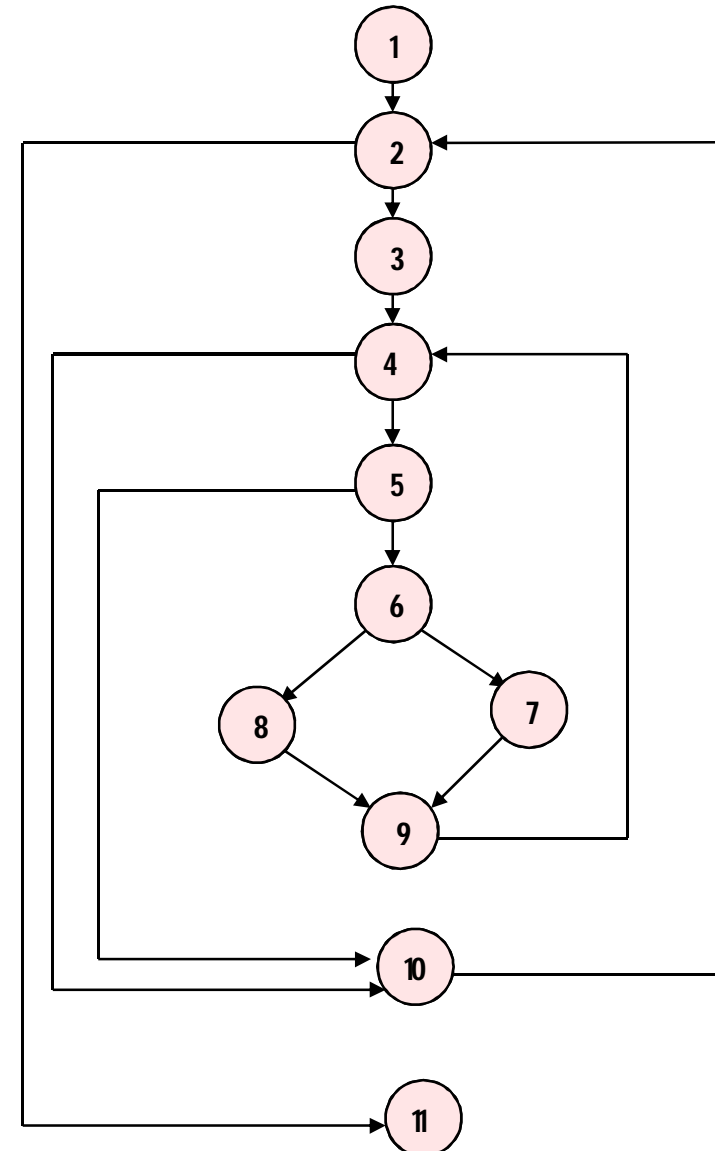
