UF2. [PAC01] Solución

Para responder a las siguientes cuestiones deberéis ayudaros del libro de texto, capítulo 3, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga UN ÚNICO DOCUMENTO PDF con las respuestas de la parte teórica y con los ejercicios y los grafos de la parte práctica.

Todas las preguntas valen lo mismo, 2 puntos cada una, por lo que cada pregunta del tipo test tiene un valor de 0,33 puntos.

Las preguntas tipo test se corregirán sólo en la caja que veréis después de las preguntas.

Recordad que la fecha límite para la entrega de esta PAC es el día 14 de NOVIEMBRE.

Información: Cualquier PAC copiada y/o en la que se haya utilizado "copy-paste" de código ya escrito será puntuada con un 0.

Actividades

Parte teórica

1. Responde a este test en la matriz de respuestas que se encuentra al final de las preguntas:

   1.1. Las pruebas estructurales

   a) Son las pruebas de caja blanca.
   b) Son las pruebas de caja negra.
   c) Son las pruebas de comportamiento.
   d) Sólo miden la entrada y salida.

   1.2. ¿Cómo se llaman las pruebas que se hacen al software en el entorno real de trabajo?

   a) Pruebas unitarias.
   b) Pruebas de integración.
   c) Pruebas de validación.
   d) Pruebas de sistema.

1.3. ¿Cómo se llaman las pruebas dónde después de probar cada módulo por separado se combinan todos y se prueba el programa completo?

a) Integración incremental.
b) Integración no incremental.
c) Prueba alfa.
d) Prueba.

1.4. ¿Cuáles de estos documentos se producen durante el proceso de prueba?

a) Plan de pruebas.
b) Especificaciones de prueba.
c) Informes de pruebas.
d) Todos los anteriores.

1.5. La prueba de camino básico…

a) Es una técnica de prueba de caja blanca.
b) Es una técnica de prueba de caja negra.
c) Permite obtener una medida de complejidad lógica.
d) Las opciones a y c son correctas.

1.6. El siguiente grafo se corresponde con una estructura

a) HACER MIENTRAS
b) REPETIR HASTA
c) CONDICIONAL
d) SECUENCIAL

MATRIZ DE RESPUESTAS

1.1	1.2	1.3	1.4	1.5	1.6
A	**C**	**B**	**D**	**D**	**A**

2. ¿Qué estrategias se siguen para probar el software? Si las pruebas de unidad funcionan ¿es necesario hacer la prueba de integración?

Las estrategias son: Prueba de Unidad, prueba de Integración, prueba de Validación y prueba del Sistema.

Si es necesario, puesto que es el siguiente paso de las pruebas de software. Se prueban diferentes partes, en las pruebas de Unidad se prueba el código fuente mientras que en las pruebas de Integración se centran en el diseño.

Parte práctica

3. Rellena en la siguiente tabla los casos de prueba tomando como referencia las reglas del análisis de valores límite:

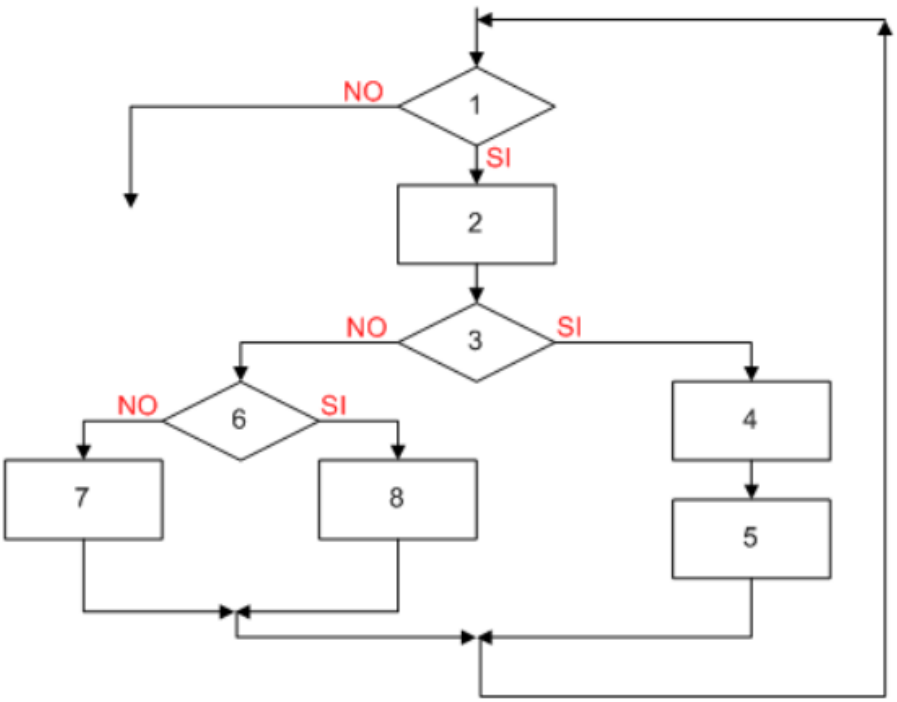
Condiciones de entrada y de salida	Casos de prueba
Una variable toma valores comprendidos entre -4 y 4 (enteros)	Para los valores: -4, 4, -3, 5, -6
El programa lee un fichero que contiene de 1 a 100 registros	Para 0, 1, 10, 100, 101 registros
El programa deberá generar de 1 a 5 listados	Para 0, 1, 5, 6 listados
El número de alumnos para calcular la nota media es 35	Para 0, 1, 35 y 36 alumnos
La función deberá devolver un array de enteros, de 1 a 10 valores	Para 1, 10, 0 y 11 enteros

