

# CURSO DE FUNDAMENTOS DE JAVA

## ENCAPSULAMIENTO EN JAVA



Por el experto: Ing. Ubaldo Acosta

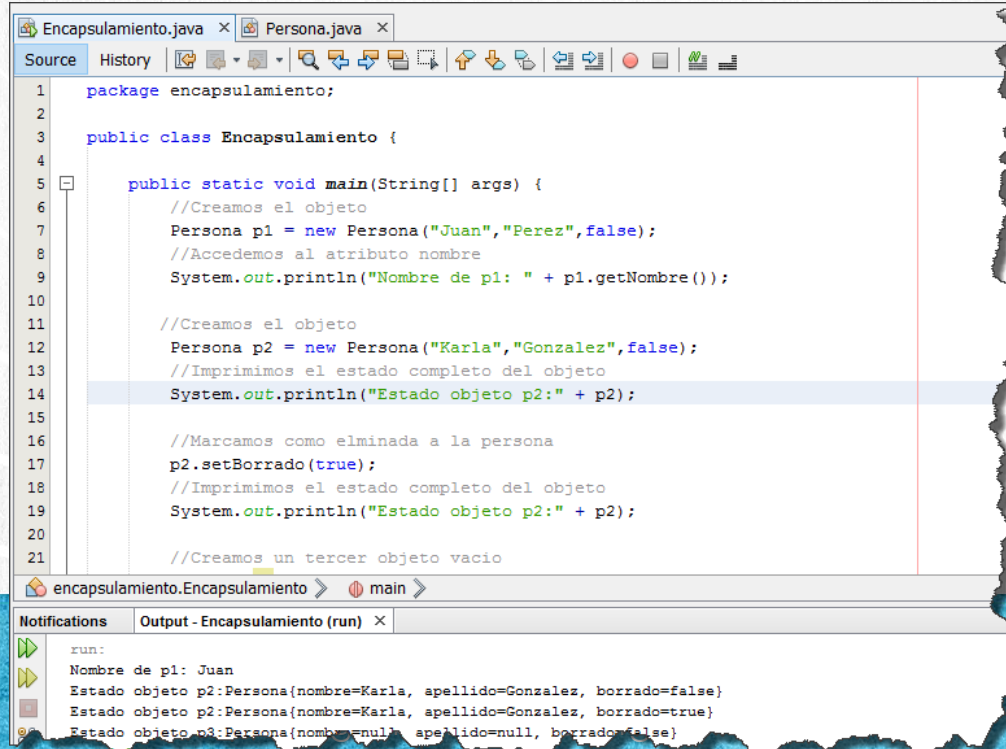


Experiencia y Conocimiento para tu vida



# OBJETIVO DEL EJERCICIO

Poner en práctica el concepto de Encapsulamiento en Java. Al finalizar deberemos observar lo siguiente:



The screenshot shows a Java IDE with two tabs: 'Encapsulamiento.java' and 'Persona.java'. The 'Encapsulamiento.java' tab is active, displaying the following code:

```
1 package encapsulamiento;
2
3 public class Encapsulamiento {
4
5     public static void main(String[] args) {
6         //Creamos el objeto
7         Persona p1 = new Persona("Juan", "Perez", false);
8         //Accedemos al atributo nombre
9         System.out.println("Nombre de p1: " + p1.getNombre());
10
11         //Creamos el objeto
12         Persona p2 = new Persona("Karla", "Gonzalez", false);
13         //Imprimimos el estado completo del objeto
14         System.out.println("Estado objeto p2:" + p2);
15
16         //Marcamos como eliminada a la persona
17         p2.setBorrado(true);
18         //Imprimimos el estado completo del objeto
19         System.out.println("Estado objeto p2:" + p2);
20
21         //Creamos un tercer objeto vacio
```

The IDE's output window at the bottom shows the following results:

```
run:
Nombre de p1: Juan
Estado objeto p2:Persona{nombre=Karla, apellido=Gonzalez, borrado=false}
Estado objeto p2:Persona{nombre=Karla, apellido=Gonzalez, borrado=true}
Estado objeto p3:Persona{nombre=null, apellido=null, borrado=false}
```



# OBJETIVO DEL EJERCICIO

Vamos a crear una clase llamada Persona, la cual aplicaremos el concepto de encapsulamiento como sigue:

Atributos  
privados

Métodos  
públicos

## Clase Persona

- nombre: String
- apellido: String
- borrado: boolean

- + getNombre( ) : String
- + setNombre( String ) : void
- + getApellido( ) : String
- + setApellido( String ) : void
- + isBorrado( ) : boolean
- + setBorrado( boolean ) : void
- + toString ( ) : String

