

HDT 10

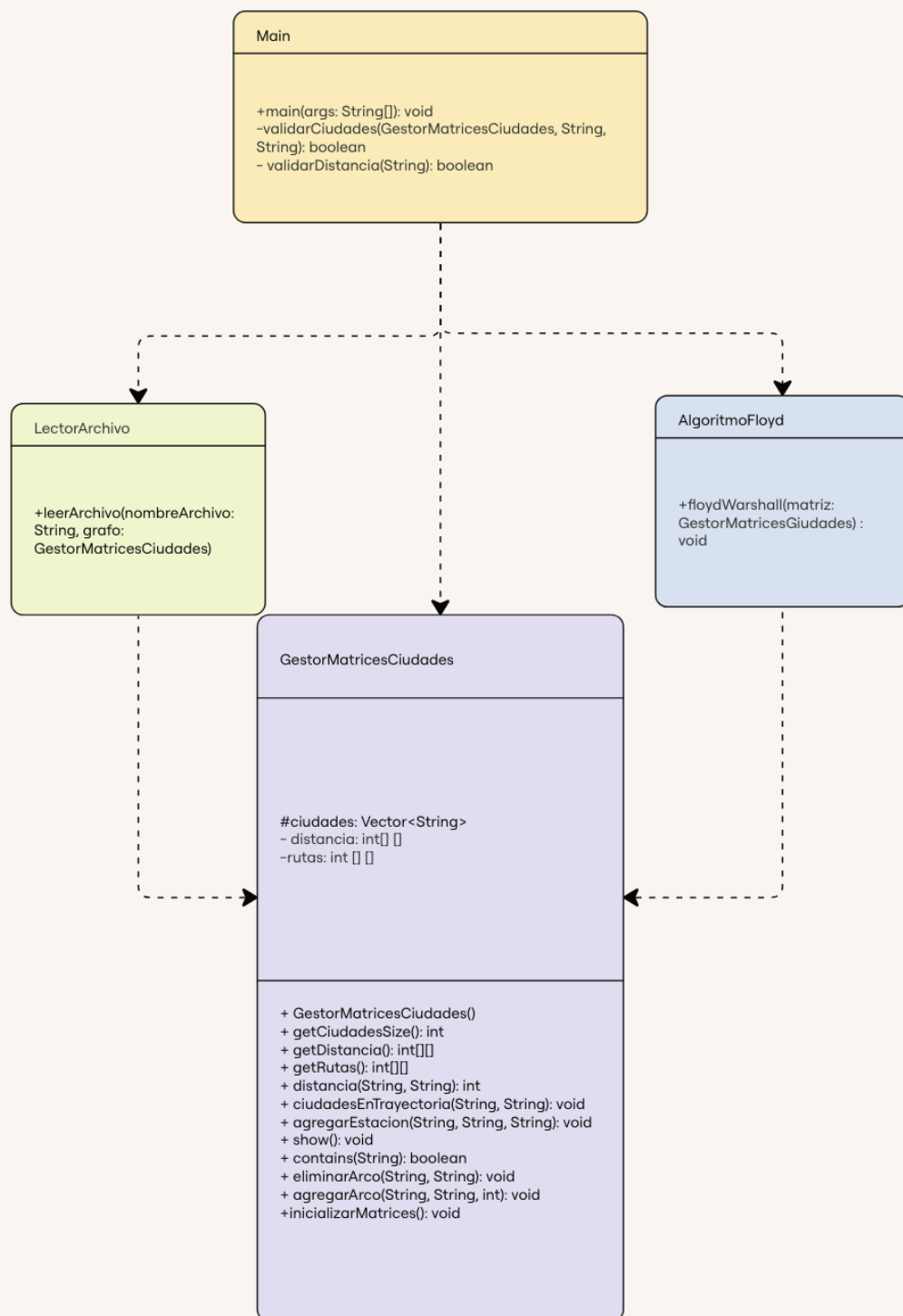
Programa de implementación de grafos para algoritmo de Floyd.

