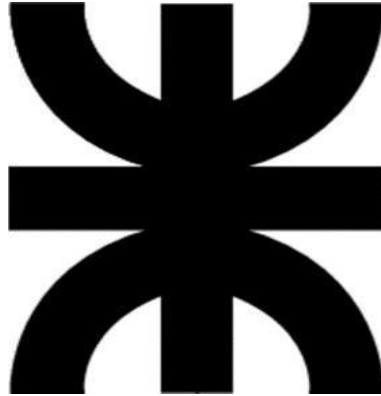


Universidad Tecnológica Nacional
Facultad Regional Tucumán



TP3:

Materia: Ingeniería del Software

Tema:

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

Apellidos y Nombres - Legajo:

- Álvarez, Marcelo Simón - 48282
- Ferjancic, Ian Ventzel - 52614
- Figueroa, Felix Ignacio - 46276
- Haro, Emanuel - 50439
- Masucci, Maribel - 50067

Docentes:

- Ing. Vicente Francisco
- Núñez Marcelo
- Pascual Facundo

Comisión: 4K3

Fecha de Presentación: 15/11/23

2023

1. Pruebas 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.

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 }
```

a)

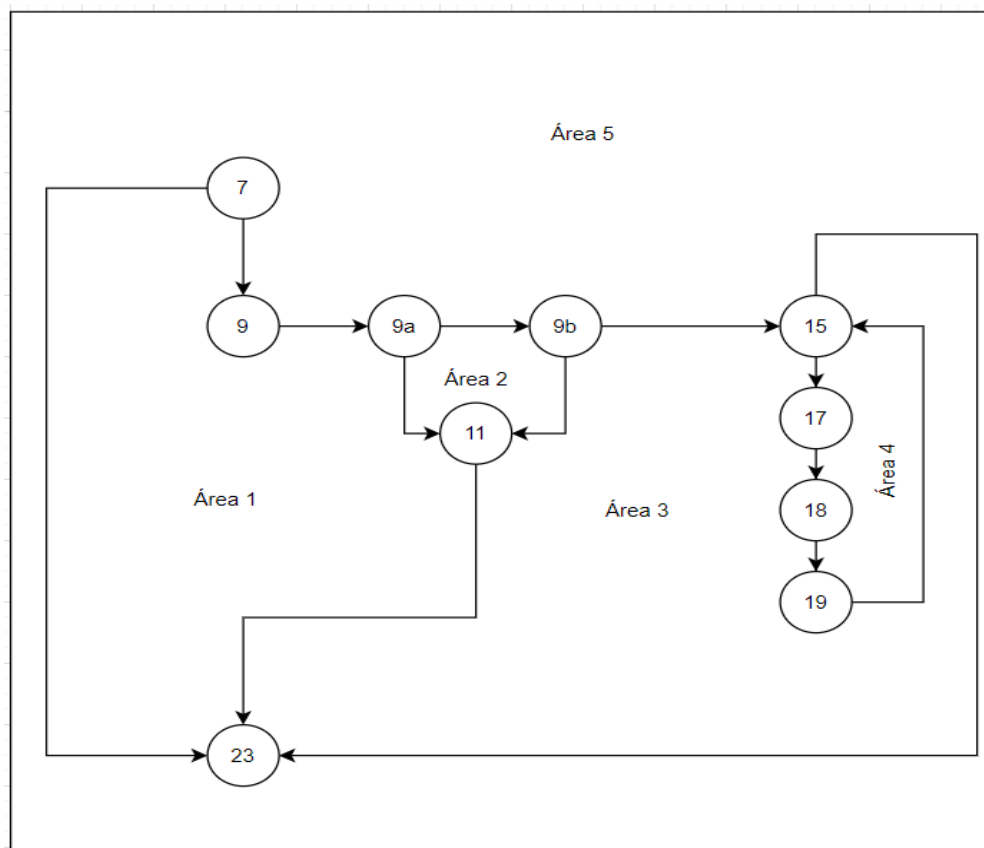
