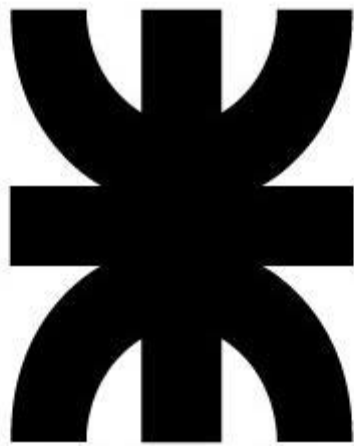


# Trabajo Práctico N°3



## **Integrantes:**

- Torres Carrasco Gonzalo Guillermo
- Albornoz Maria Agostina
- Delevaux Matheus Gerard Paul
- Gonza Gabriel Emanuel
- Madrid Maria Paula

## 1. Prueba de particiones

a) Determinar las particiones de equivalencia para un programa, cuya especificación establece, que acepta de 4 a 8 entradas que son 5 dígitos enteros mayores que 10000.

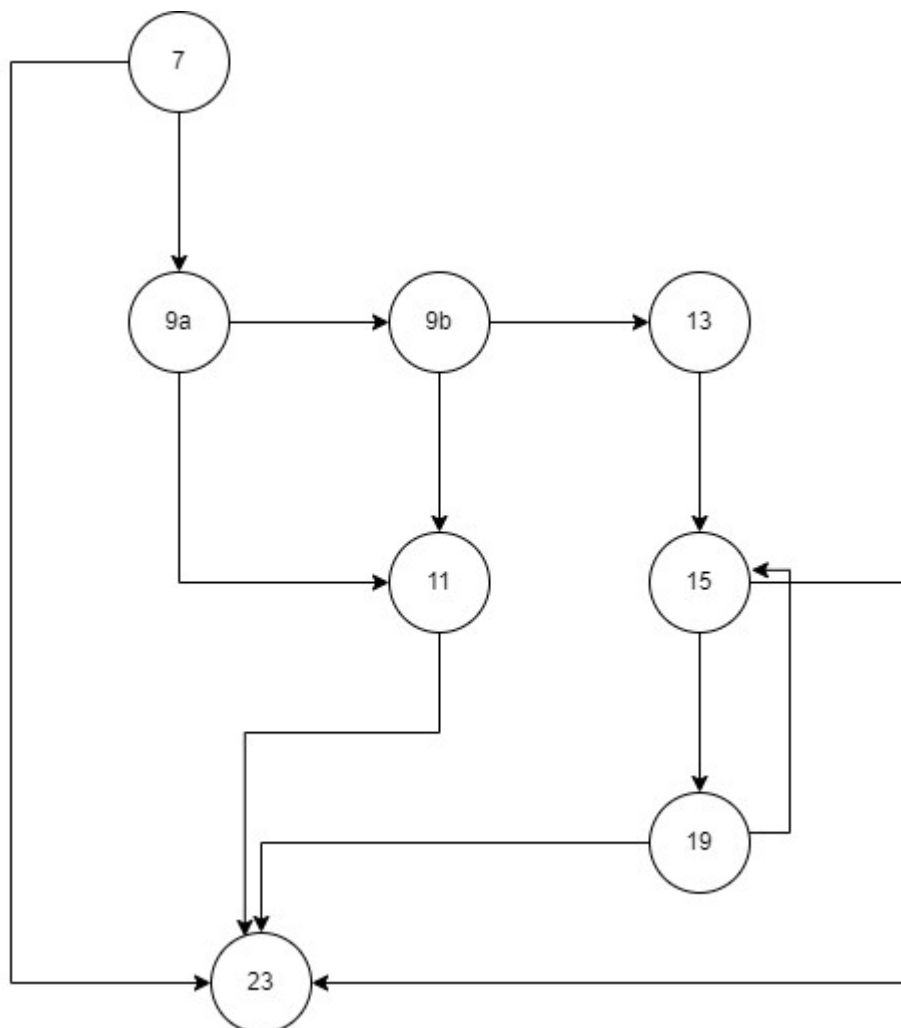
| Partición                            | Valor ejemplo | Resultado Esperado |
|--------------------------------------|---------------|--------------------|
| Entrada menor a 4                    | 3             | Inválido           |
| Entrada mayor a 4 y menor a 8        | 5             | Válido             |
| Entrada mayor a 8                    | 9             | Inválido           |
| Dígito menor a 10000                 | 9999          | Inválido           |
| Dígito mayor a 10000 y menor a 99999 | 11000         | Válido             |
| Dígito mayor a 99999                 | 100000        | Inválido           |

