

**CURSO DE FUNDAMENTOS DE JAVA**

# **EJERCICIO**

## **SOBRECARGA DE CONSTRUCTORES EN JAVA**



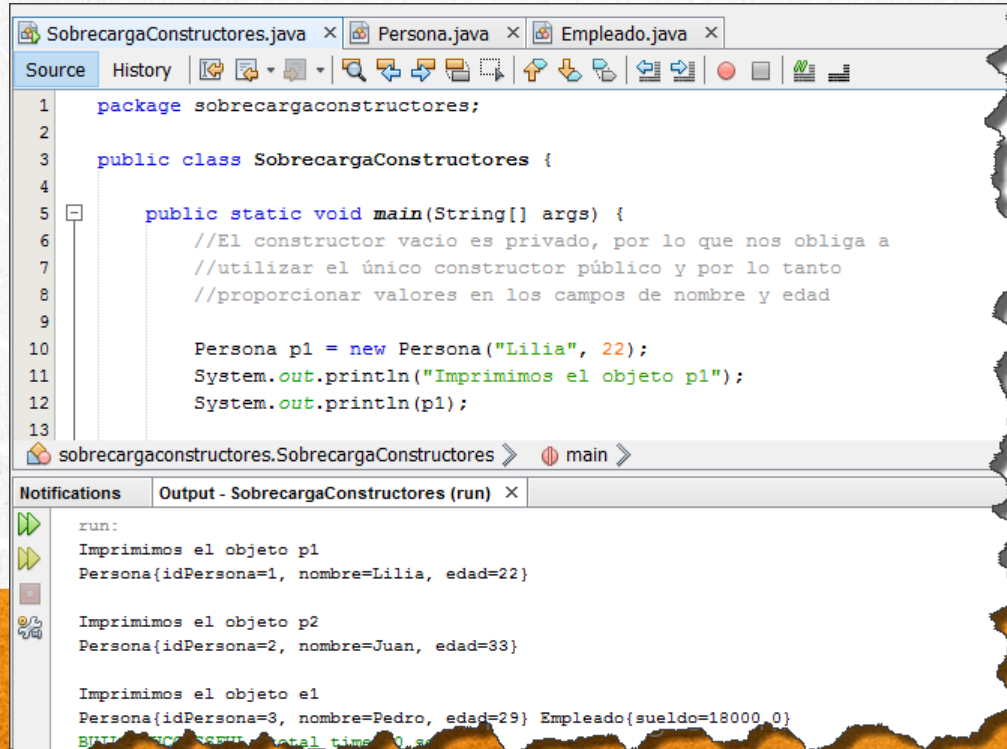
Experiencia y Conocimiento para tu vida

**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# OBJETIVO DEL EJERCICIO

Crear un ejercicio para aplicar el concepto de sobrecarga de Constructores en Java. Al finalizar deberemos observar lo siguiente:



The screenshot shows an IDE with three tabs: `SobrecargaConstructores.java`, `Persona.java`, and `Empleado.java`. The `Source` tab is active, displaying the following Java code in `SobrecargaConstructores.java`:

```
1 package sobrecargaconstructores;
2
3 public class SobrecargaConstructores {
4
5     public static void main(String[] args) {
6         //El constructor vacio es privado, por lo que nos obliga a
7         //utilizar el único constructor público y por lo tanto
8         //proporcionar valores en los campos de nombre y edad
9
10        Persona p1 = new Persona("Lilia", 22);
11        System.out.println("Imprimimos el objeto p1");
12        System.out.println(p1);
13    }
14 }
```

The `main` method is selected in the editor. Below the code editor, the `Output - SobrecargaConstructores (run)` window shows the execution results:

