

PROGRAMACIÓN II

TRABAJO PRÁCTICO 4: Ejercicios Adicionales POO

Alumna: Garcia Galfione, Fiorella

Enlace a repositorio en Github: <https://github.com/FioreGG/UTN-TUPaD-P2.git>

1. Resolución ejercicio 1

```
package ej_adicionales_poo;

public class Ej_Adicionales_POO {

    // Clase Libro - Caso practico 1
    static class Libros {
        // Atributos privados
        private String titulo;
        private String autor;

        // Atributo estatico
        private static String editorial = "Independiente";

        // constructor con titulo y autor
        public Libros(String titulo, String autor){
            this.titulo = titulo;
            this.autor = autor;
        }

        // Constructor con titulo, autor y editorial

        public Libros (String titulo, String autor, String editorial){
            this.titulo = titulo;
            this.autor = autor;
            Libros.editorial = editorial;
        }

        // Metodos sobrecargados actualizarTitulo

        public void actualizarTitulo (String nuevoTitulo){
            this.titulo = nuevoTitulo;}

        public void actualizarTitulo(String prefijo, String nuevoTitulo){
            this.titulo = prefijo + "" + nuevoTitulo;}

        public static void cambiarEditorial(String nueva){
            editorial = nueva;}
```

@Override

```
public String toString(){  
    return "Libro [Titulo=" + titulo +  
        ", Autor="+autor + ", Editorial="+editorial+ "];"
```

//Getters y setters

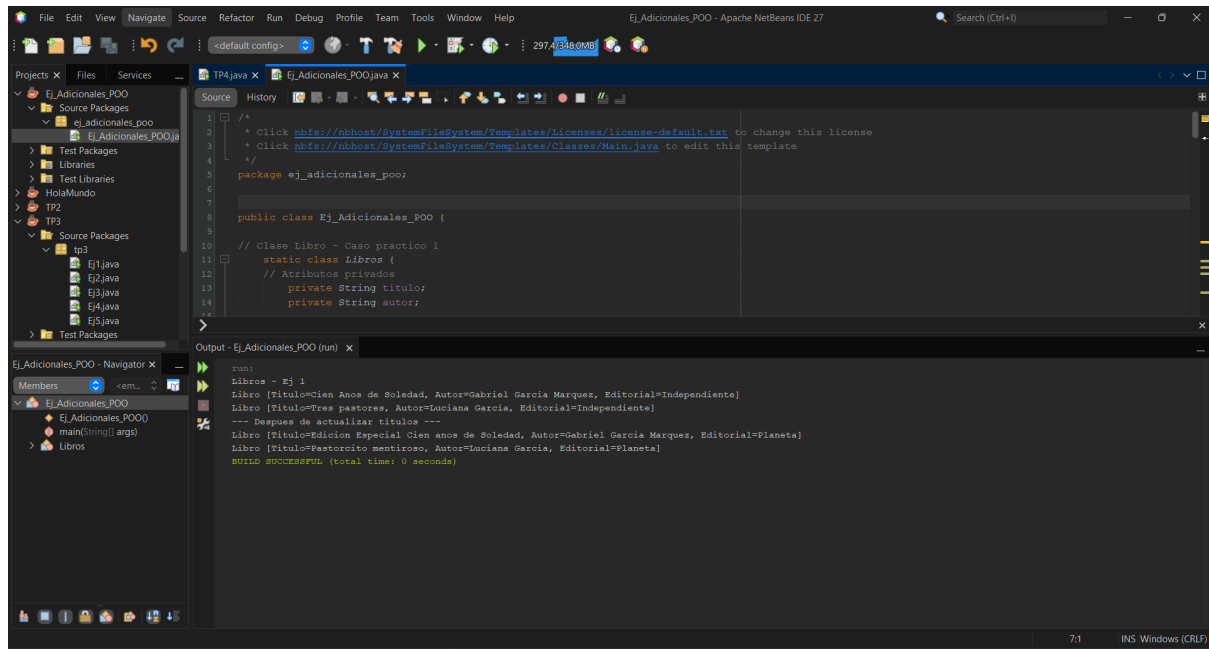
```
public String getTitulo(){return titulo;}  
public String getAutor(){return autor;}  
public String getEditorial(){return editorial;}
```

```
public void setTitulo(String titulo){this.titulo = titulo;}  
public void setAutor(String autor){this.autor = autor;}  
public void setEditorial(String editorial){this.editorial = editorial;}
```