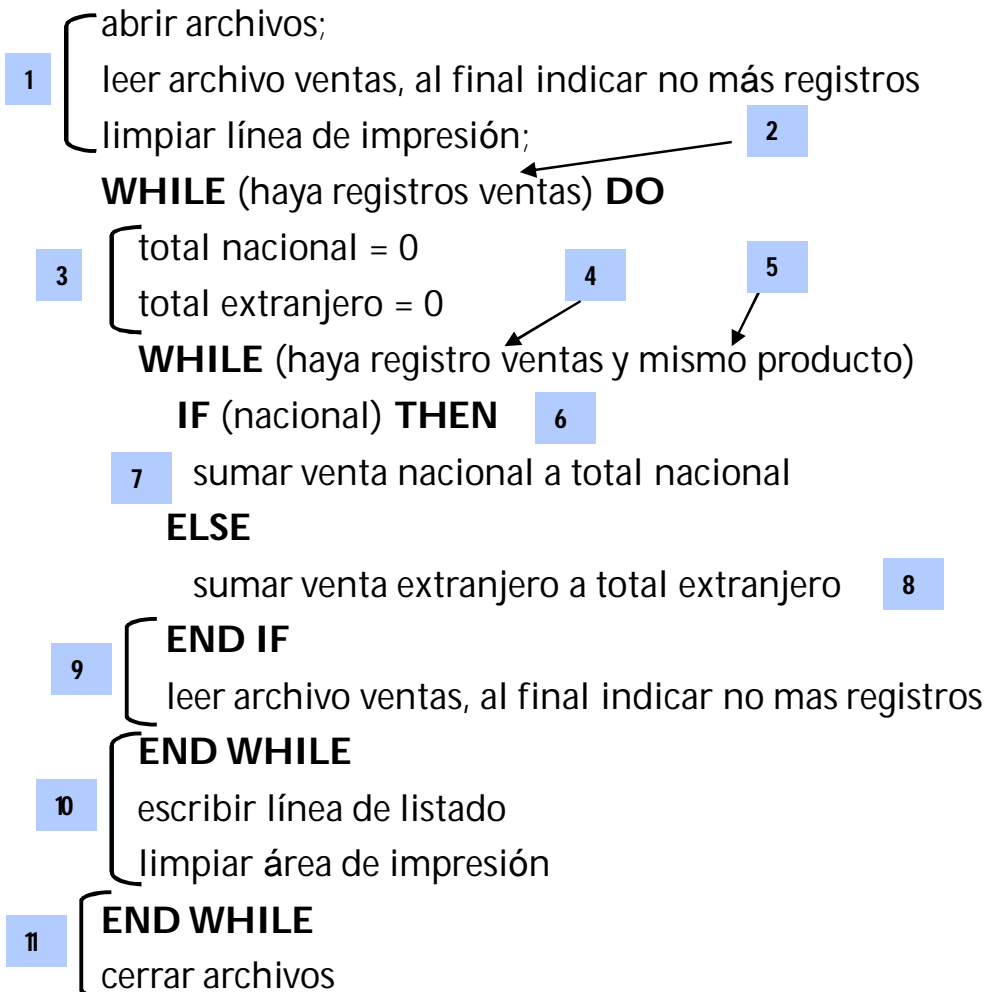


