Asignatura: Entornos de desarrollo – 12736 DAW.

Proyecto: Actividad 2. Javadoc y JUnit.

Nombre del profesor: Luis García.

Alumnos:

David Gómez Moreno.

Alberto Balaguer Gómez.

David González Alcañiz.

Zaira González Encabo.

Francisco José Soria Navarrete.

URL repositorio: https://github.com/David-Gonzalez-Alcaniz/Actividad2-Javadoc-JUnit.git

1. Introducción.

En este proyecto, nuestro equipo desarrollará una calculadora en Java**,** asegurando una correcta documentación mediante JavaDoc, utilizando IntelliJ y GitHub para la gestión del código en equipo. Cada integrante ha sido responsable de la implementación y documentación de una clase específica, siguiendo las mejores prácticas de programación y control de versiones.

Para lograr una colaboración eficiente, cada desarrollador trabajará dentro de un repositorio compartido en GitHub. Una vez implementadas y documentadas sus funciones, deberán realizar los *commits* necesarios y sincronizar su código con el repositorio remoto.

Además, contaremos con una clase principal (main) que permitirá probar las diferentes funcionalidades de la calculadora. Cada integrante deberá modificar esta clase para probar los métodos de la clase que ha implementado. Este proceso requerirá una comunicación efectiva dentro del equipo para gestionar posibles conflictos en la fusión del código.

El proyecto será evaluado considerando la calidad del código, la documentación en JavaDoc y el uso correcto de Git. Este trabajo no solo fortalecerá nuestras habilidades en IntelliJ y Git, sino que también fomentará el trabajo en equipo y la gestión de proyectos colaborativos, habilidades esenciales en el desarrollo de software.

2. Metodología.

En primer lugar, hicimos un reparto y organización del trabajo, asignando las clases y tests a cada miembro del equipo.

Cada persona se encargó de crear su clase, hacer *push* al repositorio remoto y confirmar que el resto de los compañeros del grupo lo visualizan correctamente. Después, cada integrante probó la clase de otro compañero asignado previamente, realizando las pruebas de los tests y, una vez comprobado que todo funciona correctamente, actualizar la información del repositorio.

3. Problemas y soluciones.

Al principio, tuvimos problemas al realizar la estructura del proyecto, descargando el repositorio sin haber creado la estructura previa en IntelliJ, por lo que, al querer subir las clases, nos daba error constantemente y los cambios no se mostraban visibles. La solución fue empezar de 0, crear un repositorio nuevo con su correspondiente estructura Java enlazado a IntelliJ y, finalmente, hacer el primer *commit* al repositorio para que los compañeros pudiesen

descargar el proyecto y, partiendo desde dicha estructura, poder trabajar sin dar lugar a errores.

Otro de los problemas fue dado por no estar habituados al entorno IntelliJ y, cada vez que abres el proyecto, es necesario aceptar la SDK correspondiente (en nuestro caso, Temurin-21) que nos permita acceder correctamente a los archivos, código fuente y documentación y así poder ejecutar nuestra aplicación sin errores ni líneas en rojo.

4. Conclusiones.

Hemos encontrado ciertos retos a la hora de realizar los @test, ya que era la primera vez que utilizamos este entorno de trabajo y este requiere de cierta destreza para comprobar que el código funciona correctamente. Para ello, es necesario saber cómo funciona el método *"assertEquals",* el cual nos permite comprobar el valor esperado con el resultado de la operación. Una vez comprendida la dinámica, se hace muy cómodo probar los métodos de una forma rápida.

Ha sido un trabajo desafiante, ya que los integrantes hemos tenido que sincronizar nuestro flujo de trabajo, de forma que fuese entendida por el resto de los compañeros y así perder el menor tiempo en explicar cuáles eran los avances. Gracias a los *commits* esto fue posible.

Finalmente, después de mucho trabajo, hemos conseguido adaptarnos y familiarizarnos con el nuevo entorno de trabajo y desempeñar la actividad de manera efectiva.

5. Desglose de tareas.

Alberto Balaguer:

- Creación clase división y JavaDoc correspondiente.
- Creación test módulo y realización de las correspondientes pruebas en JUnit.
- Revisión general del proyecto, generar documentación del JavaDoc y ultimar detalles.
- Participación, aporte de ideas y búsqueda de información para resolver problemas (test) y facilitar el trabajo de los demás compañeros.

David Gónzalez:

- Creación clase producto y JavaDoc correspondiente.
- Creación test suma y realización de las correspondientes pruebas en JUnit.
- Creación del menú general de la aplicación.
- Creación estructura IntelliJ y repositorio del proyecto.

- Revisión general del proyecto, generar documentación del JavaDoc y ultimar detalles.

David Gómez:

- Creación clase resta y JavaDoc correspondiente.
- Creación test producto y realización de las correspondientes pruebas en JUnit.
- Creación del documento Word explicativo de la actividad.
- Pionero en explorar el entorno IntelliJ y detectar los primeros errores de estructura.

Zaira Gonzalez:

- Creación clase modulo y JavaDoc correspondiente.
- Creación test resta y realización de las correspondientes pruebas en JUnit.
- Participación y colaboración activa en la creación/modificación de los documentos del proyecto.

Francisco Soria:

- Creación clase suma y JavaDoc correspondiente.
- Creación test división y realización de las correspondientes pruebas en JUnit.
- Creación del menú general de la aplicación.
- Revisión de fallos en la composición del menú general de la aplicación.
- Mentor del grupo en cuestiones de programación (Java).