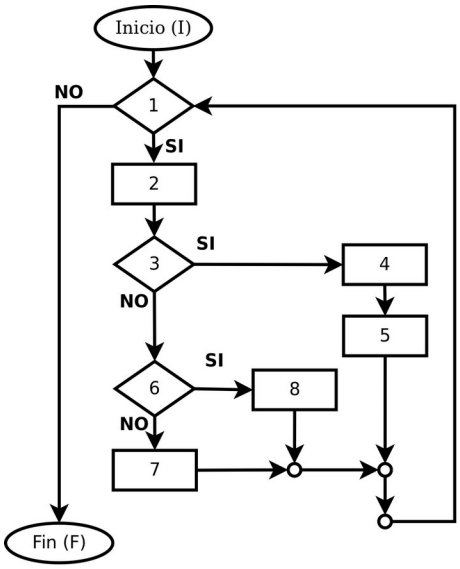
Tema 3: Pruebas del software

Práctica 1: Pruebas del software

Primero: Grafo1

A partir del siguiente diagrama de flujo, construye el grafo de flujo e indica:

- Número de nodos.
- Número de aristas.
- Número de regiones.
- Número de nodos predicado.
- Complejidad ciclomática.
- Secuencia de nodos de todos los caminos del conjunto básico



Segundo: Grafo2. Factorial

Con el siguiente método ***Java***:

- Anota la numeración de cada elemento.
- Construye el grafo de flujo.
- Calcula la complejidad ciclomática.
- Evalúa el riesgo.
- Define el conjunto básico de caminos.
- Define un caso de prueba para cada camino (valores de entrada y resultado esperado).

```java
/*
 * Método factorial: Recibe un entero y devuelve su factorial. El factorial
 * de un número es el resultado de multiplicar todos los números
 * comprendidos entre 1 y ese número. Este método resuelve el problema sin
 * llamadas recursivas
 */
public static int factorial(int n) {
    int resultado;

    resultado = 1;
    for (int i = 2; i <= n; i++) {
            resultado = resultado * i;
    }
    return resultado;
}
```

Tercero: Grafo3. Divisible

Con el siguiente método ***Java***:

- Numera los elementos.
- Construye el grafo de flujo.
- Calcula la complejidad ciclomática.
- Evalúa el riesgo.
- Define el conjunto básico de caminos.
- Define un caso de prueba para cada camino (valores de entrada y resultado esperado).

```java
/*
 * Método divisible: Recibe dos números enteros (multiplo y divisor)
 * y devuelve true si el primero es divisible por el segundo. Se
 * dice que un número (multiplo) es divisible por otro (divisor) si la
 * división es exacta, es decir, de resto 0 */
public boolean divisible(int multiplo, int divisor) {
    boolean resultado;

    if (multiplo % divisor == 0) {
            resultado = true;
    } else {
            resultado = false;
    }
    return resultado;
}
```

Cuarto: Grafo4. esPrimo

Con el siguiente método ***Java***:

- Numera los elementos.
- Construye el grafo de flujo.
- Calcula la complejidad ciclomática.
- Evalúa el riesgo.
- Define el conjunto básico de caminos.
- Define casos de prueba para cada camino (valores de entrada y resultado esperado).

```java
/*
 * Método esPrimo: Recibe un número entero y devuelve true si es
 * primo o false si no es primo. Se dice que un número es primo cuando es
 * mayor que 1 y no tiene más divisores que el 1 y a sí mismo. */
public boolean esPrimo(int n) {
    boolean primo;

    if (n <= 1) {
            primo = false;
    } else {
            primo = true;
            int i = 2;
            while (primo && i <= n / 2) {
                    if (divisible(n, i)) {
                            primo = false;
                    } else {
                            i++;
                    }
            }
    }
    return primo;
}
```

Quinto: Clases de equivalencia

Crea una tabla con casos de prueba, basados en las clases de equivalencia e incluyendo los valores límite, para los métodos ***factorial()***, ***divisible()*** y ***esPrimo()***.

Último:

Una vez terminados los ejercicios, deberás entregar un ***PDF*** llamado ***apellido1_apellido2.pdf*** con las respuestas de los 5 ejercicios. Súbelo con el enlace correspondiente de la plataforma ***Moodle***.

EL INCUMPLIMIENTO DE ESTAS INSTRUCCIONES PUEDE TENER COMO CONSECUENCIA LA NO CORRECCIÓN DE LA PRÁCTICA.