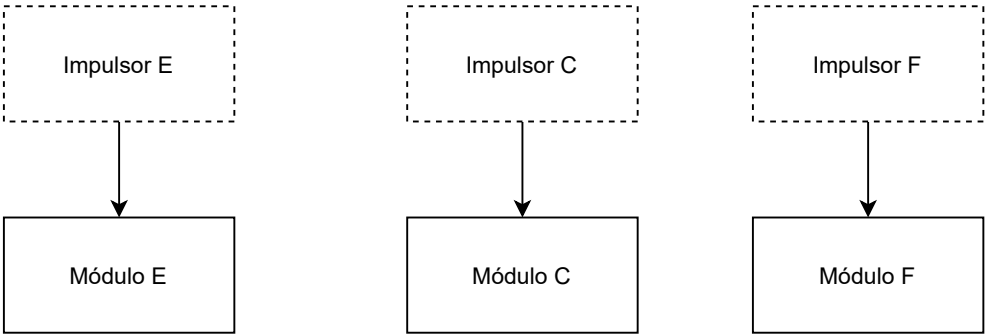
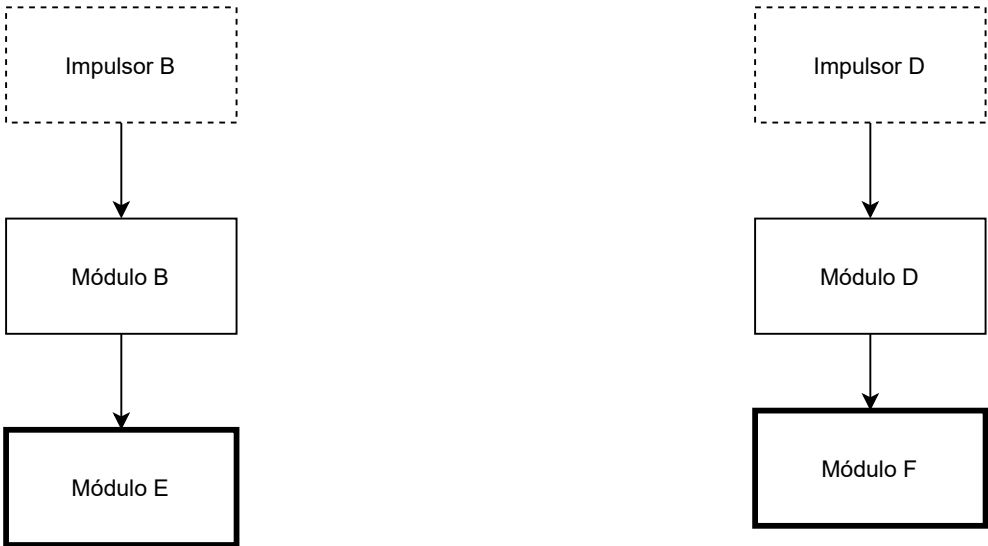