Porción	Ejemplos	Salidas esperadas	Salidas obtenidas
Entrada < 4	6666-9999-1	No aceptado	
Entrada entre (4-8)	10000-9999999-100000	Aceptado	
Entrada > 8	100000000-999999999	Aceptado	
≠int	HolaBB-@@@@@-19,25	No aceptado	

b)

Porción	Ejemplos	Salidas esperadas	Salidas obtenidas
0 > Entrada	-1 ; -10 ; -100000	No aceptado	
Entrada entre [0-1]	0 ; 1	0 ; 1	
Entrada >1	2 ; 5 ; 10	1 ; 5; 55	
Entrada ≠int	estoycansadojefe ;	No aceptado	

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.



Complejidad Ciclomática 1b):

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

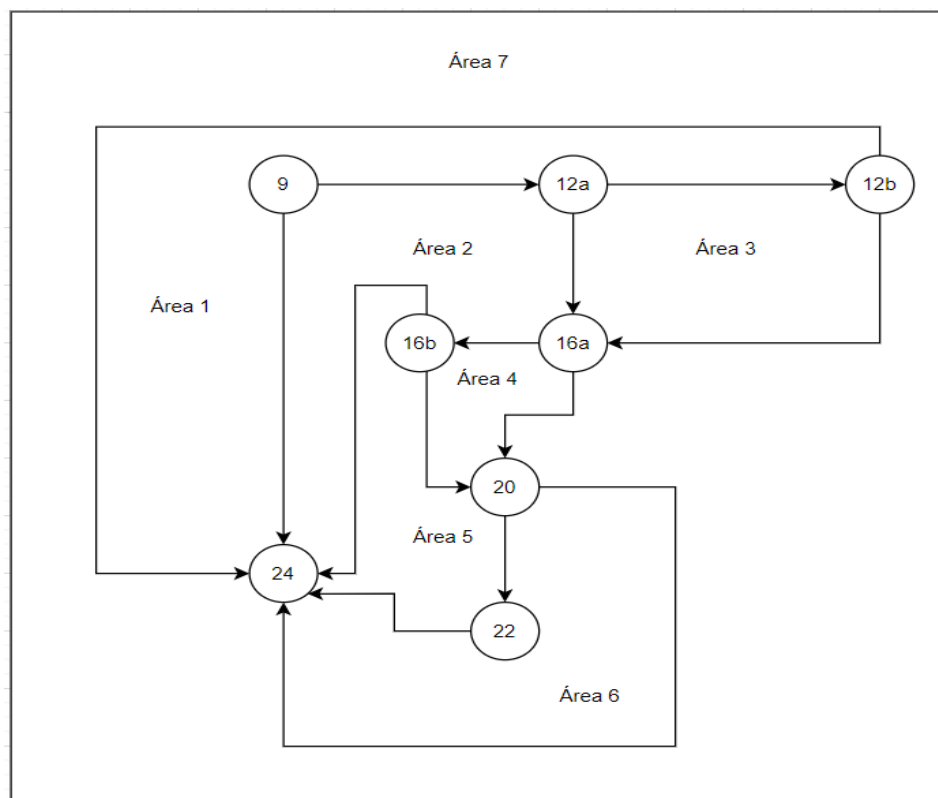
$$V(G) = A - N + 2 = 13 - 10 + 2 = 5$$

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

