

## **Informe de la Actividad 2: Javadoc y JUnit – Calculadora**

### **Calculadora en equipo:**

Segunda actividad del módulo EE.DD.

### **Integrantes del equipo:**

- ❖ Manuel Marcote Codesido - Perfil de GitHub “ManuelMarcoteC”
- ❖ Marta Guillén Escudero - Perfil de GitHub “mge1905”
- ❖ Pablo Garrido López - Perfil de GitHub “Pablo-Garridolop”
- ❖ Adrián Araujo González - Perfil de GitHub “adriovi87”

### **Repositorio GitHub compartido:**

<https://github.com/adriovi87/CalculadoraActividad2>

### **Profesor:**

- ❖ Luis García Sánchez - Perfil de GitHub “luisgs-unir”

**Fecha de entrega:** 24/02/2025

## **Introducción**

La presente actividad tiene como objetivo principal la implementación y documentación de una calculadora en Java, haciendo uso de Javadoc para la generación de documentación. Además se han realizado pruebas unitarias de cada operación utilizando JUnit, garantizando así la funcionalidad del código generado.

Como decimos, el proyecto ha consistido en dos bloques de acción:

- ❖ Por un lado se han documentado un conjunto de clases con operaciones aritméticas donde cada clase genera métodos diferentes para abordar bien casos especiales, bien un conjunto de números distinto.
- ❖ Por otro lado se han elaborado pruebas unitarias que nos permitieran validar la ejecución de los métodos implementados, con atención a casos límite y excepciones.

Este documento formal tiene por finalidad detallar el proceso seguido durante el desarrollo del proyecto, las metodologías empleadas, los retos encontrados y las soluciones aplicadas.

## Metodología

### 1. Desarrollo del Código y la Documentación

- ❖ Implementación: Se desarrollaron las clases correspondientes, cuidando la legibilidad del código y la correcta implementación de las funcionalidades requeridas. Cada clase incluyó su correspondiente comentario inicial con el nombre del autor, URL del perfil GitHub y la URL del repositorio compartido.
- ❖ Documentación con JavaDoc: Paralelamente, se generó la documentación de cada clase y método utilizando JavaDoc, siguiendo las directrices especificadas en el enunciado. Se documentaron tanto los casos generales como los casos especiales (ej. manejo de cero, desbordamiento, etc.).
- ❖ Pruebas unitarias con JUnit: Se desarrollaron pruebas unitarias para cada clase, abarcando todos los métodos y verificando tanto los escenarios normales como los casos de error. Se prestó atención a la cobertura de excepciones y condiciones límite.

### 2. Integración y Revisión

- ❖ Fusiones y resolución de conflictos: Durante el desarrollo se realizaron múltiples integraciones a través de Git, lo que implicó resolución de conflictos y sincronización de los cambios. La comunicación constante entre integrantes

facilitó la coordinación y permitió detectar y solucionar problemas de integración.

### 3. Elaboración del Documento Formal

- ❖ Redacción y estructuración: Una vez finalizada la implementación, se procedió a la redacción de este documento formal, que incluye una introducción, la metodología seguida, los problemas encontrados y las soluciones aplicadas, así como las conclusiones y el desglose de tareas.

## Problemas y soluciones

Hemos encontrado problemas en la creación del repositorio remoto, así como en el orden de subida de archivos del proyecto en su estado inicial. Creímos en un inicio que la estructura global del proyecto debía ser el esqueleto de la rama main, pero luego vimos que era más rápida y eficiente la escritura del código según se había distribuido.

También encontramos problemas a la hora de realizar pull & request debido a la diferente configuración de proyectos que escogimos individualmente. Finalmente bastó con añadir la carpeta *.idea* al archivo *.gitignore*.

Asimismo nos encontramos con fallas en la distribución y escritura del JavaDoc, desde falta de datos hasta diferentes avisos del IDE que nos comunicaron huecos en nuestra documentación. Como remedio, atajamos los warnings y los errores resaltados en consola en la generación de la documentación según las pautas sugeridas.

## Conclusiones

Ha sido todo un reto coordinarse y salir ciertamente airosos de algo tan poco intuitivo como pueden ser las primeras tomas de contacto con estas herramientas de control de versiones, así como el IDE de escritorio IntelliJ. Creemos que sin tanta

carga y con algo más de tiempo habríamos podido sacarle un mayor partido, nos toca seguir aprendiendo a utilizar estas herramientas para así poder trabajar en equipo como desarrolladores de software.

## Desglose de tareas

- ❖ Manuel Marcote Codesido - Perfil de GitHub “ManuelMarcoteC”:
  - Diseño y documentación JavaDoc de la clase Producto.
  - Diseño de pruebas unitarias TestProducto según JUnit, con su correspondiente documentación JavaDoc.
  - Diseño del menuCalculadora y diseño del menuProducto con su documentación asociada.
- ❖ Marta Guillén Escudero - Perfil de GitHub “mge1905”:
  - Diseño y documentación JavaDoc en la clase Resta
  - Diseño de las pruebas unitarias de RestaTest
  - Diseño de menuResta con documentación asociada
- ❖ Pablo Garrido López - Perfil de GitHub “Pablo-Garridolop”:
  - Diseño y documentación JavaDoc de la clase Cociente.
  - Diseño de las pruebas unitarias de CocienteTest.
  - Diseño de menuCociente con documentación asociada.
- ❖ Adrián Araujo González - Perfil de GitHub “adriovi87”
  - Diseño y documentación JavaDoc de la clase Suma.
  - Diseño de las pruebas unitarias de SumaTest.
  - Diseño de menuSuma con documentación asociada.