ALGEMESÍ
INSTITUT DE SECUNDÀRIA
St. VICENT FERRER

UD5: Diseño y realización de pruebas

Módulo: Entornos de desarrollo

ALGEMESÍ
INSTITUT DE SECUNDÀRIA
St. VICENT FERRER

INTRODUCCIÓN

•La prueba del software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

•La prueba de software es un elemento que a menudo se le conoce como verificación y validación (V & V).

•Bohem lo define:

– **Verificación:** ¿Estamos construyendo el software correctamente?

– **Validación:** ¿Estamos construyendo el producto correcto?

OBJETIVOS DE LA PRUEBA

•La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.

•Buen caso de prueba: aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

•Éxito de una prueba: si descubre un error no detectado hasta entonces.

“La prueba no puede asegurar la ausencia de defectos,
sólo puede demostrar que existen defectos en el software”.

PRINCIPIOS DE LA PRUEBA

•Las pruebas deberán planificarse mucho antes de que empiecen para garantizar la calidad de acuerdo a lo establecido en el ciclo de vida.

•Las pruebas deberán empezar por lo pequeño y progresar hacia lo grande.

•Para ser más efectivas, las pruebas deberán ser conducidas por un equipo independiente.

ALGEMESÍ
INSTITUT DE SECUNDÀRIA
St. VICENT FERRER

TIPOS DE PRUEBAS

- En los módulos, **Pruebas de unidad.**
- En la unión de los módulos, **Pruebas de integración.**
- Cuando tenemos todos unidos, **Prueba de validación.**
- Cuando el sistema está funcionando, **Prueba de sistema.**

ALGEMESÍ
INSTITUT DE SECUNDÀRIA
St. VICENT FERRER

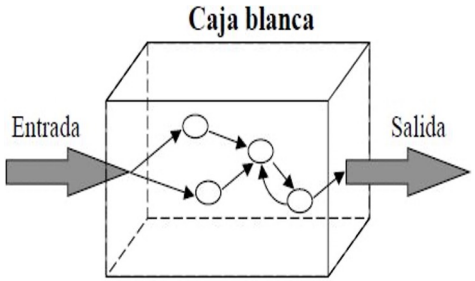
PRUEBAS DE UNIDAD

•La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software: **“el módulo”**.

•Usando la descripción del diseño procedimental como guía, se prueban los caminos de control importantes, con el fin de descubrir errores dentro del límite del módulo.

•La prueba de unidad está orientada a caja blanca y este paso se puede llevar a cabo en paralelo para múltiples módulos.

PRUEBAS DE UNIDAD:
PRUEBAS DE CAJA BLANCA



•La prueba de caja blanca denominada a veces prueba de *caja de cristal* es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.

•Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado.

•Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

PRUEBA DE CAJA BLANCA:
PRUEBA DEL CAMINO BÁSICO

•La prueba del camino básico es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe.

•Esta técnica permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico (diseño de casos de prueba) de caminos de ejecución.

•Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

PRUEBAS DE CAJA BLANCA: PRUEBA DEL CAMINO BÁSICO

- **Notación del grafo de flujo:**

–Cualquier representación del diseño procedimental se puede traducir a un grafo de flujo o grafo del programa.

–Cada círculo denominado **nodo del grafo** de flujo, representa una o más sentencias procedimentales.