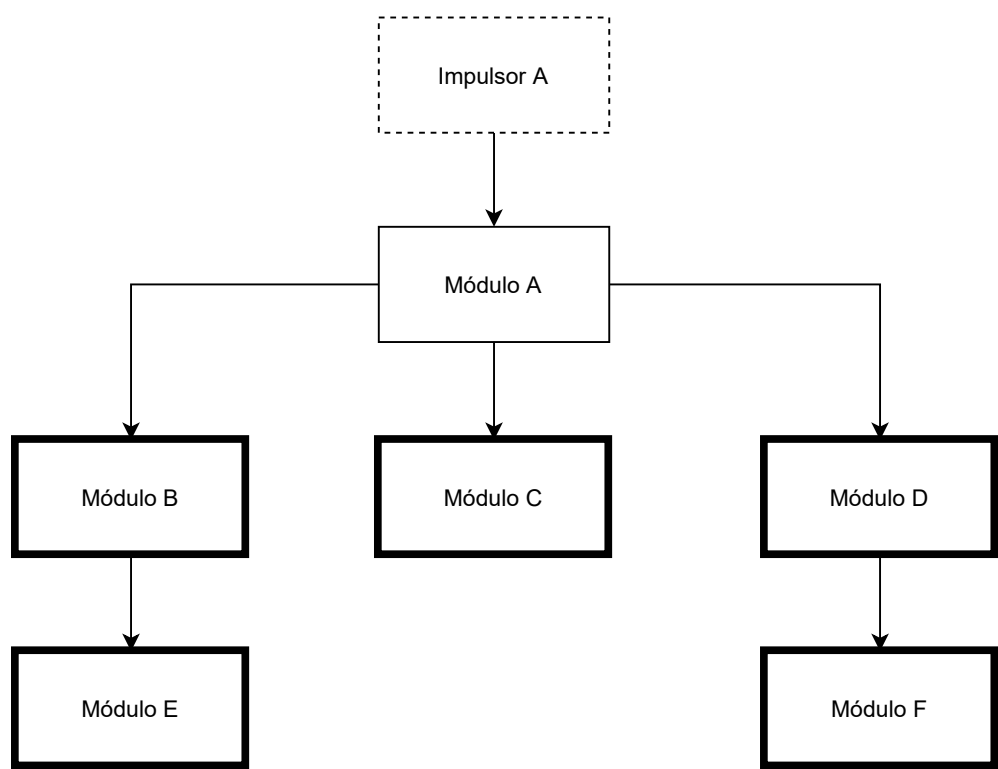
Etapas 1, 2 y 3: pruebas unitarias de E, C y F. Sus impulsores están representados con el borde punteado

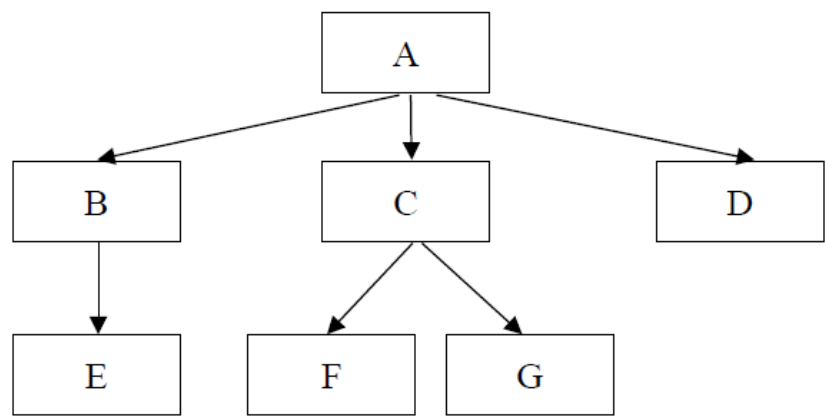


Etapas 4 y 5: pueba unitaria de B, simultaneamente con la de integración B-E y prueba unitaria de D, simultaneamente con la de integración D-F. Los impusores de B y D están representados con el borde punteado y los módulos ya probados tienen el borde grueso.

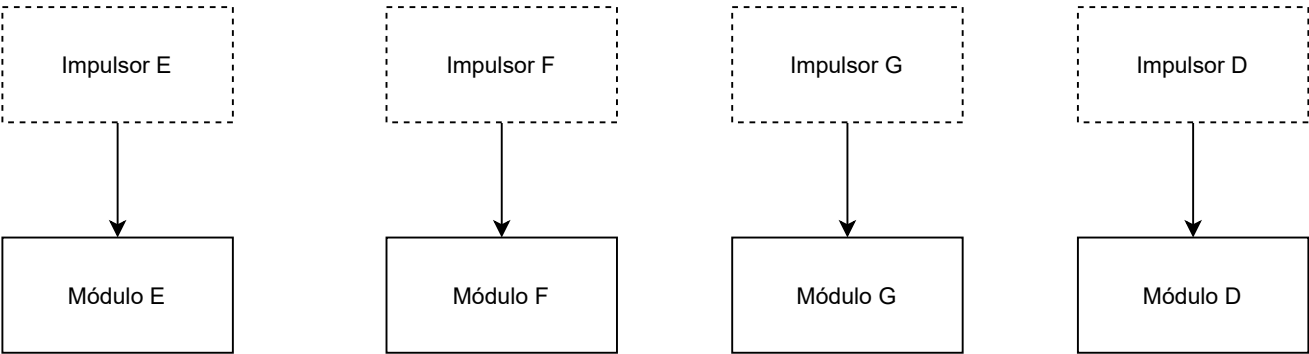


Etapla 6: prueba unitaria de A y simultaneamente la de integración A-B, B-C y A-D

