



## **Ingeniería de Software**

Comisión 4k1

---

### **TP3**

---

**Fecha de Presentación: 14/11/2023**

### **Grupo 16**

#### **Integrantes**

52596 - Paz Posse, Tobías

52679 - Figueroa, Santiago Manuel

52632 - Chacón, Luciano

#### **Profesores**

Vicente, Francisco

Torres, Mabel

<b>Pruebas de particiones</b>	<b>3</b>
Particiones para un programa	3
Número de entradas	3
Valor de las entradas:	3
Rango de valores:	3
Particiones para fibonacci	3
Valores positivos de n	4
Valores negativos de n	4
Casos límite	4
Manejo de casos base	4
<b>Pruebas de caminos</b>	<b>5</b>
Código del apartado 1b	5
Código del apartado 2a	6
Código del apartado 2b	7
<b>Pruebas de unidad</b>	<b>8</b>
<b>Automatización de pruebas de aceptación y pruebas unitarias</b>	<b>8</b>
<b>Pruebas de versión (Sistema)</b>	<b>8</b>
Listar combinaciones de artículo	8
Borrar artículo sin inventario asociado	9

# Pruebas de particiones

## Particiones para un programa

### Número de entradas

- **Partición 1:** Menos de 4 entradas.
- **Partición 2:** Entre 4 y 8 entradas.
- **Partición 3:** Más de 8 entradas.

### Valor de las entradas:

- **Partición 4:** Entradas con menos de 5 dígitos.
- **Partición 5:** Entradas con exactamente 5 dígitos.
- **Partición 6:** Entradas con más de 5 dígitos.

### Rango de valores:

- **Partición 7:** Entradas menores o iguales a 10000.
- **Partición 8:** Entradas mayores que 10000.

## Particiones para fibonacci

Teniendo en cuenta el código:

```
private static int fibonacci(int n)
{
    int actual = 0;
    int ant1, ant2;
    ant1 = 1;
    ant2 = 0;
    if (n >= 0)
    {
        if ((n == 0) || (n == 1))
        {
            actual = n;
        }
        else
        {
            for (int i = 2; i <= n; i++)
            {
                actual = ant1 + ant2;
            }
        }
    }
}
```



```
        ant2 = ant1;
        ant1 = actual;
    }
}
return actual;
}
```

### Valores positivos de n

- **Partición 1:** n igual a 0 o 1.
- **Partición 2:** n mayor que 1.

### Valores negativos de n

- **Partición 3:** n menor que 0.

### Casos límite

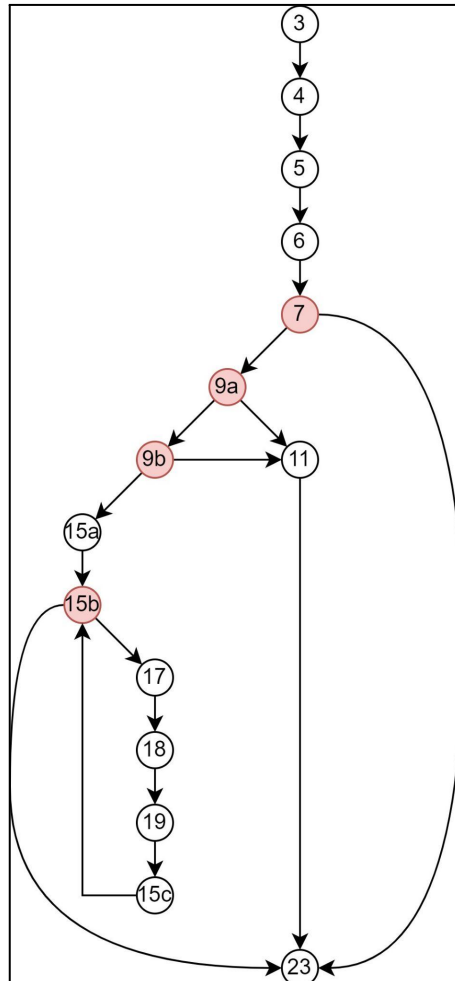
- **Partición 4:** n igual a `Int32.MaxValue`.
- **Partición 5:** n igual a `Int32.MinValue`.