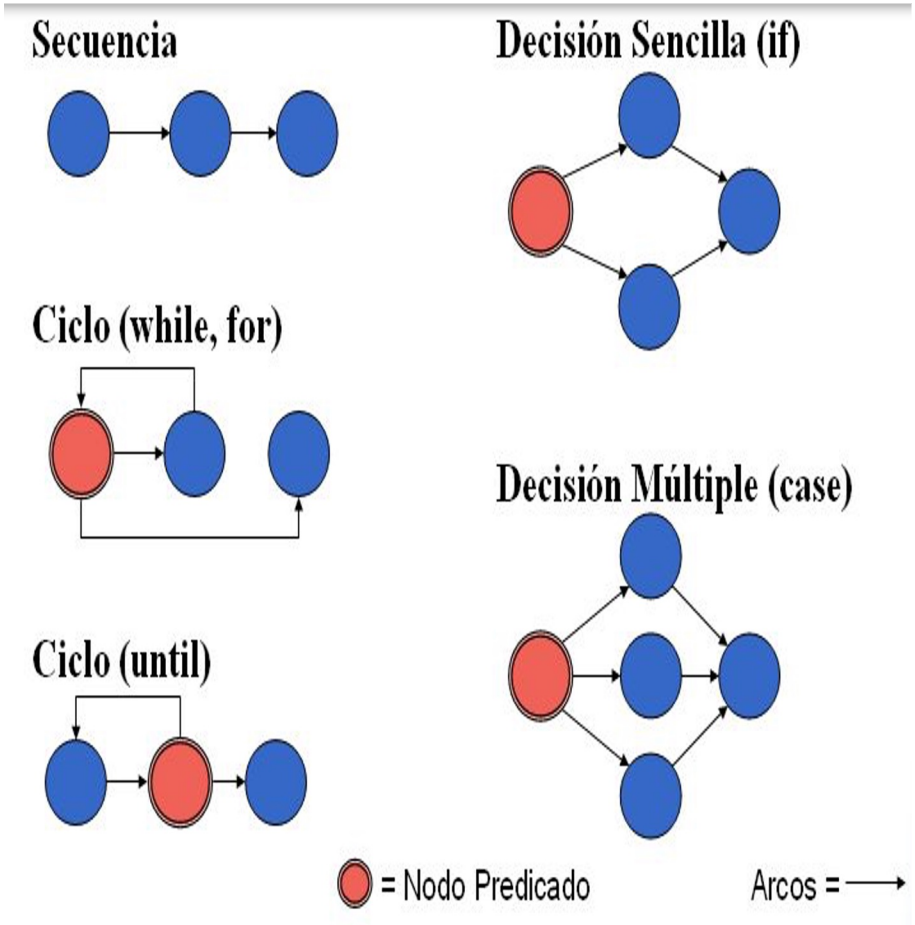
–Un solo nodo puede corresponder a una secuencia de cuadros de proceso y a un rombo de decisión.

–Las flechas del grafo denominadas **aristas o enlaces**, representan flujo de control. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental.

–Las áreas delimitadas por aristas y nodos de denominan **regiones**. Cuando contabilizamos las regiones incluimos el área exterior del grafo, contando como otra región más.

– El nodo que contiene una condición se llama nodo predicado y se caracteriza porque de él salen dos o más aristas.