b)

```
1  private static int fibonacci(int n)
2  {
3      int actual = 0;
4      int ant1, ant2;
5      ant1 = 1;
6      ant2 = 0;
7      if (n >= 0)
8      {
9          if ((n == 0) || (n == 1))
10         {
11             actual = n;
12         }
13         else
14         {
15             for (int i = 2; i <= n; i++)
16             {
17                 actual = ant1 + ant2;
18                 ant2 = ant1;
19                 ant1 = actual;
20             }
21         }
22     }
23     return actual;
24 }
```

| Enteros                       | Valor ejemplo | Resultado esperado |
|-------------------------------|---------------|--------------------|
| Valores negativos             | -1            | 0                  |
| Valores entre 0 y 1           | 1             | 1                  |
| Valores mayores o iguales a 1 | 3             | 2                  |

|                              |    |       |
|------------------------------|----|-------|
| Valores distintos de enteros | %5 | Error |
|------------------------------|----|-------|



$RC = 5 ; A = 11 ; N = 8 ; NP = 3$

$V(G) = NP + 1 = 3 + 1 = 4$

$V(G) = A - N + 2 = 11 - 8 + 2 = 5$

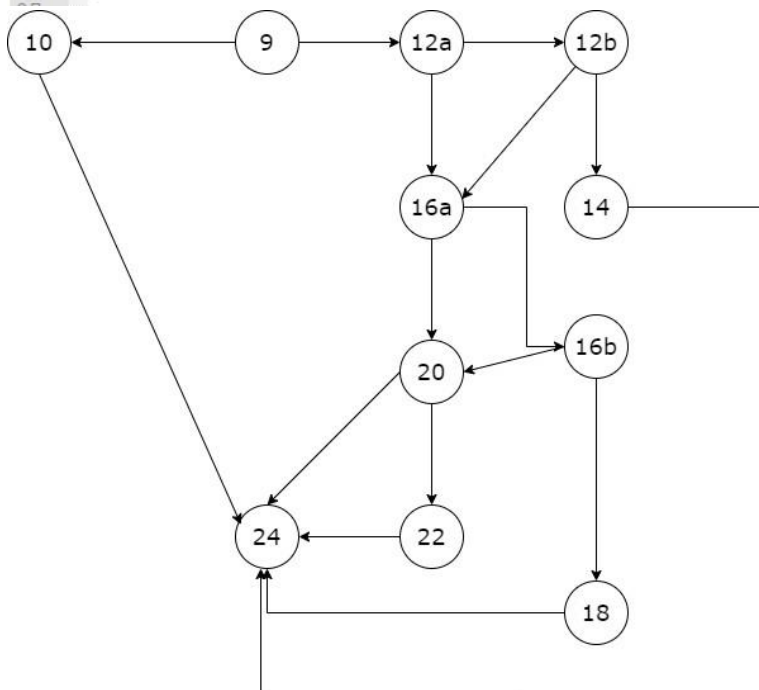
$V(G) = RC + 1 = 5 + 1 = 6$

2. Pruebas de caminos Realizar el grafo de flujo para el código del apartado b) del punto 1 y para los siguientes métodos. Calcular la complejidad ciclomática asociada por los tres métodos.

```

1 public class ReglaDeDescuento
2 {
3     private static final double porcentajeMenor = 0.03d;
4     private static final double porcentajeIntermedio = 0.05d;
5     private static final double porcentajeMayor = 0.10d;
6
7     public double Calcular(double total)
8     {
9         if (total <= 0)
10             throw new IllegalArgumentException("El total debe ser mayor a 0");
11
12         if (total > 5000 && total <= 10000)
13         {
14             return total * porcentajeMenor;
15         }
16         if (total > 10000 && total <= 25000)
17         {
18             return total * porcentajeIntermedio;
19         }
20         else if (total > 25000)
21         {
22             return total * porcentajeMayor;
23         }
24         return 0;
25     }
26 }

