



JUnit

Documentación

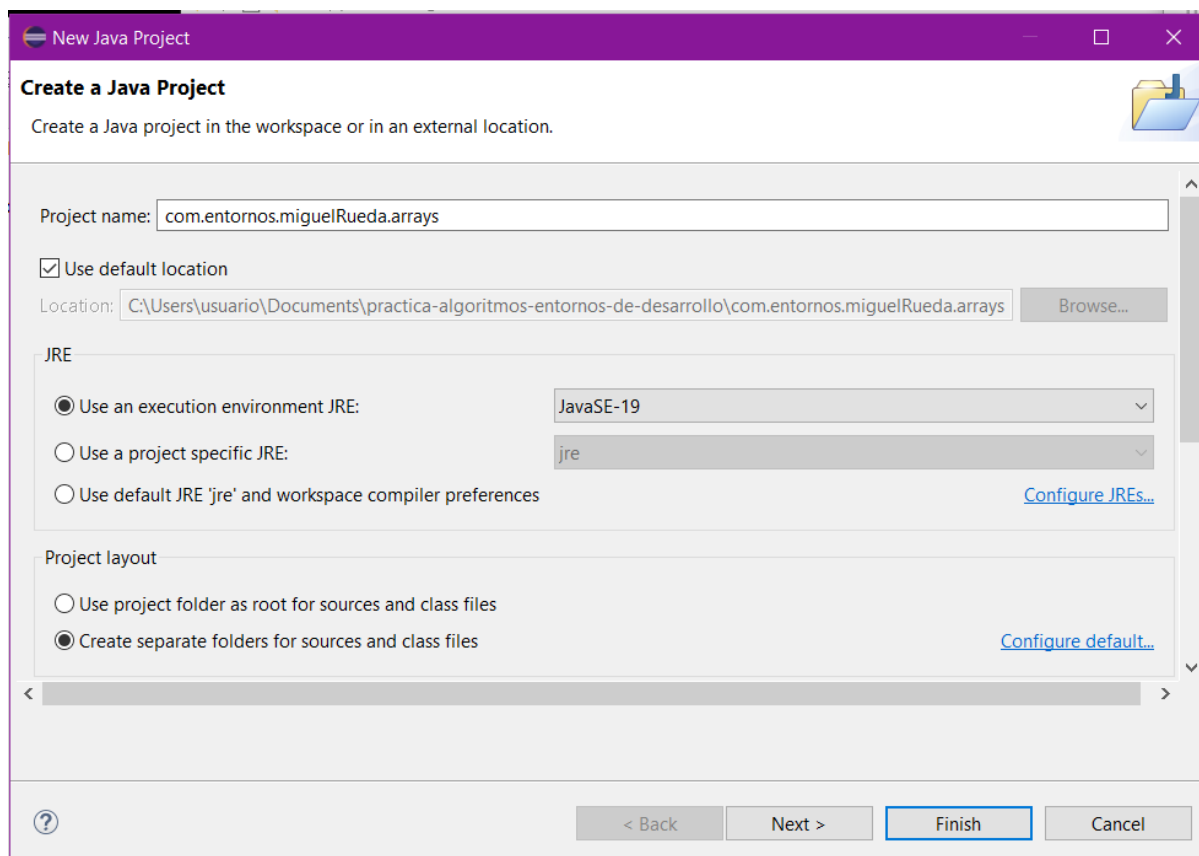
Git

**Miguel Angel Rueda
Moreno**



Antes de nada, debemos enlazar el espacio de trabajo que asignamos en clase al ordenador personal, a través del comando `git clone` y la url del repositorio, comprobar con `git fetch` y `git status` que todo vaya bien, y seguimos trabajando.

