

## TAREA 3 - ED

---

### EJERCICIO 1- Pruebas unitarias JUnit

Se pide, que crees las pruebas unitarias utilizando JUnit en Java para este código dado:

Código Java: CuentaBancaria.java

```
public class CuentaBancaria {

    private double saldo;

    // Constructor
    public CuentaBancaria(double saldoInicial) {
        if (saldoInicial < 0) {
            throw new IllegalArgumentException("El saldo inicial no puede ser negativo");
        }
        this.saldo = saldoInicial;
    }

    // Método para ingresar dinero en la cuenta
    public void ingresar(double cantidad) {
        if (cantidad <= 0) {
            throw new IllegalArgumentException("La cantidad a ingresar debe ser positiva");
        }
        saldo += cantidad;
    }

    // Método para retirar dinero de la cuenta
    public void retirar(double cantidad) {
        if (cantidad <= 0) {
            throw new IllegalArgumentException("La cantidad a retirar debe ser positiva");
        }
        if (cantidad > saldo) {
            throw new IllegalArgumentException("Saldo insuficiente");
        }
        saldo -= cantidad;
    }

    // Método para obtener el saldo actual de la cuenta
    public double obtenerSaldo() {
        return saldo;
    }
}
```

### Pasos orientativos a seguir:

- Para ello, crea la clase de pruebas: CuentaBancariaTest.java dentro del mismo paquete que la clase CuentaBancaria.
- Anota la clase para usar JUnit 4.

## TAREA 3 - ED

---

- Inicializa la cuenta bancaria que vas a utilizar para las pruebas con un saldo de 6.000€ utilizando el constructor `CuentaBancaria`.
- Escribe 5 test JUNIT para:
  - **testObtenerSaldo()**: Verifica que el saldo inicial sea 6.000€.
  - **testIngresarDinero()**: Verifica que al ingresar dinero, el saldo aumente correctamente.
  - **testRetirarDinero()**: Verifica que al retirar dinero, el saldo disminuya correctamente.
  - **testRetiroExcesivo()**: Verifica que lanzar una excepción si se intenta retirar más dinero del que hay en la cuenta.
  - **testIngresoNegativo()**: Verifica que lanzar una excepción si se intenta ingresar una cantidad negativa.

## EJERCICIO 2 - Depuración en Eclipse

Aquí hay un código Java con un pequeño error, utiliza la depuración para testearlo .

Código Java: CalculadoraFactorial.java

```
public class CalculadoraFactorial {

    public int calcularFactorial(int n) {
        if (n < 0) {
            throw new IllegalArgumentException("El factorial no está definido para números negativos");
        }

        if (n == 0 || n == 1) {
            return 1;
        }

        int resultado = 1;
        for (int i = n; i >= 1; i--) {
            resultado *= i;
        }

        return resultado;
    }

    public static void main(String[] args) {
        CalculadoraFactorial calculadora = new CalculadoraFactorial();
        int resultado = calculadora.calcularFactorial(5);
        System.out.println("El factorial de 5 es: " + resultado);
    }
}
```

**Pasos orientativos a seguir:**

## TAREA 3 - ED

---

- Abre el proyecto en Eclipse que contiene la clase `CalculadoraFactorial.java`
- Coloca un punto de ruptura (breakpoint) en la línea 18, justo antes de la llamada al método `calcularFactorial`.

Inicio de la Depuración:

- Ejecuta la aplicación en modo de depuración haciendo clic derecho en la clase y seleccionando "Debug As" > "Java Application".
- La ejecución se detendrá en el punto de ruptura.

Exploración de Variables:

- Utiliza la perspectiva de depuración en Eclipse para explorar el valor de la variable `n` y otros valores relevantes.
- Utiliza la vista "Variables" para inspeccionar los valores de las variables durante la ejecución.

Ejecución Paso a Paso:

- Utiliza los botones de control de depuración para avanzar la ejecución paso a paso.
- Observa cómo cambian los valores de las variables y sigue el flujo de ejecución.

Identificación y Corrección del Error:

- Identifica el error en el método `calcularFactorial` que causa un resultado incorrecto.
- Realiza la corrección necesaria y reinicia la ejecución para verificar que el cálculo del factorial es correcto.

Realiza capturas de pantalla de tu Eclipse en el que se vea que efectivamente puedes ver el valor de las variables en el lugar donde has puesto el punto de ruptura.

## INDICACIONES DE ENTREGA

Entrega un .pdf en el que haya capturas de pantalla justificativas de haber realizado correctamente los dos ejercicios.

En este pdf habrá capturas en las que se deberá de ver correctamente el código generado para las pruebas JUNIT (EJ1) con los casos de test y la justificación de que se han ejecutado correctamente y los pasos de ejecución del código de depuración (EJ2).

Nombre del fichero a entregar:

- Nombre-apellido1-apellido2.pdf