



Universidad César Vallejo

Pregrado

SESIÓN 01: Clases y Objetos Diagrama UML



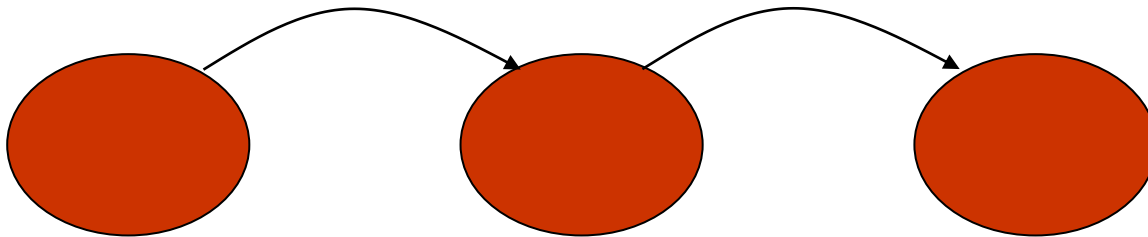
Programación Orientada a Objetos

- Metodología que define los programas en términos de "clases de objetos", objetos que son entidades que combinan **estado** (datos), **comportamiento** (procedimientos o métodos) e **identidad** (propiedad del objeto que lo diferencia del resto).
- Toma las mejores ideas de la programación estructurada y modular combinándola con nuevos y poderosos conceptos.



Paradigma estructurado

- El sistema se modela como un conjunto de procedimientos (acciones) que intercambian información.
- Datos y lógica por separado





PE vs POO

PE:

- Fija su atención en el conjunto de acciones que manipulan el flujo de datos.
- Los procesos son la parte central de este modelo.
- El mantenimiento de los desarrollos deben efectuarse minuciosamente. Se hace tediosa según el tamaño del programa.
- Los procesos que se modelan en el desarrollo de un problema plasman las operaciones necesarias para resolverlo.

POO

- Fija en la interrelación que existe entre los datos y las acciones a realizar con ellos.
- Los objetos son la parte central del modelo y los métodos que los modifican muestran el comportamiento del objeto.
- El mantenimiento de programas y aplicaciones generalmente son fáciles de realizar.
- Los objetos que se modelan en el desarrollo de un sistema se ajustan a la realidad que representa el problema.



Área Donde se Aplica la Tecnología de Objetos

- Lenguajes de programación
- Bases de datos
- Dibujo y Visualización
- Sistemas Operativos
- Interfaz de Usuario
- Metodología de análisis y diseño
- Sistemas de conmutación telefónica



Ventajas de POO

- **Uniformidad:** la representación de los objetos esta presente tanto en el diseño como en la codificación
- **Flexibilidad:** Cualquier cambio se ve reflejado en cualquier lugar donde aparecen los datos involucrados.
- **Objetivismo:** representa el mundo real.
- **Reusabilidad:** permite utilizar una clase definida previamente en aplicaciones que nos sea conveniente.
- **Mantenibilidad:** las clases, vistas como módulos independientes son fáciles de mantener.
- **Extensibilidad:** gracias a la modularidad y herencia pueden ser fácilmente extensibles al crecimiento de la aplicación.



Características de JAVA

- ✓ **Simple:** Java ofrece toda la funcionalidad de un lenguaje potente, elimina las características menos usadas y más confusas
- ✓ **Orientado a Objetos:** Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- ✓ **Distribuido:** Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos.
- ✓ **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución.



- **Seguro:** El código Java pasa muchos tests antes de ejecutarse en una máquina.
- **Interpretado:** El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto.
- **Multihilo:** Al ser MultiHilo, Java permite muchas actividades simultáneas en un programa.
- **Portable:** Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas.
- **Dinámico:** Las librerías nuevas o actualizadas no paralizarán la ejecución de las aplicaciones actuales (siempre que mantengan el API anterior).
- **Arquitectura Neutral:** el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.