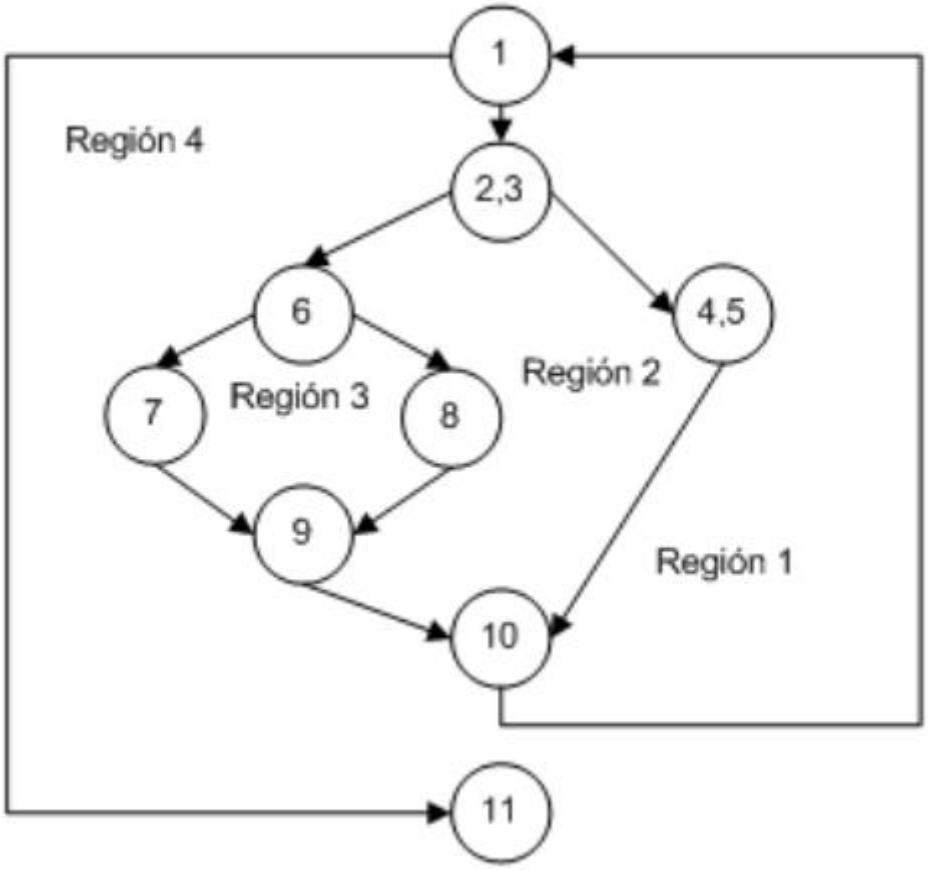
Realiza después un programa Java para probar la función que devuelve el array de enteros. Utiliza los casos de prueba que hayas definido.

```
public class ArrayEnteros {
	public static void main (String[] args) {
		int n = 1; //Valores de prueba: 0, 1, 5, 10, 54
		int[] tabla = enteros(n);

		try {
			System.out.println("Número de elementos: " +
tabla.length);

			for (int i = 0; i < n; i++) {
				System.out.println(tabla[i]);
			}
		} catch (NullPointerException e) {
```

```
                System.out.println("Array fuera de los límites");
            }
        }

        static int[] enteros (int n) {
            if (n < 1 || n > 10) {
                return null;
            }

            int[] tabla = new int[n];
            int j = 0;

            for (int i = 10; j < n; i++, j++) {
                tabla[j] = i;
            }

            return tabla;
        }
}
```

4. A partir del siguiente diagrama de flujo, construye el grafo de flujo. Indica el número de nodos, aristas, regiones, nodos predicado, la complejidad ciclomática y el conjunto de caminos independientes.