### Manejo de casos base

- **Partición 7:** n igual a 0 o 1 para verificar el manejo correcto de los casos base.

# Pruebas de caminos

## Código del apartado 1b

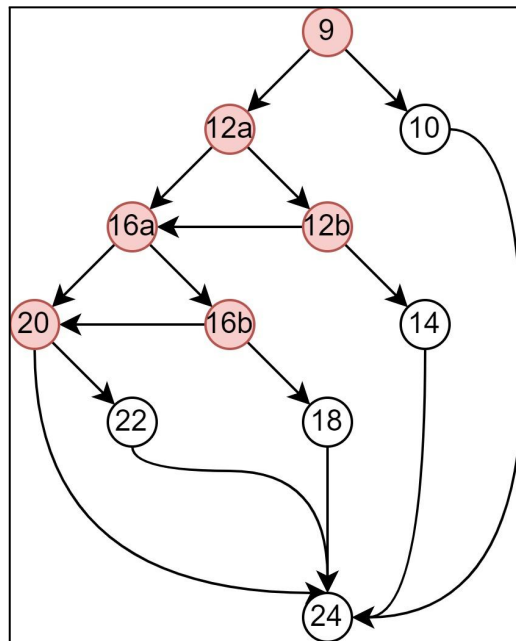


**Método 1:**  $18 \text{ [aristas]} - 15 \text{ [nodos]} + 2 = 5$

**Método 2:**  $5 \text{ [regiones cerradas]} = 5$

**Método 3:**  $4 \text{ [nodos predicados]} + 1 = 5$

## Código del apartado 2a

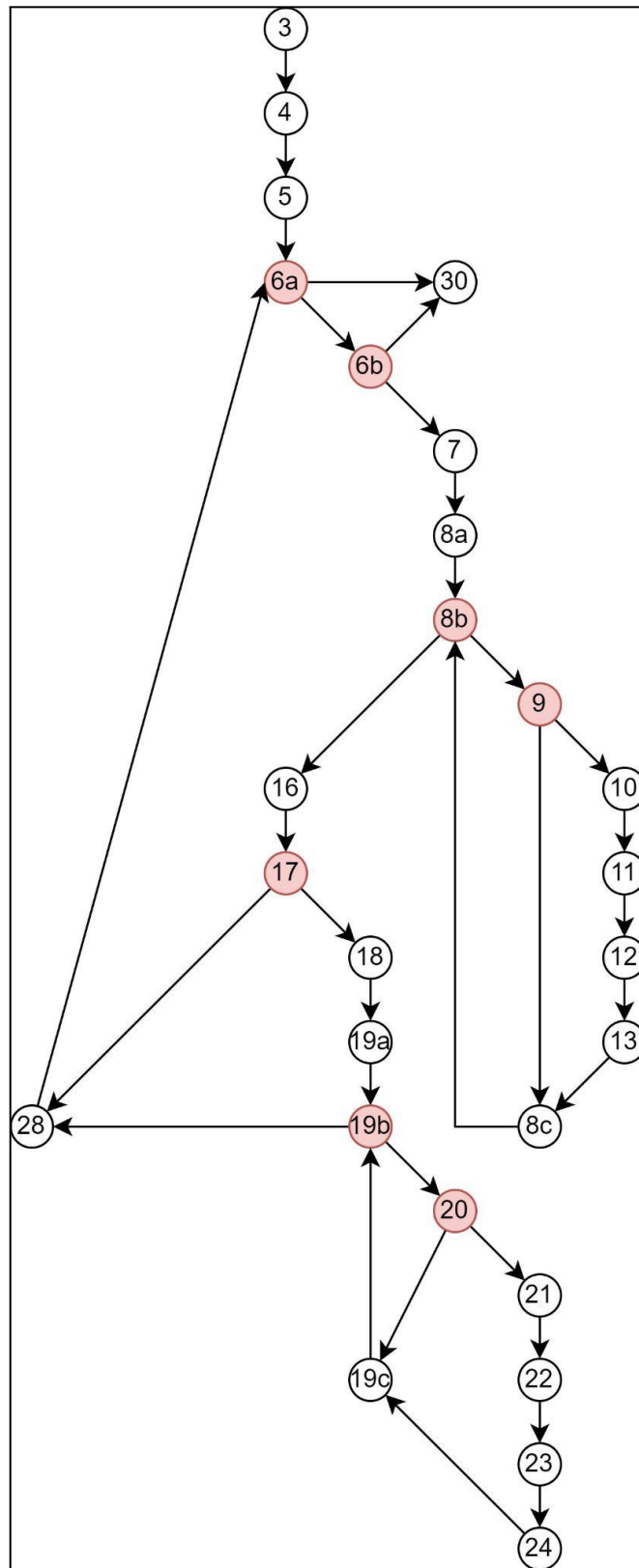


**Método 1:** 16 [aristas] - 11 [nodos] + 2 = 7

**Método 2:** 7 [regiones cerradas] = 7

**Método 3:** 6 [nodos predicados] + 1 = 7

Código del apartado 2b



**Método 1:**  $33 \text{ [aristas]} - 27 \text{ [nodos]} + 2 = 8$

**Método 2:** 8 [regiones cerradas] = 8

**Método 3:** 7 [nodos predicados] +1 = 8

## Pruebas de unidad

Las pruebas de unidad de la clase ReglaDeDescuento fueron traducidas e implementadas en el entorno de .NET y están incluidas en el repositorio. La clase está en el proyecto **ReglaDescuentoConsigna** y las pruebas están en el proyecto **ReglaDescuentoTest**.

## Automatización de pruebas de aceptación y pruebas unitarias

Está desarrollado en el entorno de .net, haciendo uso del framework Specflow.

## Pruebas de versión (Sistema)

### Listar combinaciones de artículo

Caso de Prueba	
ID: 01	Nombre: Listar combinaciones de artículo
Descripción: Se listan las combinaciones de talle y stock e información de un artículo existente	
