

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL TUCUMAN

**INGENIERÍA DE SOFTWARE**

**TRABAJO PRÁCTICO N°3**

COMISIÓN 4K1.

GRUPO N°12.

**INTEGRANTES:**

Molina, Matias Gaspar – 52560

Morales, Natalia del Valle – 47751

González Reinoso, Valentín – 50545

Navarro Bovi, José Eugenio – 50027

ESTRATEGIAS PARA CASOS DE PRUEBA, AUTOMATIZACIÓN DE PRUEBAS DE  
ACEPTACIÓN, PRUEBAS UNITARIAS Y PRUEBAS DE SISTEMA

**FECHA DE PRESENTACIÓN: 14/12/23**

**1-) Pruebas de particiones**

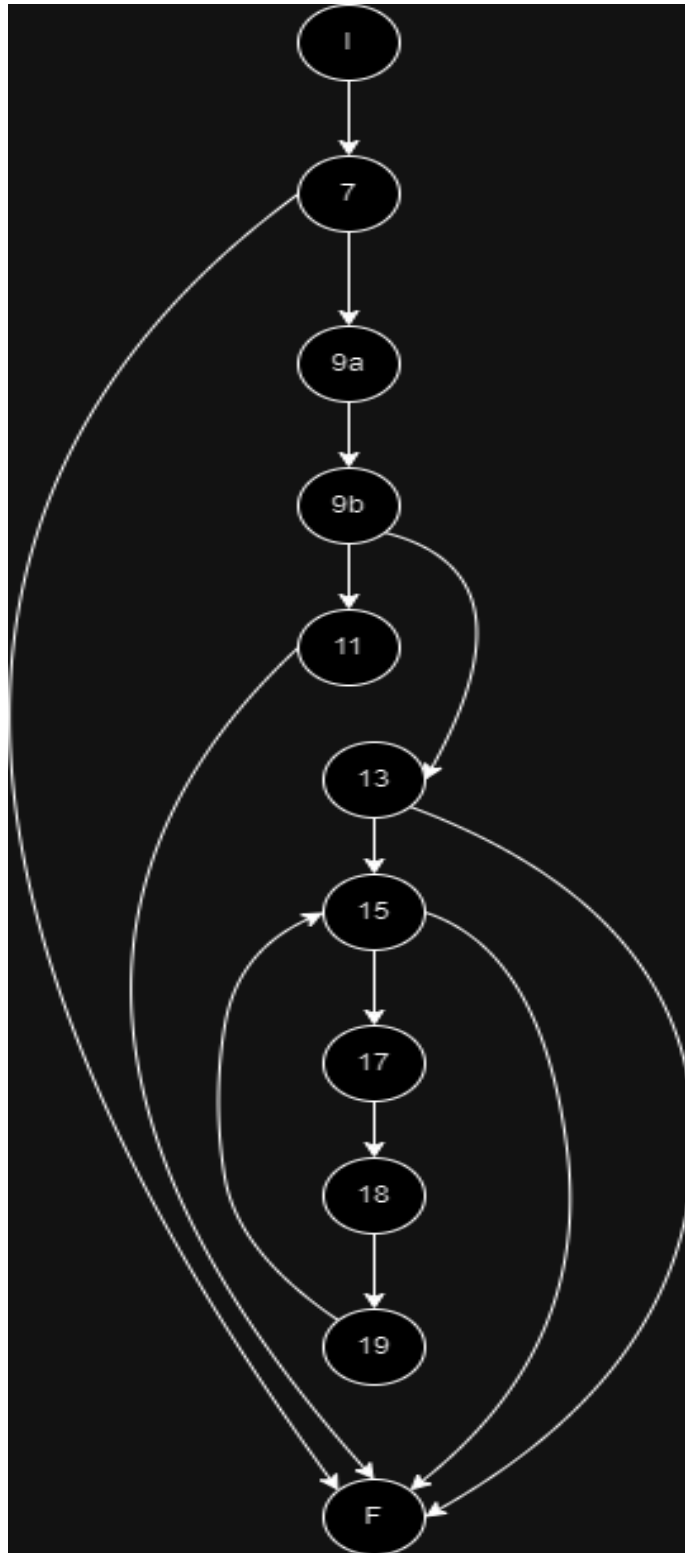
a-) Determinar las particiones de equivalencia:

↓3   ↓4	↓6	↓8   ↓9
entradas <4	entradas >= 4 y entradas <= 8	entradas >8

↓9.999	↓10.000	↓50.000	↓99.999   ↓100.000
valores <=10000	valores >10000 y <=99999	valores >99999	

b-)Determinar las particiones de equivalencia:

↓n=-1	↓n=2	↓n=0   ↓n=1	↓n=3.200.000	↓n=abc
n < 0	n > 1	n >= 0 y n <= 1	n = entero	n != entero

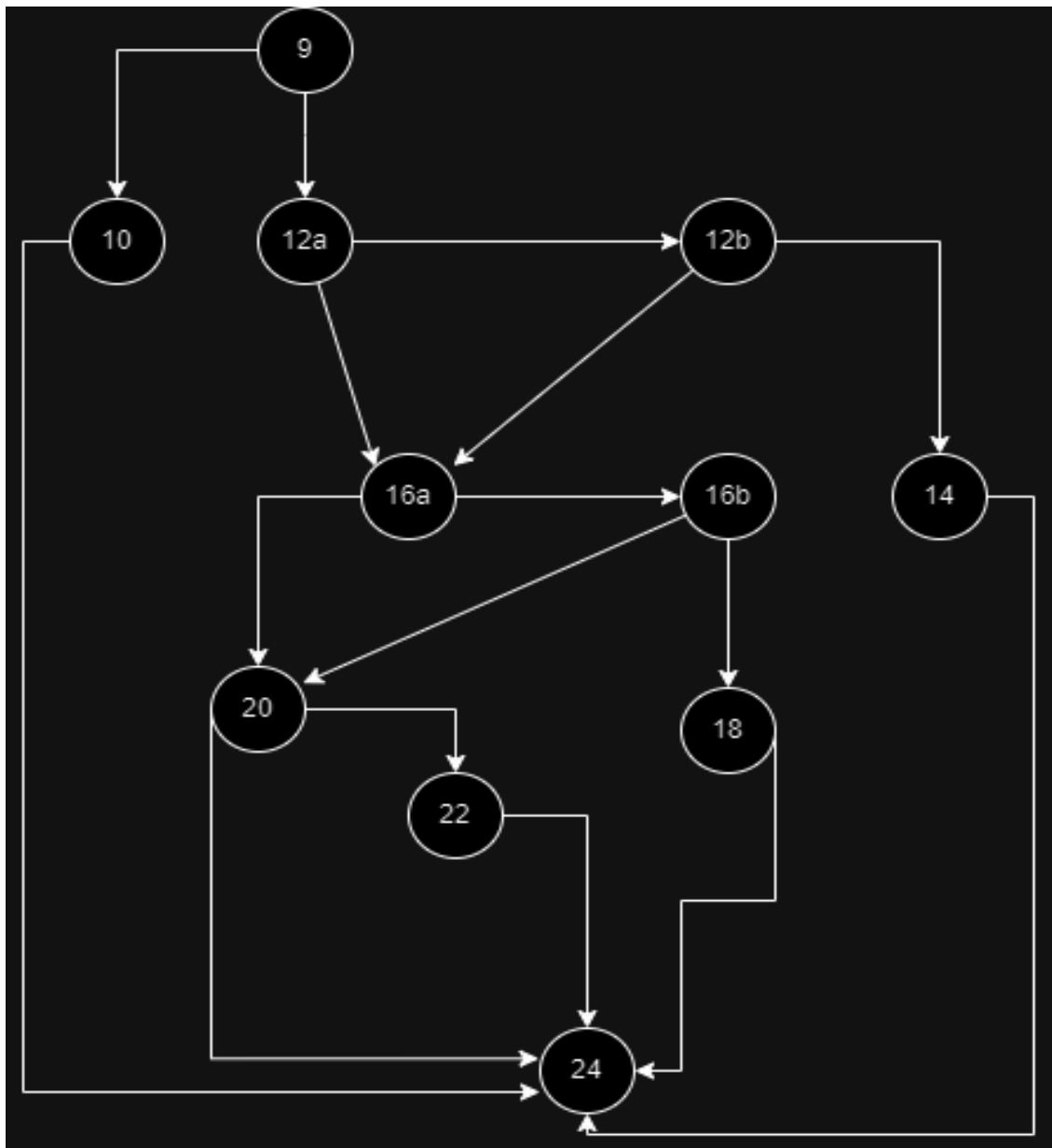
**2. Pruebas de caminos****a)**

1:  $V(G) = a - n + 2 = 14 - 11 + 2 = 5$

2:  $V(G) = R = 5$

3:  $V(G) = p + 1 = 4 + 1 = 5$

b)