Captura del nuevo proyecto de Java:



Commit a los cambios y subida al repositorio.

```
C:\Users\usuario\Documents\practica-algoritmos-entornos-de-desarrollo>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   com.entornos.miguelRueda.arrays/.classpath
    new file:   com.entornos.miguelRueda.arrays/.gitignore
    new file:   com.entornos.miguelRueda.arrays/.project
    new file:   com.entornos.miguelRueda.arrays/.settings/org.eclipse.core.resources.prefs
    new file:   com.entornos.miguelRueda.arrays/.settings/org.eclipse.jdt.core.prefs
    new file:   com.entornos.miguelRueda.arrays/src/module-info.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

C:\Users\usuario\Documents\practica-algoritmos-entornos-de-desarrollo>git commit -m "Creacion del proyecto de Java"
[main 48cc939] Creacion del proyecto de Java
```

Creamos una clase abstracta "Arrays.java" con los siguientes métodos:

→ media - Recibe un array de números enteros y devuelve la media con decimales de dicho array.

```
4  /**
5   * Devolver la media de una serie de numeros enteros
6   * @param Recibe por parametro una estructura array de numeros enteros
7   * @return La división entre el sumatorio de los numeros y la cantidad de valores. La media.
8   */
9  public static double media (int [] numeros) {
10     int sumatorio=0;
11     for(int numero:numeros) {
12         sumatorio+=numero;
13     }
14     return (double)sumatorio/numeros.length;
15 }
```

Test correcto.

```
11
12 @TestInstance(Lifecycle.PER_CLASS)
13 class testMedia {
14
15     static int [] arrayNegativo;
16     static int [] arrayPositivos;
17     static int [] arrayCorto;
18
19
20     @BeforeAll
21     void preparar() {
22         arrayNegativo= new int [] {-72, -10, -1, -90, -23, -28};
23         arrayPositivos = new int [] {24, 72, 31, 99, 10, 51};
24         arrayCorto = new int [] {15, 10, 6, 1};
25     }
26
27
28     @Test
29     void testMedia() {
30         double resultado1=MisArrays.media(arrayNegativo);
31         double resultadoEsperado1=-224.0/6;
32         double resultado2=MisArrays.media(arrayPositivos);
33         double resultadoEsperado2=287.0/6;
34         double resultado3=MisArrays.media(arrayCorto);
35         double resultadoEsperado3=32.0/4;
36
37         assertEquals(resultadoEsperado1, resultado1);
38         assertEquals(resultadoEsperado2, resultado2);
39         assertEquals(resultadoEsperado3, resultado3);
40     }
41 }
```




→ mediana - Recibe un array de números enteros y devuelve la mediana con decimales de dicho array.

```
17- /**
18-  * Devolver la mediana de una serie de numeros enteros en un array
19-  * @param Recibe por parametro una estructura array de numeros enteros
20-  * @return La mediana de la serie de numeros recibidos
21-  */
22- public static double mediana (int [] numeros) {
23-     if(numeros.length%2==0) {
24-         return ((double)numeros[(numeros.length/2)-1]+numeros[numeros.length/2])/2;
25-     }
26-     else {
27-         return (double)numeros[numeros.length/2];
28-     }
29- }
```

Test correcto.

Runs: 2/2 Errors: 0 Failures: 0

> testMedia [Runner: JUnit 5] (0.015 s)

Failure Trace

```
26
27
28- @Test
29- void testMedia() {
30-     double resultado1=MisArrays.media(arrayNegativo);
31-     double resultadoEsperado1=-224.0/6;
32-     double resultado2=MisArrays.media(arrayPositivos);
33-     double resultadoEsperado2=287.0/6;
34-     double resultado3=MisArrays.media(arrayCorta);
35-     double resultadoEsperado3=32.0/4;
36-
37-     assertEquals(resultadoEsperado1, resultado1);
38-     assertEquals(resultadoEsperado2, resultado2);
39-     assertEquals(resultadoEsperado3, resultado3);
40- }
41- @Test
42- void testMediana() {
43-     double resultado1=MisArrays.mediana(arrayNegativo)
44-     double resultadoEsperado1=-91.0/2;
45-     double resultado2=MisArrays.mediana(arrayPositivos)
46-     double resultadoEsperado2=130.0/2;
47-     double resultado3=MisArrays.mediana(arrayCorta);
48-     double resultadoEsperado3=8.0;
49-
50-     assertEquals(resultadoEsperado1, resultado1);
51-     assertEquals(resultadoEsperado2, resultado2);
52-     assertEquals(resultadoEsperado3, resultado3);
53- }
```

→ maximo - Recibe un array de números enteros y devuelve el valor máximo de dicho array.

```
30  /**
31   * Devolver el numero mayor de una serie de valores enteros
32   * @param Recibe por parametro un array de numeros enteros
33   * @return Devuelve el numero mayor
34   */
35  public static int maximo (int [] numeros) {
36      int mayor=numeros[0];
37      for (int numero:numeros) {
38          if (numero>mayor) mayor=numero;
39      }
40      return mayor;
41  }
```

Test correcto.

