



Clase presencial



Sesion5A: Programación Orientada a Objetos

Instructor: David Paúl Porras Córdova

@iscodem



Contenido de Agenda

- Clases y objetos
 - ▣ Clases
 - ▣ Objetos
 - ▣ Atributos, métodos y constantes
 - ▣ Constructores
 - ▣ Interfaces y diferencia con clases
- Principios básicos de la POO
 - ▣ Encapsulamiento
 - ▣ Herencia
 - ▣ Polimorfismo



Clases y objetos

- Clases
- Objetos
- Atributos, métodos y constantes
- Constructores
- Interfaces y diferencia con clases

1. Clases

Una Clase es una plantilla o prototipo que define los atributos y métodos de un objeto.

ClaseCelular
-color -peso -ancho -alto -grosor
+enviaMensaje() +recibeMensaje() +seleccionDeTono() +seleccionDeFondo() +llamada()



2. Objetos

Un objeto es una unidad de código con atributos y métodos predefinidos.





Contenido de Agenda

- Programación estructurada
- Programación Modular
 - ▣ Funciones
 - ▣ Procedimientos
 - ▣ Métodos
- Buenas prácticas de programación



Contenido de Agenda

Paradigma: Ejemplo ilustrativo, enunciado modelo

- Programación imperativa
- Programación funcional
- Programación lógica

Nuevo Paradigma:

- Programación Orientada a Objetos (POO)

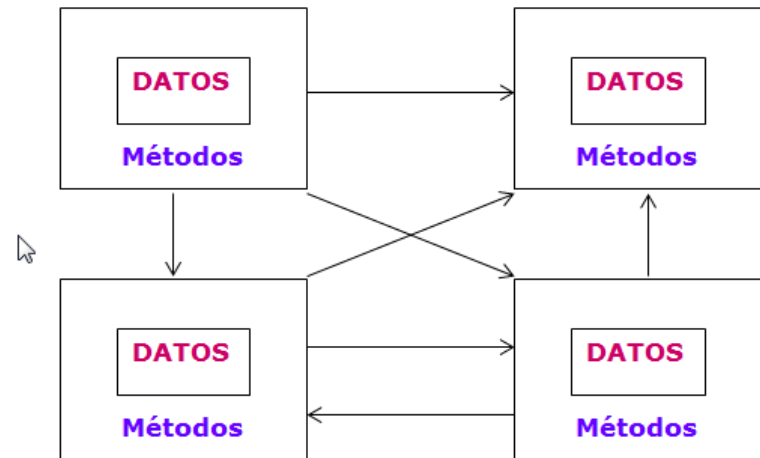


Programación Orientada a Objetos.

Programación modular



Programación Orientado a Objetos





Ventajas de la Programación Orientada a Objetos.

- ❑ Facilita la creación de software de calidad pues sus características potencian:
 - ❑ La mantención
 - ❑ La extensión y
 - ❑ La reutilización del software generado bajo este paradigma
- ❑ La POO se basa en la idea natural de un mundo lleno de objetos y que la resolución de problemas se realiza mediante el modelo de objetos



Programación Orientada a Objetos.

- La visión de Objetos:
 - Mesas
 - Sillas
 - Computadores
 - Autos
 - Cuentas bancarias
 - Partidos de fútbol
 - Perros, etc



Programación Orientada a Objetos.

