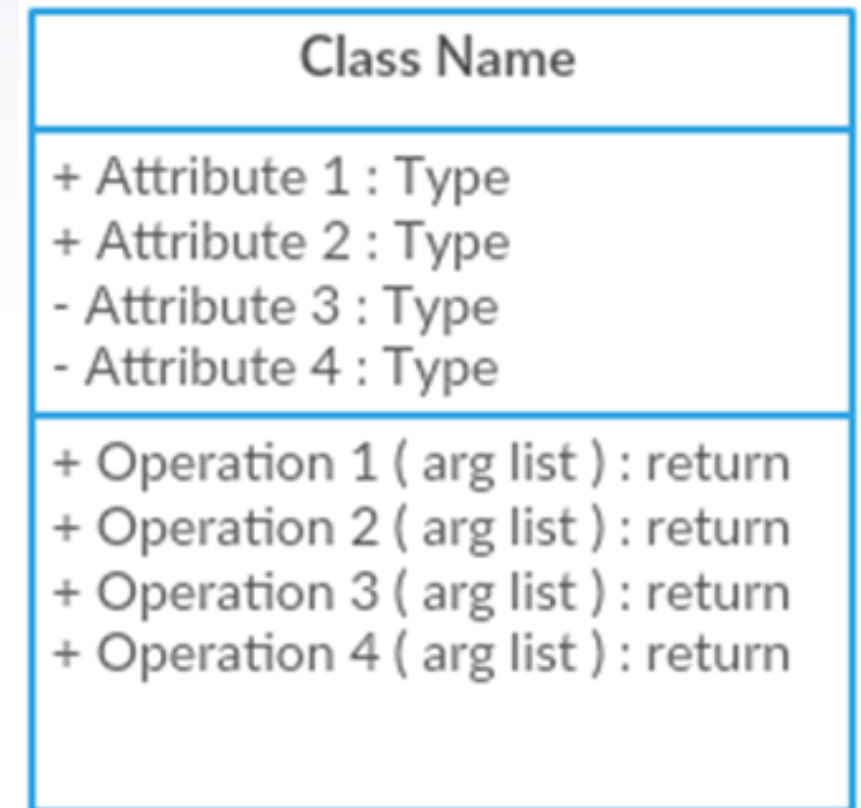










Diagrama de Clases

- Un software se compone de diversos objetos de diferentes clases.
- Las clases coexisten debido a las relaciones que mantienen entre sí.
- La relevancia del diagrama radica en ilustrar las interacciones que existen entre las clases que forman parte del sistema.

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/ Implementation	
Dependency	
Aggregation	
Composition	



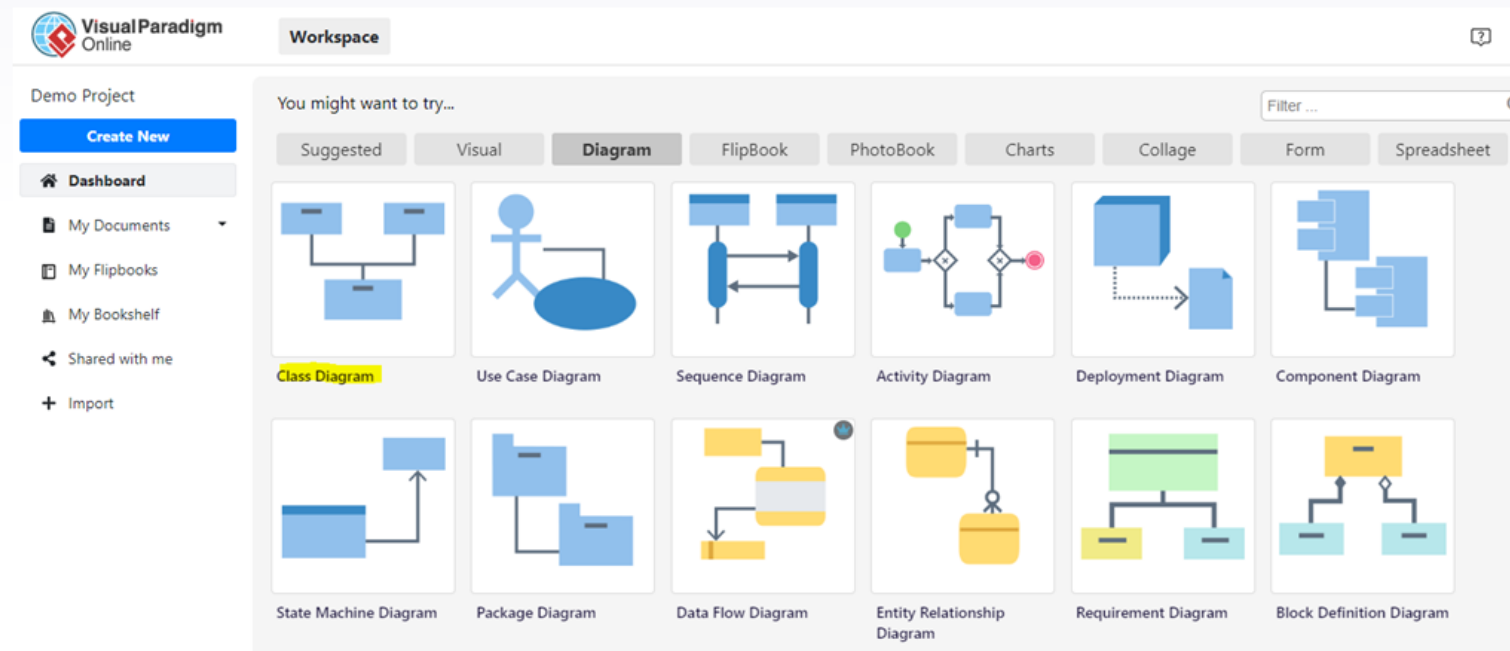
Procedimientos para crear un Diagrama de Clases

1. Reconozca las clases y asigne un nombre distintivo a cada una.
2. Determine las interacciones entre las clases.
3. Emplee los conectores apropiados para ilustrar la relación.
4. Incluya los atributos y métodos correspondientes en cada clase.



Creación de Diagramas Online

<https://online.visual-paradigm.com/drive/#diagramlist:proj=0&dashboard>





Creación de Diagramas

- Otras opciones gratuitas para crear diagramas de clase
 - Lucidchart (plataforma en línea)
 - StarUML (software descargable)

PREGRADO

Ingeniería de Sistemas de Información

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería



Universidad Peruana
de Ciencias Aplicadas

Prolongación Primavera 2390,
Monterrico, Santiago de Surco

Lima 33 - Perú
T 511 313 3333

<https://www.upc.edu.pe>

exígete, innova

UPC

