### Trabajo de Entornos de Desarrollo

# ACTIVIDAD #2

### Desarrollo de una calculadora en Java en entorno colaborativo.



Presentado por:   
 Nelson Rosales, Kyle Lamm

GitHub Respositorio:

<https://github.com/nelson2411/actividad_two_javaDoc_jUnit.git>

Introducción:

* En esta actividad, crearemos una calculadora sencilla que tendrá clases individuales como sumar, restar, dividir y multiplicar. Dentro de estas clases, probaremos múltiples funciones como el uso de diferentes tipos (números enteros, números reales) y diferentes cantidades de argumentos.

Metodología:

* En esta actividad, combinaremos varias herramientas para crear y ejecutar nuestro código. En primer lugar, se crea un repositorio en el que se puede derivar la base de nuestro trabajo y se establece un lugar de trabajo común. A continuación, cada miembro del equipo clonará una copia del repositorio en su sistema local. A continuación, cada miembro creará el código que le haya sido asignado utilizando el IDE de su elección (Intellij, Eclipse, etc.) teniendo en cuenta que debe utilizar el formato correcto para crear comentarios compatibles con Javadoc. A continuación, cada miembro utilizará Git para enviar su código al repositorio utilizando una rama separada. Después, las ramas se fusionarán y una vez más se extraerán del repositorio a su directorio de trabajo local y, a continuación, se crearán e implementarán las pruebas. Las pruebas en esta actividad se llevarán a cabo a través del framework Junit. Después de crear e implementar las pruebas, los miembros del equipo enviarán su trabajo al repositorio para su revisión y el ciclo se repetirá tantas veces como sea necesario hasta que el trabajo esté terminado a satisfacción de cada miembro.

Problemas y Soluciones:

* Uno de los problemas con los que nos encontramos fue el de enviar y fusionar correctamente nuestro trabajo en el repositorio desde nuestros espacios de trabajo locales. Esto se resolvió comunicándonos y ayudándonos mutuamente a entender dichas confusiones y dificultades y utilizando adecuadamente los comandos y procesos correctos de Git y GitHub.

Conclusiones:

* A veces, la tarea puede resultar difícil cuando se utiliza GitHub y los comandos y procesos correctos para garantizar que el código se ha clonado correctamente y no hay discrepancias. Sin embargo, después de aclimatarnos al proceso, la tarea se hizo más fácil y GitHub fue una buena herramienta para crear dispositivos de seguridad y asegurar que se evitan errores y discrepancias. Junit también ha demostrado ser una gran herramienta para crear y probar el código escrito y encontrar errores que no eran fácilmente visibles o difíciles de detectar.

Desglose de Tareas:

* Kyle Lamm ha creado la clase Cociente y probado la clase Suma que fue escrita por Danielrguezm.