¿Qué son las Clases?

- Es un conjunto de objetos que comparten una estructura común y un comportamiento común
- Una clase es una descripción de un grupo de objetos con propiedades en común (atributos), comportamiento similar (operaciones), la misma manera de relacionarse entre objetos (asociaciones y agregados) y una semántica en común.



Clase

Persona
nombre apellido materno apellido paterno sexo edad
comer() beber() dormir()

← Nombre

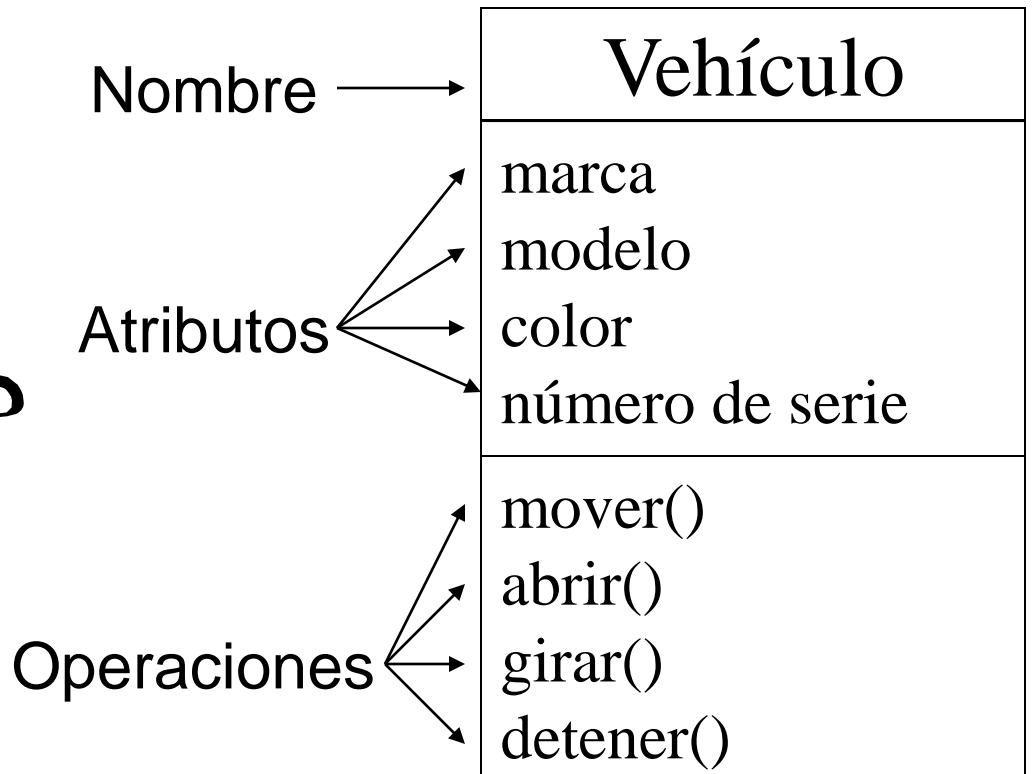
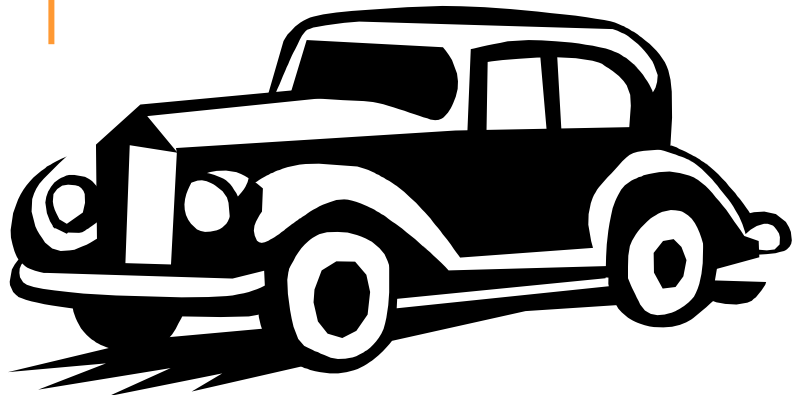
←
←
←
←
← Atributos

