

ILERNA

Online

Videotutoría 5: Grafos y otras pruebas

Módulo 05: Entornos de Desarrollo



RESUMEN VT04

¿Qué habéis aprendido la VT anterior?

Objetivos VT 4

- Grafos
- Otras Pruebas

Complejidad ciclomática

- Métrica del software que nos proporciona una medida cuantitativa de la complejidad lógica de un programa. Nos establecerá el número de casos de prueba que deberán ejecutarse para que las sentencias sean ejecutadas al menos una vez.
- La **complejidad ciclomática** $V(G)$ se podrá calcular de 3 formas:
 - 1) $V(G)$ = Número de regiones del grafo
 - 2) $V(G)$ = Aristas – Nodos + 2
 - 3) $V(G)$ = Nodos predicado + 1

En la pac nos debe salir entre 1 y 10.

Complejidad ciclomática	Evaluación del riesgo
Entre 1 y 10	Programas o métodos sencillos, sin mucho riesgo
Entre 11 y 20	Programas o métodos más complejos, riesgo moderado
Entre 21 y 50	Programas o métodos complejos, alto riesgo
Mayor que 50	Programas o métodos no testeables, muy alto riesgo

Cálculo del camino básico

Esta complejidad ciclomática me va a determinar cuál es la cota inferior del número de pruebas que yo tengo que realizar, como mínimo, todos los caminos existentes al menos una vez por cada uno y una vez por cada arista del grafo.

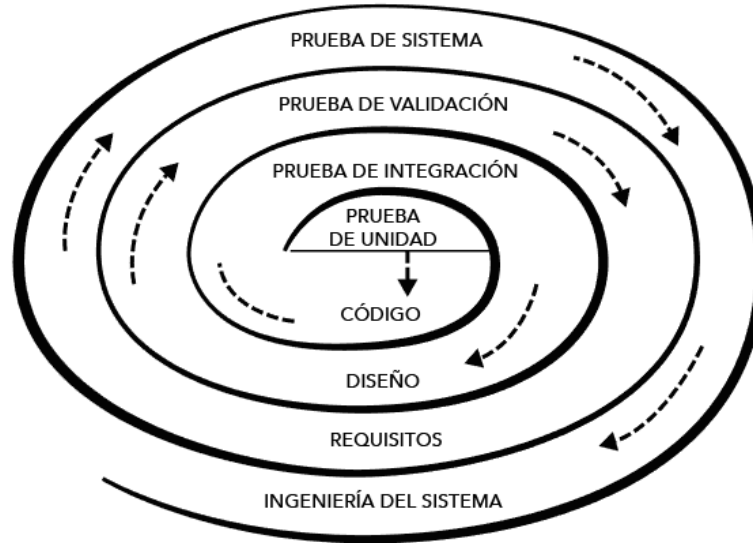
Prueba del Camino Básico (Pruebas de Caja Blanca)

Ejemplo

```
1:  do while queden registros
    leer registros;
2:  if campo 1 del registro = 0
3:    then procesar registro;
        guardar en buffer;
        incrementar contador;
4:    elseif campo 2 del registro = 0
5:      then reinicializar contador;
6:      else procesar registro;
        guardar en archivo;
7a:   endif
    endif
7b:  enddo
8:0: end
```

No tiene nodo propio porque pertenece al nodo del do while. Eso nos da una pista de que se trata de un Do {//CONDICION} WHILE (condición). y no de un DO WHILE {//CONDICION} (condición).

Estrategias de prueba SW



Estrategias de prueba SW

- **Pruebas de unidad:** En esta prueba vamos a comprobar cada módulo para eliminar cualquier tipo de error en la interfaz o en la lógica interna. Utiliza ambas técnicas, tanto la prueba de la caja negra como la de la blanca. P.ej. JUNIT
- **Pruebas de integración:**
 - 1) Integración **no incremental** o big bang. Comprobación de cada módulo por separado y después se prueba de forma conjunta. Se detectan muchos errores y la corrección es difícil.
 - 2) Integración **incremental**. En este caso el programa se va creando y probando en pequeñas secciones por lo que localizar los fallos es más sencillo. En esta integración podemos optar por dos estrategias:
 - a. Ascendente. Se comienza con los módulos más bajos del programa.
 - b. Descendente. Se empieza por el módulo principal descendiendo por la jerarquía de control.