Vianka Vanessa Castro Ordoñez - 23201

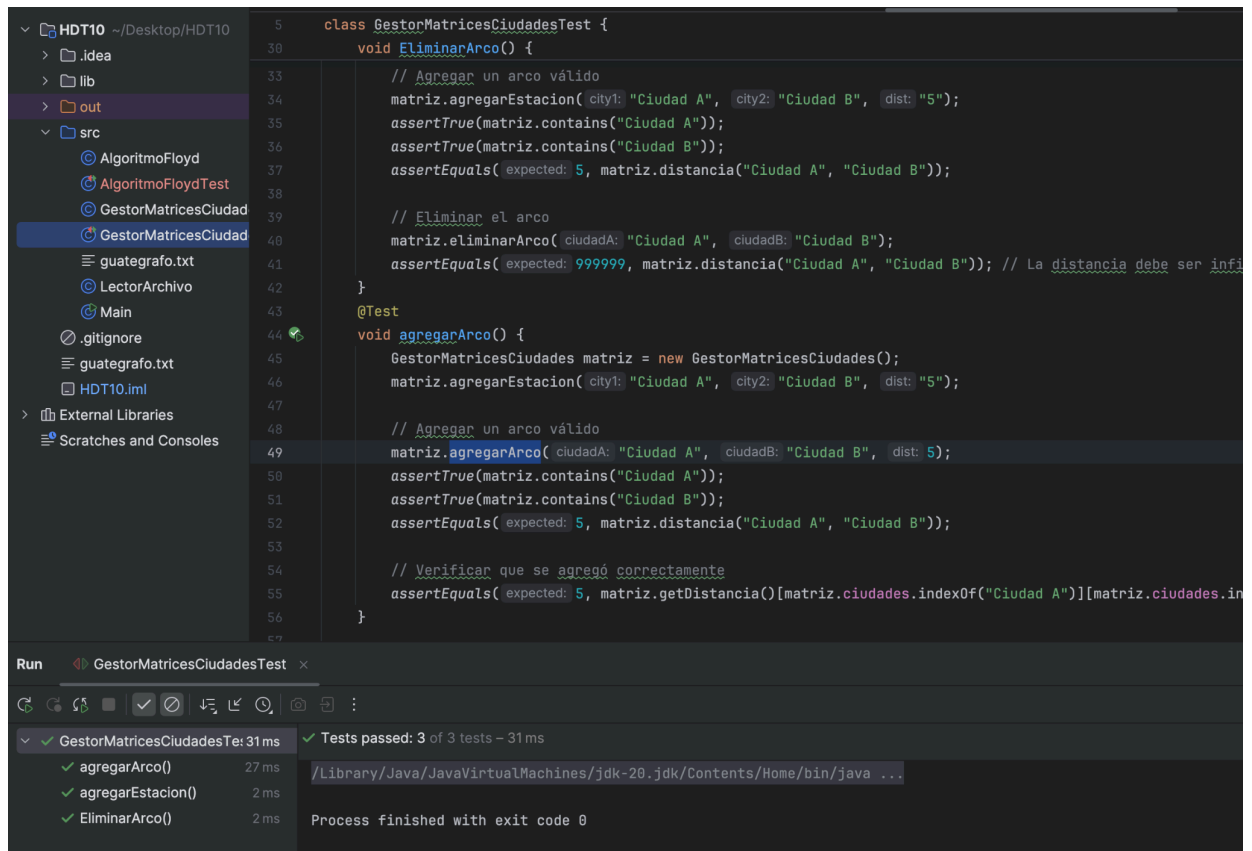
Repositorio Github

<https://github.com/Vann06/HDT10>

UML



Pruebas Unitarias JUnit



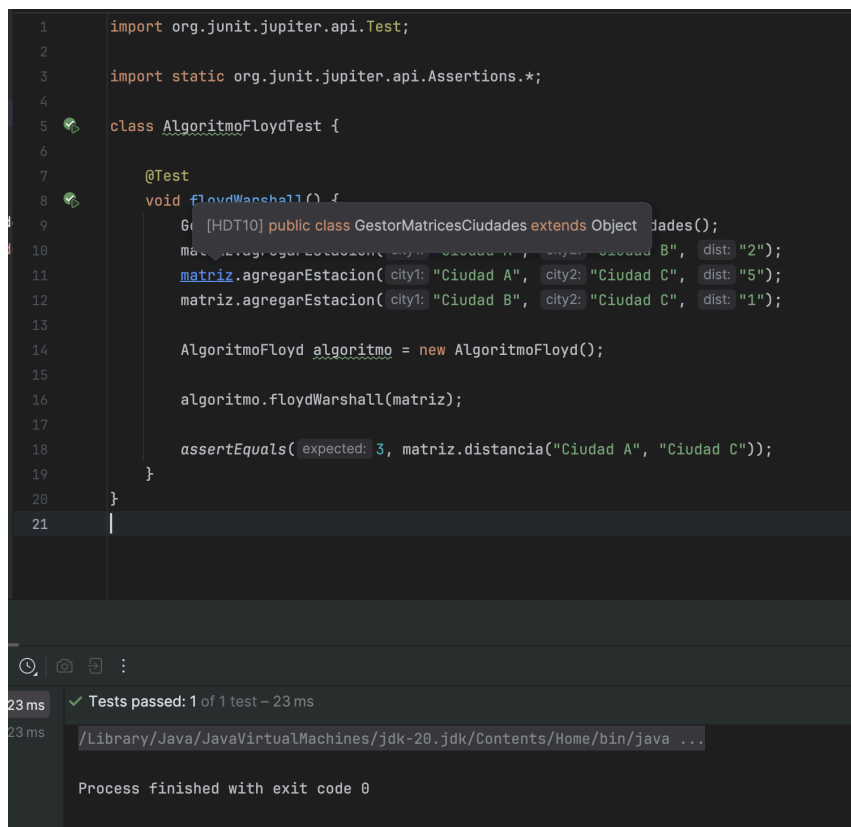
```
5 class GestorMatricesCiudadesTest {
30 void EliminarArco() {
33 // Agregar un arco válido
34 matriz.agregarEstacion( city1: "Ciudad A", city2: "Ciudad B", dist: "5");
35 assertTrue(matriz.contains("Ciudad A"));
36 assertTrue(matriz.contains("Ciudad B"));
37 assertEquals( expected: 5, matriz.distancia("Ciudad A", "Ciudad B"));
38
39 // Eliminar el arco
40 matriz.eliminarArco( ciudadA: "Ciudad A", ciudadB: "Ciudad B");
41 assertEquals( expected: 999999, matriz.distancia("Ciudad A", "Ciudad B")); // La distancia debe ser infinita
42 }
43 @Test
44 void agregarArco() {
45 GestorMatricesCiudades matriz = new GestorMatricesCiudades();
46 matriz.agregarEstacion( city1: "Ciudad A", city2: "Ciudad B", dist: "5");
47
48 // Agregar un arco válido
49 matriz.agregarArco( ciudadA: "Ciudad A", ciudadB: "Ciudad B", dist: 5);
50 assertTrue(matriz.contains("Ciudad A"));
51 assertTrue(matriz.contains("Ciudad B"));
52 assertEquals( expected: 5, matriz.distancia("Ciudad A", "Ciudad B"));