Los objetos se pueden Organizar según su necesidad

Mamíferos : Perro, Ciervo

Teléfonos : Fijo, Celular

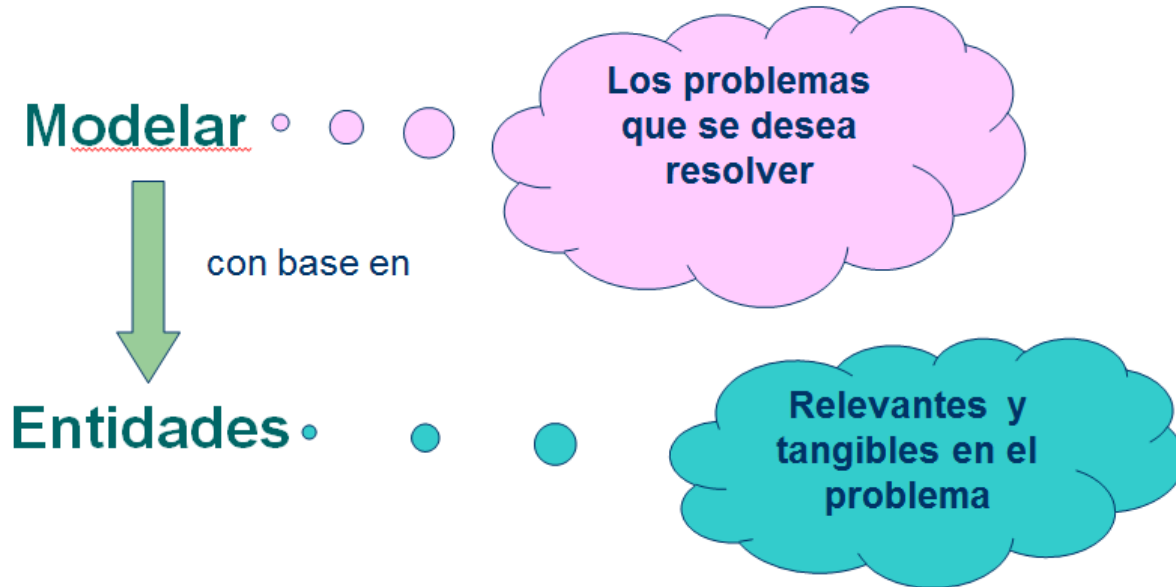
Deportes : Fútbol, Tenis

Vehículos : Automóvil, Camión



METODOLOGÍA de POO

La base de esta tecnología es:

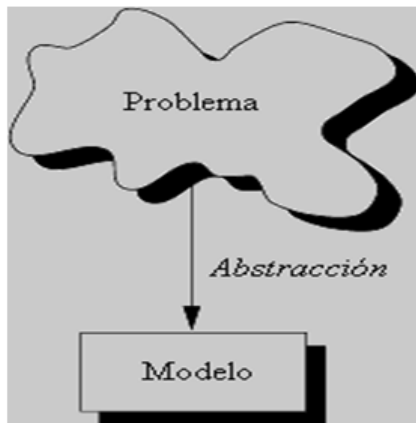




ABSTRACCIÓN

El modelo define una *perspectiva abstracta* del problema

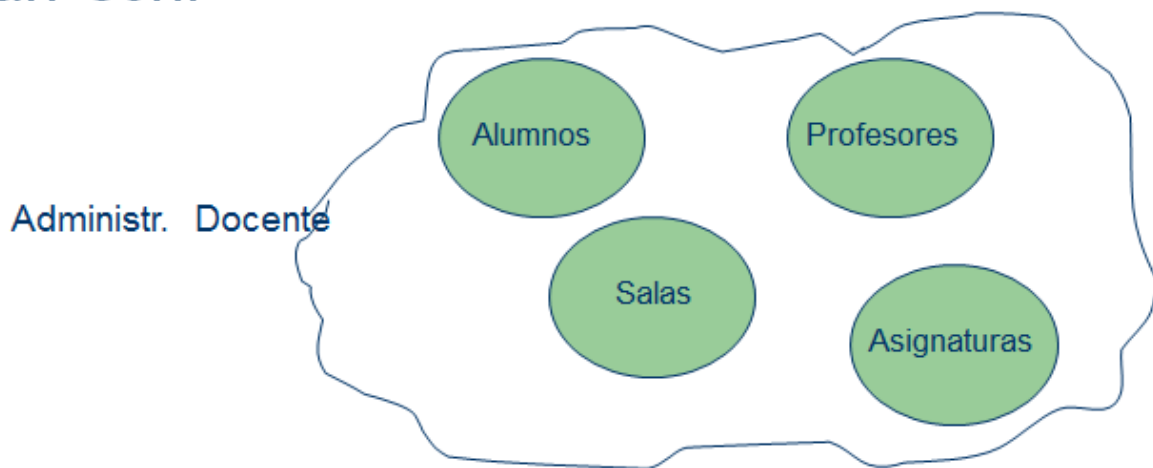
- Los *datos* que son afectados
- Las *operaciones* que se aplican sobre los datos





Ejemplo

Para la administración docente, las entidades que participan son:





¿Cómo crear un ambiente orientado a objetos?

