**Pruebas de Software**



**TP Integrador 2020**

***Docente a cargo:* Eduardo Diez**

***Instructor Ayudante: Pablo San Román***

***Alumno: Oscar Ruina***

***DNI: 35639606***

**2-Planilla de especificación de casos de prueba**

Test Unitarios.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Nombre** | **Condiciones de Entrada** | **Pre-Condición** | **Resultado Esperado** | **Resultado** |
| 1 | testGetValordeDificultad | -Objeto Dificultad creado.  -3 variables tipo entero creadas | Inicializo las tres variables con los valores 10,15 y 7  Instancio el objeto dificultad pasando como parámetro las variables. | True, al verificar que el método getValorDeDificultad() devuelva un valor igual a la suma de las tres variables anteriores. | True |
| 2 | testIgualA | -2 Objetos de tipo Posición creados.  -4 variables numéricas creadas | -inicializo las variables(los pares x y con igual valor en las 4 variables).  -Instancio los objetos de tipo Posición con las variables. | True, al verificar que el método igualA() devuelve true. | True |
| 3 | testDistanciaHasta | -2 Objetos de tipo Posición creados.  -5 variables numéricas creadas | -Inicializo las variables, calculo la distancia entre las variables.  -Instancio los objetos de tipo Posición con las variables. | True, al verificar que el método distanciaHasta() devuelve un valor igual a la variable distancia calculada anteriormente | True |
| 4 | testPuedeAterrizarHelicoptero | -Objetos Posición, Helipuerto, Helicóptero creados.  -2 variables numéricas creadas | -inicializar las variable con valores numéricos.  -instanciar los objetos de tipo Posición y Helipuerto. | True, al verificar que el método puedeAterrizarHelicoptero() devuelve true al ser un objeto tipo Helicoptero el que puede aterrizar. | True |
| 5 | testPuedeAterrizarAvionSimple | -Objetos Posición, Helipuerto, Avión Simple creados.  -2 variables numéricas creadas | -inicializar las variable con valores numéricos.  -instanciar los objetos de tipo Posición y Helipuerto. | False, al verificar que el método puedeAterrizarAvionSimple() devuelve false al ser un objeto tipo AvionSimple el que no puede aterrizar. | False |

Test de Integración.

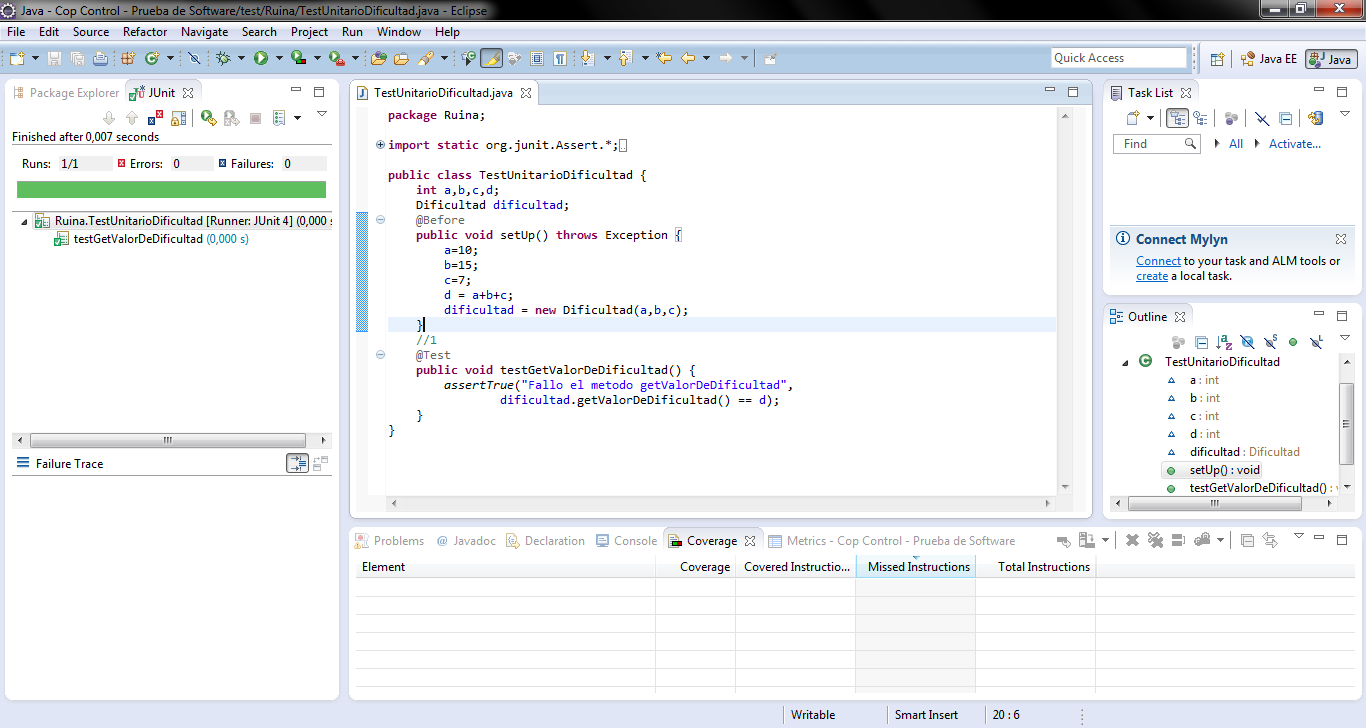
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Nombre** | **Condiciones de Entrada** | **Pre-Condición** | **Resultado Esperado** | **Resultado** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**3-Captura de Pantalla de los resultados**