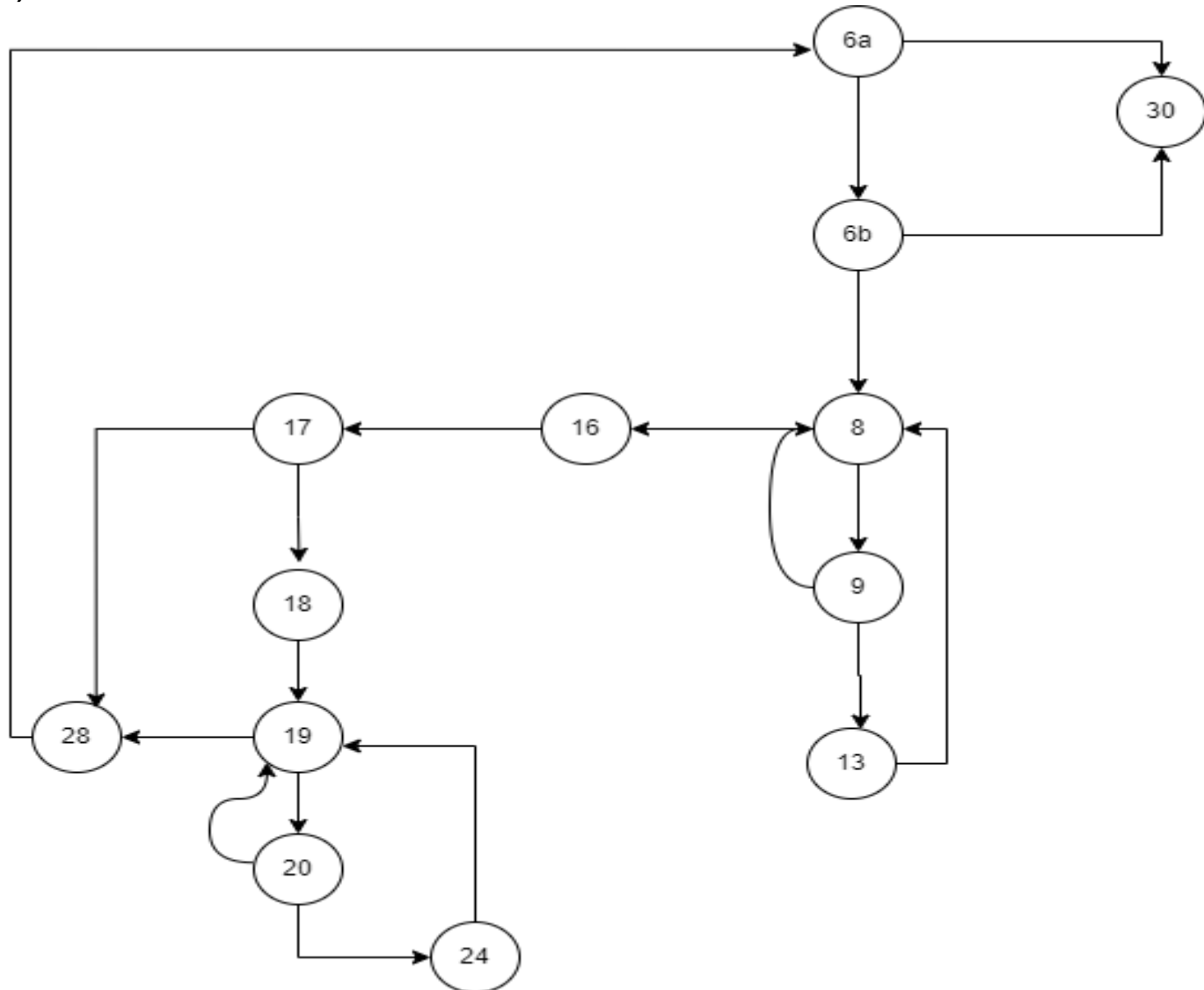


1:  $V(G) = a - n + 2 = 16 - 11 + 2 = 7$

2:  $V(G) = R = 7$

3:  $V(G) = p + 1 = 6 + 1 = 7$

c-)



