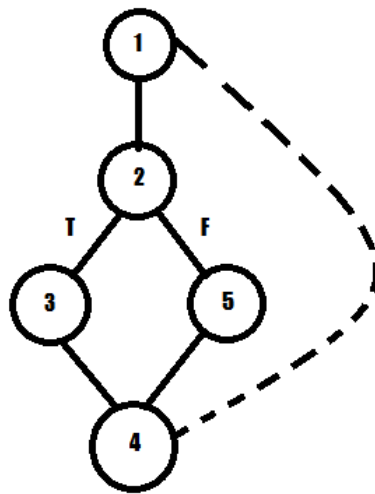


Exercicios repaso

Exercicio 1



Aristas 5 Nodos 5

$$5 - 5 + 2 = 2$$

Rexións cerradas = 2

Camiños:

1-2-3-4

1-2-5-4

Exercicio 2

Añadida condición igual
que 0 prezo será igual a 0.

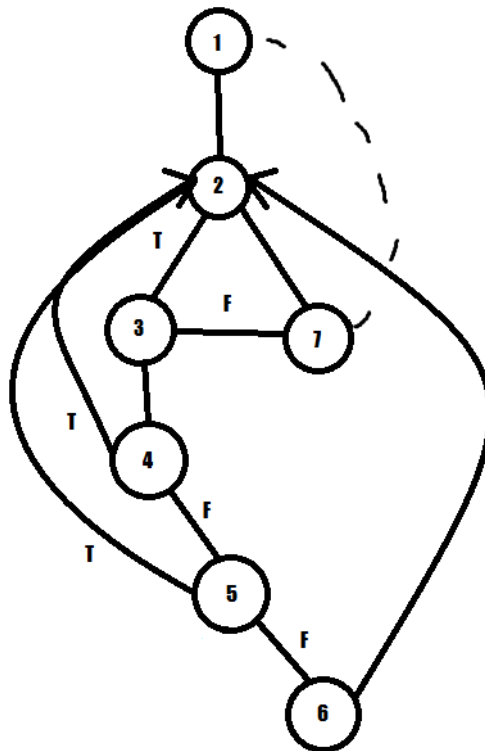
```
public int calcularPrezoKWh() {  
    int prezoKWh = 0;  
    if (consumoKWh != 0) {  
        if (consumoKWh <= 300) {  
            prezoKWh = 9;  
        } else if (consumoKWh <= 600) {  
            prezoKWh = 8;  
        } else if (consumoKWh <= 1000) {  
            prezoKWh = 6;  
        } else if (consumoKWh <= 2000) {  
            prezoKWh = 5;  
        }  
    }  
    return prezoKWh;  
}
```

Exercicio 3

```

package buscarcaracter;
public class OperacionsArrays {
    public boolean busca(char c, char[] v) { 1
        int a, z, m;
        a = 0;
        z = v.length - 1;
        boolean resultado=false;
        while (a <= z && resultado==false) { 2,3
            m = (a + z) / 2;
            if (v[m] == c) { 4
                resultado=true;
            }
            else
            {
                if (v[m] < c) { 5
                    a = m + 1;
                }
                else{ 6
                    z = m - 1;
                }
            }
        }
        return resultado; 7
    }
}

```



Aristas 10 Nodos 7

$10 - 7 + 2 = 5$

Rexións cerradas 5

1-2-7

1-2-3-7

1-2-3-4-2-3-7

1-2-3-4-5-2-7

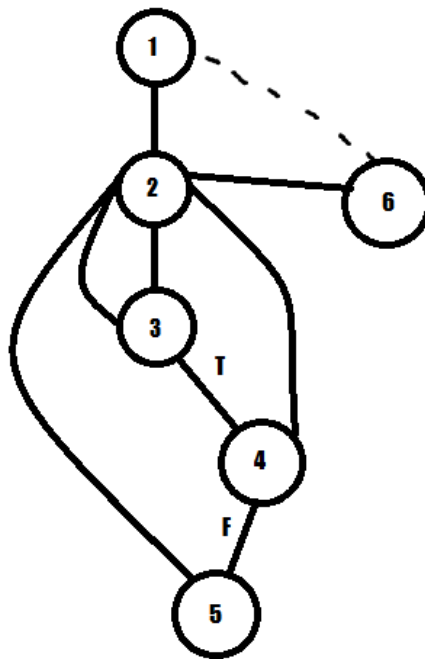
1-2-3-4-5-6-2-7

Exercicio 4

```

package proxecto_acronimos;
public class Acronimos {
    public String obterAcronimo(String cadena){ 1
        String resultado="";
        char caracter;
        int n=cadena.length();
        for(int i=0;i<n;i++){ 2
            caracter=cadena.charAt(i);
            if(caracter!=' '){ 3
                if (i==0){ 4
                    resultado=resultado+caracter+'.';
                }
            } else{ 5
                if(cadena.charAt(i-1)==' '){
                    resultado=resultado+caracter+'.';
                }
            }
        }
        return resultado; 6
    }
}

```



$$7 - 6 + 2 = 3$$

1-2-3-4-5-6

1-2-3-2-6

1-2-3-4-2-6

Exercicio 5

```
1  import static org.junit.Assert.assertEquals;
2
3  import org.junit.Test;
4
5  public class CalculadoraTest {
6      @Test
7      public void test1() {
8          Calculadora calctest = new Calculadora();
9          int resultado = calctest.suma(a: 1, b: 1);
10         assertEquals(expected: 2, resultado, delta: 0);
11     }
12
13     @Test
14     public void test2() {
15         Calculadora calctest = new Calculadora();
16         int resultado = calctest.resta(a: 10, b: 5);
17         assertEquals(expected: 5, resultado, delta: 0);
18     }
19
20     @Test
21     public void test3() {
22         Calculadora calctest = new Calculadora();
23         int resultado = calctest.multiplicacion(a: 2, b: 4);
24         assertEquals(expected: 8, resultado, delta: 0);
25     }
26
27     @Test
28     public void test4() {
29         Calculadora calctest = new Calculadora();
30         int resultado = calctest.division(a: 8, b: 2);
31         assertEquals(expected: 4, resultado, delta: 0);
32     }
33 }
34
```