```



RC = 6 ; A = 16 ; N = 12 ; NP= 4

$V(G) = NP + 1 = 4 + 1 = 5$

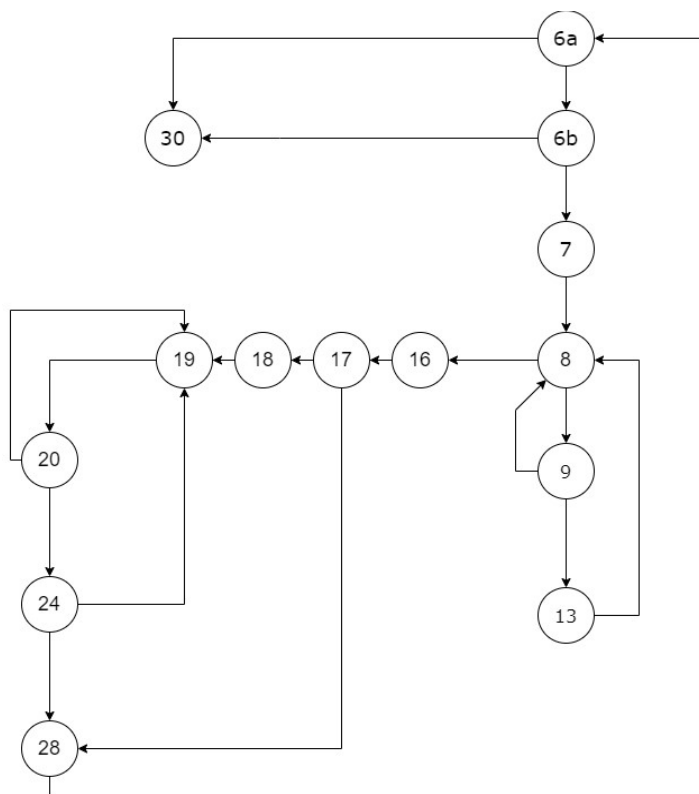
$V(G) = A - N + 2 = 16 - 12 + 2 = 6$

$V(G) = RC + 1 = 6 + 1 = 7$

```

1  public static int[] cocktailSort(int[] numbers)
2  {
3      boolean swapped = true;
4      int i = 0;
5      int j = numbers.length - 1;
6      while(i < j && swapped){
7          swapped = false;
8          for(int k = i; k < j; k++){
9              if(numbers[k] > numbers[k + 1]){
10                 int temp = numbers[k];
11                 numbers[k] = numbers[k + 1];
12                 numbers[k + 1] = temp;
13                 swapped = true;
14             }
15         }
16         j--;
17         if(swapped){
18             swapped = false;
19             for(int k = j; k > i; k--){
20                 if(numbers[k] < numbers[k - 1]){
21                     int temp = numbers[k];
22                     numbers[k] = numbers[k - 1];
23                     numbers[k - 1] = temp;
24                     swapped = true;
25                 }
26             }
27         }
28         i++;
29     }
30     return numbers;
31 }

```



RC = 7 ; A = 20 ; N = 14 ; NP= 7

$V(G) = NP + 1 = 7+1=6$

$V(G) = A - N + 2 = 20 - 14 + 2 = 8$

$V(G) = RC + 1 = 7 + 1 = 8$

3. Las pruebas se encuentran Adjuntadas.

4. Las pruebas se encuentran Adjuntadas.

5. Pruebas de Versión (sistema) Para el caso de uso Realizar Venta diseñar 2 (dos) casos de prueba. Los casos se deben preparar en la plantilla que se adjunta.

| Caso de prueba                     |                                |
|------------------------------------|--------------------------------|
| <b>ID:</b> 0001                    | <b>Nombre:</b> Buscar artículo |
| <b>Descripción:</b>                |                                |
| <b>Prioridad:</b> Alta             | <b>CU/HU:</b> Realizar Venta   |
| <b>Módulo/Funcionalidad:</b> Venta |                                |
| <b>Diseñado por:</b>               | <b>Fecha:</b> 13/11/2023       |
| <b>Ejecutado por:</b>              | <b>Fecha:</b> 13/11/2023       |

**Precondiciones:**

Vendedor autenticado y autorizado.

Punto de venta habilitado.

| Paso | Acción                       | Resultado Esperado  | Paso/Fallo | Comentarios |
|------|------------------------------|---|------------|-------------|
| 1    | Ingresar código del producto | se muestra en tabla detalle del producto (marca, categoría)<br>lista los talles y colores disponibles con su respectivo stock |            |             |

| Caso de prueba  |                                    |
|---|------------------------------------|
| <b>ID:</b> 0002   | <b>Nombre:</b> Cliente inexistente |
| <b>Descripción:</b> Realizar una nueva ventana con un cliente inexistente |                                    |
| <b>Prioridad:</b> Alta  | <b>CU/HU:</b> Realizar Venta       |
| <b>Módulo/Funcionalidad:</b> Existencia de Cliente                        |                                    |
| <b>Diseñado por:</b>  | <b>Fecha:</b> 13/11/2023           |
| <b>Ejecutado por:</b>   | <b>Fecha:</b> 13/11/2023           |

**Precondiciones:**

Vendedor autenticado y autorizado.  
Punto de venta habilitado.  
Nueva venta iniciada

| Paso | Acción  | Resultado Esperado  | Paso/Fallo | Comentarios  |
|------|---|---|------------|--|
| 1    | Ingresar el código del producto                           | Visualizar datos del producto, colores y talles disponibles |            |  |
| 2    | Seleccionar color, talla e ingresar cantidad del producto | Visualizar precio unitario y subtotal a pagar               |            |  |
| 3    | Finalizar carga de productos                              | Visualizar total a pagar                                    |            | El importe considerado es mayor a \$92.720 por lo tanto requiere factura A |
| 4    | Ingresar cuit del cliente                                 | Mensaje de error, indicando que el cuit no está registrado. |            |  |