Validación y Verificación de software

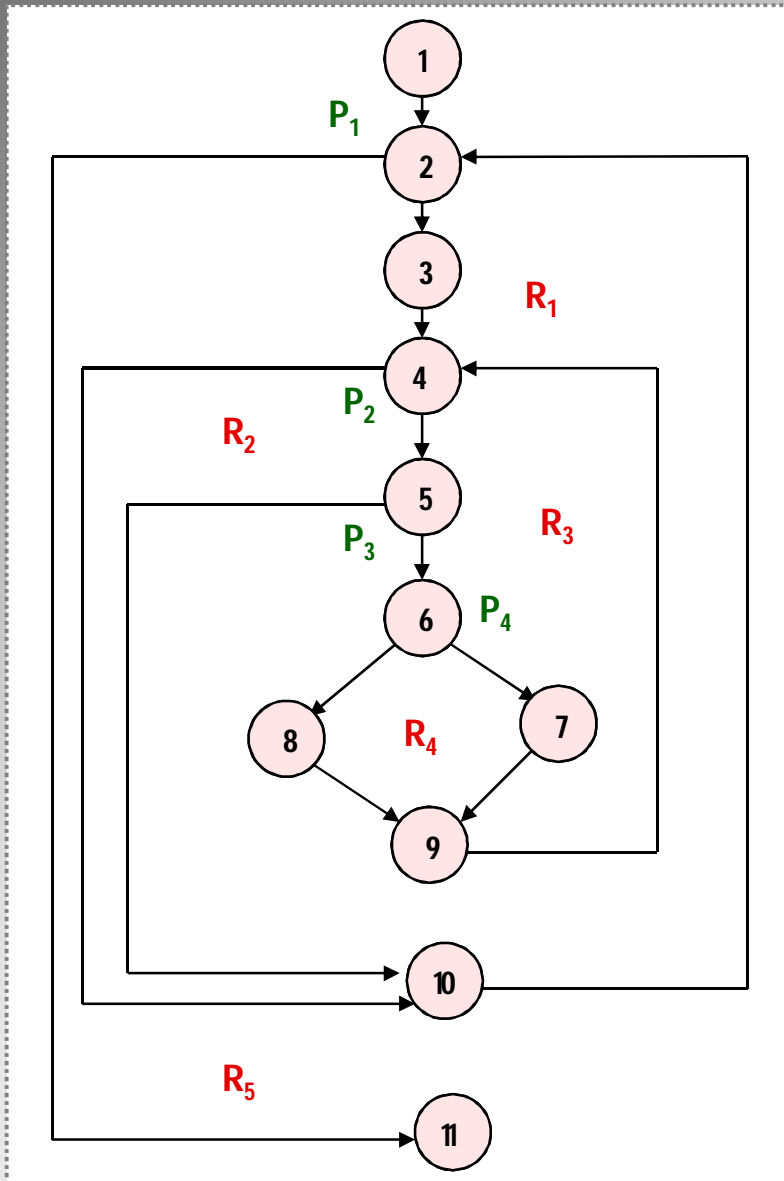
Solución de ejercicios

EJERCICIO 2

1 Representar el grafo de flujo



EJERCICIO 2



2 Calcular complejidad ciclomática

- $V(G) = 5$ regiones
- $V(G) = 14$ aristas - 11 nodos + 2 = 5
- $V(G) = 4$ nodos predicado + 1 = 5

3 Determinar conjunto básico

Camino 1: 1-2-11

Camino 2: 1-2-3-4-10-2-...

Camino 3: 1-2-3-4-5-10-2-...

Camino 4: 1-2-3-4-5-6-7-9-4-...

Camino 5: 1-2-3-4-5-6-8-9-4-...

EJERCICIO 3

1 Representar el grafo de flujo

BEGIN

1

leer datos de persona

código de barras = código de barras + 1

3

IF edad \geq 30 and persona = "VIP" **THEN**

2

IF cargo = "Director" **THEN**

4

5 color de tarjeta = "Dorado"

ELSE

6 color de tarjeta = "Rojo"

7

END IF

8

ELSE

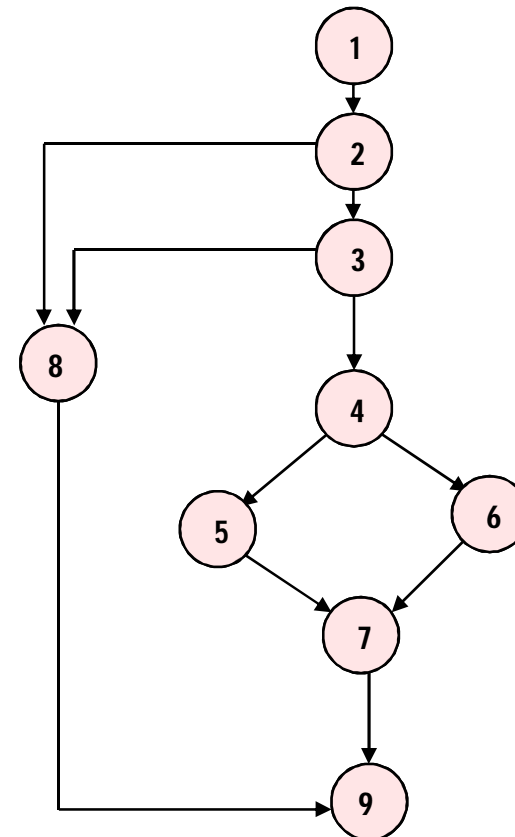
color de tarjeta = "Blanco"

