

ENTERPRISE JAVA DEVELOPER

# JAVA ORIENTADO A OBJETOS

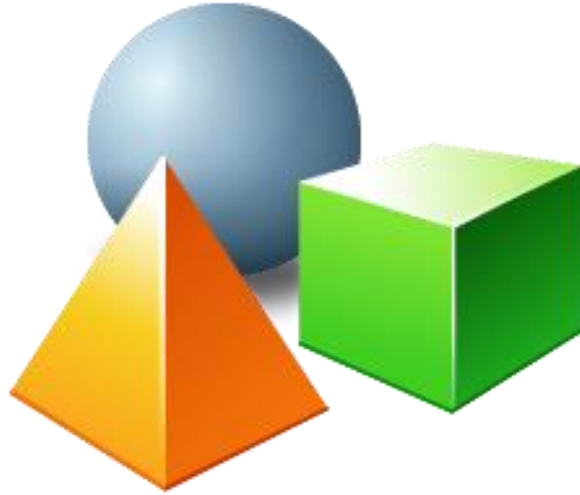
## CLASES Y OBJETOS

Eric Gustavo Coronel Castillo

[www.youtube.com/DesarrollaSoftware](http://www.youtube.com/DesarrollaSoftware)

[gcoronelc.blogspot.com](http://gcoronelc.blogspot.com)





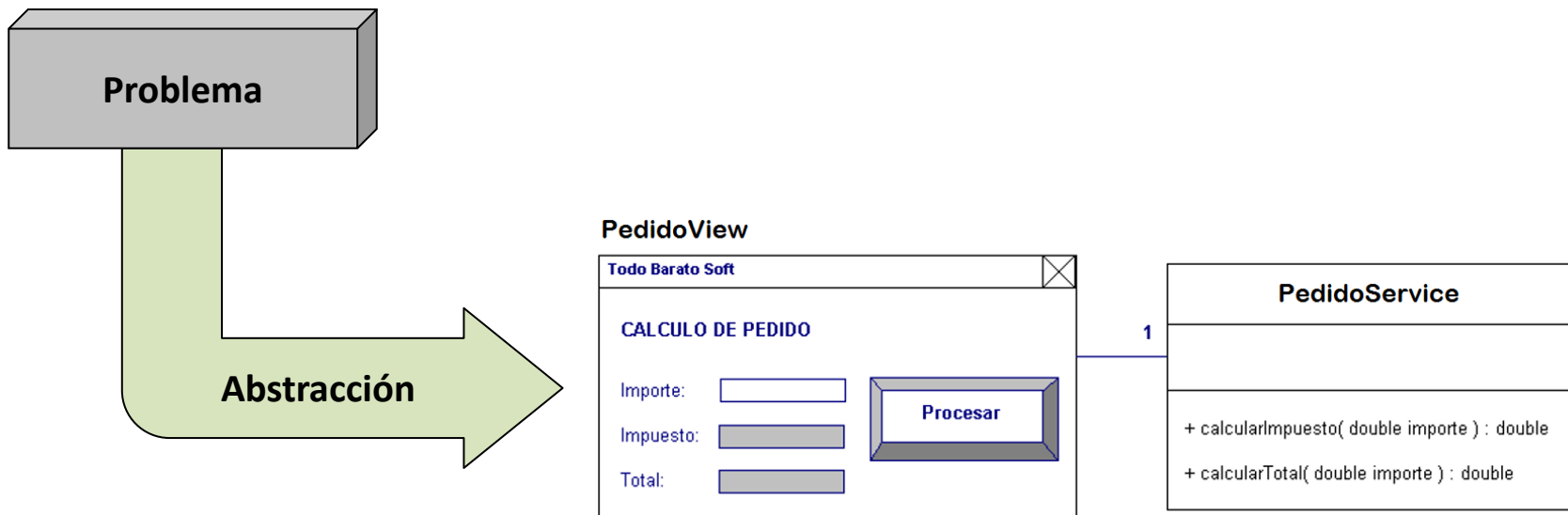
## Temas

- Objetivo
- Abstracción
- Definición de clase y objeto
- Implementación de clases
- Creación y uso de objetos
- Paquetes
- Proyecto ejemplo



# OBJETIVO

Entender los conceptos de Clase y Objeto, y su aplicación en la solución de problemas sencillos.





# ABSTRACCIÓN

Consiste en capturar, percibir y clasificar las características (datos- atributos) y comportamientos (operaciones) necesarias (relevantes) del mundo real (proceso a sistematizar) para dar solución al problema.



