¡Bienvenidos!

Pruebas de Software

UNIVERSIDAD DE
SAN BUENAVENTURA
CALI

GreenSQA
Software Quality Assurance

Técnicas de Evaluación de Calidad de Software

PROGRAMACIÓN TEMAS

TEMAS

- Pruebas de caja blanca
- Importancia de las pruebas de caja blanca
- Tipos de pruebas de caja blanca
- Definición y ejemplo de cada tipo

ACTIVIDADES

- Revisión de Preguntas de la Guía de Estudio
- Evaluación de los temas estudiados mediante preguntas de selección múltiple.
- Desarrollo taller practico en clase

PREGUNTAS PARA RESOLVER

¿Qué son las pruebas de caja blanca?

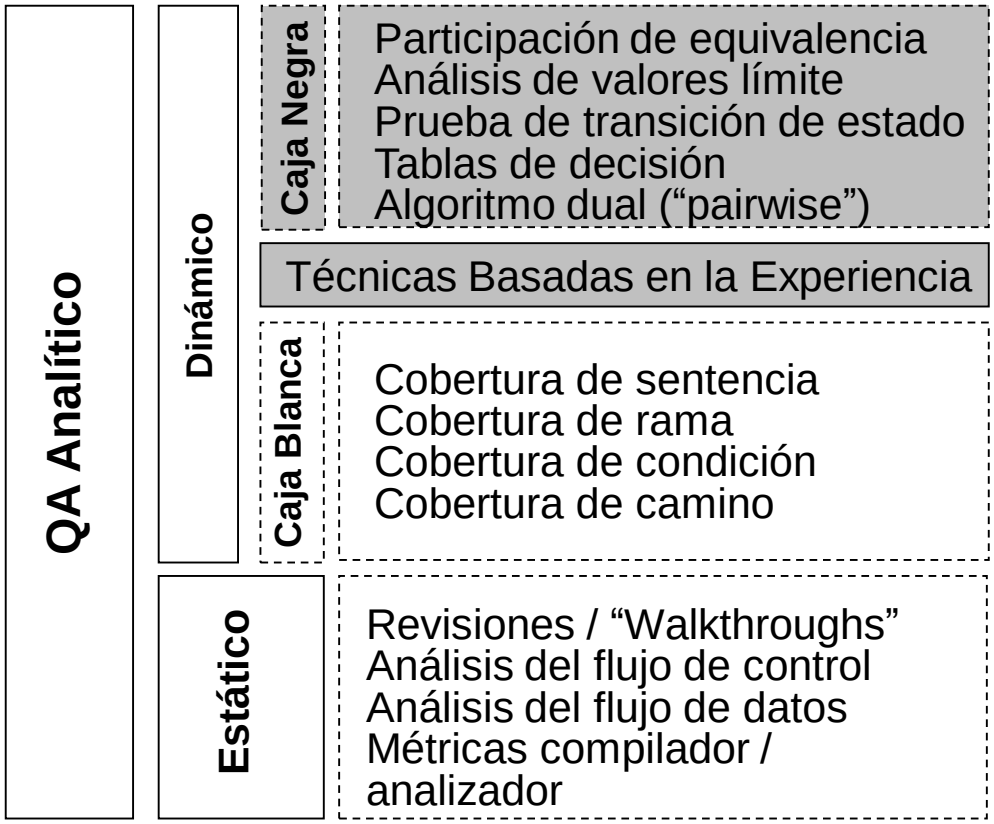
¿Por qué son importantes las pruebas de caja blanca?

¿Cuáles son los tipos de pruebas de caja blanca?

¿Cuáles es la diferencia entre las pruebas de caja blanca y las pruebas de caja negra?

Pruebas Dinámicas

Las pruebas dinámicas se dividen en dos categorías o grupos: **caja blanca** y caja negra.



La agrupación se realiza en función del carácter básico del método utilizado para obtener los casos de prueba .

Cada grupo tiene sus propias técnicas para diseñar casos de prueba.

Pruebas de caja blanca y su importancia

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.

2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.