```

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 }

```



Complejidad Ciclomática ReglaDeDescuento:

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

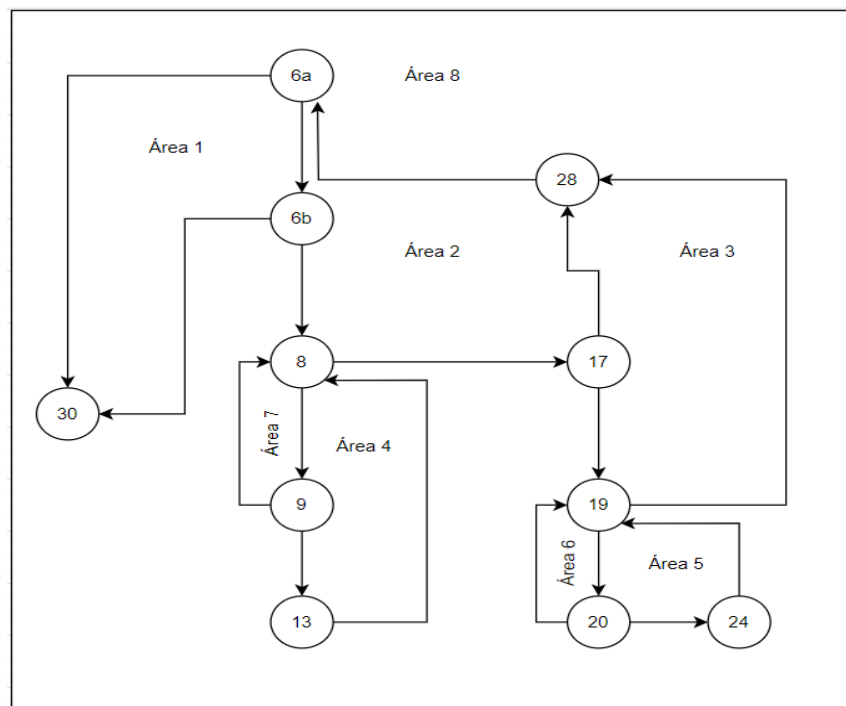
$$V(G) = A - N + 2 = 13 - 8 + 2 = 7$$

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

```

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             i++;
28         }
29     }
30     return numbers;
31 }

```



Complejidad Ciclomática CocktailSort:

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

$$V(G) = A - N + 2 = 17 - 11 + 2 = 8$$

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

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.

Plantilla para caso de prueba

Caso de Prueba	
ID: 1	Nombre: Realizar pago con tarjeta
Descripción: Realizar pago con tarjeta de crédito/débito	
Prioridad: Media	CU / HU: Realizar venta
Módulo / Funcionalidad: Pago con tarjeta	
Diseñado por: Grupo 11 – 4K3	Fecha: 15/11/2023
Ejecutado por:	Fecha:
Precondiciones: Vendedor autenticado y con permisos para realizar la operación El sistema debe estar conectado al sistema externo AFIP La venta está en curso	

Paso	Acción	Resultado Esperado	Pasó / Falló	Comentarios
1	Se selecciona forma de pago con tarjeta	Mostrar los campos: Nro de Tarjeta, Fecha Vto, Código de Seguridad y Monto.		
2	Ingresar los datos de la tarjeta	Nro de Tarjeta, Fecha Vto, Código de Seguridad		
3	Confirmar el pago	Número de operación exitoso y mensaje "Pago realizado exitosamente"		

ID: Identificador | CU: Caso de Uso | HU: Historia de Usuario

Caso de Prueba	
ID: 2	Nombre: Realizar venta simple
Descripción: Realizar venta con pago en efectivo y consumidor final. (Sin conexión al sistema externo de pago con tarjeta).	
Prioridad: Alta	CU / HU: Realizar venta
Módulo / Funcionalidad: Realizar venta	
Diseñado por: Grupo 11 – 4K3	Fecha: 15/11/2023
Ejecutado por:	Fecha:
Precondiciones: El vendedor debe estar autenticado en el sistema. Deben existir artículos cargados en el sistema. El sistema debe esta conectado al sistema externo AFIP	

Paso	Acción	Resultado Esperado	Pasó / Falló	Comentarios
1	Se ingresa el código del artículo.	El sistema muestra marca, color, talla, categoría y stock del respectivo artículo.		
2	Se selecciona el color y talla del artículo. Se agrega a la línea de venta.	El sistema registra una nueva línea de venta y presenta descripción del artículo., precio de venta (Precio de Venta = Neto Gravado + IVA), subtotal y total.		
3	Se elige forma de pago en efectivo y consumidor final.	El sistema solicita ingresar la cantidad de dinero entregada por el cliente.		
4	Se ingresa la cantidad de dinero entregada por el cliente	El sistema calcula el vuelto.		
5	Se confirma la venta.	El sistema imprime el comprobante, registra la venta y actualiza el stock.		

ID: Identificador | CU: Caso de Uso | HU: Historia de Usuario