END IF

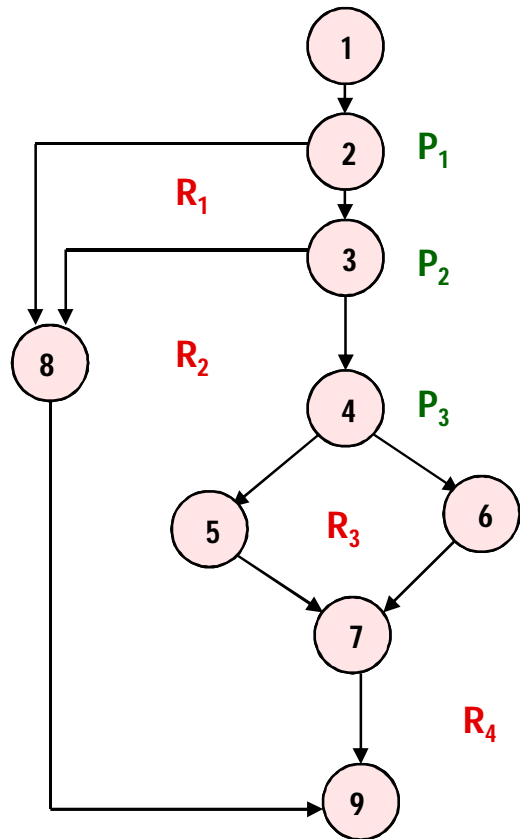
9

imprimir código de barras, nombre, color de tarjeta

END



EJERCICIO 3



2 Calcular complejidad ciclomática

- $V(G) = 4$ regiones
- $V(G) = 11$ aristas - 9 nodos + 2 = 4
- $V(G) = 3$ nodos predicado + 1 = 4

3 Determinar conjunto básico

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

EJERCICIO 3

4 Preparar casos de prueba

Camino 1: 1-2-8-9

edad < 30

Resultados esperados

color de tarjeta = "Blanco"

Camino 2: 1-2-3-8-9

edad >= 30 \wedge persona \neq "VIP"

Resultados esperados

color de tarjeta = "Blanco"

Camino 3: 1-2-3-4-5-7-9

edad >= 30 \wedge persona = "VIP" \wedge cargo = "Director"

Resultados esperados

color de tarjeta = "Dorado"

Camino 3: 1-2-3-4-6-7-9

edad >= 30 \wedge persona = "VIP" \wedge cargo \neq "Director"

Resultados esperados

color de tarjeta = "Rojo"

BEGIN

1

leer datos de persona

código de barras = código de barras + 1

3

IF edad >= 30 and persona = "VIP" THEN

2

IF cargo = "Director" THEN

4

5 color de tarjeta = "Dorado"

ELSE

6 color de tarjeta = "Rojo"

7

END IF

8

ELSE

color de tarjeta = "Blanco"

END IF

9

imprimir código de barras, nombre, color de tarjeta

END

EJERCICIO 6

Código de área

Número de tres dígitos que no comience por 0

Prefijo

Número de tres dígitos que no comience por 0 ó 1

Sufijo

Número de cuatro dígitos que comience por 2

Contraseña

String de 6 caracteres alfanuméricos

Órdenes

"comprobar". "depositar", "pagar factura", ...

EJERCICIO 6

Definir clases de equivalencia

Código de área ➔ Requiere un miembro de un conjunto

+ Clase válida Numérico

+ Clase inválida No Numérico

Rango de valores para Código de área ➔ Requiere un rango

+ Clase válida $100 \leq \text{Código de área} \leq 999$

+ Clase inválida Código de área < 100

+ Clase inválida Código de área > 999

Prefijo ➔ Requiere un miembro de un conjunto

+ Clase válida Numérico

+ Clase inválida No Numérico

Rango de valores para Prefijo ➔ Requiere un rango

+ Clase válida $200 \leq \text{Prefijo} \leq 999$

+ Clase inválida Prefijo < 100

+ Clase inválida $L > 999$

EJERCICIO 6

Sufijo

.....► Requiere un miembro de un conjunto

✚ Clase válida Numérico

✚ Clase inválida No Numérico

Rango de valores para Sufijo

.....► Requiere un rango

✚ Clase válida $2000 \leq \text{Sufijo} \leq 2999$

✚ Clase inválida $\text{Sufijo} < 2000$

✚ Clase inválida $\text{Sufijo} > 2999$

Contraseña

.....► Requiere un miembro de un conjunto

✚ Clase válida Alfanumérico

✚ Clase inválida No Alfanumérico

Longitud de contraseña

.....► Requiere un rango

✚ Clase válida Longitud = 6

✚ Clase inválida Longitud < 6

✚ Clase inválida Longitud > 6

Órdenes

.....► Requiere un miembro de un conjunto

✚ Clase válida "comprobar", "depositar", "pagar factura", ...

✚ Clase inválida Ninguna orden válida