3. Ejecuten todos los bucles en sus límites operacionales.

4. Ejerciten las estructuras internas de datos para asegurar su validez.

Pruebas de caja blanca y su importancia

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Pruebas de caja blanca y su importancia

La teoría establece que todas las partes de un programa deberían ser ejecutadas por lo menos una vez.

El grado de cobertura de un programa se mide con el uso de herramientas (por ejemplo, analizadores de cobertura):

- La instrumentación del código se lleva a cabo con el objeto de contar la ejecución de caminos, es decir se insertan contadores en el código del programa del objeto de prueba.

- Estos contadores son inicializados en cero, cada ejecución del camino específico incrementará el contador correspondientemente.

- Los contadores que mantienen el valor cero tras las pruebas indican las partes del programa que aún no han sido ejecutadas.

Diferencia entre pruebas de caja negra y caja blanca

CAJA NEGRA:

Las pruebas basadas en la especificación, también conocidas como las pruebas de caja negra o como las pruebas de comportamiento, son las que son diseñadas desde la especificación del sistema. En las pruebas de comportamiento, nosotros ignoramos el funcionamiento interno del sistema y nos enfocamos acerca de cómo este se supone que se debe comportar.

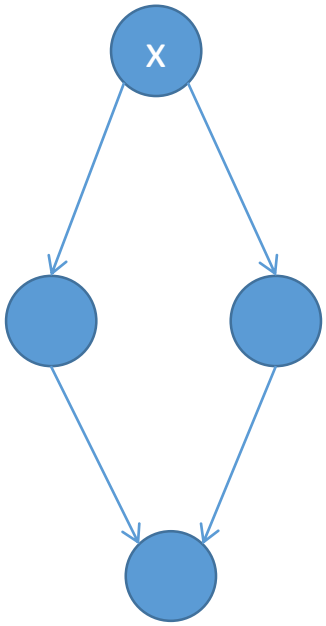
CAJA BLANCA:

Las pruebas basadas en la estructura, las cuáles son llamadas también pruebas estructurales o de caja blanca, se basan en la estructura interna del sistema o en un componente del sistema; p.ej., nosotros examinamos cómo el sistema funciona y qué hace. Específicamente, seleccionamos las entradas, las precondiciones, las condiciones, los eventos u otros estímulos basados en una parte de la estructura del sistema que estos ejercerán.