**CURSO DE FUNDAMENTOS DE JAVA**

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

# 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:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

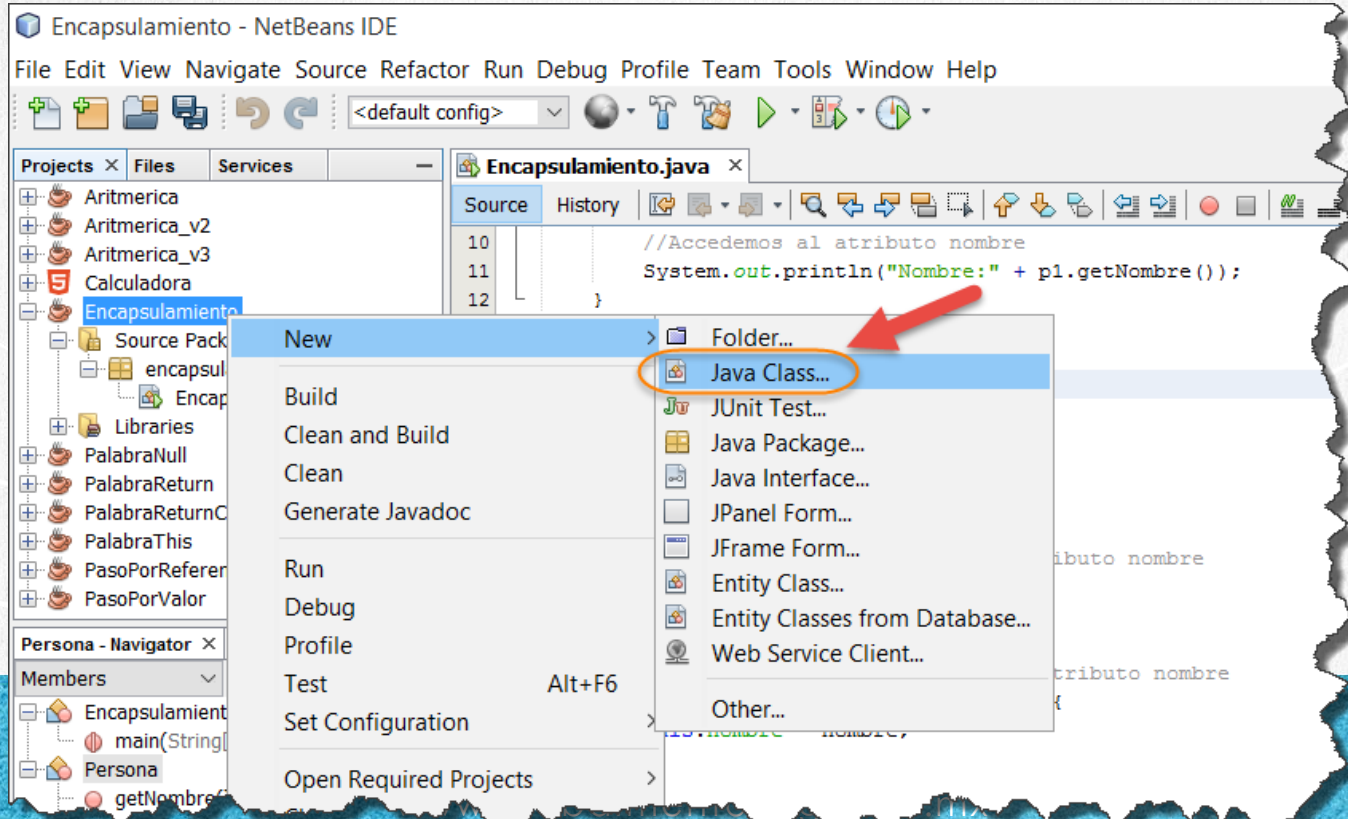
Libraries Folder:

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

☒ Create Main Class

# PASO 2. CREACIÓN CLASE PERSONA

Vamos a crear la clase Persona:





# PASO 2. CREACIÓN CLASE PERSONA (CONT)

Vamos a crear 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

# PASO 3. MODIFICAMOS EL CÓDIGO

## Archivo Persona.java:

```
package encapsulamiento;
```

```
public class Persona {  
    //Constructor Vacio  
    public Persona() {  
    }  
    //Constructor con 3 argumentos  
    public Persona(String nombre, String apellido, boolean borrado) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.borrado = borrado;  
    }  
}
```

```
//Atributos privados
```

```
private String nombre;  
private String apellido;  
private boolean borrado;
```

```
//Métodos publicos para acceder y/o modificar los atributos
```

```
public String getNombre() {  
    return nombre;  
}
```

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

```
public String getApellido() {  
    return apellido;  
}
```

```
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}
```

```
public boolean isBorrado() {  
    return borrado;  
}  
  
public void setBorrado(boolean borrado) {  
    this.borrado = borrado;  
}  
  
//Método que imprime el estado del objeto  
@Override  
public String toString() {  
    return "Persona{" + "nombre=" + nombre + ", apellido=" +  
    apellido + ", borrado=" + borrado + '}';  
}
```