Notación UML



## Persona

- + Nombre : String
- + Edad : Integer
- + Profesion : String

- + Caminar()
- + Correr()
- + Cantar() : String

## Animal

- + Raza : String
- + Genero : String

- + Comer()

## Transporte

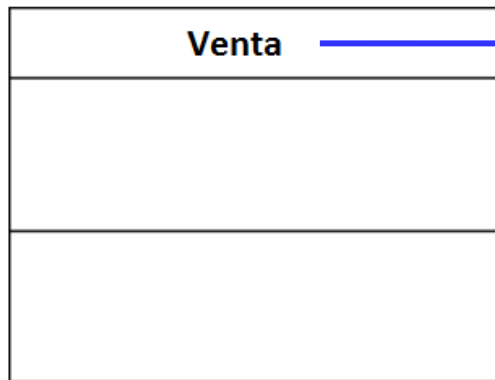
- + Tipo : String
- + Marca : String
- + Año : Integer

- + Encender() : Boolean
- + Acelerar(Velocidad : Integer)

# DEFINICIÓN DE CLASE Y OBJETO

## CLASE

- Una clase define un tipo de objeto en particular.
- Por ejemplo, la clase Empleado define a todos los trabajadores de una empresa.



→ Nombre de la Clase

### Ejemplos de Nombres de Clase

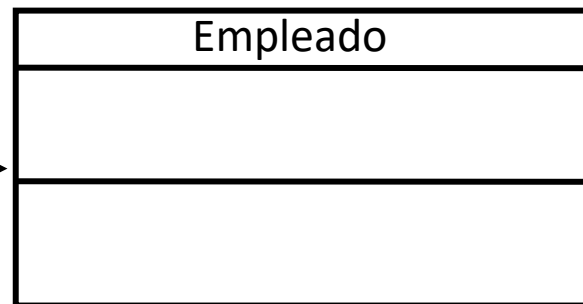
- Cliente
- Factura
- NotaCredito
- Guia
- Pedido
- Matricula
- CuentaMaestra



# DEFINICIÓN DE CLASE Y OBJETO

## OBJETO

- Un objeto es una instancia de una clase.
- Por ejemplo, cada trabajador de una empresa es una instancia de la clase Empleado.



# DEFINICIÓN DE CLASE Y OBJETO

## Notación UML de OBJETO



objPersona1 : Persona

+ Nombre : Jennifer  
+ Edad : 34  
+ Profesion : Cantante

+ Caminar()  
+ Correr()  
+ Cantar():String



objPersona2 : Persona

+ Nombre : Zidane  
+ Edad : 40  
+ Profesion : Futbolista

+ Caminar()  
+ Correr()  
+ Cantar():String

## Notación UML de Clase

Persona

+ Nombre : String  
+ Edad : Integer  
+ Profesion : String

+ Caminar()  
+ Correr()  
+ Cantar() : String

Instancia

Instancia

# IMPLEMENTACIÓN DE CLASES

## SINTAXIS

```
public class NombreClase {  
  
    // Definición de variables  
  
    // Definición de métodos  
  
}
```

El nombre del archivo debe tener el mismo nombre de la clase.

Por ejemplo, si la clase se llama **Producto** el nombre del archivo que contiene a la clase se debe llamar **Producto.java**.



# IMPLEMENTACIÓN DE CLASES

## ATRIBUTOS

- Representa un dato del objeto.
- Cada atributo de un objeto tiene un valor que pertenece a un dominio de valores determinado.
- En Java se implementan creando variables a nivel de clase.

Venta
<ul style="list-style-type: none"><li>- id : Integer</li><li>- fecha: Date</li><li>- cliente: String</li><li>- importe: Double</li><li>...</li></ul>

```
public class Venta {  
  
    // Variables que implementación de atributos  
    private Integer id;  
    private Date fecha;  
    private String cliente;  
    private Double importe;  
  
}
```

# IMPLEMENTACIÓN DE CLASES

## OPERACIONES

- Son servicios proporcionado por un objeto que pueden ser solicitados por otros objetos.
- Determinan el comportamiento del objeto.
- La implementación en Java se realiza mediante métodos.

Venta
<ul style="list-style-type: none"><li>- id : Integer</li><li>- fecha: Date</li><li>- cliente: String</li><li>- importe: Double</li><li>...</li></ul>
<ul style="list-style-type: none"><li>+ buscar() : boolean</li><li>+ insertar() : void</li><li>+ modificar() : void</li><li>+ eliminar() : void</li><li>...</li></ul>

```
public class Venta {  
  
    // Implementación de atributos  
    private Integer id;  
    ...  
  
    // Implementación de operaciones  
    public boolean buscar( ... ) {  
        ...  
    }  
    ...  
}
```



# IMPLEMENTACIÓN DE CLASES

## DEFINICIÓN DE MÉTODOS

```
public <tipo> nombreMétodo ( [ parámetros ] ) {
```

```
    // Implementación
```

```
    [ return valorRetorno; ]
```

```
}
```

- <tipo>** Determina el tipo de dato que retorna el método, si no retorna ningún valor se utiliza **void**.
- return** Esta sentencia finaliza la ejecución del método, se acompaña de un valor cuando el método debe retornar un resultado.