1:  $V(G) = a - n + 2 = 24 - 20 + 2 = 6$

2:  $V(G) = R = 8$

3:  $V(G) = p + 1 = 7 + 1 = 8$

### 3-) Pruebas de Unidad (Unitarias)

```
public class ReglaDeDescuentoTest {  
    José Eugenio Navarro Bovi  
    @ParameterizedTest  
    @ValueSource(doubles = {4999, 5000})  
    public void CalcularDescuentoConValoresMenoresOIgualA5000NoAplicaDescuento(double total){  
        double esperado = 0;  
        ReglaDeDescuento reglaDeDescuento = new ReglaDeDescuento();  
        double resultado = reglaDeDescuento.Calcular(total);  
        assertEquals(esperado, resultado);  
    }  
  
    @Test  
    public void CalcularDescuentoConValoresMenoresOIgualQueCeroArrojaExcepcion(){  
        String excepcionEsperada = "El total debe ser mayor a 0";  
        ReglaDeDescuento reglaDeDescuento = new ReglaDeDescuento();  
        try{  
            reglaDeDescuento.Calcular( total: -5);  
        } catch(Exception e){  
            assertEquals(excepcionEsperada, e.getMessage());  
        }  
    }  
  
    @ParameterizedTest  
    @ValueSource(doubles = {5500, 8000, 5000, 4999, 10001, 10000})  
    public void CalcularDescuentoConValoresMayoresA5000YMenoresOIgualA10000AplicaPorcentajeMenor(double total){  
        double porcentajeMenor = 0.03d;  
        double esperado = total * porcentajeMenor;  
        ReglaDeDescuento reglaDeDescuento = new ReglaDeDescuento();  
        double resultado = reglaDeDescuento.Calcular(total);  
        assertEquals(esperado, resultado);  
    }  
  
    @ParameterizedTest  
    @ValueSource(doubles = {9999, 10000, 25000, 25001})  
    public void CalcularDescuentoConValoresMayoresA10000YMenoresOIgualA25000AplicaPorcentajeMedio(double total){  
        double porcentajeIntermedio = 0.05d;  
        double esperado = total * porcentajeIntermedio;  
        ReglaDeDescuento reglaDeDescuento = new ReglaDeDescuento();  
        double resultado = reglaDeDescuento.Calcular(total);  
        assertEquals(esperado, resultado);  
    }  
}
```

```

@ParameterizedTest
@ValueSource(doubles = {24999, 25000, 25500, 28000})
public void CalcularDescuentoConValoresMayoresA25000AplicaPorcentajeMayor(double total){
    double porcentajeMayor = 0.1d;
    double esperado = total * porcentajeMayor;
    ReglaDeDescuento reglaDeDescuento = new ReglaDeDescuento();
    double resultado = reglaDeDescuento.Calcular(total);
    assertEquals(esperado, resultado);
}
}

```

## 5. Pruebas de Versión (sistema)

Caso de Prueba	
<b>ID:</b> 01	<b>Nombre:</b> Listar combinaciones de artículos existentes.
<b>Descripción:</b> Listar artículos para una venta en curso.	
<b>Prioridad:</b> Alta	<b>CU / HU:</b> Realizar venta.
<b>Módulo / Funcionalidad:</b> Agregar Artículos a la venta.	
<b>Diseñado por:</b> G12	<b>Fecha:</b> 14/11/2023
<b>Ejecutado por:</b>	<b>Fecha:</b>

### Precondiciones:

Existe una venta en curso

Paso	Acción	Resultado Esperado	Pasó/Falló	Comentarios
1	Introducir código del artículo	Mostrar en la interfaz de usuario marca, descripción y precio del artículo; y se lista las combinaciones de talle, color y cantidad disponibles.		

Caso de Prueba	
<b>ID:</b> 02	<b>Nombre:</b> Monto inferior a \$92700.
<b>Descripción:</b> Recibir pago en efectivo de una venta cuyo monto es menor a \$92700.	
<b>Prioridad:</b> Alta.	<b>CU / HU:</b> Realizar venta.
<b>Módulo / Funcionalidad:</b> Pago en Efectivo	
<b>Diseñado por:</b> G12	<b>Fecha:</b> 14/11/2023
<b>Ejecutado por:</b>	<b>Fecha:</b>

**Precondiciones:**

Existe una venta en curso.

Existe un cliente consumidor final asociado a la venta.

Se tiene el monto total de la venta.

Paso	Acción	Resultado Esperado	Pasó/Falló	Comentarios
1	Selecciono la opción de Pago en Efectivo.	La venta cambia al estado "Pagada"		Venta debe tener una fecha y un estado "pendiente de pago".