//clase de prueba (main)

```
}  
public static void main(String[] args) {  
    System.out.println("Libros - Ej 1 ");  
  
    // Crear libros  
    Libros l1 = new Libros("Cien Anos de Soledad", "Gabriel Garcia Marquez");  
    Libros l2 = new Libros("Tres pastores", "Luciana Garcia");  
  
    //Mostrar libros  
    System.out.println(l1);  
    System.out.println(l2);  
  
    // Actualizar titulo  
    l2.actualizarTitulo("Pastorcito mentiroso");  
    l1.actualizarTitulo("Edicion Especial ", "Cien anos de Soledad");  
  
    //Actualizar editorial  
    Libros.cambiarEditorial("Planeta");  
  
    //Mostrar titulos actualizados  
    System.out.println("--- Despues de actualizar titulos ---");  
    System.out.println(l1);  
    System.out.println(l2);  
  
}
```

}



2. Resolución ejercicio 2

package ej_adicionales_poo;

```
public class Alumnos {  
    // Clase Alumnos - Caso practico 2  
    static class Alumno {  
        // Atributos privados  
        private String nombre;  
        private double promedio;  
  
        // Atributo estatico  
        private static double notaAprobacion = 6;
```

```
        // constructor  
        public Alumno(String nombre){  
            this.nombre = nombre;  
            this.promedio = 0;  
        }
```

```
        // Constructor con nombre y promedio
```

```
        public Alumno (String nombre, double promedio){  
            this.nombre = nombre;  
            this.promedio = promedio;  
        }
```

```

// Metodos sobrecargados actualizarPromedio

public void actualizarPromedio (double nuevoPromedio){
    this.promedio = nuevoPromedio;}

public void actualizarPromedio(double [] notas){
    double suma = 0;
    for (double n : notas){
        suma += n;}
    this.promedio = notas.length > 0 ? suma / notas.length : 0;}

public boolean aprobo(){
    return this.promedio >= notaAprobacion;}

public static void cambiarNotaAprobacion(double nueva){
    notaAprobacion = nueva;}

@Override

public String toString(){
    return "Alumno [Nombre =" + nombre +
        ", Promedio =" + promedio + ", Aprobo?? =" + (aprobo() ? "Si":"No")+"]";}
}

public static void main(String[] args) {
    System.out.println("Alumnos - Ej 2 ");

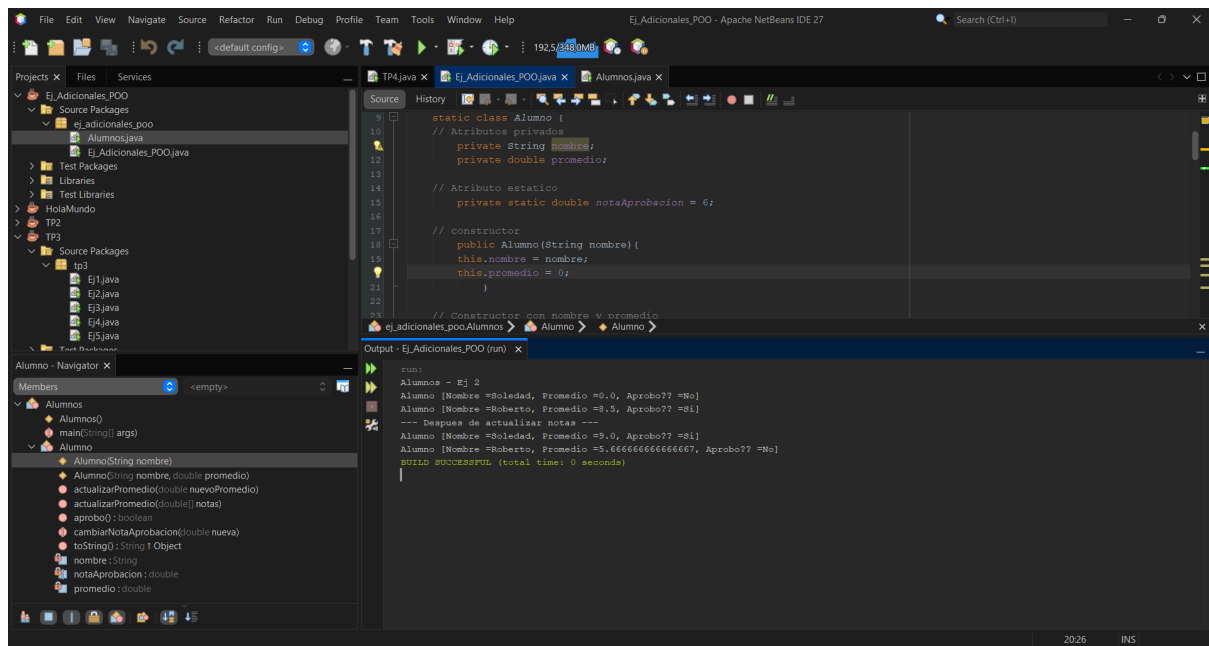
    // Crear alumnos
    Alumno a1 = new Alumno("Soledad");
    Alumno a2 = new Alumno("Roberto", 8.5);

    //Mostrar alumnos
    System.out.println(a1);
    System.out.println(a2);

    // Actualizar promedio
    a1.actualizarPromedio(9.0);
    a2.actualizarPromedio(new double []{4,6,7});

    //Mostrar alumnos actualizados
    System.out.println("--- Despues de actualizar notas ---");
    System.out.println(a1);
    System.out.println(a2);
    }}

