**Conociendo el proyecto**

1. El proyecto BlueJ “calculadora” contiene una construcción parcial del sistema. Revisen el directorio donde se encuentra el proyecto. Describan el contenido considerando los directorios y las extensiones de los archivos.

* Tenemos 12 archivos los cuales terminan con .java, .ctxt y .class.

2. Exploren el proyecto en BlueJ ¿Cuántas clases tiene? ¿Cuál es la relación entre ellas? ¿Cuál es la clase principal? ¿Cómo la reconocen? ¿Cuáles son las clases “diferentes”? ¿Cuál es su propósito?

* Tiene 3 clases la cuales son SuperCalculadora, Velocidad y Ángulo. Entre ellas existe la relación de uso. La clase principal es SuperCalculadora ya que esta genera las demás clases. la clase diferente es AnguloTest ya que esta maneja un tipo de formato diferente; tipo test siendo su propósito hacer pruebas a la clase Angulo.

Para las siguientes dos preguntas sólo consideren las clases “normales”:

3. Generen y revisen la documentación del proyecto; ¿está completa la documentación de cada clase? (Detalle el estado de documentación de cada clase: encabezado y métodos)

* La clase SuperCalculadora no está documentada, la clase Velocidad tiene documentacion pero no completa. Y la clase Angulo maneja más información en su documentación.

4. Revisen el código del proyecto, ¿en qué estado está cada clase? (Detalle el estado de codificación)

* Todas las clases están incompletas, porque en varios de sus métodos no se encuentran implementados correctamente.

**Ingeniería reversa**

MDD MODEL DRIVEN DEVELOPMENT

1. Genere el diagrama de clases correspondiente a Calculadora con todos sus elementos. (No incluya la clase de pruebas)

2. ¿Qué tipos de contenedores tienen sus colecciones? Consulte el API Java 1¿Qué diferencias hay entre ellos?

**Conociendo Pruebas en BlueJ**

**De TDD → BDD (TEST → BEHAVIOUR DRIVEN DEVELOPMENT)**

1. Revisen el código de la clase AnguloTest. ¿cuáles etiquetas tiene (componentes con símbolo @)? ¿cuántos métodos tiene? ¿cuantos métodos son de prueba? ¿cómo los reconocen?

* Before, Test, After
* Tiene en total 7 metodos, de prueba hay 5 metodos. Los reconocemos por medio de la etiqueta @Test.

2. Ejecuten los tests de la clase AnguloTest. (click derecho sobre la clase, Test All) ¿cuántos tests se ejecutan? ¿cuántos pasan las pruebas? ¿por qué?

* Se ejecutan 5 tests, ninguno pasa porque en todos se generan fallas.