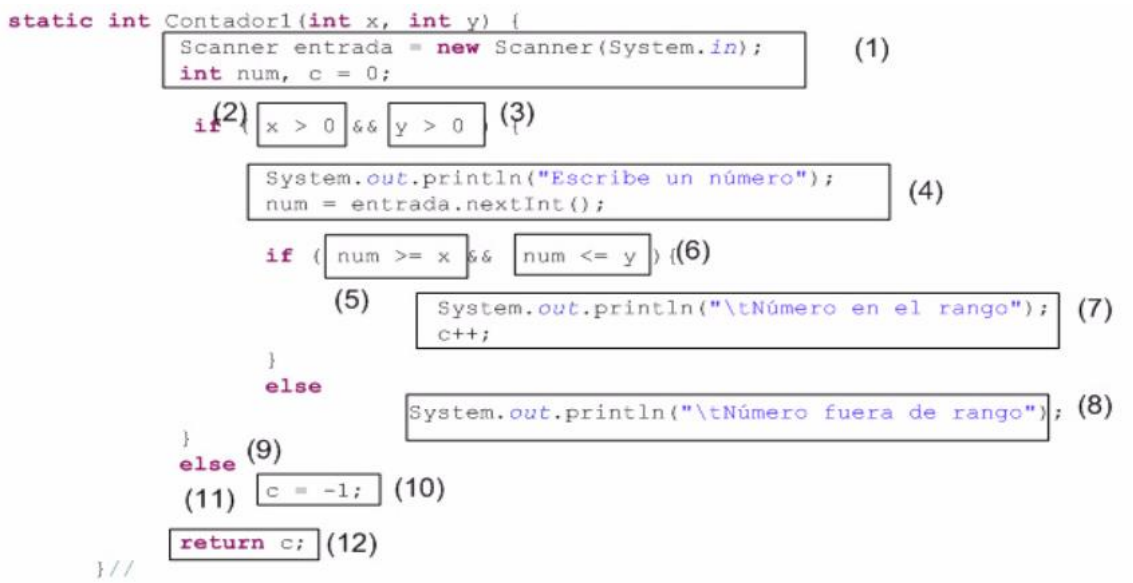
Nº Aristas = 11
Nº Nodos = 9
Nodos predicado = 3 → Nodo que contiene una condición (1; 2,3; 6)
Nº regiones = 4