```



3. Resolución ejercicio 3

package ej_adicionales_poo;

public class Productos {

// Clase Producto - Caso práctico 3

static class Producto {

// Atributos privados

private String nombre;

private double precioBase;

// Atributo estatico

private static double IVA = 0.21;

// constructor con nombre y precioBase

public Producto(String nombre, double precioBase){

this.nombre = nombre;

this.precioBase = precioBase;

}

// Constructor con nombre solamente

public Producto (String nombre){

this.nombre = nombre;

this.precioBase = 100000;

}

// Metodos sobrecargados aplicarDescuento

```

public void aplicarDescuento (double porcentaje){
    precioBase -= precioBase * (porcentaje/100);}

public void aplicarDescuento(double porcentaje, double precioMinimo){
    double descuento = precioBase*(porcentaje/100);
    double nuevoPrecio = precioBase - descuento;
    if (nuevoPrecio < precioMinimo){
        precioBase = precioMinimo;
    } else {
        precioBase = nuevoPrecio;}
    }

public double calcularPrecioFinal(){
    return precioBase * (1+IVA);}

public static void cambiarIVA(double nuevo){
    IVA = nuevo;}

@Override

public String toString(){
    return "Producto [Nombre = " + nombre +
        ", Precio Base = "+precioBase + " Precio Final con IVA " +
    calcularPrecioFinal()+ "];"
}

public static void main(String[] args) {
    System.out.println("Productos - Ej 2 ");

    // Crear productos
    Producto p1 = new Producto("Mesa carpintera");
    Producto p2 = new Producto("Tablon de 7", 85000.5);

    //Mostrar productos
    System.out.println(p1);
    System.out.println(p2);

    //Aplicar descuentos

    p1.aplicarDescuento(50);
    p2.aplicarDescuento(5, 550000);
    //Mostrar productos
    System.out.println("Productos con descuentos ");
    System.out.println(p1);
    System.out.println(p2);
}

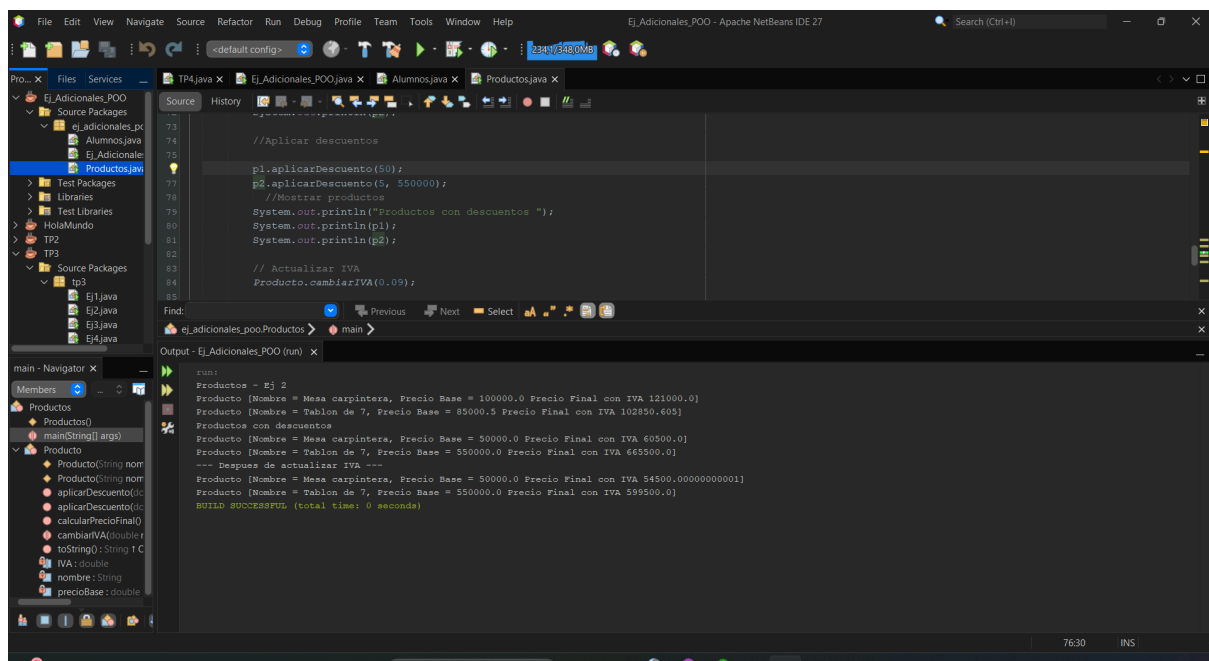
```

```
// Actualizar IVA
Producto.cambiarIVA(0.09);
```

```
//Mostrar productos actualizados
System.out.println("--- Despues de actualizar IVA ---");
System.out.println(p1);
System.out.println(p2);
```

```
}
```

```
}
```



