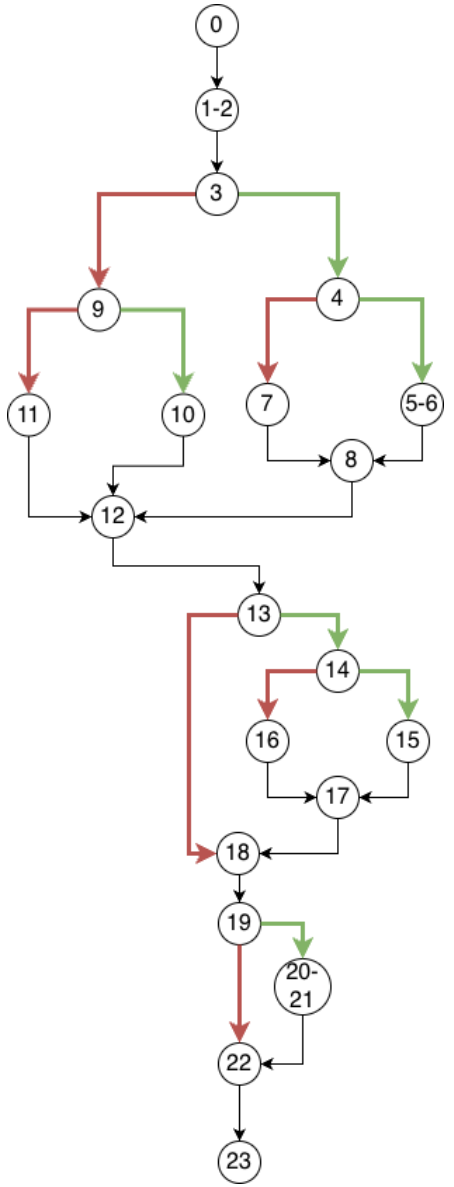
EXAMEN UT03 – TEST UNITARIOS

NOMBRE: _____

CÓDIGO 01

```
0: public static String procesarInventario(
         int cantidad,
         double precioUnitario,
         boolean esPerecedero) {
1:       double costoTotal = 0.0, descuento = 0.0;
2:       String resultado;
3:       if (cantidad > 0) {
4:          if (precioUnitario > 0) {
5:                costoTotal = cantidad *
                  precioUnitario;
6:                resultado =("Costo total: " +
                  costoTotal);
            } else {
7:                resultado = ("Error: Precio unitario
                  inválido");
8:          }
9:       } else if (cantidad == 0) {
10:          resultado =("Advertencia: Cantidad en
               inventario es cero");
           } else {
11:          resultado = ("Error: Cantidad negativa no
                permitida");
12:    }
13:    if (esPerecedero) {
14:       if (costoTotal > 100) {
15:           descuento = costoTotal * 0.15;
           } else {
16:           descuento = costoTotal * 0.05;
17:        }
18:    }
19:    if (costoTotal > 0.0) {
20:       double costoFinal = costoTotal - descuento;
21:       resultado =("Costo total: " + costoFinal);
22:    }
23:    return (resultado);
24: }
```

GRAFO DE FLUJO



Aplicando la fórmula, calcula aquí la complejidad del grafo V(G):

6+1 = 7

Indica los nodos predicado:

3 – 4 – 9 – 13 – 14 – 19

HAY QUE HACER POR TANTO 7 TEST

```
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class ProcesarInventarioTest {

    // 1) cantidad > 0, precioUnitario > 0, no perecedero (sin descuento)
    @Test
    void testCantidadPositivaPrecioPositivoNoPerecedero() {
        String resultado = Examen.procesarInventario(5, 10.0, false);
        // costoTotal = 50, sin descuento
        assertEquals("Costo total: 50.0", resultado);
    }

    // 2) cantidad > 0, precioUnitario > 0, perecedero y costoTotal <= 100
(descuento 5%)
    @Test
    void testPerecederoConDescuento5Porciento() {
        String resultado = Examen.procesarInventario(5, 10.0, true);
        // costoTotal = 50 -> 5% de descuento = 2.5 -> costoFinal = 47.5
        assertEquals("Costo total: 47.5", resultado);
    }

    // 3) cantidad > 0, precioUnitario > 0, perecedero y costoTotal > 100
(descuento 15%)
    @Test
    void testPerecederoConDescuento15Porciento() {
        String resultado = Examen.procesarInventario(20, 10.0, true);
        // costoTotal = 200 -> 15% de descuento = 30 -> costoFinal = 170
        assertEquals("Costo total: 170.0", resultado);
    }

    // 4) cantidad > 0, precioUnitario <= 0 (error de precio)
    @Test
    void testPrecioUnitarioInvalido() {
        String resultado = Examen.procesarInventario(5, 0.0, true);
        assertEquals("Error: Precio unitario inválido", resultado);
    }

    // 5) cantidad == 0 (advertencia), no perecedero
    @Test
    void testCantidadCeroNoPerecedero() {
        String resultado = Examen.procesarInventario(0, 10.0, false);
        assertEquals("Advertencia: Cantidad en inventario es cero", resultado);
    }

    // 6) cantidad == 0 (advertencia), perecedero (no hay descuento porque
costoTotal=0)
    @Test
    void testCantidadCeroPerecedero() {
        String resultado = Examen.procesarInventario(0, 10.0, true);
        assertEquals("Advertencia: Cantidad en inventario es cero", resultado);
    }

    // 7) cantidad < 0 (error), perecedero o no da igual
    @Test
    void testCantidadNegativaNoPerecedero() {
        String resultado = Examen.procesarInventario(-1, 10.0, false);
        assertEquals("Error: Cantidad negativa no permitida", resultado);
    }

}
```