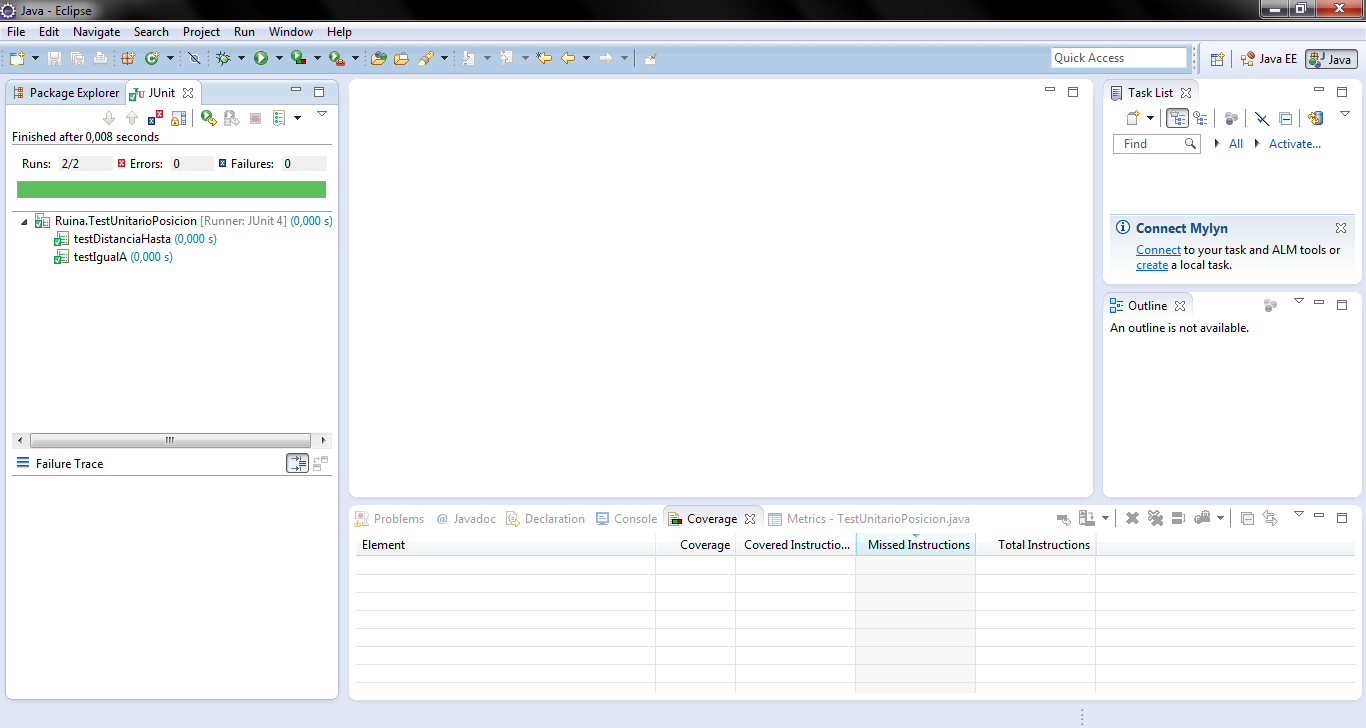
Coloco el resultado del test más el código.