4. Resolución ejercicio 4

```
package ej_adicionales_poo;
```

```
public class Cuentas {
```

```
// Clase Cuentas - Caso práctico 4
```

```
static class Cuenta {
```

```
// Atributos privados
```

```
private int numero;
```

```
private String titular;
```

```
private double saldo;
```

```
private static int ultimoNumero = 100;
```

```
private static int totalCuentas = 0;
```

```

// Constructor con titular (saldo=0)
public Cuenta(String titular) {
    this.numero = ++ultimoNumero;
    this.titular = titular;
    this.saldo = 0;
    totalCuentas++;
}

// Constructor con titular y saldo inicial
public Cuenta(String titular, double saldoInicial) {
    this.numero = ++ultimoNumero;
    this.titular = titular;
    this.saldo = saldoInicial;
    totalCuentas++;
}

// Sobrecarga consultarSaldo
public double consultarSaldo() {
    return saldo;
}

public double consultarSaldo(double cotizacionDolar) {
    if (cotizacionDolar <= 0) return 0;
    return saldo / cotizacionDolar;
}

// Depositar y extraer
public void depositar(double monto) {
    if (monto > 0) {
        saldo += monto;
    }
}

public boolean extraer(double monto) {
    if (monto > 0 && monto <= saldo) {
        saldo -= monto;
        return true;
    }
    return false;
}

public static int mostrarTotalCuentas() {
    return totalCuentas;
}

@Override
public String toString() {
    return "Cuenta [Numero=" + numero +
        ", Titular=" + titular +

```



```

        ", Saldo=" + String.format("%.2f", saldo) + "];
    }
}

public static void main(String[] args) {
    Cuenta c1 = new Cuenta("Maria");
    Cuenta c2 = new Cuenta("Jose", 2000);
    System.out.println(c1);
    System.out.println(c2);
    c1.depositar(500);
    c2.extraer(300);
    System.out.println("Despues de operaciones:");
    System.out.println(c1);
    System.out.println(c2);
    System.out.println("Saldo c2 en dolares (cotizacion 1360): " +
c2.consultarSaldo(1360));
    System.out.println("Total cuentas creadas: " + Cuenta.mostrarTotalCuentas());
}
}

```

The screenshot shows an IDE window titled "Ej_Adicionales_POO - Apache NetBeans IDE 27". The main editor displays the source code of the program, which is the same code shown in the previous block. The left sidebar shows the project structure with "Ej_Adicionales_POO" as the root, containing "ej_adicionales_poo" and "Cuentas.java". The bottom pane shows the output of the program, which is as follows:

```

run:
Cuenta [Numero=101, Titular=Maria, Saldo=0,00]
Cuenta [Numero=102, Titular=Jose, Saldo=2000,00]
Despues de operaciones:
Cuenta [Numero=101, Titular=Maria, Saldo=500,00]
Cuenta [Numero=102, Titular=Jose, Saldo=1700,00]
Saldo c2 en dolares (cotizacion 1360): 1.25
Total cuentas creadas: 2
BUILD SUCCESSFUL (total time: 0 seconds)

```