AMENTOS DE JAVA

entoring.com.mx

# PASO 4. MODIFICAMOS EL CÓDIGO

## Archivo Encapsulamiento.java:

```
package encapsulamiento;

public class Encapsulamiento {

    public static void main(String[] args) {
        //Creamos el objeto
        Persona p1 = new Persona("Juan","Perez",false);
        //Accedemos al atributo nombre
        System.out.println("Nombre de p1: " + p1.getNombre());

        //Creamos el objeto
        Persona p2 = new Persona("Karla","Gonzalez",false);
        //Imprimimos el estado completo del objeto
        System.out.println("Estado objeto p2:" + p2);

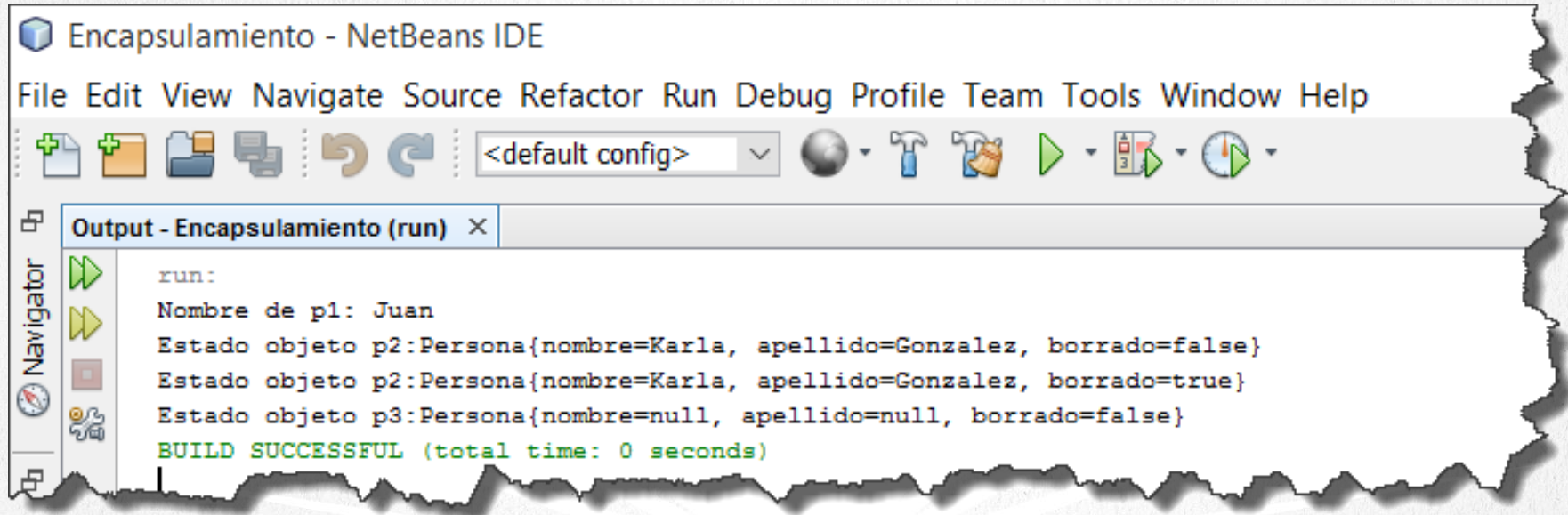
        //Marcamos como eliminada a la persona
        p2.setBorrado(true);
        //Imprimimos el estado completo del objeto
        System.out.println("Estado objeto p2:" + p2);

        //Creamos un tercer objeto vacio
        Persona p3 = new Persona();
        //Imprimimos el estado completo del objeto
        System.out.println("Estado objeto p3:" + p3);
    }
}
```



# PASO 5. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto:



# TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Modificar más atributos e imprimir el estado de cada objeto y también imprimir atributos de manera individual con el objetivo de entender a detalle el manejo del concepto de encapsulamiento.



# CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos puesto en práctica el concepto de Encapsulamiento. Hemos visto cómo utilizar los modificadores de acceso `private` y `public` para lograr este objetivo.

Es importante entender a detalle este concepto, ya que es la base para muchas de las clases que estaremos desarrollando a lo largo de este y los siguientes cursos.

Nota: El método `toString()` nos permite convertir un objeto Java en una cadena, por ello lo utilizamos para poder mandar a imprimir el estado del objeto en Java. Para entender a detalle la explicación de este método necesitamos haber estudiado los temas de herencia y polimorfismo, sin embargo aplicarlo, como hemos visto, es realmente simple, solo debemos agregarlo a nuestra clase y concatenar los atributos para mostrar el estado de nuestro objeto, es decir, los valores de cada atributo, eso es todo lo que debemos hacer para implementarlo, sin embargo para entender su funcionamiento a detalle lo estudiaremos posteriormente.



# CURSO ONLINE

# FUNDAMENTOS DE JAVA

---

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

**CURSO DE FUNDAMENTOS DE JAVA**

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