

Boletín 1. Ejercicios de Pruebas de Software

Ejercicio 1: Dado el siguiente programa en java:

```
import java.util.Scanner;

public class Maximo {

    public static void main(String[] args) {

        int x = 0;
        int y = 0;
        int z = 0;
        int max = 0;
        Scanner sc = new Scanner(System.in);

        ① System.out.println("Introducir valor de A: ");
        x = sc.nextInt();
        System.out.println("Introducir valor de B: ");
        y = sc.nextInt();
        System.out.println("Introducir valor de C: ");
        z = sc.nextInt();

        ② if ((x>y) && (x>z)) {
            ③ max = x; ④
        }
        else{
            if (z>y) { ⑤
                max = z; ⑥
            }
            else{
                max = y; ⑦
            }
        }

        System.out.println ("El máximo es: " + max); ⑧
    }
}
```

Se pide:

- Obtener el grafo de flujo
- Calcular la complejidad ciclomática de McCabe $V(G)$
- Definir conjuntos de caminos básicos y realización de las pruebas mínimas.

Ejercicio 2: Dado el siguiente fragmento de programa en java:

```
if ((a>1) && (b>5) && (c<2)) {  
    x = x + 1;  
}  
else {  
    x = x - 1;  
}  
System.out.println ("El valor de x es: " + x);
```

Se pide:

- Obtener el grafo de flujo
- Calcular la complejidad ciclomática de McCabe $V(G)$
- Definir conjuntos de caminos básicos y realización de las pruebas mínimas.

Ejercicio 3: Dado el siguiente fragmento de programa en java:

```
if ((a>1) || (b>5) || (c<2)) {  
    x = x + 1;  
}  
else {  
    x = x - 1;  
}  
System.out.println ("El valor de x es: " + x);
```

Se pide:

- Obtener el grafo de flujo
- Calcular la complejidad ciclomática de McCabe $V(G)$
- Definir conjuntos de caminos básicos y realización de las pruebas mínimas.

Ejercicio 4: Realiza el grafo de flujo, calcula la complejidad ciclomática, define el conjunto básico de caminos y elabora los casos de prueba para la siguiente función Java:

```
static int Contador (int x, int y) {
    ① Scanner entrada= new Scanner (System.in);
    int num c= 0;
    if (x>0 && y>0) {
        ④ System.out.println ("Escribe un número");
        num= entrada.nextInt();
        if (num>= x && num<= y) {
            ⑤ System.out.println ("\tNúmero en el rango");
            ⑦ c++;
        }
        else
            System.out.println ("\t Número fuera de rango"); ⑧
    }
    else
        c= -1; ⑨
    ⑩ entrada.close();
    return c;
} //
```

Ejercicio 5: Recupera el ordinograma de los números primos

Se pide:

- Obtener el grafo de flujo
- Calcular la complejidad ciclomática de McCabe $V(G)$

Ejercicio 6: Dado el siguiente programa en Java

```
① { Scanner sc = new Scanner(System.in);
    System.out.println("Escoja una opción");
    int opcion = sc.nextInt();
    ② switch (opcion) {
        case 1:
            ③ {
                int multi = 1;
                int total = 0;
                ④ while (total<100) {
                    ⑤ total = 11 * multi;
                    ⑥ if (total <= 100) {
                        ⑦ System.out.printf("%d %n", total);
                    }
                }
            }
        }
    }
```



```

    }
    multi++; ⑧
  }
  break; ⑨
}
case 2:
⑩ {
  int cont = 1;
  ⑪ for (int i=2; i<=1000; i++){
    ⑫ if ( esPrimo(i) ){
      ⑬ if (cont<5){
        ⑭ System.out.printf("%5d", i);
        cont++;
      }
      else{
        ⑮ System.out.printf("%5d\n", i);
        cont = 1;
      }
    }
  }
  break; ⑯
}
case 3:
⑰ {
  System.out.println("Este caso parece fácil, pero tiene
  trampa...");
}
default:
⑱ {
  System.out.println("Opción incorrecta");
  break;
}
}
⑲ System.out.println("FIN del programa.");

```

Se pide:

*Obtener el grafo de flujo y Calcular la complejidad ciclomática de McCabe
V(G)*

NOTA IMPORTANTE:

Debido a que las *sentencias tipo CASE* pueden convertirse en IFs encadenados, hay que tener en cuenta cómo se calcula la complejidad ciclomática contando nodos predicados: Cada valor con el que se compara la variable es equivalente a un nodo predicado ("default" no se cuenta como opción).

Sea el siguiente fragmento de código:

```

switch (opcion) {
    case 1:{
        break;
    }
    case 2:{
        break;
    }
    case 3:{
        break;
    }
    default:{
        break;
    }
}

```

$$V(G) = \text{Nodos Predicados} + 1 = 3 + 1 = 4$$

Ejercicio 7: Dado el siguiente programa en java:

```

① { Scanner sc = new Scanner(System.in);
    System.out.println("Escoja una opción");
    int opcion = sc.nextInt();
    ② switch (opcion) {
        case 1:
            ③ {
                ④ calcularMultiplosOnce();
                break;
            }
        case 2:
            ⑤ {
                SmostrarPrimosDosAMil();
                ⑥ break;
            }
        case 3:
            ⑦ {
                System.out.println("Este caso parece fácil, pero tiene
                trampa...");
            }
        default:
            ⑧ {
                System.out.println("Opción incorrecta");
                ⑨ break;
            }
    }
    ⑩ System.out.println("FIN del programa.");
}

```

Se pide:

Obtener el grafo de flujo y Calcular la complejidad ciclomática de McCabe

V(G)