Prioridad: Alta	CU / HU: Agregar Artículo a la venta
Módulo / Funcionalidad: Realizar Venta	
Diseñado por: Grupo 16	Fecha: 13/11/2023
Ejecutado por:	Fecha:

<b>Precondiciones:</b> El vendedor debe estar autenticado en el sistema Existe una venta en proceso Existe un artículo con sus combinaciones				
Paso	Acción	Resultado Esperado	Pasó /Falló	Comentarios
1	Ingresar el código del artículo	Mostrar la Marca y Descripción del artículo y mostrar las combinaciones de Talle, Color y Stock del artículo		



## Borrar artículo sin inventario asociado

Caso de Prueba	
<b>ID: 02</b>	<b>Nombre: Borrar artículo sin inventario asociado</b>
<b>Descripción: Se realiza un borrado lógico de un artículo que no cuenta con stock</b>	
<b>Prioridad: Alta</b>	<b>CU / HU: Baja lógica de artículo existente</b>
<b>Módulo / Funcionalidad: Administración de Artículos</b>	
<b>Diseñado por: GRUPO 16</b>	<b>Fecha: 13/11/23</b>
<b>Ejecutado por:</b>	<b>Fecha:</b>

<b>Precondiciones:</b> <b>Administrador autenticado y con permisos para borrar artículos</b> <b>Artículos activos cargados</b> <b>Combinaciones de artículos sin stock en inventario</b>				
Paso	Acción	Resultado Esperado	Pasó / Falló	Comentarios
1	Ingresar el código del artículo.	Visualizar la información del artículo con sus combinaciones.		
2	Seleccionar la opción de borrar el artículo.	Visualizar un mensaje de confirmación.		
3	Confirmar	Mensaje de borrado exitoso. Registro almacenado correctamente en la base de datos.		