Estrategias de prueba SW

- **Pruebas de Validación:** Conseguiremos la prueba de validación cuando el programa funcione de acuerdo con las expectativas expuestas por el cliente
 - Prueba **Alfa**: realizada por el cliente o usuario en el lugar de desarrollo. Usará el programa bajo la observación del desarrollador que irá registrando los errores.
 - Prueba **Beta**: realizada por los usuarios finales en su lugar de trabajo sin la presencia del desarrollador. En este caso será el usuario el que registre los errores y se los comunique al desarrollador para que realice las modificaciones correspondientes
- Pruebas de **sistema**:
 - Prueba de **recuperación**: se fuerza el fallo del software y que la recuperación se realice correctamente.
 - Prueba de **seguridad**: se comprueba que el sistema esté protegido frente a acciones ilegales.
 - Prueba de resistencia (**Stress**). se realizan acciones que requieran una gran cantidad de recursos.

Clases de equivalencia

- Prueba de caja negra que divide los valores de los campos de entrada de un programa en clases de equivalencia.
- Se podrá definir dos tipos de clases de equivalencia:
 - **Clase válidas:** valores de entrada válidos.
 - **Clase no válidas:** valores de entrada no válidos.

Clases de equivalencia: Ejemplo

Considérese una aplicación bancaria, donde el usuario puede conectarse al banco por Internet y realizar una serie de operaciones bancarias. Una vez accedido al banco con las consiguientes medidas de seguridad (clave de acceso y demás), la información de entrada del procedimiento que gestiona las operaciones concretas a realizar por el usuario requiere la siguiente entrada:

- Código del banco. En blanco o número de tres dígitos. En este último caso, el primero de los tiene que ser mayor que 1.
- Código de sucursal. Un número de cuatro dígitos. El primero de ellos mayor de 0.
- Número de cuenta. Número de cinco dígitos.
- Clave personal. Valor alfanumérico de cinco posiciones.
- Orden. Este valor se introducirá según la orden que se desee realizar. Puede estar en blanco o ser una de las dos cadenas siguientes: o “Talonario” o “Movimientos”

Clases de equivalencia: Ejemplo

Condición de Entrada	Tipo	Clase Equivalencia Válida	Clase Equivalencia No Válida
Código banco	Lógica (puede estar o no) Si está es Rango	1: En blanco 2: $100 \leq \text{Código banco} \leq 999$	3: Un valor no numérico 4: Código banco < 100 5: Código banco > 999
Código sucursal	Rango	6: $1000 \leq \text{Código sucursal} \leq 9999$	7: Código sucursal < 1000 8: Código sucursal ≥ 9999
Nº Cuenta	Valor	9: Cualquier número de cinco dígitos	10: Número de menos de cinco dígitos 11: Número de más de cinco dígitos
Clave	Valor	12: Cualquier cadena de caracteres alfanuméricos de 5 posiciones	13: Cadena de menos de cinco posiciones 14: Cadena de más de cinco posiciones
Orden	Conjunto, con comportamiento distinto	15: "" 16: "Talónario" 17: "Movimientos"	18: Cadena distinto de blanco y de las válidas 19: "Talónarios" 20: "Movimiento"

Análisis de valores límite

Prueba de caja negra donde se basa en la hipótesis de que suelen ocurrir más errores en los valores extremos de los campos de entrada.

Complementa a la técnica anterior. Además, no solo estará centrado en las condiciones de entrada, sino que definen también las clases de salida.

Ejemplo: para un rango de valores enteros que estén comprendidos entre 5 y 15, tenemos que escribir casos de prueba para 5, 15, 4 y 16.