- ❑ Identificar los objetos importantes.
- ❑ Organizar los objetos en jerarquías.
- ❑ Otorgar, a los objetos, atributos relevantes que describan sus características.
- ❑ Proveer, a cada objeto, las funciones inherentes a su comportamiento.



Conceptos de POO

La POO trata de aproximarse al modo de actuar del hombre y no al de la máquina

- Conceptos fundamentales que sustentan la POO:
 - Clase
 - Objeto
 - Instancia
 - Atributos
 - Métodos



Clase

Una clase es como un tipo de dato creado por el usuario, que posee dos categorías de miembros:

- **Atributos** (Datos) ⇒ Estado
- **Métodos** (Algoritmos) ⇒ Comportamiento



Programación Orientada a Objetos.

En general, es posible crear una clase a partir de cualquier objeto que esté a nuestro alrededor. Por ejemplo:

- Persona
- Automóvil
- Mascota



Instancias - Objetos.

- Una **Instancia** es una ocurrencia de la clase
- Al momento de crear un objeto se produce la instanciación
- Un **Objeto** es una instancia de una Clase específica



Instancias - Objetos.

EJEMPLO:

Suponer que existe la clase **Vehículo**

- El auto patente WF-4631 es una instancia de la clase **Vehículo**, o sea, un objeto de esa clase
- La camioneta patente ZT-9283 es otra instancia de la clase **Vehículo**



Atributos

Son los datos que caracterizan a los objetos de una clase y determinan el estado de un objeto.

- Marca
- Año
- Color
- Patente, etc.



Métodos

- ❑ Representan todas aquellas acciones que se pueden realizar sobre un objeto de cierta clase
- ❑ En la implementación, estos métodos son segmentos de código en la forma de funciones
- ❑ La clase Vehículo puede incluir los métodos:
 - ❑ Encender
 - ❑ Acelerar
 - ❑ Virar
 - ❑ Frenar



Principios de la POO

- Propiedades, que lo llevan a ser *un estilo de desarrollo* que permite crear *código re-utilizable*

➤ Encapsulamiento

➤ Herencia

➤ Polimorfismo



Programación Orientada a Objetos.

ENCAPSULAMIENTO

Abstracción de Datos



Encapsulamiento

- ❑ Proceso por el que se ocultan:
 - ❑ Las estructuras de datos
 - ❑ Los detalles de la implementación
- ❑ Permite considerar a los objetos como "cajas negras", evitando que otros objetos accedan a detalles que NO LES INTERESA
- ❑ Una vez creada la clase, las funciones usuarias no requieren conocer los detalles de su implementación



Encapsulamiento

- Toda clase tiene un conjunto de **atributos** y **métodos** asociados a ella.
- Todos ellos están **encapsulados** o contenidos dentro de la misma clase, de manera que son **miembros** de dicha clase
- Esos métodos y atributos pueden ser utilizados por **otras** clases **sólo si** la clase que los encapsula les brinda los **permisos** necesarios para ello



Encapsulamiento

Atributos de una Cuenta Corriente:

- Número
- Saldo

¿Cómo se almacenan estos datos?

Métodos:

- Depositar
- Girar
- Conocer el saldo



Programación Orientada a Objetos.