←
←
← Operaciones



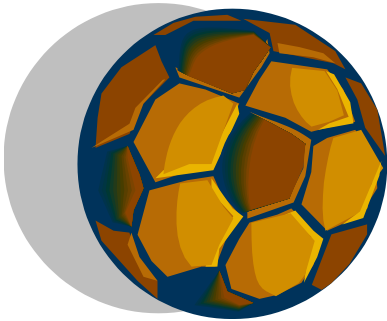
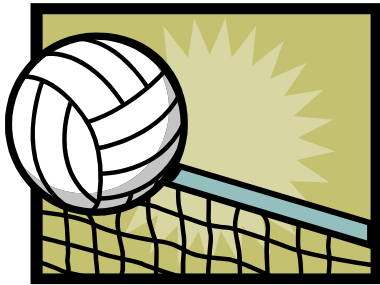


Clase





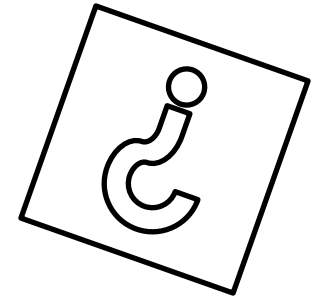
Clase



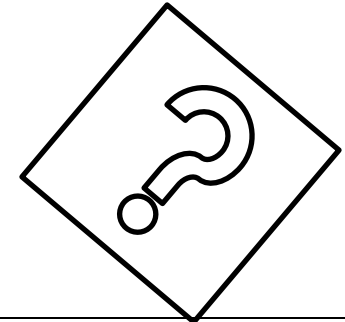
Nombre →

Pelota

Atributos →



Operaciones →





Clases y Objetos

Clase

generalización
de características
(atributos y
comportamientos)

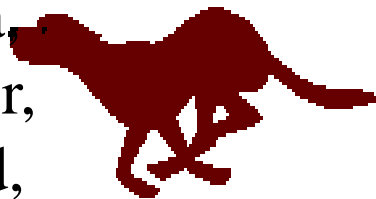
Objeto

Instancia de una clase
distinguible por sus
características
específicas

Mundo Real

Perro

Raza,
Color,
Edad,
Corre,



.....

Tino

Pastor Alemán
Marrón
7 meses
Veloz



.....

En OOP

Clase

Define
datos y
métodos

Objetos

Ocupa
espacio,
se crea y
se destruye



CLASES Y OBJETOS - EJEMPLO

ATRIBUTOS

METODOS

DNI

Numero
Primer_apellido
Segundo_apellido
Pre_Nombres
Ubigeo
Sexo
.....

Verificar_caducidad(f_caducidad)
Verificar_Ubigeo(Ubigeo)

CLASE DNI

004

Ra

Laz

Ma

020

Ma

....

Ver

Ver

Ver

Ver

10267855

Tol

Lu

Ef

02

Ma

....

Ver

Ver

Ver

10457898

Alvarado

Trrujillo

Jose Carlos

020101

Masculino

.....

Verificar_caducidad(05-12-2011)

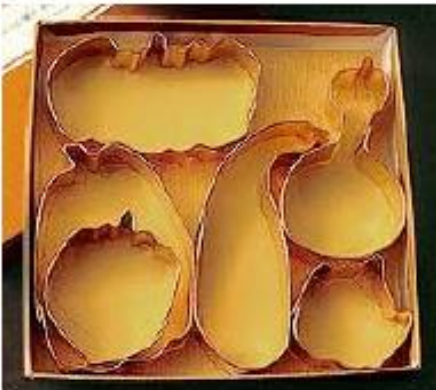
Verificar_Ubigeo(020101)