The screenshot shows an IDE window with a JUnit test runner on the left and source code on the right. The test runner indicates that the test 'testMedia' passed successfully after 0.167 seconds, with 3/3 runs, 0 errors, and 0 failures. The source code on the right includes the 'maximo' method, which is highlighted in blue. The code is as follows:

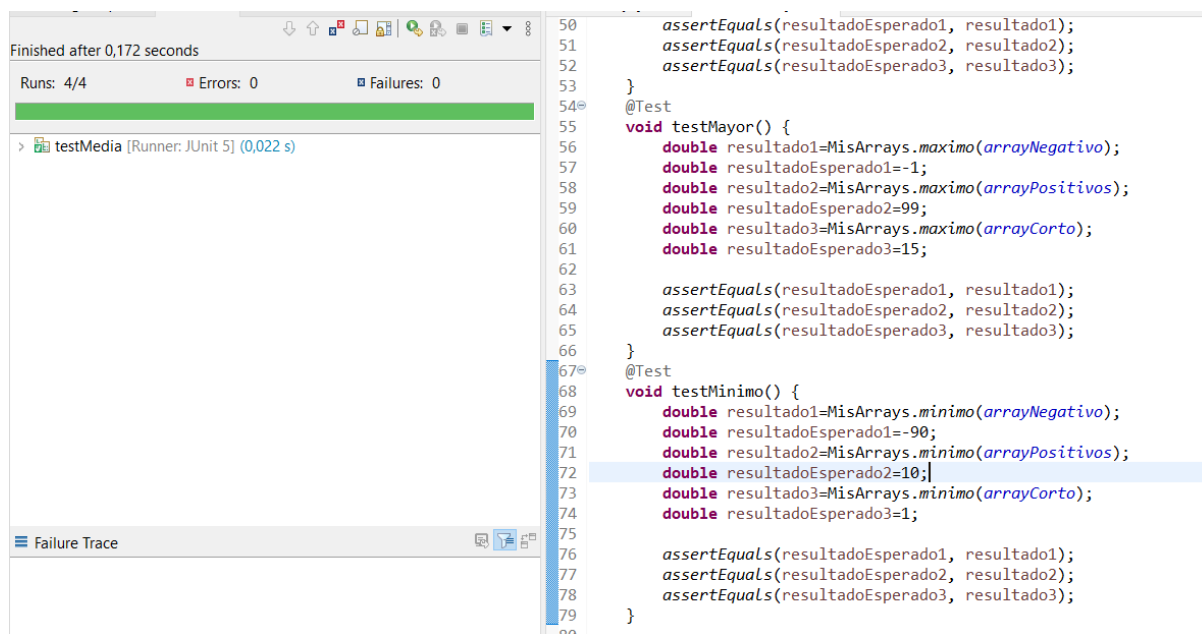
```
13  }
14  return (double)sumatorio/numeros.length;
15  }
16
17  /**
18   * Devolver la mediana de una serie de numeros enteros en un array
19   * @param Recibe por parametro una estructura array de numeros enteros
20   * @return La mediana de la serie de numeros recibidos
21   */
22  public static double mediana (int [] numeros) {
23      if(numeros.length%2==0) {
24          return ((double)numeros[(numeros.length/2)-1]+numeros[numeros.length/2])/2;
25      }
26      else {
27          return (double)numeros[numeros.length/2];
28      }
29  }
30
31  /**
32   * Devolver el numero mayor de una serie de valores enteros
33   * @param Recibe por parametro un array de numeros enteros
34   * @return Devuelve el numero mayor
35   */
36  public static int maximo (int [] numeros) {
37      int mayor=-9999;
38      for (int numero:numeros) {
39          if (numero>mayor) mayor=numero;
40      }
41      return mayor;
42  }
```



→ minimo - Recibe un array de números enteros y devuelve el valor mínimo de dicho array.