HERENCIA

Organización jerárquica

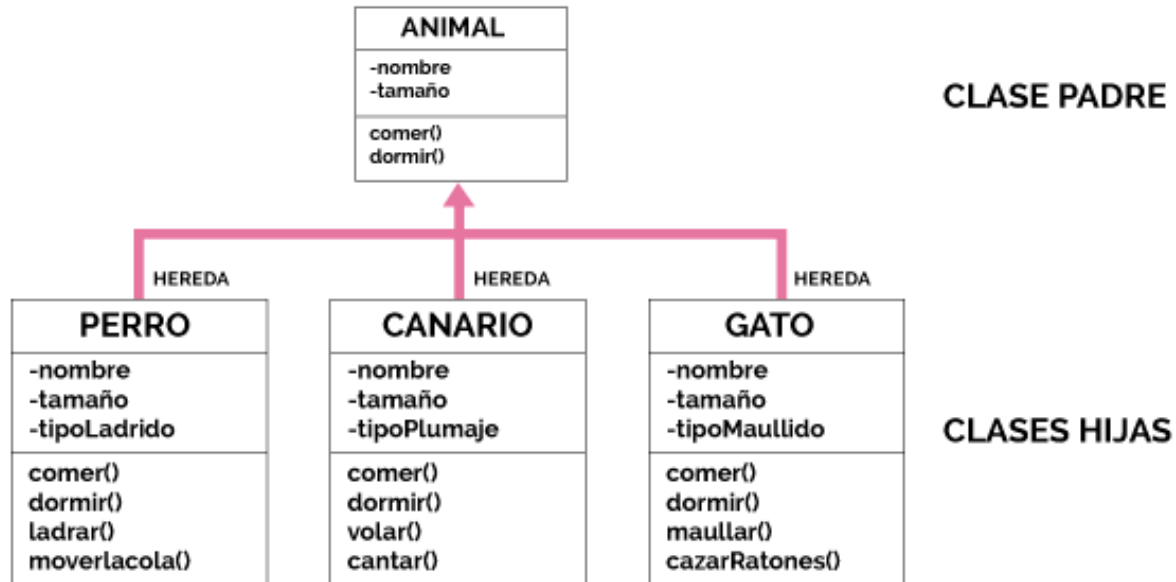


Herencia

- ❑ Permite reutilizar código creando **nuevas** clases a partir de las existentes (construidas y depuradas)
- ❑ Compromete una relación de jerarquía (**es-un**)
- ❑ Una nueva clase se generará agregando atributos y/o código a una clase existente
- ❑ Una clase (derivada) **puede heredar** de otra clase (base):
 - Atributos y
 - Métodos



Herencia





Programación Orientada a Objetos.

POLIMORFISMO





Polimorfismo

- Capacidad que permite a dos **clases diferentes** responder de **forma distinta** a un **mismo mensaje**.
- Esto significa que dos clases que tengan un método con el mismo nombre y que respondan al mismo tipo de mensaje (es decir, que reciban los mismo parámetros), ejecutarán acciones distintas



Polimorfismo

Ejemplo 1:

Al presionar el acelerador esperamos que aumente la velocidad del auto, independiente de si se tiene un:

- Motor con carburador
- Motor con inyección electrónica



Polimorfismo

Ejemplo 2:

Si se tienen las clases **Entero** y **Char**, ambas responderán de manera distinta al mensaje **"Sucesor"**

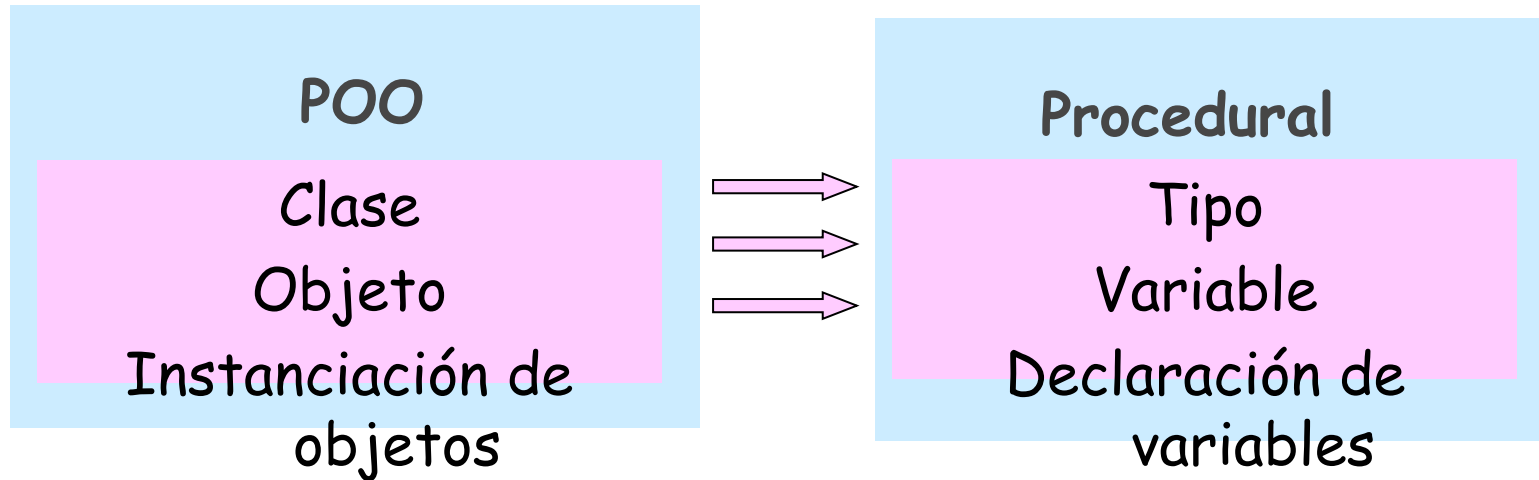




Objetos

En la **POO** se dice que un objeto: "integra datos y algoritmos"

