## VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE

## RELACIÓN DE EJERCICIOS

1 Considere la siguiente especificación en LDP del procedimiento *ordenar*. Construya el grafo de flujo, calcule la complejidad ciclomática y determine un conjunto básico de caminos independientes.

```
procedimiento: ordenar
 DO WHILE queden registros
   leer registros;
   IF campo 1 del registro = 0 THEN
     procesar registro;
     guardar en buffer;
     incrementar contador;
   ELSE
     IF campo 2 del registro = 0 THEN
      reinicializar contador;
     ELSE
       procesar registro;
      quardar en archivo;
     END IF
   END IF
 END DO
```

2 Utilizando la técnica del camino básico, elabore un conjunto de casos de prueba para el siguiente programa

```
abrir archivos:
leer archivo ventas, al final indicar no más registros
limpiar línea de impresión;
WHILE (haya registros ventas) DO
  total nacional = 0
  total extranjero = 0
  WHILE (haya registro ventas y mismo producto)
    IF (nacional) THEN
     sumar venta nacional a total nacional
   ELSE
     sumar venta extranjero a total extranjero
   END IF
   leer archivo ventas, al final indicar no mas registros
  END WHILE
  escribir línea de listado
  limpiar área de impresión
END WHILE
cerrar archivos
```

**3** Utilizando la técnica del camino básico, elabore un conjunto de casos de prueba para el siguiente programa

```
BEGIN

leer datos de persona
código de barras = código de barras + 1
IF edad >= 30 and persona ="VIP" THEN
IF cargo = "Director" THEN
color de tarjeta = "Dorado"
ELSE
color de tarjeta = "Rojo"
END IF
ELSE
color de tarjeta = "Blanco"
END IF
imprimir código de barras, nombre, color de tarjeta
END
```

**4** Utilizando la técnica del camino básico, elabore un conjunto de casos de prueba para la siguiente especificación de la función *transforma* 

```
integer transforma (int, x)
 integer tmp
   IF x es par THEN
     tmp = x * 2
   ELSE
     IF x es múltiplo de 3 THEN
       tmp = x * 3
     ELSE
       IF x es múltiplo de 5 THEN
        tmp = x * 5
      ELSE
         temp = x
       END IF
   END IF
 END IF
 return tmp
```

5 Utilizando un diagrama de flujo o un LDP, diseñe un programa que satisfaga la siguiente descripción:

"Programa que lee tres valores enteros. Los tres valores se interpretan como representación de los tres lados de un triángulo. El programa imprime un mensaje indicando si el triángulo es escaleno (tres lados desiguales), isósceles (dos lados iguales y uno desigual) o equilátero (tres lados iguales)."

Obtenga un grafo de flujo para el programa y aplique la técnica del camino básico para desarrollar casos de prueba que garanticen que se comprueban todos los enunciados del programa.

**6** Considere los datos contenidos en una aplicación de automatización bancaria. El usuario puede "llamar" al banco usando su ordenador, dar su contraseña de seis dígitos y continuar con una serie de órdenes clave que desencadenarán varias funciones bancarias. El software proporcionado por la aplicación bancaria acepta datos en la siguiente forma:

Código de área — un 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 — valor alfanumérico de seis dígitos
Órdenes — "comprobar", "depositar", "pagar factura", ...

Elabore un conjunto de casos de prueba utilizando la técnica de la partición de equivalencia

7 Elabore un conjunto de casos de prueba utilizando la técnica de la partición de equivalencia para la función *transforma* 

integer transforma (int, x)

poscondiciones

si x es par devuelve el valor x \* 2

en caso contrario, si x es múltiplo de 3 devuelve el valor x \* 3

en caso contrario, si x es múltiplo de 5 devuelve el valor x \* 5

en caso contrario devuelve el valor x