3. Estudie las etiquetas encontradas en 1. Expliquen en sus palabras su significado.

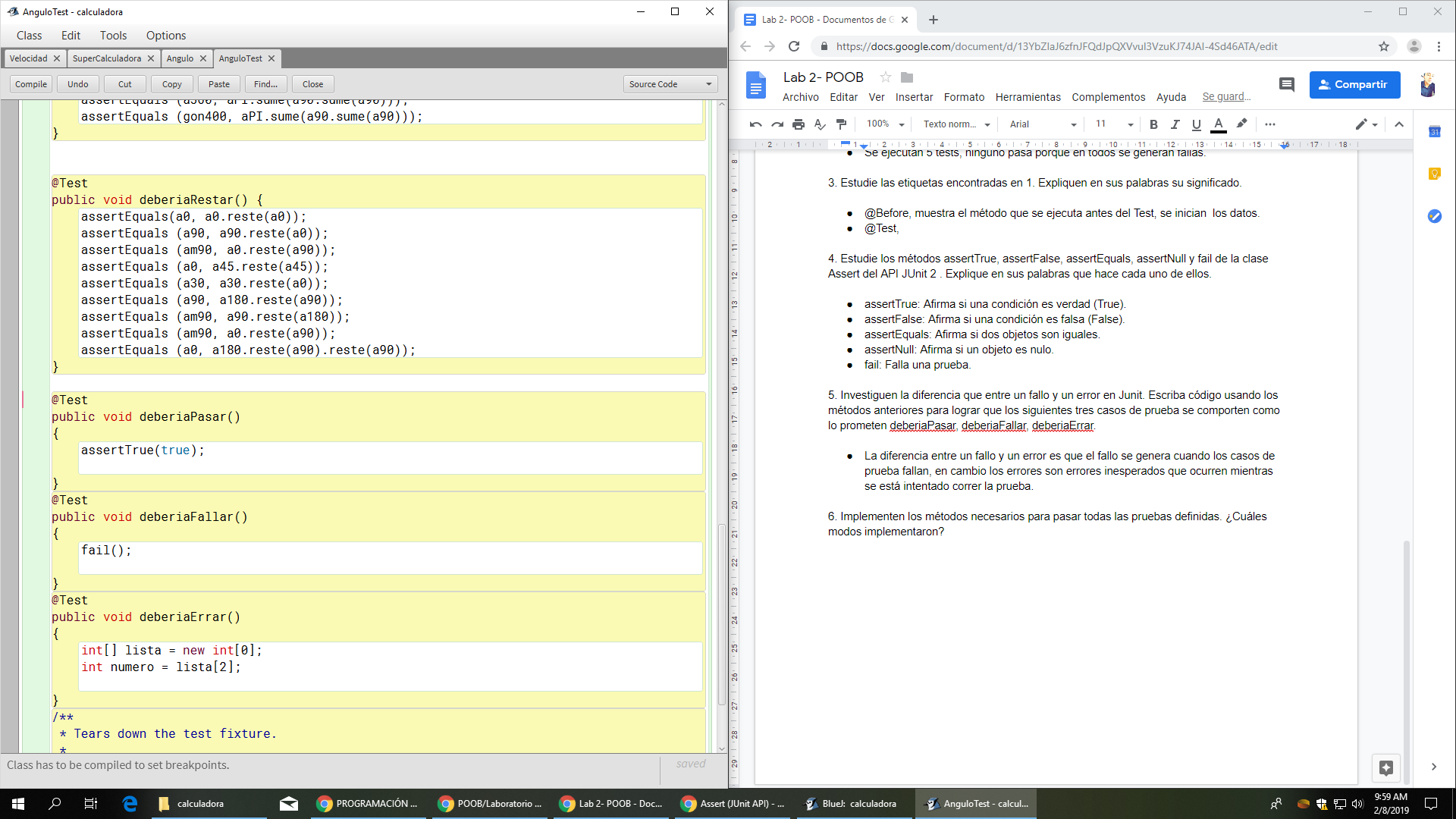
* @Before, muestra el método que se ejecuta antes del Test, se inician los datos.
* @Test,

4. Estudie los métodos assertTrue, assertFalse, assertEquals, assertNull y fail de la clase Assert del API JUnit 2 . Explique en sus palabras que hace cada uno de ellos.

* assertTrue: Afirma si una condición es verdad (True).
* assertFalse: Afirma si una condición es falsa (False).
* assertEquals: Afirma si dos objetos son iguales.
* assertNull: Afirma si un objeto es nulo.
* fail: Falla una prueba.

5. Investiguen la diferencia que entre un fallo y un error en Junit. Escriba código usando los métodos anteriores para lograr que los siguientes tres casos de prueba se comporten como lo prometen deberiaPasar, deberiaFallar, deberiaErrar.

* La diferencia entre un fallo y un error es que el fallo se genera cuando los casos de prueba fallan, en cambio los errores son errores inesperados que ocurren mientras se está intentado correr la prueba.



6. Implementen los métodos necesarios para pasar todas las pruebas definidas. ¿Cuáles modos implementaron?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ciclo | CalculadoraTest | Calculadora | Velocidad | Angulo |
| 1 |  | adicione  consulte | toString | toString |
| 2 |  | elimine  duplicar |  |  |
| 3 |  | calcule  calcule | sume  reste  vectorial  escalar |  |
| 4 |  |  | componenteH  componenteV | seno  coseno |
| 5 |  | calcule |  |  |

**1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)**

10 horas en total, 5 horas por cada persona.

**2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?**

El estado actual es terminado, y completo. Ya que cumple con todos los requerimientos que se piden en el laboratorio.

**3. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?**

Las práctica XP más útil para nosotros fue realizar el código de prueba primero antes que todo, ya que este método de práctica genera un mayor orden y entendimiento al momento de realizar el código final.

**4. ¿Cuál consideran fue el mayor logro? ¿Por qué?**

El mayor logro fue poder establecer clases de una manera más ordenada y más limpias sin necesidad de sobrecargar cada clase.

**5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?**

El mayor problema técnico fue el manejo y la creación del stack. Para resolver esto nos informamos con la documentación adecuada sobre este.

**6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?**

Realizamos un buen trabajo manejando una buena distribución de tareas, sin sobrecargar a la otra persona. Nos comprometemos a documentarnos más acerca de las características de java.