

Los integrantes del equipo son:

MARIA CARMEN CORREA HERAS
JUAN RAMÓN VARÓ NÚÑEZ

VERÓNICA BONIS MARTÍN

A continuación se presentan las tareas a realizar y el método de trabajo elegido:

Requerimiento 1

Para resolver el primer requerimiento mantenemos una corta reunión (a través de Discord, ya que esta aplicación nos permite compartir texto, pantalla y voz) en la que definimos la estructura de las pruebas a realizar en cada clase y métodos:

- Clase 1: Suma: Daniel del Río Pérez → (Juan Ramón Varó)
- Clase 2: Resta: Juan Ramón Varó
- Clase 3: Producto: Verónica Bonis Martín
- Clase 4: Cociente: María del Carmen Correa Heras

En este caso testaremos cada clase individualmente (prueba unitaria) que nos servirá para ver si el código funciona correctamente. Cada miembro del equipo plantea su código incluyendo las siguientes anotaciones utilizada para las pruebas unitarias:

- @Test: Anotación que deben llevar todos los test.
- @Before: métodos que serán invocados antes de cada test
- @BeforeClass: métodos que serán invocados antes de ejecutar los test.
- @After: métodos que serán invocados después de cada test
- @AfterClass: métodos que serán invocados después de ejecutar los test.
- @Ignore: sirve para marcar que un test no debe ser tenido en cuenta a la hora de ejecutar los test unitarios.

CREACIÓN DE LOS TEST UNITARIOS

Primero creamos la **carpeta Test** al mismo nivel que la carpeta src en nuestro proyecto de Eclipse.

Una vez creado, hacemos click derecho en la carpeta Test recién creada, y seleccionamos la opción **build path**, a través de la cual entramos a otro grupo de opciones en el cual elegimos la opción de configuración, que se llama Configure Build Path -> source -> añadimos la carpeta test al classpath.

Debemos configurar la carpeta Test y debemos darnos cuenta de las librerías dependientes que necesitamos activas en nuestro paquete de pruebas, y marcarlas.

Creamos tantas carpetas de pruebas como clases hay en nuestro proyecto, haciendo click derecho en la carpeta Test -> new -> JUnit Test Case, y nombrándolos despectivamente,

nosotros le hemos llamado a los métodos de prueba con la palabra Test seguida del nombre del método al que hace referencia:

- TestSuma
- TestResta
- TestCociente
- TestProducto

En cada método probamos el caso genérico y los casos especiales con la anotación @test y aserciones, cuya librería importamos. La aserción que hemos utilizado ha sido la **assertEquals()** con parámetros, según fuera el tipo. En algunos casos de double requería incluir un parámetro epsilon ya que los double pueden no ser exactamente iguales y permite establecer lo cerca que tienen que estar. Si no lo hacíamos así daba error.

Los métodos before y afterall, before y aftereach, en realidad los dejamos presentes, pero no van a probar nada en nuestro proyecto. Está comentado dentro del código.

También importamos las clases suma, resta producto y cociente, la que corresponda.

Subimos todo mediante Git al repositorio común de Github.

<https://github.com/carwenblue/EntornosDesarrolloTelefonica>