

ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

SOBRECARGA

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com





Temas

- Objetivo
- Firma de un Método
- Sobrecarga de Métodos
- Sobrecarga de Constructores
- Proyectos Propuestos
- Bibliografía



OBJETIVO

Aplicar la sobrecarga para disponer de diversas versiones de métodos y constructores que se puedan aplicar dependiendo de las necesidades que se tengan o se proyecten tener.

```
❏ CASE_INSENSITIVE_ORDER Comparator<String>
❏ copyValueOf(char[] data) String
❏ copyValueOf(char[] data, int of... String
❏ format(String format, Object..... String
❏ format(Locale l, String format,... String
❏ valueOf(Object obj) String
❏ valueOf(boolean b) String
❏ valueOf(char c) String
❏ valueOf(char[] data) String
❏ valueOf(double d) String
❏ valueOf(float f) String
❏ valueOf(int i) String
❏ valueOf(long l) String
❏ valueOf(char[] data, int offset... String
❏ class
```



FIRMA DE UN METODO

- La firma de un método esta definida por:
 - Nombre del método
 - Parámetros del método
 - La cantidad de parámetros
 - El tipo de dato de sus parámetros
 - Orden de los parámetros

- En una clase no puede existir dos métodos con la misma firma.



FIRMA DE UN METODO

```
public class DemoService {
```

```
    public void procesar(double value1, int value2) {
```

method procesar(double,int) is already defined in class DemoService

(Alt-Enter shows hints)

```
    }
```

```
    public void procesar(double value1, int value2) {
```

```
    }
```

```
}
```



SOBRECARGA DE MÉTODOS

- Es la implementación de varios métodos con el mismo nombre, pero que se diferencian en:
 - La cantidad de parámetros
 - El tipo de dato de sus parámetros
 - Orden de los parámetros

- Por lo tanto podemos afirmar que los métodos tienen diferente firma.



SOBRECARGA DE MÉTODOS

```
public class Clase1 {  
  
    public void operacion1() {  
        // Implementación  
    }  
    public void operacion1(int param1) {  
        // Implementación  
    }  
    public void operacion1(String param1) {  
        // Implementación  
    }  
    public void operacion1(int param1, String param2) {  
        // Implementación  
    }  
  
}
```

Clase1
+ operacion1() + operacion1(param1:int) + operacion1(param1:String) + operacion1(param1:int, param2:String)



SOBRECARGA DE MÉTODOS

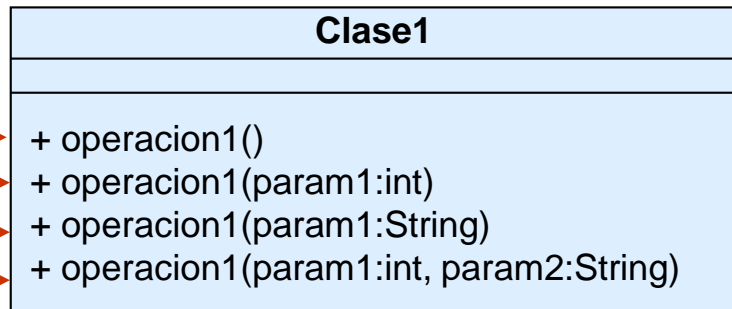
```
Clase1 obj = new Clase1();
```

```
obj.operacion1();
```

```
obj.operacion1(15);
```

```
obj.operacion1("abc");
```

```
obj.operacion1(15,"abc");
```





SOBRECARGA DE CONSTRUCTORES

```
public class Clase1 {  
  
    public Clase1(){  
        // implementación  
    }  
    public Clase1(int param1){  
        // implementación  
    }  
    public Clase1(String param1){  
        // implementación  
    }  
    public Clase1(int param1, String param2){  
        // implementación  
    }  
  
}
```

Clase1
+ Clase1() + Clase1(param1:int) + Clase1(param1:String) + Clase1(param1:int, param2:String)



SOBRECARGA DE CONSTRUCTORES

Clase1 obj = new Clase1();
Clase1 obj = new Clase1(20);
Clase1 obj = new Clase1("abc");
Clase1 obj = new Clase1(20,"abc");

Clase1
+ Clase1() + Clase1(param1:int) + Clase1(param1:String) + Clase1(param1:int, param2:String)



PROYECTO EJEMPLO

La empresa EduTec necesita de una librería que permita calcular el promedio de un conjunto de números.

Se sabe que pueden ser 2, 3, 4 o 5 números.

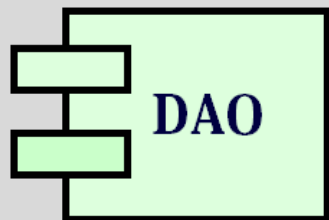
A usted se le ha encargado que desarrolle la librería que necesita Edutec y construya una aplicación de prueba.



CODIGO FUENTE

EUREKA-CS-ORACLE-JDBC

APLICACIÓN JAVA



JDBC

A light blue rounded rectangular box with a dark blue border, representing the JDBC driver component.

CONEXIÓN

A large, light gray double-headed arrow with a black outline, indicating a bidirectional connection between the application and the database.

Esquema

EUREKA

ORACLE XE 11g

Dirección de descarga: <https://goo.gl/TDgc5R>

ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

Gracias

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