PRUEBAS DE CAJA BLANCA: PRUEBA DEL CAMINO BÁSICO



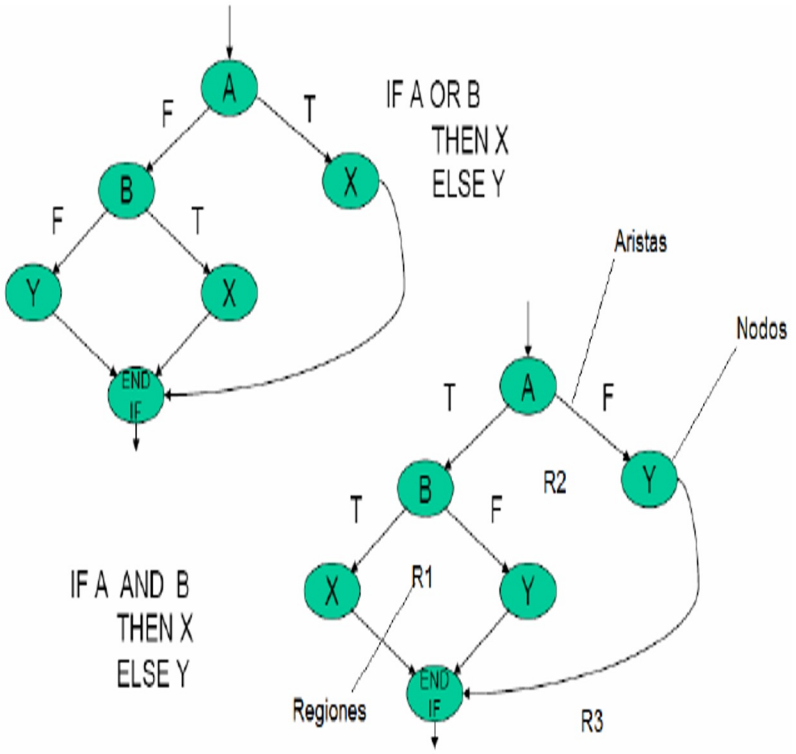


Figura 5: Ejemplos de Grafos de Flujo

PRUEBA DEL CAMINO BÁSICO: COMPLEJIDAD CICLOMÁTICA (VG)

•Define el número de caminos independientes del conjunto básico de un programa y nos da un límite inferior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

•Un **camino independiente** es cualquier camino del programa que introduce por lo menos un nuevo conjunto de sentencias de procesamiento o una nueva condición. En términos del grafo de flujo, un camino independiente se debe mover por lo menos por una arista que no haya sido recorrida anteriormente a la definición de un camino.

•La **complejidad ciclomática V(G)** se puede calcular de tres formas:

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.

2. Aristas - Nodos + 2, es decir **V (G) = A – N + 2**.

3. Nodos Predicado + 1 (un nodo predicado es el que representa una condicional if o case, es decir, que de él salen varios caminos).

•Por tanto se deben preparar los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

EJEMPLO: PRUEBA DEL CAMINO BÁSICO

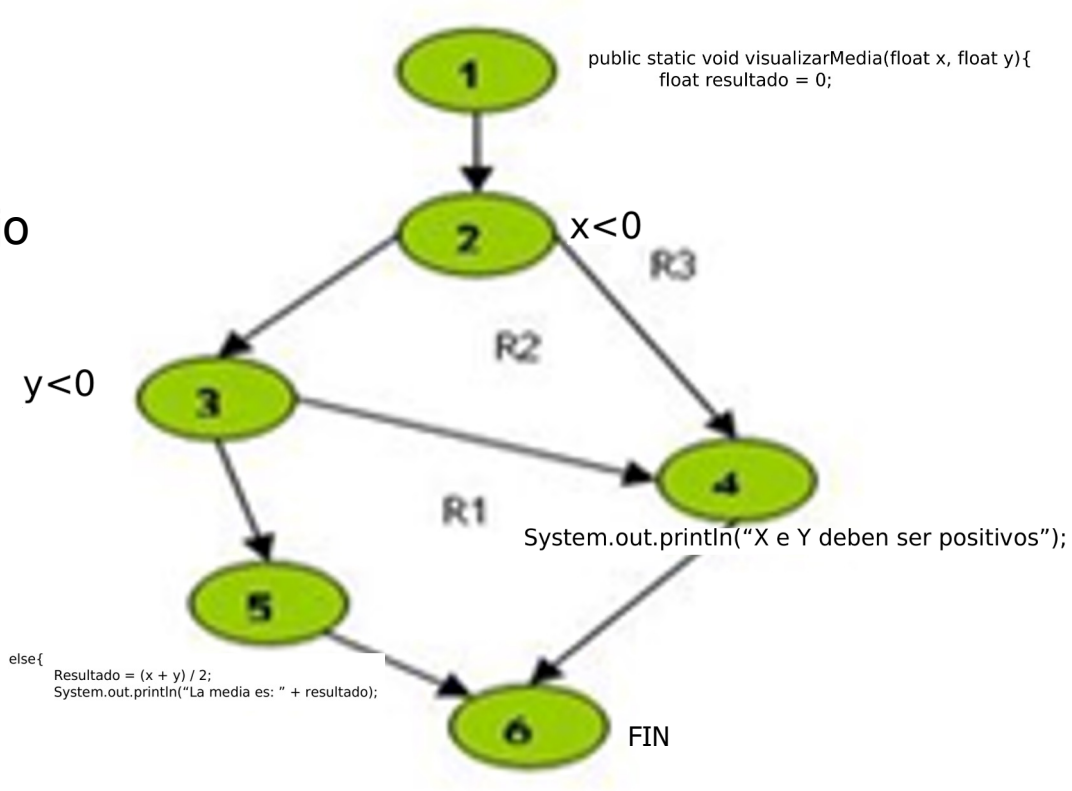
Diseñar el conjunto de casos de prueba mediante el método de la complejidad ciclomática para el siguiente código:

```
public static void visualizarMedia(float x, float y){
        float resultado = 0;
                if ((x<0) || (y<0)){
                System.out.println("X e Y deben ser positivos");
                }
                else{
                Resultado = (x + y) / 2;
                System.out.println("La media es: " + resultado);
                }
}
```