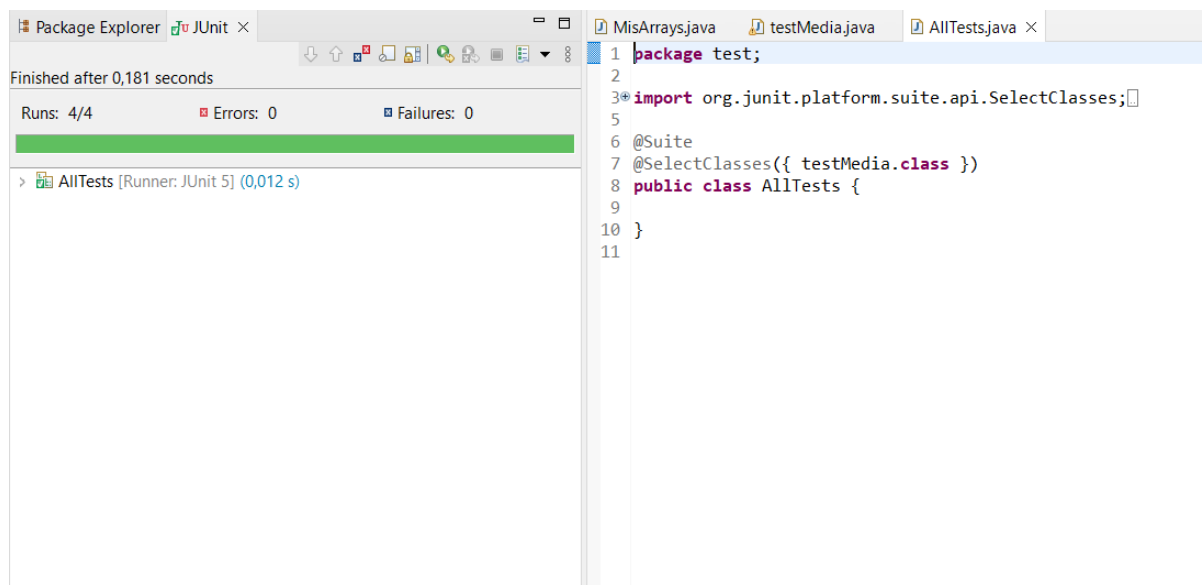
```
42  /**
43   * Devolver el numero menor de una serie de valores enteros
44   * @param Recibe por parametro un array de numeros enteros
45   * @return Devuelve el numero menor
46   */
47  public static int minimo (int [] numeros) {
48      int minimo=numeros[0];
49      for (int numero:numeros) {
50          if (numero<minimo) minimo=numero;
51      }
52      return minimo;
53  }
54  }
```

Test correcto.

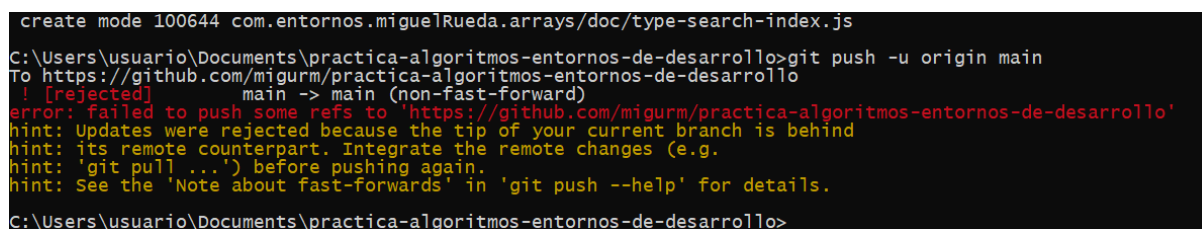




Suite.



Alejandra ayer cuando me fuí a casa intenté seguir trabajando en el repositorio que cree por la mañana en la clase, la cosa es que me lo debí de cargar, porque ni pude subir nada desde allí, y hoy al volver a clase e intentar empezar a subir los métodos uno a uno desde el pc de aquí tampoco me está dejando y no se muy bien si crear un nuevo repositorio o no sé.



Este es el error que me está dando desde el pc de clase.

Te mando el proyecto de java con su javadoc y los test.

