






Pruebas

Caja negra

```
1 package ficha;
2
3 import static org.junit.Assert.assertEquals;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 import org.junit.Before;
8 import org.junit.jupiter.api.Test;
9
10
11 class Ficha_AlumnoTest {
12     private List<String> esperado = new ArrayList<String>();
13
14
15     @Before
16     public void inicializa() {
17         esperado.add("Antonella");
18         esperado.add("Raconto");
19         esperado.add("666666999");
20         esperado.add("prueba@hotmail.com");
21         esperado.add("C/ Gran Vía nº34");
22     }
23
24
25     @Test
26     public void test_correcto() {
27         List<String> resultado = Ficha_alumno.añadir_alumno("Antonella", "Raconto", "666666999", "prueba@hotmail.com", "C/ Gran Vía nº34");
28         inicializa();
29         System.out.println(esperado);
30         assertEquals(esperado, resultado);
31     }
32
33
34     @Test
35     public void test_incorrecto() {
36
37         List<String> esperado = new ArrayList<String>();
38
39         esperado.add("Juana");
40         esperado.add("Raconto");
41         esperado.add("666666999");
42         esperado.add("prueba@hotmail.com");
43         esperado.add("C/ Gran Vía nº34");
44
45         List<String> resultado = Ficha_alumno.añadir_alumno("Antonella", "Raconto", "666666999", "prueba@hotmail.com", "C/ Gran Vía nº34");
46         assertEquals(esperado, resultado);
47     }
48 }
49 }
```

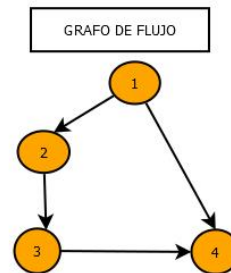
Runs: 2/2  Errors: 0  Failures: 1

▼  Ficha_AlumnoTest [Runner: JUnit 5] (0,005 s)

-  test_incorrecto() (0,088 s)
-  test_correcto() (0,005 s)

Caja blanca

```
1 package ficha;
2
3 import java.util.ArrayList;
4
5
6
7 public class Ficha_alumno {
8
9     public static List<String> añadir_alumno(String nombre, String apellidos, String telefono, String email, String direccion) {
10
11         List<String> alumnos = new ArrayList<String>();
12
13         alumnos.add(nombre);
14         alumnos.add(apellidos);
15         alumnos.add(telefono);
16         alumnos.add(email);
17         alumnos.add(direccion);
18
19
20         return alumnos;
21     }
22
23
24     public static void main(String[] args) {
25
26         Scanner teclado = new Scanner(System.in);
27         int opcion;
28         List<String> resultado;
29
30         System.out.println("1. Añadir alumno");
31         System.out.println("2. Salir\n");
32         System.out.println("Selecciona la opcion deseada: ");
33         opcion = teclado.nextInt();
34
35
36         switch(opcion) {
37             case 1:
38                 System.out.println("Añade un nuevo alumno");
39                 System.out.println("-----\n");
40
41                 System.out.println("Nombre: ");
42                 String nombre = teclado.next();
43
44                 System.out.println("Apellidos: ");
45                 String apellidos = teclado.next();
46
47                 System.out.println("Teléfono: ");
48                 String telefono = teclado.next();
49
50                 System.out.println("Email: ");
51                 String email = teclado.next();
52
53                 System.out.println("Direccion: ");
54                 String direccion = teclado.next();
55
56                 resultado = añadir_alumno(nombre, apellidos, telefono, email, direccion);
57                 System.out.println("Añadido nuevo alumno:\n" + resultado);
58
59             case 2:
60                 System.out.println("SALIR");
61                 System.exit(0);
62                 teclado.close();
63                 break;
64         }
65     }
66 }
```



Caminos básicos

- Camino 1: 1 - 2 - 3 - 4
- Camino 2: 1 - 4

Casos de prueba

- Camino 1: 1
- Camino 2: 2

Complejidad ciclomática

Aristas - Nodos + 2

- Aristas: 4
- Nodos: 4

La complejidad ciclomática es 2.