OBJETO DNI



Clases vs. Objetos

Clase



Molde de Galletas

Objetos:

Instancias de clases



Galletas



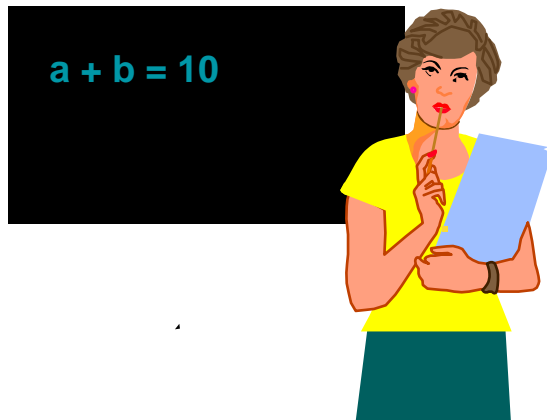
¿Qué es un Objeto?

- Son llamados también “instancias”, es un concepto, una abstracción o una cosa con límites bien definidas y significado para una aplicación
- Un objeto es algo que tiene:
 - Estado
 - Comportamiento
 - Identidad



Un Objeto tiene Estado

- El estado de un objeto se compone de los valores de los atributos y los enlaces (referencias) que mantiene un objeto en un momento determinado. Los objetos interactúan unos con otros y como consecuencia de esas interacciones cambian de estado (es decir, cambian el valor de sus atributos o sus enlaces con otros objetos).



Nombre:	Carmen Reyes
Nº Empleado:	2587
Fecha de Contr.:	21 de marzo 1987
Condicion:	Contratada



Estado

Persona
Juan García Rodríguez Masculino 15
comer() beber() dormir()

Persona
Juan García Rodríguez Masculino 32
comer() beber() dormir()

Vehículo
Ford Mustang Blanco SX-54852545
mover() abrir() girar() detener()

Vehículo
Ford Mustang Rojo SX-54852526
mover() abrir() girar() detener()



Un Objeto tiene Comportamiento

- El comportamiento de un objeto es cómo actúa y reacciona un objeto, en función de sus cambios de estado y paso de mensajes.
- El comportamiento de un objeto es modelado por un conjunto de mensajes a los que puede responder (las operaciones que el objeto puede realizar)



Paso de Mensajes

- Es la acción de efectuar una llamada a un método.
- Medio de colaboración entre objetos



:MandoADistancia

Conectar ()



Canal (8)

:Televisor



Principios de la POO

- Abstracción
- Modularidad
- Encapsulamiento
- Herencia
- Polimorfismo



Declaración en JAVA NetBeans:

- Clases
- Objetos
- Métodos
- Acceso a datos
- Atributos





Clase

- Es una plantilla para un objeto.
- Define la estructura de un objeto y su interfaz funcional, es decir, define los métodos y atributos de un grupo de objetos de características similares.

```
class NombreClase{  
    //...  
    //cuerpo de la clase  
    //...  
}
```



```
class Empleado{
```

```
    //Atributos
```

```
    public String nombres, apellidos;
```

```
    private double sueldo;
```

```
    public int edad;
```

```
    //Constructor de la clase
```

```
    public Empleado(String nom, String ape, double s, int ed)
```

```
{        //Actualizando la información de la clase
```

```
    this.nombres=nom;
```

```
    this.apellidos=ape;
```

```
    this.sueldo=s;
```

```
    this.edad=ed;
```

```
}
```

```
    //Métodos Propios de la clase
```

```
    public void trabajar(String n, int horas, int dias){
```

```
}
```

```
    private double asistir(int faltas){
```

```
        return 0;
```

```
}
```

```
}
```




Atributos

- Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables.

```
tipo nombre;
```



Métodos

- Los métodos son subrutinas que definen la interfaz de una clase, sus capacidades y comportamiento.
- Los métodos pueden tener una lista de parámetros que permiten comunicar información entre métodos.

```
tipo Nombre_del_metodo (parametros)
```