En la programación estructurada, variables y funciones están separadas.



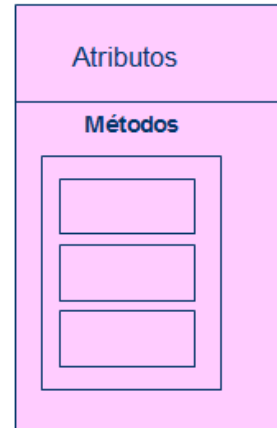


Objetos.

Cada objeto ***es responsable*** de inicializarse y destruirse en forma correcta

Un ***objeto*** consta de:

- Tiempo de vida
- Estado
- Comportamiento





Tiempo de vida de un Objetos.

- ❑ La duración de un objeto en un programa siempre está limitada en el tiempo.
- ❑ La mayoría de los objetos sólo existen durante una parte de la ejecución del programa.
- ❑ Los objetos son creados mediante un mecanismo denominado instanciación.
- ❑ Los objetos dejan de existir cuando son destruidos.



Estado de un Objeto

- ❑ Queda definido por sus **atributos**.
- ❑ Con él se definen las **propiedades del objeto**, y el estado en que se encuentra en un momento determinado de su existencia



Comportamiento de un objeto

- Queda definido por los **métodos**
- Los prototipos de los **métodos** definidos en la interfaz de una clase permiten a otros objetos, que forman parte de la aplicación, interactuar con los objetos de esa clase.



Programación Orientada a Objetos.

CLASES



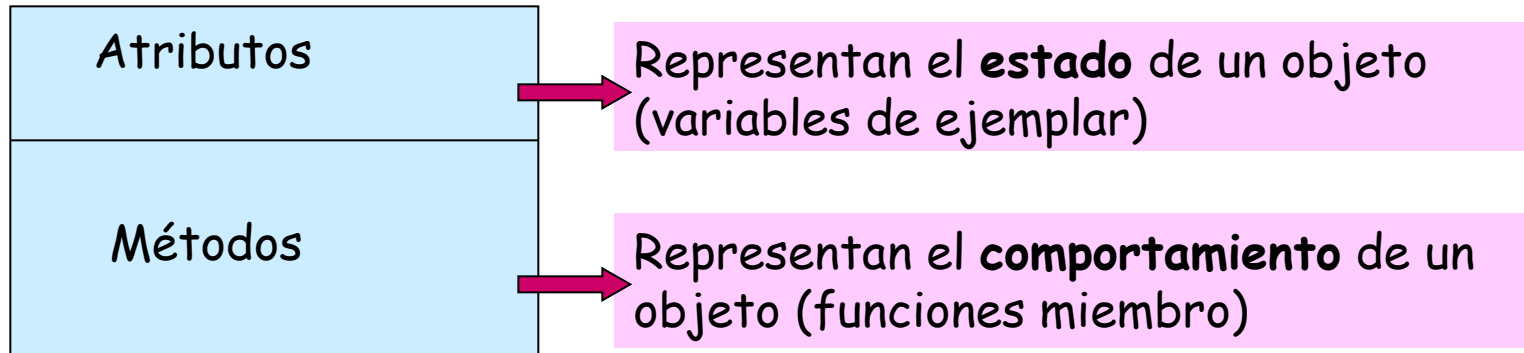
Clases

- Las clases ***son abstracciones*** que representan a un conjunto de objetos con un:
 - Comportamiento e
 - Interfaz común
- Es la ***implementación*** de ***un tipo de dato*** (considerando los objetos como instancias de las clases)



Clases

- ✚ Permiten definir y representar colecciones de objetos
- ✚ Proveen un **modelo** para la creación de objetos
- ✚ Los elementos que componen la clase son





Ejemplo de Instancia de un Objeto

- **Clase:** Cuenta corriente
 - Atributos:
 - Número
 - Nombre
 - Saldo
 - Métodos:
 - Depositar
 - Girar
 - Consultar saldo



Programación Orientada a Objetos.