EJEMPLO: PRUEBA DEL CAMINO BÁSICO

Solución:

1.- Conversión al grafo de flujo

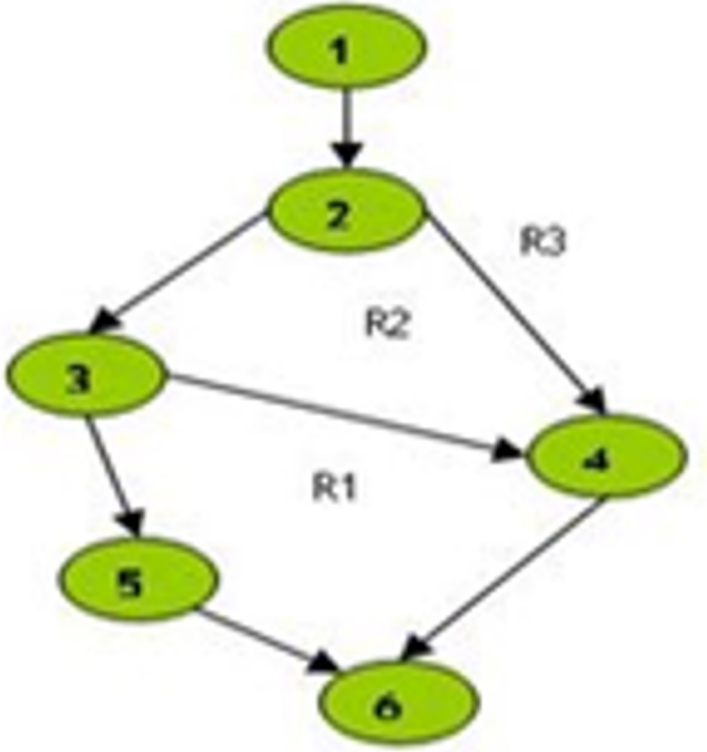


Diapositiva con código para facilitar el entendimiento del GF.

EJEMPLO: PRUEBA DEL CAMINO BÁSICO

Solución:

1.- Conversión al grafo de flujo



EJEMPLO: PRUEBA DEL CAMINO BÁSICO

Solución:

2.- Cálculo de la Complejidad Ciclomática

V(G) = 3 regiones = 3

V(G) = 7A – 6N + 2 = 3

V(G) = 2NP + 1 = 3

EJEMPLO: PRUEBA DEL CAMINO BÁSICO

Solución:

3.- Conjunto de caminos básicos:

Habrá tantos caminos básicos como grados de complejidad posee el código

Camino 1: 1-2-3-5-6	Escoger algún X e Y tal que NO se cumpla la condición x<0 || y<0 x=10, y=20 visualizarMedia(10, 20)	Visualiza: La media es 15
Camino 2: 1-2-3-4-6	Escoger algún X tal que NO se cumpla la condición x<0 y escoger algún Y que SÍ cumpla la condición y<0 x=10, y=-20 visualizarMedia(10, -20)	Visualiza: X e Y deben ser positivos
Camino 3: 1-2-4-6	Escoger algún X tal que SÍ se cumpla la condición x<0 (Y puede ser cualquier valor) x=-10, y=20 visualizarMedia(-10, 20)	Visualiza: X e Y deben ser positivos