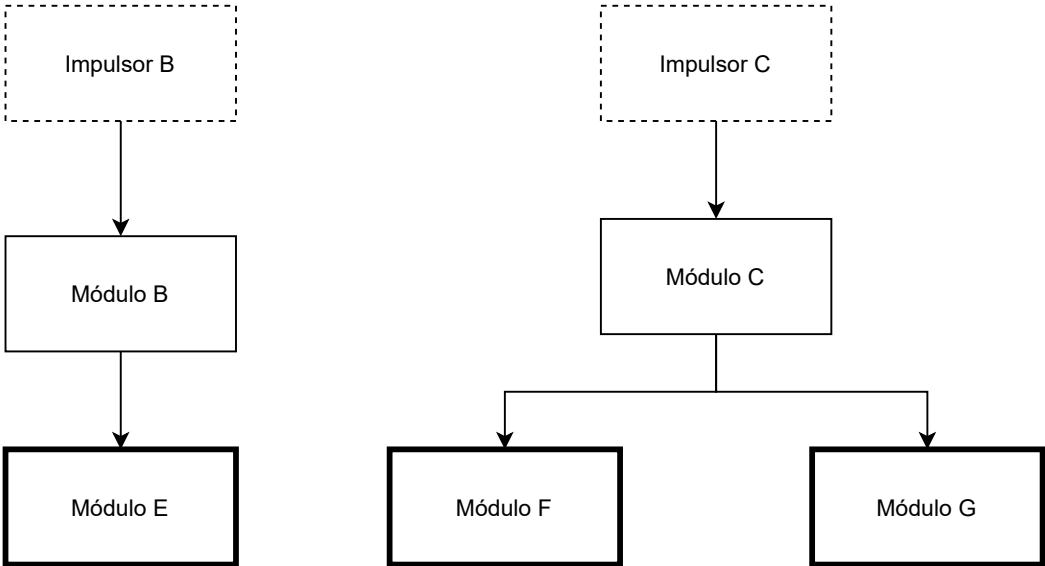


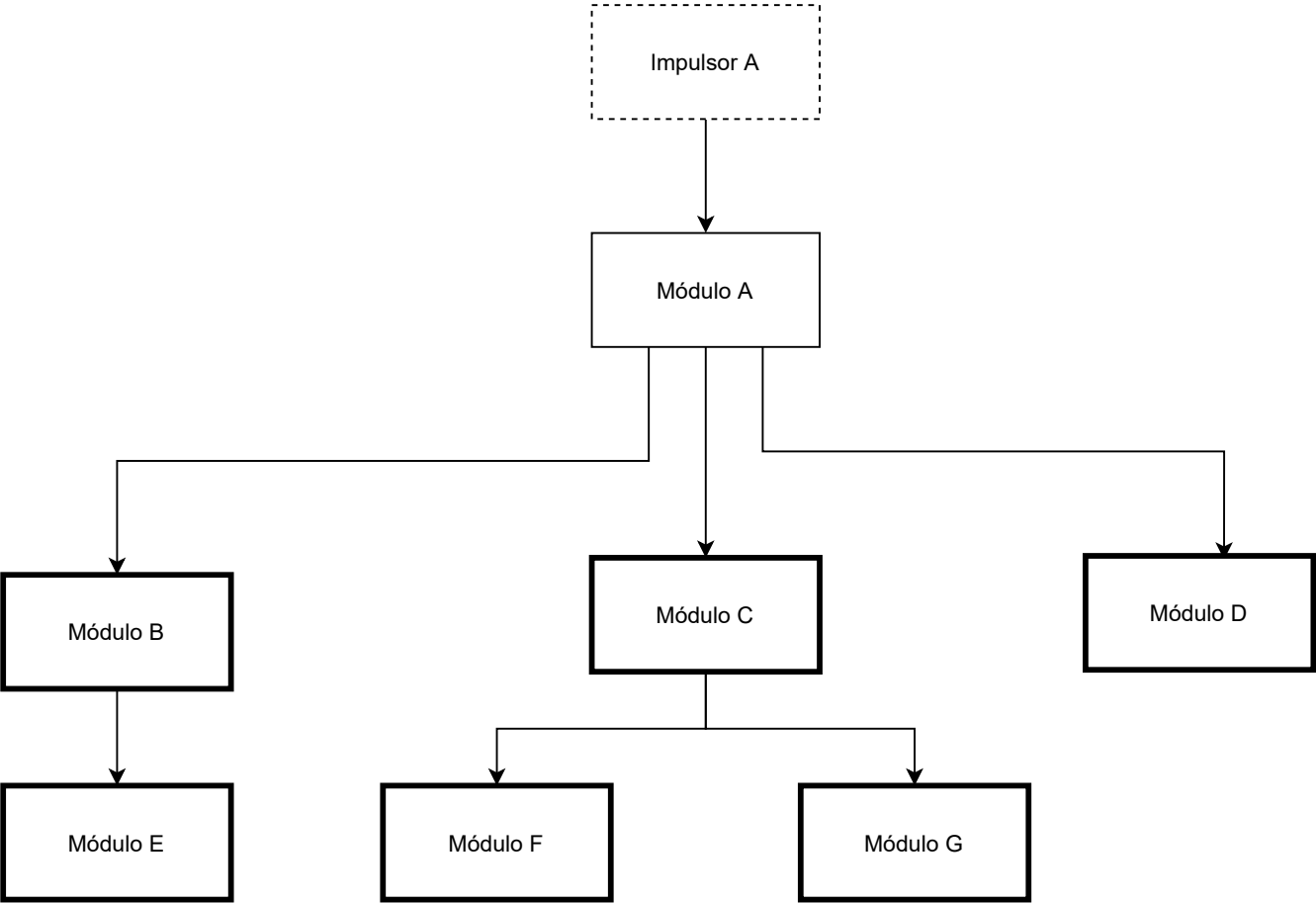


Etapas 1, 2, 3 y 4



Etapas 5 y 6



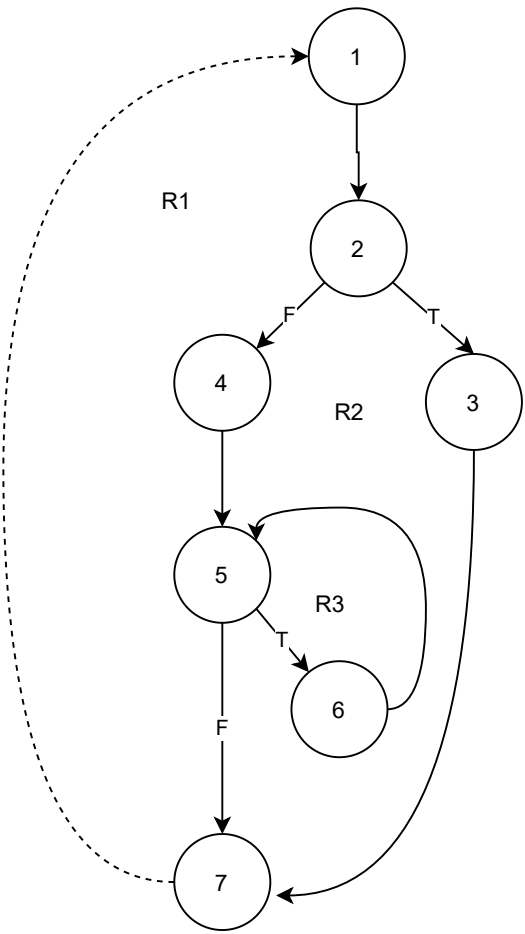


Realizar a proba da clase ContarPalabras (getNumPalabras):

```
public int getNumPalabras() {  
    int contador = 1, pos;  
    frase = frase.trim();  
    if (frase.isEmpty()) {  
        contador = 0;  
    } else {  
        pos = frase.indexOf(" ");  
        while (pos != -1) {  
            contador++;  
            pos = frase.indexOf(" ", pos + 1);  
        }  
    }  
    return contador;  
}
```

Compejidad
 $V(G)=a-n+2=8-7+2=3$
 $V(G)=r=3$
 $V(G)=c+1=2+1=3$

Caminos
1-2-3-7
1-2-4-5-6-5...
1-2-4-5-7



Clases de equivalencia

Entrada	Clases válidas	Clases inválidas
String cadena	(1) String	

Análisis de valores límite (AVL)

AVL1. Cadena con 0 caracteres.

