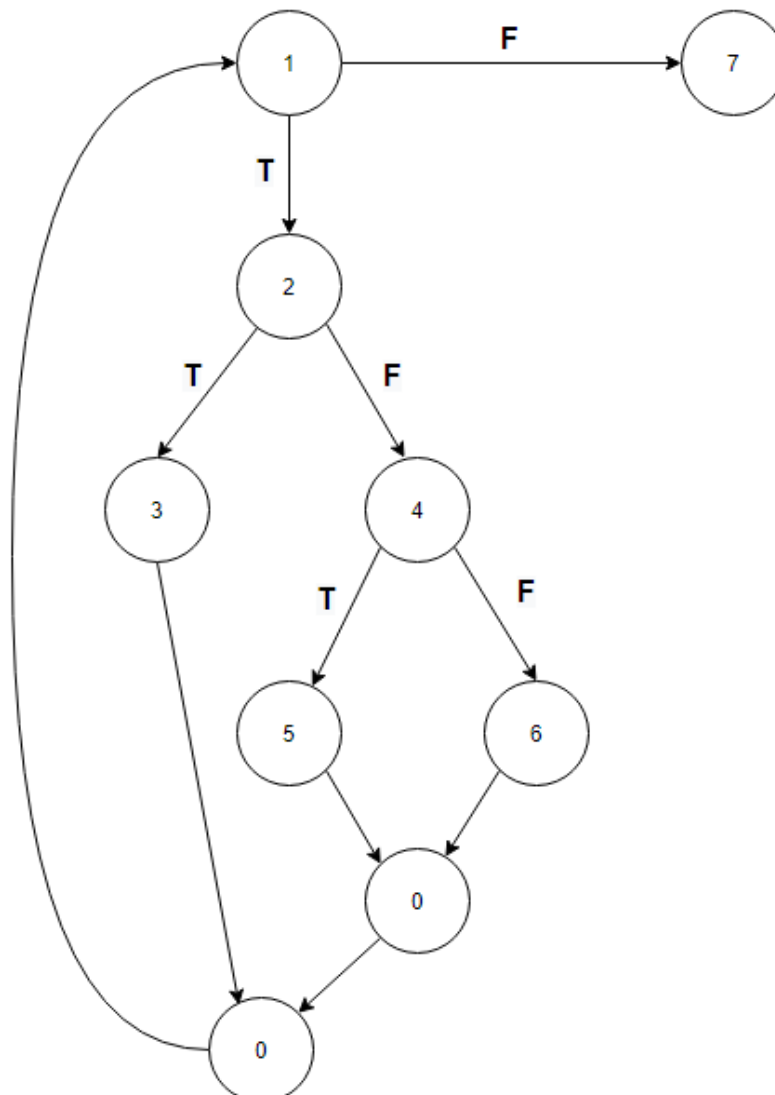


ACTIVIDADES 1 TEMA 6 PRUEBAS DEL SOFTWARE

Ejercicio 1. Dado el siguiente pseudocódigo:

```
/* Algoritmo de búsqueda binaria. Datos de entrada, llave y Lista. Lmin es la longitud mínima  
y lmax la longitud del array */  
Lee llave  
x1=lmin  
x2=lmax  
resp=-1  
mientras (x1<x2-1)  
  ix=(x1+x2)/2  
  si (Lista[ix]==llave) entonces  
    resp=ix  
    x1=ix  
  x2= ix  
  de otro modo si (Lista[ix]<llave) entonces  
    x1=ix  
  de otro modo x2=ix  
fin del mientras  
regresa resp
```

1. Obtener el grafo de flujo relacionado al anterior pseudocódigo.



2. Calcular la complejidad ciclomática.

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

$$V(G) = 11 - 9 + 2 = 4$$

3. Rellenar la tabla.

CAMINO	DATO DE PRUEBA	RESULTADO ESPERADO	RESULTADO REAL
1,7	llave = 1, $x_1 = 0$, $x_2 = 1$, resp=-1	resp = -1	resp= -1
1,2,3,1,7	llave = 1, $x_1 = 0$, $x_2 = 2$, resp=-1	resp = ix = 1 $x_1 = ix = 1$ $x_2 = ix = 1$	resp= 1 $x_1 = 1$ $x_2 = 1$
1,2,4,5,1,7	llave = 5, $x_1 = 0$, $x_2 = 3$, resp=-1	$x_1 = ix = 1,5$ resp = -1	$x_1 = 1,5$ resp= -1
1,2,4,6,1,7	llave = 1, $x_1 = 1$, $x_2 = 3$, resp=-1	$x_2 = ix = 2$ resp= -1	$x_2 = 2$ resp= -1

Ejercicio 2. Realiza la prueba de la siguiente condición usando las tres técnicas conocidas:

```
If ( x<10 && y>20)
    Z= 7
Else
    Z = 8
```

Cobertura de decisiones:

IF verdadero: $x=9$, $y= 21$
IF falso: $x = 11$, $y= 1$

Cobertura de condiciones:

Caso 1: $x=9$, $y= 21$ (resultado IF=V)
Caso 2: $x = 11$, $y= 1$ (resultado IF=F)

Cobertura de condiciones múltiple:

Caso 1 (V-V): $x=9$, $y= 21$ (resultado IF=V)
Caso 2 (F-F): $x = 11$, $y= 1$ (resultado IF=F)
Caso 3 (F-V): $x = 11$, $y= 21$ (resultado IF=F)
Caso 4 (V-F): $x = 9$, $y= 1$ (resultado IF=F)

Ejercicio 3. Realiza la prueba de bucles del siguiente fragmento. La inicialización de **s** es un dato de entrada que se le pregunta al usuario:

```
do
{
    s = s+1
    s = Console.WriteLine("¿Quieres seguir? (no=10,sí=otro número)");
    if(s == 10)
        break;
    Console.WriteLine("Introduce un número: ");
    numero = Int.Parse(Console.ReadLine());
    if( numero % 2 == 0)
        Console.WriteLine("El número es par");
}
while ( s < 10 );
```

- Saltar el bucle sin entrar en él:

No se puede saltar el bucle ya que es un do-while y tiene que entrar por él al menos una vez.

- Hacer m pasos por el bucle con $m < n$

Si $m = 3$

s = 5 en la primera iteración, después el usuario introducirá un número, si cumple la condición mostrará en pantalla "El número es par". El bucle continuará ya que $s < 10$, **s = 8** en la segunda iteración (s sigue siendo < 10 , por tanto, el bucle continúa), **s = 10** en la tercera iteración, de esta forma finalizará cuando entre en la condición: **if(s == 10)** y ejecute el **break**;

- Pasar una vez por el bucle:

Cuando $s \geq 10$ entra una vez y se sale del bucle por la propia condición (**while(s<10)**). Por ejemplo $s = 22$.

- Pasar 2 veces por el bucle.

s = 8, el usuario tendrá que introducir un número, si el número es par (**numero%2==0**), por ejemplo numero = 4: cumple la condición de **if(numero%2==0)** y mostraría por pantalla "El número es par", seguido de esto sería **s = s+1**. En la siguiente iteración el usuario le asigna el valor **s = 10**, por tanto, entra en la condición **if(s==10)** y se sale del bucle. Número de iteraciones: 2.

- Pasar n-1, n y n+1 veces por el bucle.

Siempre y cuando el usuario dentro del bucle no escriba 10 en el **s = int.Parse(Console.ReadLine())**, el bucle se va a seguir ejecutando. Finalizará cuando el usuario le dé un valor mayor que 9 a **s**.

Ejercicio 4. Se va a realizar una entrada de datos de un empleado por pantalla gráfica, se definen 3 campos de entrada y una lista para elegir el oficio. La aplicación acepta los datos de esta manera:

- Empleado: número de 3 dígitos que no empieza por 0
- Departamento: en blanco o número de dos dígitos
- Oficio: Analista, Diseñador, Programador o Elige oficio

Si la entrada es correcta el programa asigna un salario (que se muestra en pantalla) a cada empleado según estas normas:

- S1 si Oficio es Analista se asigna 2500
- S2 si el Oficio es Diseñador se asigna 1500
- S3 si el Oficio es Programador se asigna 2000

Si la entrada no es correcta el programa muestra un mensaje indicando la entrada incorrecta.

- ER1 si el Empleado no es correcto
- ER2 si el Departamento no es correcto
- ER3 si no se ha elegido Oficio

Representa las clases de equivalencia para cada condición de entrada mediante una tabla.

Condición de entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Empleado	Rango	1: $999 \leq \text{numero} \leq 100$	ER1 2: $999 \geq \text{numero} \leq 100$
Departamento	Rango	3: $99 \leq \text{departamento} \leq 10$ 4: (en blanco)	ER2 5: $99 \geq \text{departamento} \leq 10$
Oficio	Conjunto con comportamiento distinto	6: "(cualquier oficio)" 7: "Analista" 8: "Diseñador" 9: "Programador"	ER3 10: (en blanco)

Ejercicio 5. A partir de la tabla (Definición de Clases de Equivalencia) obtenida en la actividad anterior, debes obtener los casos de prueba correspondientes. Para ello, deberás de representarlos en otra tabla donde cada fila representa un caso de prueba con:

- Los códigos de las clases de equivalencia que se aplican (los que asignaste en la anterior, aunque también se podría haber asignado un número a cada clase)
- Los valores asignados a las condiciones de entrada
- El resultado esperado según el enunciado del problema

Para obtener los casos de prueba deberás de tener en cuenta estas dos reglas:

- Los casos de prueba deben de cubrir la mayor cantidad de clases de equivalencia válidas (podría escribirse solo uno que cubriera todas las clases de equivalencia válidas)
- Se escribe un solo caso de prueba por cada clase de equivalencia inválido. Si se prueban múltiples clases de equivalencia no válidas en el mismo caso de prueba, pueden darse alguna de estas situaciones:
 - que algunas de estas pruebas nunca se ejecuten porque la primera enmascara a las otras o
 - que alguna de estas pruebas termina la ejecución del caso de prueba.

Los casos de prueba se van añadiendo a la tabla hasta que todas las clases de equivalencias válidas y no válidas hayan sido cubiertas.

CASOS DE PRUEBA

Nº Caso	Clases de equivalencia	Dato de prueba	Resultado esperado	Resultado real
1	numero >= 100 y numero <=999	numero = 432	válido	válido
2	numero < 100 o numero > 999	numero = 4556	no válido	ER1 no válido
3	Departamento > 0 y Departamento <100	Departamento = 44	válido	válido
4	Departamento en blanco		válido	válido
5	Departamento < 0 o Departamento > 99	Departamento = 342	no válido	ER2 no válido
7	Oficio = "Analista"	Analista	2500	S1 válido
8	Oficio = "Diseñador"	Diseñador	1500	S2 válido
9	Oficio = "Programador"	Programador	2000	S3 válido
6	Oficio = "Administrador"	Administrador	válido	válido
10	Oficio no elegido		no válido	ER3 no válido