Clase, Atributos y Métodos

```
class Empleado{
```

```
    String nombres, apellidos;  
    double sueldo;  
    int edad;
```

```
    void trabajar(String n, int horas, int dias){
```

```
    }  
    double asistir(int faltas){  
        return 0;  
    }
```

```
}
```

Clase

*Atributos o
Variables*

Métodos



Objeto

- Un objeto es una instancia o variable de una clase. La **declaración** de una referencia a objeto se realiza:

```
Clase objeto;
```



Método Constructor

- Un constructor es un método que inicia un objeto inmediatamente después de su creación, evitándonos tener que iniciar las variables. Tiene el mismo nombre de la clase a la que pertenece.

```
class Empleado {  
    Empleado() {
```

```
}
```

```
    Empleado(int x, int y) {
```

```
}
```

```
}
```

Constructor



Operador new

- Permite crear un nuevo objeto invocando al método constructor de la clase, y devuelve una referencia a ese objeto, proporcionando almacenamiento persistente.

```
Clase objeto = new metodoConstructor() ;
```



Operador punto (.)

- Las variables de instancia:

• `NombreDelObjeto.variableDeInstancia`

`NombreDelObjeto.método()`



Modificadores de Acceso

- Los modificadores de acceso definen niveles de visibilidad entre miembros (variables y métodos) de una clase y otros objetos.

```
modificador_de_acceso tipo nombre_variable;  
  
modificador_de_acceso tipo nombre_metodo( ){  
}
```




- **public:** Especifica que las variables y métodos de una clase podrán ser accedidas por todos los objetos, aunque sean de clases de otros paquetes diferentes al de la clase.
- **protected:** Especifica que las variables y métodos de una clase podrán ser accedidas, además de los métodos de la misma clase, por los métodos de todas las clases del mismo paquete y por las subclases de esta aunque no se encuentren en el mismo paquete.
- **private:** Especifica que las variables y métodos de una clase podrán ser accedidas sólo por los métodos de la misma clase.



Modificadores de acceso para variables y métodos

Un miembro tiene acceso a:				
Puede ser accedido por métodos de:	public	protected	omitido	private
■ La misma clase	Si	Si	Si	Si
■ Subclases del mismo paquete	Si	Si	Si	-
■ Otras clases del mismo paquete	Si	Si	Si	-
■ Subclases en otros paquetes	Si	Si	-	-
■ Otras clases de otros paquetes	Si	-	-	-

```
class Empleado{
```

```
    //Atributos
```

```
    public String nombres, apellidos;
```

```
    private double sueldo;
```

```
    public int edad;
```

```
    //Constructor de la clase
```

```
    public Empleado(String nom, String ape, double s, int ed)
```

```
    {        //Actualizando la información de la clase
```

```
        this.nombres=nom;
```

```
        this.apellidos=ape;
```

```
        this.sueldo=s;
```

```
        this.edad=ed;
```

```
    }
```

```
    //Métodos Propios de la clase
```

```
    public void trabajar(String n, int horas, int dias){
```

```
    }
```

```
    private double asistir(int faltas){
```

```
        return 0;
```

```
    }
```

```
}
```



Examen de Entrada: Construya la siguiente clase

- Implemente una clase de nombre Empleado con los siguientes atributos: código, nombres, área laboral (Sistemas, Administración, Marketing), sueldo, horas extras y afiliación (AFP o SNP). Además considere atributos de valores comunes para todos los empleados, para los porcentajes de descuento por afiliación a una AFP (11% del sueldo), por afiliación al sistema nacional de pensiones (6% de sueldo) y por salud (3% del sueldo). Considere métodos de cálculo para el monto de horas extras ($\text{sueldo} \times \text{horas extras} / 240$). Considere el cálculo de los descuentos así como del Sueldo bruto y el sueldo neto.



Universidad **César Vallejo**

Licenciada por Sunedu
para que puedas salir adelante