AVL2. Cadena con 1 carácter visible.

AVL3. Cadena con 1 carácter no visible (espacio en blanco).

AVL4. Cadena con 1 carácter no visible (tabulador).

Conjetura de errores

CE1. Cadena con n caracteres visibles y empieza por espacios en blanco.

CE2. Cadena con n caracteres visibles y acaba por espacios en blanco.

CE3. Cadena con n caracteres visibles y con más de un espacio en blanco separando las palabras.

Casos de prueba

Identificación del caso de prueba	Entrada	Identificación de Clase/camino/AVL/CE	Salida
	String cadena		
C1	""	AVL1/1-2-3-7	0
C2	"Hola"	AVL2/1-2-4-5-7	1
C3	" "	AVL3/1-2-4-5-6...-7	0
C4	"(TAB)"	AVL4	0
C5	" Hola mundo"	CE1	2
C6	"Hola mundo "	CE2	2
C7	"Hola mundo"	CE3	2

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.teis.contarpalabras;
7
8   import org.junit.jupiter.api.AfterEach;
9   import org.junit.jupiter.api.AfterAll;
10  import org.junit.jupiter.api.BeforeEach;
11  import org.junit.jupiter.api.BeforeAll;
12  import org.junit.jupiter.api.Test;
13  import static org.junit.jupiter.api.Assertions.*;
14
15  /**
16   *
17   * @author a20estefaniapc
18   */
19  public class ContarPalabrasTest {
20
21      public ContarPalabrasTest() {
22      }
23
24      @org.junit.jupiter.api.BeforeAll
25      public static void setUpClass() throws Exception {
26      }
27
28      @org.junit.jupiter.api.AfterAll
29      public static void tearDownClass() throws Exception {
30      }
31
32      @org.junit.jupiter.api.BeforeEach
33      public void setUp() throws Exception {
34      }
35
36
37
38
39
40
41
42  /**
43   * Test of getNumPalabras method, of class ContarPalabras.
44   */
45
46  @org.junit.jupiter.api.Test
47  public void testGetNumPalabrasC1() {
48      System.out.println("C1: Cadena con 0 caracteres");
49      ContarPalabras instance = new ContarPalabras("");
50      int expResult = 0;
51      int result = instance.getNumPalabras();
52      System.out.println("C1: valor esperado: "+expResult+" Valor obtenido: "+result);
53      assertEquals(expResult, result);
54      // TODO review the generated test code and remove the default call to fail.
55      //fail("The test case is a prototype.");
56
57  }
58
59  @org.junit.jupiter.api.Test
60  public void testGetNumPalabrasC2() {
61      System.out.println("C2: Cadena con al menos 1 carácter visible");
62      ContarPalabras instance = new ContarPalabras("Hola");
63      int expResult = 1;
64      int result = instance.getNumPalabras();
65      System.out.println("C2: valor esperado: "+expResult+" Valor obtenido: "+result);
66      assertEquals(expResult, result);
67      // TODO review the generated test code and remove the default call to fail.
68      //fail("The test case is a prototype.");
69
70  }
71

```

```

72 @org.junit.jupiter.api.Test
73 public void testGetNumPalabrasC3() {
74     System.out.println("C3: Cadena con 1 carácter no visible (espacio en blanco).");
75     ContarPalabras instance = new ContarPalabras(" ");
76     int expectedResult = 0;
77     int result = instance.getNumPalabras();
78     System.out.println("C3: valor esperado: "+expectedResult+" Valor obtenido: "+result);
79     assertEquals(expectedResult, result);
80     // TODO review the generated test code and remove the default call to fail.
81     //fail("The test case is a prototype.");
82 }

```