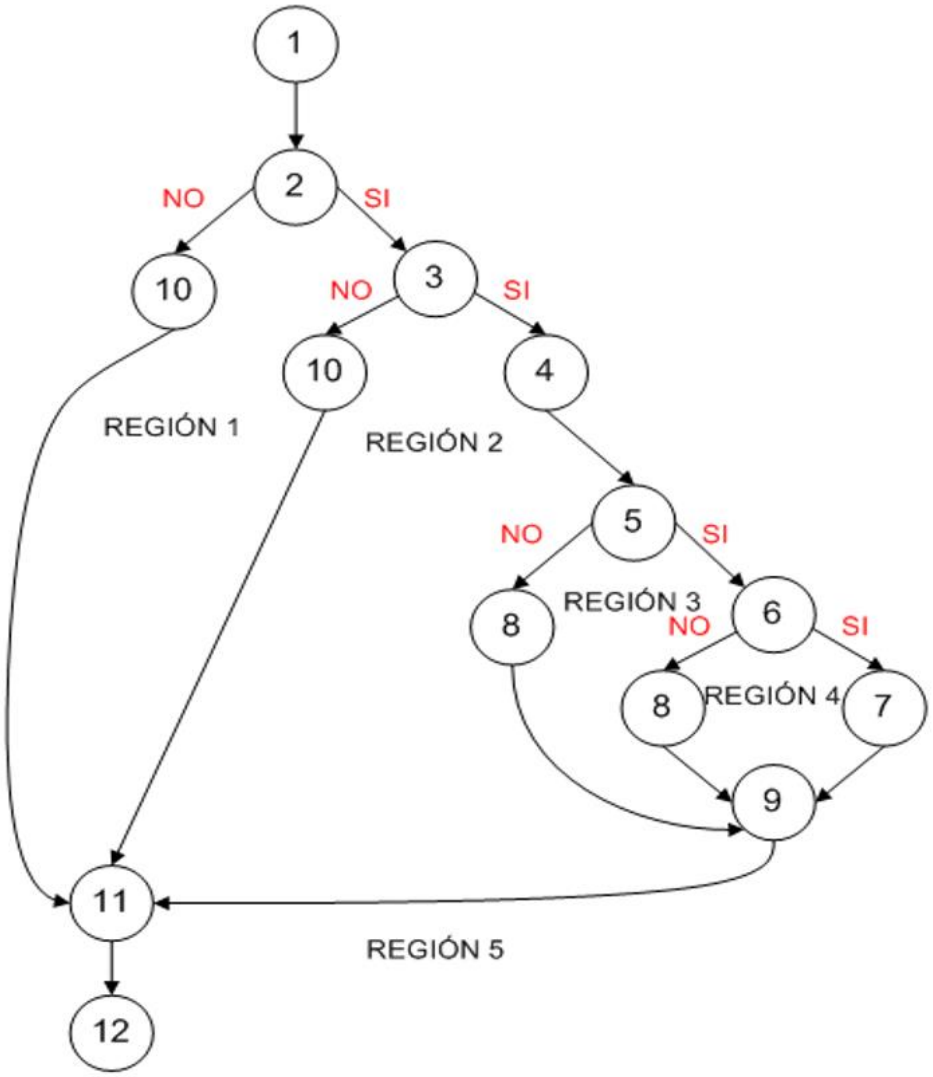
V(G) = 4, tenemos 4 caminos básicos, sin mucho riesgo.

Camino 1: 1 - 11
Camino 2: 1 - 2,3 - 4,5 - 10 - 1 - 11
Camino 3: 1 - 2,3 - 6 - 8 - 9 - 10 - 1 - 11
Camino 4: 1 - 2,3 - 6 - 7 - 9 - 10 - 1 - 11

5. Realiza el grafo de flujo, calcula la complejidad ciclomática, define el conjunto básico de caminos, elabora los casos de prueba para cada camino y evalúa el riesgo para la siguiente función Java:

```
static int Contador1(int x, int y) {
  Scanner entrada = new Scanner(System.in);
  int num, c = 0;
  if (x > 0 && y > 0) {
      System.out.println("Escribe un número");
      num = entrada.nextInt();
      if (num >= x && num <= y){
           System.out.println("\tNúmero en el rango");
           c++;
      }
      else
           System.out.println("\tNúmero fuera de rango");

      }
  else
      c = -1;
  return c;
}//
```





Nº Aristas = 17
Nº Nodos = 12
Nodos predicado = 4 → Nodo que contiene una condición (2; 3; 5; 6)
Nº Regiones = 5
V(G) = 5, tenemos 5 caminos básicos, sin mucho riesgo

Camino 1: 1 - 2 - 10 - 11 - 12
Camino 2: 1 - 2 - 3 - 10 - 11 - 12
Camino 3: 1 - 2 - 3 - 4 - 5 - 8 - 9 - 11 - 12
Camino 4: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9 - 11 - 12
Camino 5: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 11 - 12

Camino	Caso de prueba	Resultado esperado
Camino 1	Escoger algún X tal que NO se cumple la condición X>0 X = -1 Contador (-1, 10)	C = -1 Devuelve -1
Camino 2	Escoger algún X e Y tal que se cumpla la condición X > 0 y NO se cumpla Y > 0 X = 1, Y=-10 Contador (1, 10)	C = -1 Devuelve -1
Camino 3	Escoger algún X e Y tal que se cumpla la condición X > 0 e Y > 0 X = 1, Y = 10, num 5 Contador (1,10)	Número en el rango C = 1 Devuelve 1
Camino 4	Escoger algún X e Y tal que se cumpla la condición X > 0 e Y > 0 X = 1, Y = 10, num = 15 Contador (1,10)	Número en el rango C = 0 Devuelve 0
Camino 5	Escoger algún X e Y tal que se cumpla la condición X > 0 e Y > 0 X = 1, Y = 10, num = -15 Contador (1,10)	Número fuera de rango C = 0 Devuelve 0