□ **Clase:**

Cuenta corriente

□ **Instanciación:**

Cuenta Corriente A, B

Objeto: A

Num: 1234

Nombre: Juan

Saldo: 350.000

Métodos

Depositar

Girar

Consultar

Objeto: B

Num: 9876

Nombre: María

Saldo: 450.600

Métodos

Depositar

Girar

Consultar



Programación Orientada a Objetos.

MENSAJES



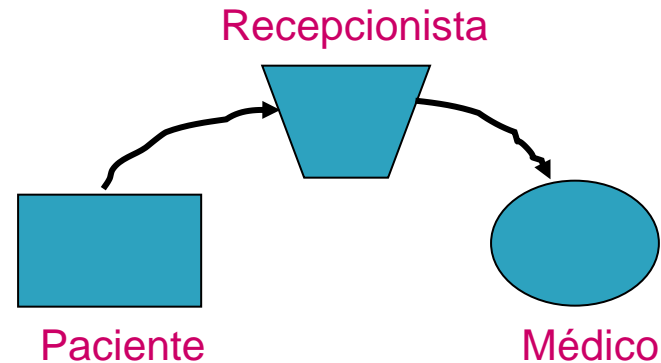
Mensajes

- Mecanismo por el cual **se solicita** una acción sobre el objeto
- Un programa en ejecución es una colección de objetos que se crean, **interactúan** y se destruyen
- La **interacción** se basa en **mensajes** que son enviados de un objeto a otro, de modo que el emisor le pide al receptor la ejecución de un método



Mensajes

- Un objeto invoca un método como una reacción al recibir un mensaje
- La interpretación del mensaje dependerá del receptor





Programación Orientada a Objetos.

MÉTODOS



Métodos

- Un método es una función miembro de una clase
- Establece el comportamiento del objeto
- Opera directamente sobre el objeto que lo invocó
- Recibe, como **parámetro implícito**, el objeto que lo invocó



Mensajes

- Si el método requiere otros objetos de la clase, éstos deberán ser pasados como parámetros explícitos y el método sólo podrá acceder en forma indirecta a estos objetos

- Ejemplos:
 - Depositar
 - Girar
 - Consultar



Programación Orientada a Objetos.

RESUMEN



Programación Orientada a Objetos.

- ❑ Un objeto (agente emisor) envía un mensaje a otro objeto (agente receptor)
- ❑ El mensaje tiene codificada la petición de una acción
- ❑ El mensaje incluye la información (argumentos) necesaria para satisfacer la petición
- ❑ Si el receptor acepta el mensaje, acepta la responsabilidad de ejecutar la acción indicada
- ❑ En respuesta a un mensaje, el receptor ejecuta un método para satisfacer la petición



Clases y Ejemplares

- Todos los objetos son ejemplares de una clase
- La clase del receptor determina el método que se activa como respuesta a un mensaje
- Todos los objetos de una clase usan el mismo método en respuesta a mensajes similares



Programación Orientada a Objetos.

- Una variable de ejemplar es una variable interna mantenida por un ejemplar
- Cada ejemplar tiene su propia colección de variables de ejemplar
- Las variables de ejemplar sólo son modificables por los métodos definidos en la clase
- Un objeto es la combinación de estado y comportamiento



Clases y Métodos

- ❑ El estado lo determinan las variables de ejemplar
- ❑ El comportamiento lo determinan los métodos
- ❑ Desde el exterior, los clientes sólo pueden ver el comportamiento de los objetos
- ❑ Desde el interior, los métodos proporcionan el comportamiento apropiado mediante las modificaciones del estado



Clases y Métodos

- ❑ La interfaz describe la forma en que un objeto se conecta con el mundo
- ❑ La implementación describe cómo se logra la responsabilidad prometida en la interfaz
- ❑ Una clase se puede concebir como un registro con dos variedades de campos: datos y procedimientos
- ❑ Los datos constituyen las variables de ejemplar
- ❑ Los procedimientos constituyen los métodos