53
54 // Verificar que se agregó correctamente
55 assertEquals( expected: 5, matriz.getDistancia()[matriz.ciudades.indexOf("Ciudad A")][matriz.ciudades.indexOf("Ciudad B")]);
56 }
57 }
```

Run GestorMatricesCiudadesTest

Tests passed: 3 of 3 tests - 31 ms

- agregarArco() 27 ms
- agregarEstacion() 2 ms
- EliminarArco() 2 ms

Process finished with exit code 0



```
1 import org.junit.jupiter.api.Test;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class AlgoritmoFloydTest {
6
7     @Test
8     void FloydWarshall() {
9         GestorMatricesCiudades matriz = new GestorMatricesCiudades();
10        matriz.agregarEstacion( city1: "Ciudad A", city2: "Ciudad B", dist: "2");
11        matriz.agregarEstacion( city1: "Ciudad A", city2: "Ciudad C", dist: "5");
12        matriz.agregarEstacion( city1: "Ciudad B", city2: "Ciudad C", dist: "1");
13
14        AlgoritmoFloyd algoritmo = new AlgoritmoFloyd();
15
16        algoritmo.FloydWarshall(matriz);
17
18        assertEquals( expected: 3, matriz.distancia("Ciudad A", "Ciudad C"));
19    }
20 }
21
```

Tests passed: 1 of 1 test - 23 ms

Process finished with exit code 0