

---

## EXEMPLO

### PROBAS UNITARIAS

### CREACIÓN DE CASOS DE PROBA

---

Neste exercicio, analizaremos un pequeno programa en Java para calcular a complexidade ciclomática de McCabe, definir as clases de equivalencia, realizar un análise de valores límite e establecer unha conxectura de erros, ademais de xerar os casos de proba correspondentes.

```
public class Comparador {  
    public String compararNumeros(int a, int b) {  
        if (a > b) {  
            return "O primeiro número é maior";  
        } else if (a < b) {  
            return "O segundo número é maior";  
        } else {  
            return "Os dous números son iguais";  
        }  
    }  
}
```

## 1 Complexidade Ciclomática de McCabe

Neste código hai 2 decisións (if-else), polo que hai 3 camiños de execución. Podes crear o grafo e resolver esta solución.

Definición dos camiños segundo McCabe

- **Camiño 1:**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$
- **Camiño 2:**  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$
- **Camiño 3:**  $1 \rightarrow 2 \rightarrow 5$
- 

## 2 Clases de Equivalencia

Entrada	Clases válidas	Clases inválidas
int a	(1) Calquera número enteiro	(2) Cadea de caracteres
int b	(3) Calquera número enteiro	(4) Cadea de caracteres

## 3 Análise de Valor Límite (AVL)

AVL	a	b	Resultado esperado
AVL1	-2147483648	0	"O segundo número é maior"

AVL2	2147483647	0	"O primeiro número é maior"
AVL3	100	100	"Os dous números son iguais"

## 4 Conxetura de Erros (CE)

CE	Descrición
CE1	Entrada dunha cadea de caracteres en a ou b
CE2	Entrada dun valor nulo en a ou b

## 5 Creación dos Casos de Proba

Para comezar a táboa de casos de proba, o primeiro paso **sempre debe ser a análise dos camiños de execución**. Isto débese a que cada camiño representa unha posible ruta de execución dentro do código, e o noso obxectivo é asegurarnos de que todos os camiños están cubertos con polo menos un caso de proba.

Unha vez temos os camiños claros, **podemos engadir as clases de equivalencia para diversificar os valores de entrada**. Isto significa que, para cada camiño, podemos incluír diferentes tipos de valores dentro das súas clases de equivalencia correspondentes.

### 5.1 Orde correcta para crear a táboa de casos de proba:

1. **Determinar os camiños de execución.** Isto permítenos definir os casos mínimos necesarios para cubrir todas as posibilidades.
2. **Asociar os valores das clases de equivalencia.** Dentro de cada camiño, podemos escoller diferentes valores representativos das clases de equivalencia.
3. **Engadir valores límite.** Para verificar o comportamento do código nos extremos posibles dos datos de entrada.
4. **Engadir casos de erro (conxetura de erros).** Estes casos axúdannos a asegurar que o programa manexa correctamente entradas inesperadas.

ID	a	b	Clases / AVL / CE	Camiño	Resultado esperado
C1	5	3	(1)(3)	1-2-3-5	"O primeiro número é maior"
C2	2	6	(1)(3)	1-2-4-5	"O segundo número é maior"
C3	10	10	(1)(3)	1-2-5	"Os dous números son iguais"
C4	-2147483648	0	AVL1	1-2-4-5	"O segundo número é maior"
C5	2147483647	0	AVL2	1-2-3-5	"O primeiro número é maior"
C6	"cadea"	5	(2)(3) CE1	-	Erro de tipo
C7	5	"cadea"	(1)(4) CE1	-	Erro de tipo
C8	null	3	CE2	-	Excepción
C9	3	null	CE2	-	Excepción