Notación de grafos

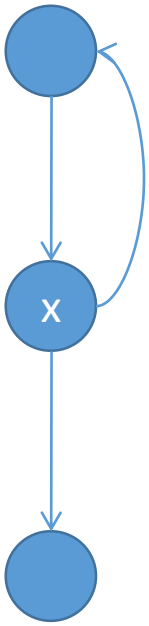
Secuencia

If



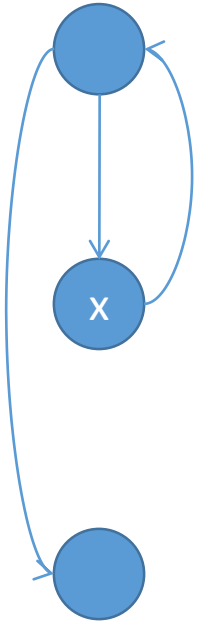
Si x entonces …

Repeat



Hacer … hasta x

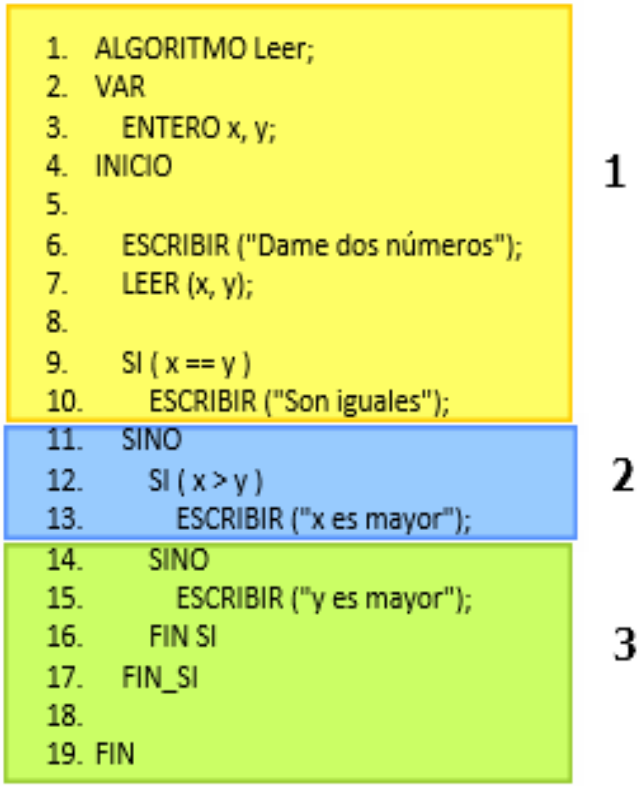
While



Mientras x hacer …

PSEUDOCÓDIGO

```
1.  ALGORITMO Leer;
2.  VAR
3.     ENTERO x, y;
4.  INICIO
5.
6.     ESCRIBIR ("Dame dos números");
7.     LEER (x, y);
8.
9.     SI ( x == y )
10.       ESCRIBIR ("Son iguales");
11.    SINO
12.       SI ( x > y )
13.          ESCRIBIR ("x es mayor");
14.       SINO
15.          ESCRIBIR ("y es mayor");
16.       FIN SI
17.    FIN_SI
18.
19. FIN
```

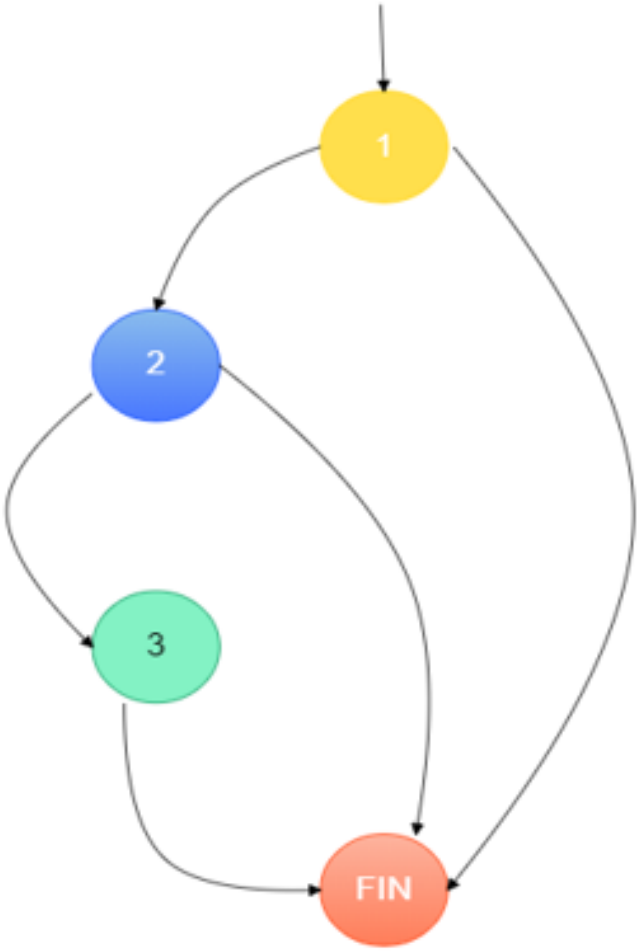


```
1.  ALGORITMO Leer;
2.  VAR
3.     ENTERO x, y;
4.  INICIO
5.
6.     ESCRIBIR ("Dame dos números");
7.     LEER (x, y);
8.
9.     SI ( x == y )
10.       ESCRIBIR ("Son iguales");
11.    SINO
12.       SI ( x > y )
13.          ESCRIBIR ("x es mayor");
14.       SINO
15.          ESCRIBIR ("y es mayor");
16.       FIN SI
17.    FIN_SI
18.
19. FIN
```