```

85 @org.junit.jupiter.api.Test
86 public void testGetNumPalabrasC4() {
87     System.out.println("C4: Cadena con 1 carácter no visible (tabulador).");
88     ContarPalabras instance = new ContarPalabras("\t");
89     int expectedResult = 0;
90     int result = instance.getNumPalabras();
91     System.out.println("C4: valor esperado: "+expectedResult+" Valor obtenido: "+result);
92     assertEquals(expectedResult, result);
93     // TODO review the generated test code and remove the default call to fail.
94     //fail("The test case is a prototype.");
95 }

```

```

97 @org.junit.jupiter.api.Test
98 public void testGetNumPalabrasC5() {
99     System.out.println("C5: Cadena con n caracteres visibles y empieza por espacios en blanco. ");
100     ContarPalabras instance = new ContarPalabras("  Hola mundo");
101     int expectedResult = 2;
102     int result = instance.getNumPalabras();
103     System.out.println("C5: valor esperado: "+expectedResult+" Valor obtenido: "+result);
104     assertEquals(expectedResult, result);
105     // TODO review the generated test code and remove the default call to fail.
106     //fail("The test case is a prototype.");
107 }

```

```

111 @org.junit.jupiter.api.Test
112 public void testGetNumPalabrasC6() {
113     System.out.println("C6: Cadena con n caracteres visibles y acaba por espacios en blanco");
114     ContarPalabras instance = new ContarPalabras("Hola mundo ");
115     int expectedResult = 2;
116     int result = instance.getNumPalabras();
117     System.out.println("C6: valor esperado: "+expectedResult+" Valor obtenido: "+result);
118     assertEquals(expectedResult, result);
119     // TODO review the generated test code and remove the default call to fail.
120     //fail("The test case is a prototype.");
121 }

```

```

124 @org.junit.jupiter.api.Test
125 public void testGetNumPalabrasC7() {
126     System.out.println("C7: Cadena con n caracteres visibles y con más de un espacio en blanco separando las palabras");
127     ContarPalabras instance = new ContarPalabras("Hola mundo");
128     int expectedResult = 2;
129     int result = instance.getNumPalabras();
130     System.out.println("C7: valor esperado: "+expectedResult+" Valor obtenido: "+result);
131     assertEquals(expectedResult, result);
132     // TODO review the generated test code and remove the default call to fail.
133     //fail("The test case is a prototype.");
134 }

```


Source History

```
43  * Test of getNumPalabras method, of class ContarPalabras.
44  */
45
46  @org.junit.jupiter.api.Test
47  public void testGetNumPalabrasC1() {
48      System.out.println("C1: Cadena con 0 caracteres");
49      ContarPalabras instance = new ContarPalabras("");
50      int expectedResult = 0;
51      int result = instance.getNumPalabras();
52      System.out.println("C1: valor esperado: "+expectedResult+" Valor obtenido: "+result);
53      assertEquals(expectedResult, result);
54      // TODO review the generated test code and remove the default call to fail.
55      //fail("The test case is a prototype.");
56  }
57
58
59  @org.junit.jupiter.api.Test
60  public void testGetNumPalabrasC2() {
61      System.out.println("C2: Cadena con al menos 1 carácter visible");
```

com.teis.contarpalabras.ContarPalabrasTest > testGetNumPalabrasC6 >

Test Results

com.teis:ContarPalabras:jar:1.0-SNAPSHOT (Unit) >

Tests passed: 85,71 %

6 tests passed, 1 test failed. (0,065 s)

- com.teis.contarpalabras.ContarPalabrasTest Failed
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC4 passed (0,035 s)
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC1 passed (0,0 s)
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC3 passed (0,0 s)
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC5 passed (0,001 s)
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC7 Failed: expected: <2> but was: <5>
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC2 passed (0,0 s)
- com.teis.contarpalabras.ContarPalabrasTest.testGetNumPalabrasC6 passed (0,001 s)

C7: Cadena con n caracteres visibles y con más de un espacio en blanco sepa
C7: valor esperado: 2 Valor obtenido: 5