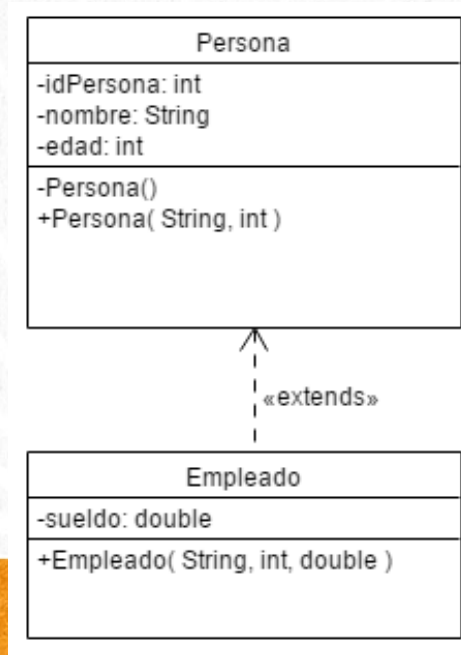
```
run:
Imprimimos el objeto p1
Persona{idPersona=1, nombre=Lilia, edad=22}

Imprimimos el objeto p2
Persona{idPersona=2, nombre=Juan, edad=33}

Imprimimos el objeto e1
Persona{idPersona=3, nombre=Pedro, edad=29} Empleado{sueldo=18000.0}
BUILD SUCCESSFUL (total time: 0.5s)
```

# DIAGRAMA DE CLASES

El siguiente es un diagrama de Clases del ejercicio, creado con la herramienta <http://www.umlet.com/umletino/umletino.html>:





# PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto:

New Java Application

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name:

Project Location:  Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:  Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > **Finish** Cancel Help

# PASO 2. CREAMOS LA CLASE PERSONA

New Java Class

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

< Back   Next >   **Finish**   Cancel   Help

**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 3. MODIFICAMOS EL CÓDIGO

## Archivo Persona.java:

```
package sobrecargaconstructores;

public class Persona {

    private int idPersona;
    private String nombre;
    private int edad;
    private static int contadorPersonas;

    //Constructor sin argumentos y privado
    //Asigna el idPersona
    private Persona() {
        this.idPersona = ++contadorPersonas;
    }

    //Constructor completo sobrecargado
    public Persona(String nombre, int edad) {
        //Se manda a llamar el constructor vacio para que se asigne el idPersona
        this();
        this.nombre = nombre;
        this.edad = edad;
    }

    @Override
    public String toString() {
        return "Persona{" + "idPersona=" + idPersona + ", nombre=" + nombre + ", edad=" + edad + '}';
    }
}
```

# PASO 4. CREAMOS LA CLASE EMPLEADO

New Java Class

×

Steps

1. Choose File Type

2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

< Back

Next >

**Finish**

Cancel

Help

# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo Empleado.java:

```
package sobrecargaconstructores;

public class Empleado extends Persona {

    private double sueldo;

    public Empleado(String nombre, int edad, double sueldo) {
        super(nombre, edad); //Super debe ser la primera linea
        this.sueldo = sueldo;
    }

    public double getSueldo() {
        return sueldo;
    }

    public void setSueldo(double sueldo) {
        this.sueldo = sueldo;
    }

    @Override
    public String toString() {
        //Primero mandamos a llamar el método toString de la clase Persona
        //para que podamos observar los valores de la clase Padre,
        //y despues imprimimos los valores de la clase hija
        return super.toString() + " Empleado{sueldo=" + sueldo + "}";
    }
}
```