1

2

3



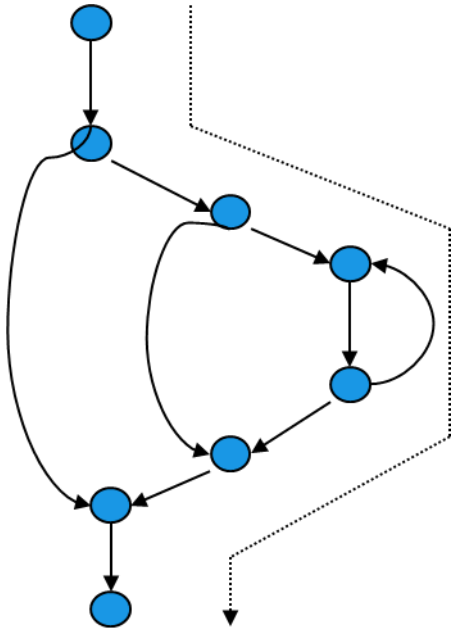
Tipos de pruebas de caja blanca

Cobertura de sentencia

Cobertura de sentencia (“statement coverage”), el foco de atención es la sentencia del código de un programa. El objetivo de la prueba (Criterio de salida) es lograr la cobertura de un porcentaje especifico de todas las sentencias, denominado cobertura de sentencia.

Se evalúa el siguiente segmento de código de un programa que está representado por el diagrama de flujo de control (imagen a la derecha).

```
If (i>0) {
                j=f (i);
if (j>10){
for (k=i; k >10; k -- ){
      ……
                }
     }
}
```



- Todas las sentencias de este programa pueden ser alcanzadas haciendo uso de este camino a la derecha.
- Un solo caso de prueba será suficiente para alcanzar el 100% de cobertura de sentencia.
- Si hay código muerto en el programa, no se podrá lograr una cobertura del 100%

Cobertura de decisión

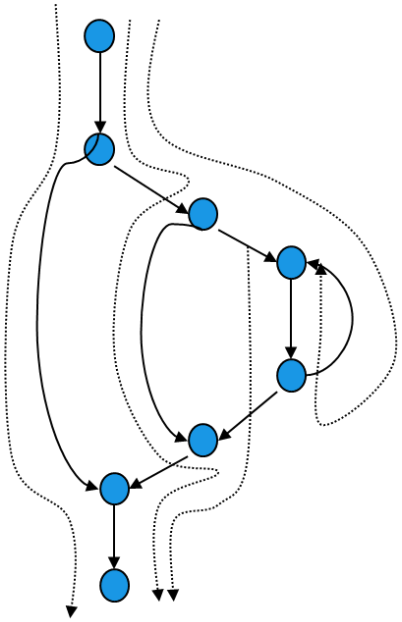
Cobertura de decisión = cobertura de rama (“decisión coverage = branch coverage”)

- En lugar de las sentencias, la cobertura de decisión se centra en el flujo de control en un segmento del programa (No los nodos sino las aristas del diagrama del flujo de control)

- El propósito de esta prueba (Criterio de salida) es lograr la cobertura de un porcentaje especifico de todas las decisiones, denominado cobertura de decisión)

Cobertura de decisión

- El diagrama de flujo de control representa el segmento del programa objeto de la evaluación
- Tres caminos diferentes conducen a través del diagrama de este segmento de programa:
  - La primera sentencia “if” conduce a dos direcciones diferentes.
  - El camino de la primera sentencia “if” se divide nuevamente en dos caminos diferentes, uno de los cuales contiene un bucle.
  - Solamente se puede alcanzar las aristas a través de una combinación de los tres caminos posibles.
  - Son necesarios tres casos de prueba para alcanzar una cobertura de decisión del 100%.
  - Una cobertura de decisión del 100% siempre incluye una cobertura de sentencia del 100%.



- Son necesarios tres casos de prueba para alcanzar una cobertura de decisión del 100%.
- Una cobertura de decisión del 100% siempre incluye una cobertura de sentencia del 100%.

Cobertura de condición

- Hay tres tipos de cobertura de condición:
  - Cobertura de condición simple (“simple condition coverage”).
  - Cobertura de condición múltiple (“multiple condition coverage”).
  - Mínima cobertura de condición múltiple (“minimum multiple condition coverage”).

Cobertura de Condición

- Se tiene en cuenta la complejidad de una condición que esté constituida por múltiples condiciones atómicas:
  - Una condición atómica no puede ser dividida en sentencias condicionales mas pequeñas.

- Éste método tiene por objetivo detectar defectos que resulten de la implementación de condiciones múltiples (condiciones combinadas).
  - Las condiciones múltiples están constituidas por condiciones atómicas, que se combinan con el uso de operadores lógicos como:
    - Ejemplo: ((a>2) OR (b<6))
  - Las condiciones atómicas no tienen operadores lógico, sólo contienen operadores relacionales y el operador NOT (=, >,<, etc.)

Cobertura de Condición Simple

Cada sub.-condición atómica de una sentencia condicional combinada tiene que tomar, al menos una vez, los valores lógicos verdadero (“true”) así como falso (“false”).

Considerar la siguiente condición. a>2 OR b <6 Los casos de prueba para la cobertura de condición simple podrán ser por ejemplo		
a=6 (true)	b=9 (false)	a>2 OR b<6 (true)
a=1 (false)	b=2 (true)	a>2 OR b<6 (true)

- Este ejemplo se utiliza para explicar la cobertura de condición utilizado una expresión como una condición múltiple.
- Con sólo dos casos de prueba se puede lograr una cobertura de condición simple.
- Cada sub-condición ha tomado los valores “true” y “false”.
- Sin embargo, el resultado combinado es verdadero (“true”) en ambos casos.
  - True OR false = true
  - False OR true = true

Cobertura de Condición Múltiple

Deben formar parte de las pruebas todas las combinaciones que puedan ser creadas utilizando permutaciones de las sub condiciones atómicas.

Considerar la siguiente condición.

a>2 OR b <6

Los casos de prueba para la cobertura de condición múltiple podrían ser por ejemplo

a=6 (true)	b=9 (false)	a>2 OR b<6 (true)
a=6 (true)	b=2 (true)	a>2 OR b<6 (true)
a=1 (false)	b=2 (true)	a>2 OR b<6 (true)
a=1 (false)	b=9 (false)	a>2 OR b<6 (false)

- Este ejemplo se utiliza para explicar la cobertura de condición utilizado una expresión como una condición múltiple.
- Con cuatro casos de prueba se puede lograr una cobertura de condición múltiple
  - Se han creado las combinaciones de los valores verdadero (“true”) y falso (“false”).
  - Se han logrado todo los posibles resultados de la condición múltiple.
- El número de casos de prueba se incrementa de forma exponencial:
  - n= número de condiciones atómicas.
  - 2 ^n= número de casos de prueba.

Mínima Cobertura de Condición Múltiple

Todas las combinaciones que puedan ser creadas utilizando los resultados lógicos de cada sub-condición deben ser parte de las pruebas, sólo si el cambio del resultado de una sub-condición cambia el resultado de la condición combinada.

Considerar la siguiente condición.

a>2 OR b <6

Los casos de prueba para la cobertura de condición múltiple podrían ser por ejemplo

a=6 (true)	b=9 (false)	a>2 OR b<6 (true)
a=6 (true)	b=2 (true)	a>2 OR b<6 (true)
a=1 (false)	b=2 (true)	a>2 OR b<6 (true)
a=1 (false)	b=9 (false)	a>2 OR b<6 (false)

- Este ejemplo se utiliza para explicar la cobertura de condición utilizando una expresión con una condición múltiple.
- Los cambios de una sub-condición cambian el resultado global para tres de cuatro casos de prueba.
  - Sólo para el caso nº2 (true OR true =true) el cambio en la sub-condición no resultará en un cambio en la condición global.¡este caso de prueba puede ser omitido!.
- El número de casos de prueba se puede reducir a un valor entre n+1 y 2n

Cobertura de Condición

Conclusiones generales:

- La **mínima cobertura de condición múltiple** es incluso mejor, debido a:
  - Reduce el número de casos de prueba de (n+1) a (2n).
  - Las coberturas de sentencia y decisión también son cubiertas.
  - Tiene en cuenta la complejidad de las sentencias de decisión.

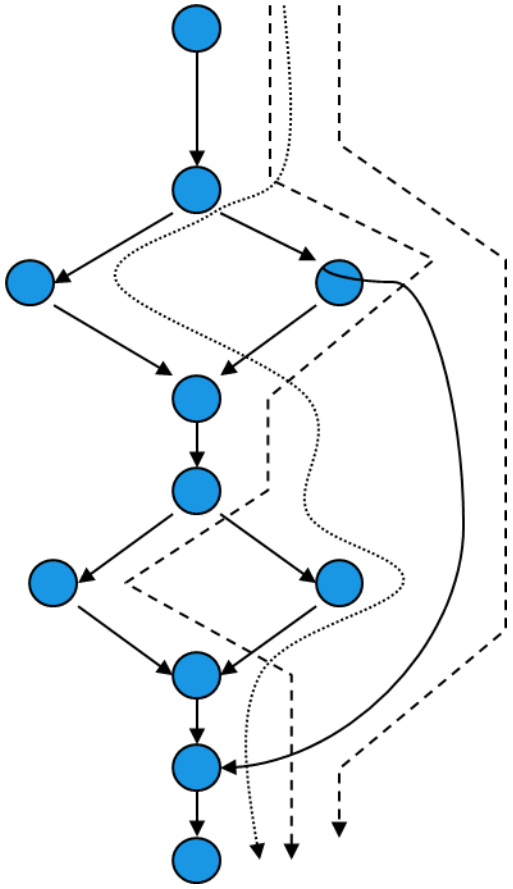
- **Todas las decisiones complejas deben ser probadas, la mínima cobertura de condición múltiple es adecuada para lograr este objetivo.**

Cobertura de Camino

- El foco del análisis de cobertura es el diagrama del flujo de control.
- Las sentencias son nodos.
- El flujo de control esta representado por las aristas.
- Cada camino es una vía única desde el inicio al fin del diagrama del flujo de control.
- El objetivo de esta prueba (Criterio de salida) es alcanzar un porcentaje definido de cobertura de camino.

Cobertura de Camino

- El diagrama de flujo de control de la imagen a la derecha, representa el segmento de programa a ser evaluado. Contiene dos sentencias “if”.
- Tres caminos diferentes conducen a través del diagrama de este segmento de programa logran una cobertura de decisión completa.
- Sin embargo, pueden ser ejecutados cinco posibles caminos distintos.
  - Son necesarios cinco casos de pruebas para lograr un 100% de cobertura de camino.



Cobertura de Camino

Conclusiones generales

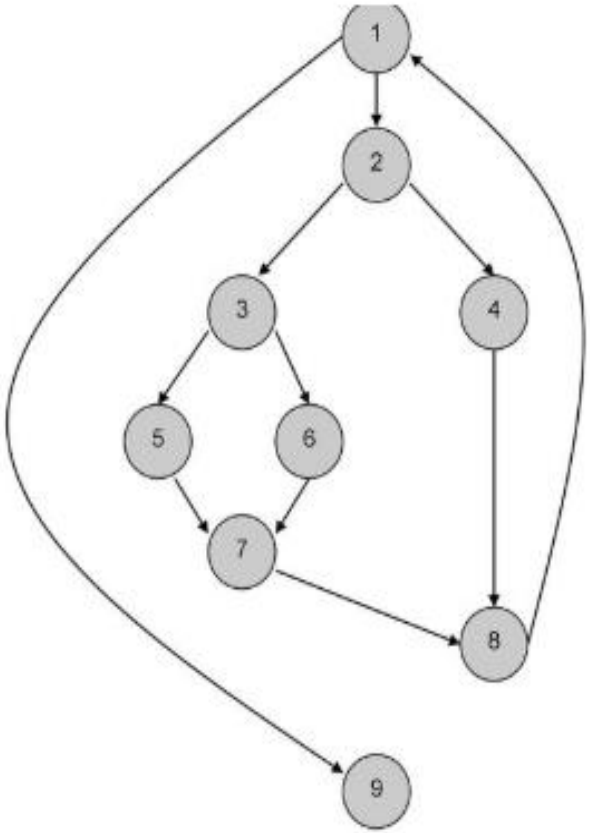
- El 100% de cobertura de camino sólo se puede lograr en programas muy simples.
  - Un solo bucle puede conducir a una explosión de casos de prueba dado que, todo número posible de ejecuciones de un bucle constituye un nuevo caso de prueba.
  - Teóricamente es posible un número indefinido de caminos.

- Cobertura de camino es más exhaustiva que la cobertura de sentencia y de decisión pus cada posible camino a través del programa es ejecutado.

- **100% de cobertura de camino incluye 100% de cobertura de decisión, que a su vez incluye 100% de cobertura de sentencia.**

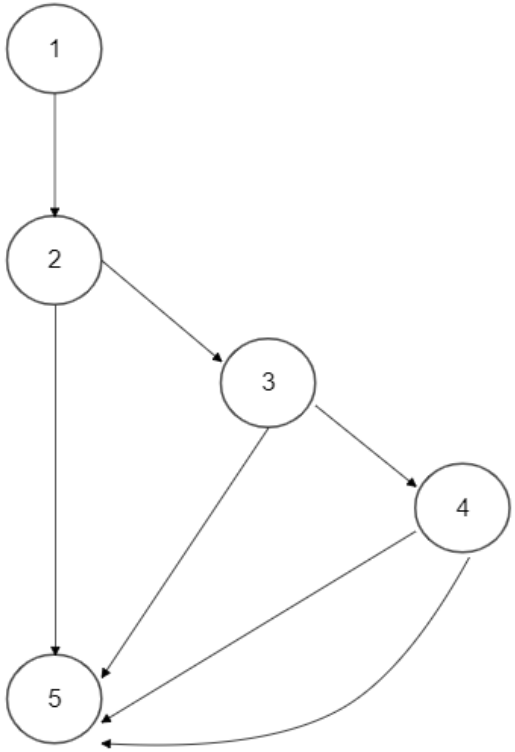
Ejercicio en clase

¿Cuántos casos de pruebas son necesarios para cubrir el 100% de cobertura de sentencia?, justificar su respuesta dibujando cada camino



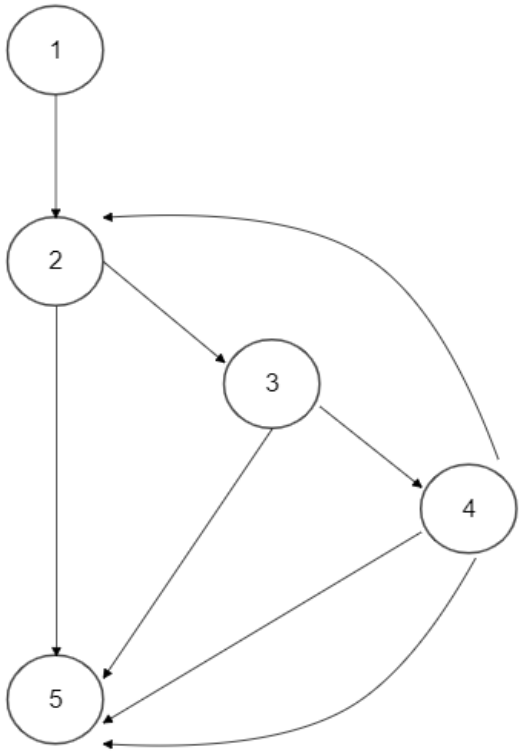
Ejercicio en clase

¿Cuántos casos de pruebas son necesarios para cubrir el 100% de cobertura de decisión? , justificar su respuesta dibujando cada camino



Ejercicio en clase

¿Cuántos casos de pruebas son necesarios para cubrir el 100% de cobertura de camino? , justificar su respuesta dibujando cada camino



¡Gracias!

UNIVERSIDAD DE
SAN BUENAVENTURA
CALI

GreenSQA
Software Quality Assurance