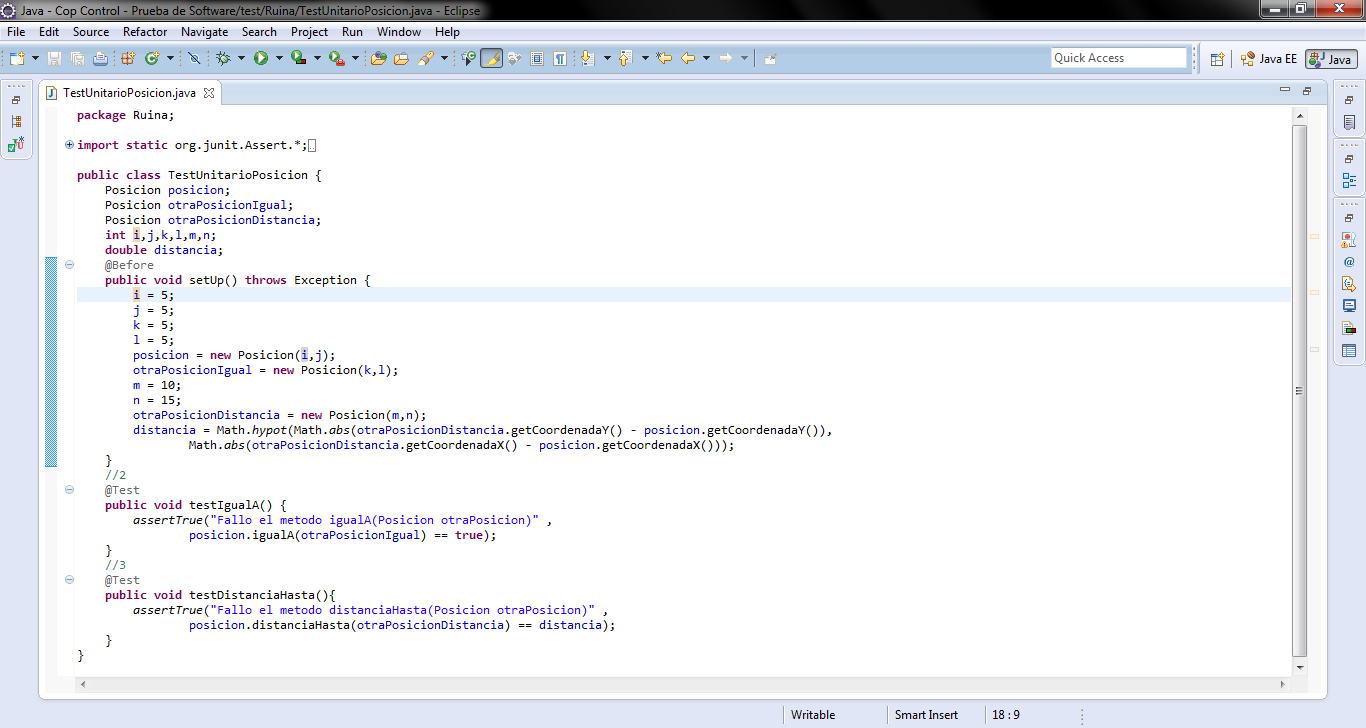
**Test Unitarios:**

Test 1:

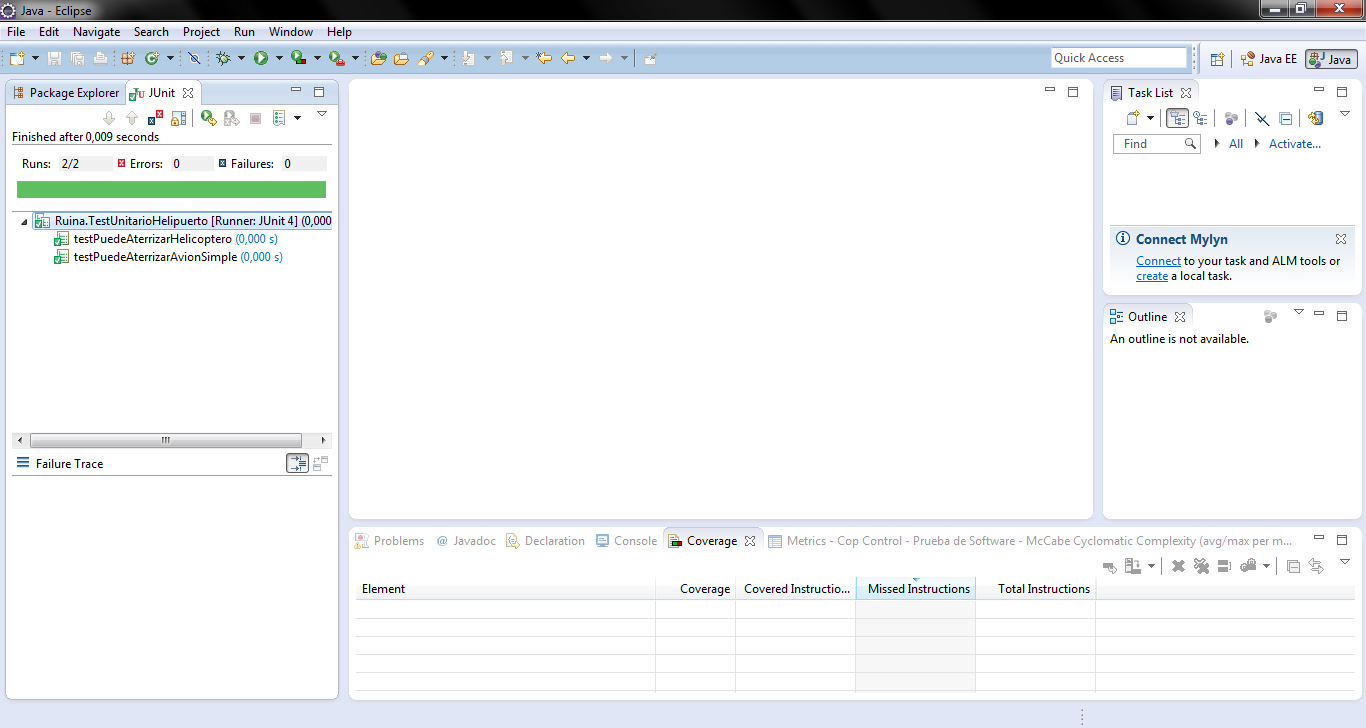


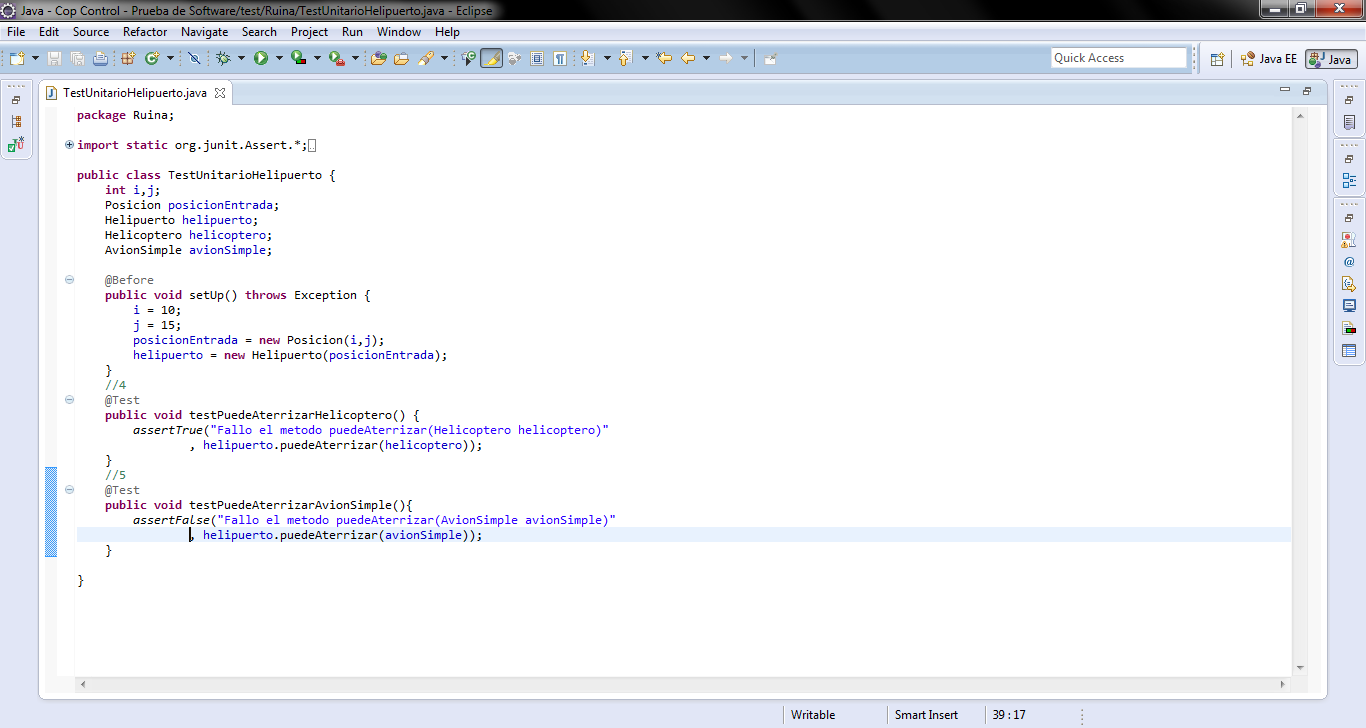
Test 2 y Test 3:





Test 4 y 5





**5-Metodos con más de 24 líneas de código**

moverHacia – getVecinos - generarPosicionExtremaAlAzar

**6-Metodos con más de 4 parámetros**

Intersects – intersects

**7-Metodos con complejidad ciclo matica mayor a 4**

moverHacia – generarPosicionExtremoRecta – generarPosicionExtremaAlAzar – crearAvionAlAzar-chocoConOtroAvion – nuevaVistaDePista -nuevaVistaDeAvion