# CREACIÓN Y USO DE OBJETOS

---

## OPERADOR NEW

NombreClase variable = **new** NombreClase();

ó

NombreClase variable = null;  
variable = new NombreClase();

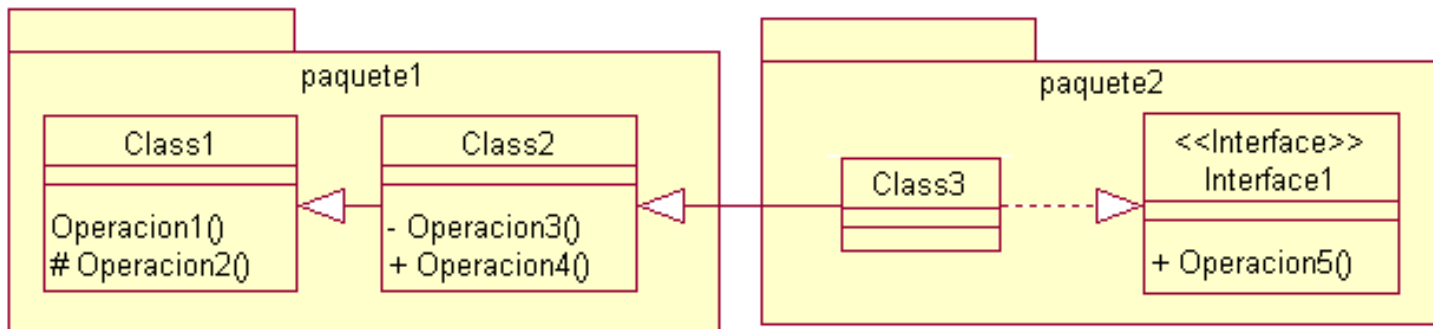
## ACCESO A LOS MÉTODOS

variable.nombreMétodo ( ... )



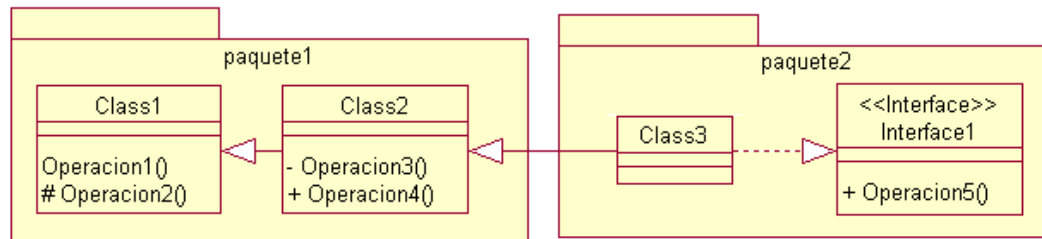
# PAQUETES (PACKAGES)

- Organiza y agrupa un conjunto de clases, interfaces, subpaquetes y otros.
- La creación de paquetes evita el conflicto de nombre de clases, además un paquete puede contener clases, campos y métodos que están disponible sólo dentro del paquete.
- Observe la siguiente figura usando notación UML, y responda **¿Qué operaciones (métodos) tendrá la clase Class3?**



# PAQUETES (PACKAGES)

Para definir un paquete se usa la instrucción **package** y para utilizar clases de otro paquete, indique la ruta del paquete antes del nombre de la clase o la instrucción **import**.



```
// Definiendo un paquete
package paquete1;

// Clase asociada al paquete
public class Class1() { . . . };
```

```
// Definiendo un paquete
package paquete2;

// Interface asociada al paquete
public interface Interface1() { . . . };
```

```
// Definiendo un paquete
package paquete1;

// Clase asociada al paquete
public class Class2 extends Class1
{ . . . };
```

```
// Definiendo un paquete
package paquete2;

// Importando todas las clases del paquete
import paquete1.*;

// Clase asociada al paquete
public class Class3 extends Class2 implements Interface1
{ . . . };
```



# PROYECTO EJEMPLO

---

La empresa "Todo Barato" necesita facilitar la elaboración de los pedidos que realizan sus empleados a sus proveedores, el problema radica al momento de calcular el impuesto.

La empresa ha solicitado a su departamento de sistemas elaborar un programa en Java que permita ingresar el importe del pedido, y calcule el impuesto y el total que se debe pagar al proveedor.





ENTERPRISE JAVA DEVELOPER

# JAVA ORIENTADO A OBJETOS

**Gracias**

Eric Gustavo Coronel Castillo  
[gcoronelc.blogspot.com](http://gcoronelc.blogspot.com)





[gcoronelc.github.io](https://gcoronelc.github.io)



## FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



## PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos



## CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas



## PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC