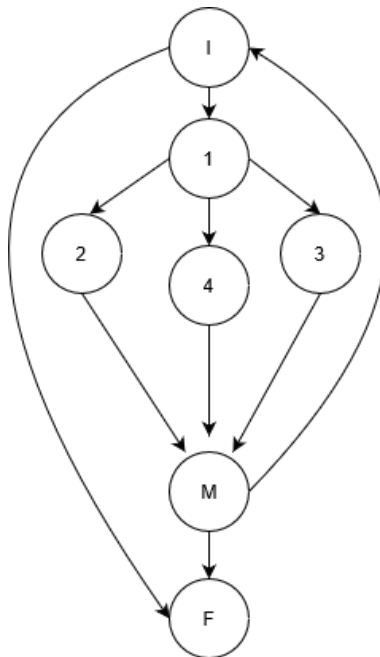


Entornos de desarrollo

Ejercicio 1. Dado el siguiente pseudocódigo:

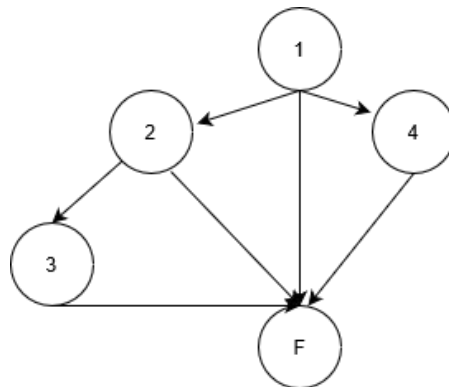
1. Obtener el grafo de flujo relacionado al anterior pseudocódigo
2. Calcular la complejidad ciclomática
3. Rellenar la tabla



$$V(G) = 10 \text{ aristas} - 7 \text{ nodos} + 2 = 5$$

Camino	Dato de prueba	Resultado esperado	Resultado real
Lista[ix]==llave	x1=2, x2=4	resp=ix	3
Lista[ix]<llave	x1=2, x2=4	x1=ix	2
Lista[ix]>llave	x1=2, x2=4	x2=ix	4

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



Por cobertura de decisiones:

IF Verdadero: $x=8, y=22$

IF Falso: $x=12, y=18$

Cobertura de condiciones:

Caso 1: $x=8, y=22$, (resultado if= V)

Caso 2: $x=12, y=18, Z=8$ (resultado if= F)

Cobertura de condiciones múltiple:

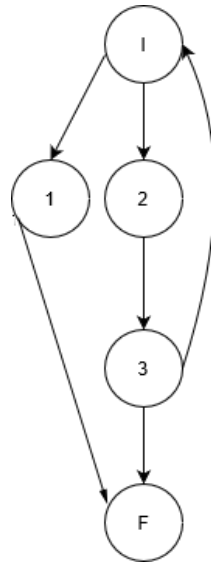
Caso 1 (V-V): $x=8, y=22$ (if = V)

Caso 2 (V-F): $x=8, y=18$ (if = F)

Caso 3 (F-V): $x=12, y=22$ (if = F)

Caso 4: (F-F): $12, y=18, Z=8$ (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:



- Pasar por alto totalmente el bucle.

Cuando s sea mayor que 10, no entra al bucle.

- Pasar una sola vez por el bucle.

Cuando s sea menor que 10, por ejemplo 8:

$8=8+1;$

$9<10$, entonces sigue el bucle y continúa con la sentencia if.

Introduce otro número, por ejemplo, el 5, pasa a otro if donde se hace $\%2$.

El número no es par.

- Pasar dos veces por el bucle.

$9=9+1$

$10==10$

Fin del bucle, no pasa a sentencia if.

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

Mientras que s sea menor que 10, entrará al bucle y sumará cada iteración 1 al número preguntado al usuario. Si el número es distinto a 10, se pedirá otro número y se comprobará si es par.

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 empiece por 0
- Departamento: en blanco o número de dos dígitos
- Oficio: Analista, Diseñador, Programador o Elige oficio

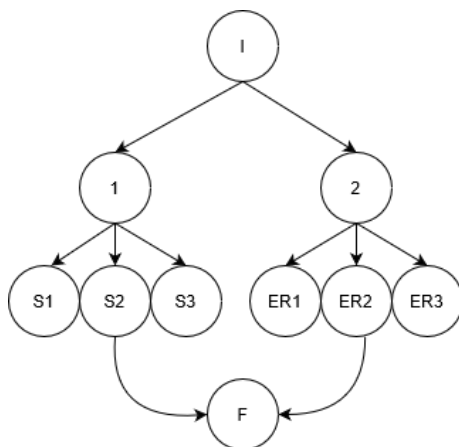
Si la entrada es correcta el programa asigna un salario (que se muestra en pantalla) a cada empleado según estas normas:

- 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,



Entrada	Rango de valores	Caso de prueba	Número de valores	Conjunto de valores	Resultado esperado
Empleado	- No empiece por 0 (válido). - Empiece por 0 (inválido)	ER1	- Valores < 3 dígitos (válido) - Valores > 3 dígitos (inválido)	- (empleado: x ∈ {"a"}) (válido) - (empleado: x ∉ {"a"}) (inválido)	- No empieza por 0 (válido) - Empieza por 0 (inválido)
Departamento	- Valor vacío o dígitos = 2 dígitos (válido) - Número < 0 > 2 dígitos (inválido)	ER2	- Valor vacío = 2 Dígitos (válido) - Menor o mayor a 2 (inválido)	- (dpto: x ∈ {"a","b"}) (válido) - (dpto: x ∉ {"a","b"}) (inválido)	- Longitud 2 dígitos (válido) - Longitud < 0 > 2 dígitos (inválido)
Oficio	= Analista o = Diseñador o = Programador	- ER3 - S1 S2 S3	Clases predefinidas	- (oficio: x ∈ {"a","d","p"}) (válido) - (oficio: x ∉ {"a","d","p"}) (inválido)	- Analista (válido) - Clase no definida (inválido)

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 actividad 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 equivalencia válidas y no válidas hayan sido cubiertas.

Clases de equivalencia	Dato de prueba	Resultado esperado	Resultado real
Dígitos empleado = 3	123	válido	válido
Dígitos empleado != 3	23 1234	inválido	ER1
Dígitos empleado empieza por 0	012	inválido	ER1

Departamento en blanco		válido	válido
Dígitos departamento = 2	12	válido	válido
Dígitos departamento != 2	123 1	inválido	ER2
Oficio = Analista	Analista	válido	válido
Oficio = Diseñador	Diseñador	válido	válido
Oficio = Programador	Programador	válido	válido
Oficio = “ “	Publicista	válido	válido
Oficio no definido		inválido	ER3
Oficio = Analista	Analista	2500	S1
Oficio = Diseñador	Diseñador	1500	S2
Oficio = Programador	Programador	2000	S3