# PASO 6. MODIFICAMOS EL CÓDIGO

## Archivo SobrecargaConstructores.java:

```
package sobrecargaconstructores;

public class SobrecargaConstructores {

    public static void main(String[] args) {
        //El constructor vacío es privado, por lo que nos obliga a
        //utilizar el único constructor público y por lo tanto
        //proporcionar valores en los campos de nombre y edad

        Persona p1 = new Persona("Lilia", 22);
        System.out.println("Imprimimos el objeto p1");
        System.out.println(p1);

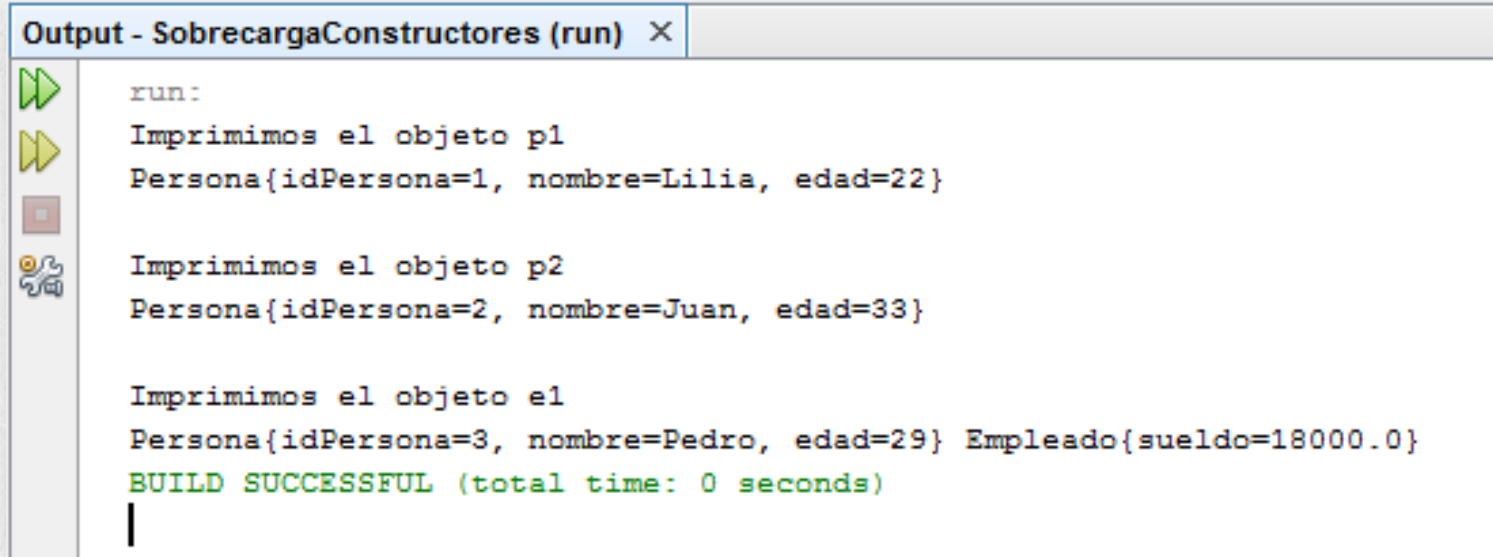
        //Creamos un nuevo objeto de tipo persona
        Persona p2 = new Persona("Juan", 33);
        System.out.println("\nImprimimos el objeto p2");
        System.out.println(p2);

        //Creamos un objeto empleado
        Empleado e1 = new Empleado("Pedro", 29, 18000);
        System.out.println("\nImprimimos el objeto e1");
        System.out.println(e1);

    }
}
```

# PASO 7. EJECUTAMOS EL PROYECTO

Ejecutamos nuestro proyecto. El resultado es:



```
Output - SobrecargaConstructores (run) X
run:
Imprimimos el objeto p1
Persona{idPersona=1, nombre=Lilia, edad=22}

Imprimimos el objeto p2
Persona{idPersona=2, nombre=Juan, edad=33}

Imprimimos el objeto e1
Persona{idPersona=3, nombre=Pedro, edad=29} Empleado{sueldo=18000.0}
BUILD SUCCESSFUL (total time: 0 seconds)
```

# TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Probar con distintos valores y verificar el resultado.
- Crear más variables tanto de objetos Persona, Empleado y verificar resultados.



Experiencia y Conocimiento para tu vida

**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el concepto de Sobrecarga de Constructores.
- Además vimos varios temas relacionados más, que tienen que ver con el buen diseño de nuestras clases.
- Ya estamos empezando a aplicar varios temas que hemos venido trabajando, así que es importante que vayan quedando claros, ya que los seguiremos utilizando en las siguientes lecciones.



**CURSO ONLINE**

# **FUNDAMENTOS DE JAVA**

---

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)