HITO 1 DEL 3º TRIMESTRE DE ENTORNOS DE DESARROLLO

Alejandro Cortés Díaz

DISEÑO DE APLICACIONES MULTIPLATAFORMA – ENTORNOS DE DESARROLLO 12/05/2025

Índice

¿Qué son las pruebas estructurales?

Las pruebas estructurales, también conocidas como pruebas de caja blanca, se centran en la estructura interna del código fuente. El objetivo principal es verificar que todos los caminos posibles dentro del código funcionen correctamente, es decir, que la lógica del programa sea correcta**.**

Objetivos principales:

- Verificar todos los flujos de control posibles del código.
- Garantizar la cobertura de instrucciones, decisiones y condiciones.
- Identificar errores en condiciones lógicas, bucles, llamadas a funciones, etc.

Técnicas comunes:

- Cobertura de código: mide qué porcentaje del código ha sido ejecutado durante las pruebas.
- Pruebas de flujo de control: analizan el flujo de ejecución (if, while, for, etc.).
- Pruebas de caminos lógicos: prueban todos los caminos posibles del programa.

¿Qué son las pruebas funcionales?

Las pruebas funcionales**,** también llamadas pruebas de caja negra, evalúan si el software cumple con los requisitos funcionales especificados, sin tener en cuenta cómo está implementado internamente.

Objetivos principales:

- Comprobar que las funcionalidades funcionan según lo esperado.
- Validar entradas, salidas y comportamiento del sistema.
- Identificar errores en la lógica de negocio desde el punto de vista del usuario.

Técnicas comunes:

- Pruebas de caja negra**:** validan la funcionalidad sin mirar el código.
- Pruebas de aceptación del usuario**:** verifican que el software satisface las necesidades del cliente.
- Pruebas de regresión: aseguran que nuevas funcionalidades no rompen las existentes.

Comparación entre pruebas estructurales y funcionales

- Mientras que el enfoque de la caja blanca es la lógica interna del código, la caja negra se centra en las funcionalidades externas del sistema.

- La caja blanca tiene acceso al código fuente mientras que la negra no.

- La caja blanca está basada en la implementación mientras que la negra en los requisitos de usuario.

- La caja blanca hace uso de detalles referentes al código fuente mientras que la caja negra no tiene mucho conocimiento sobre el diseño interno del elemento.

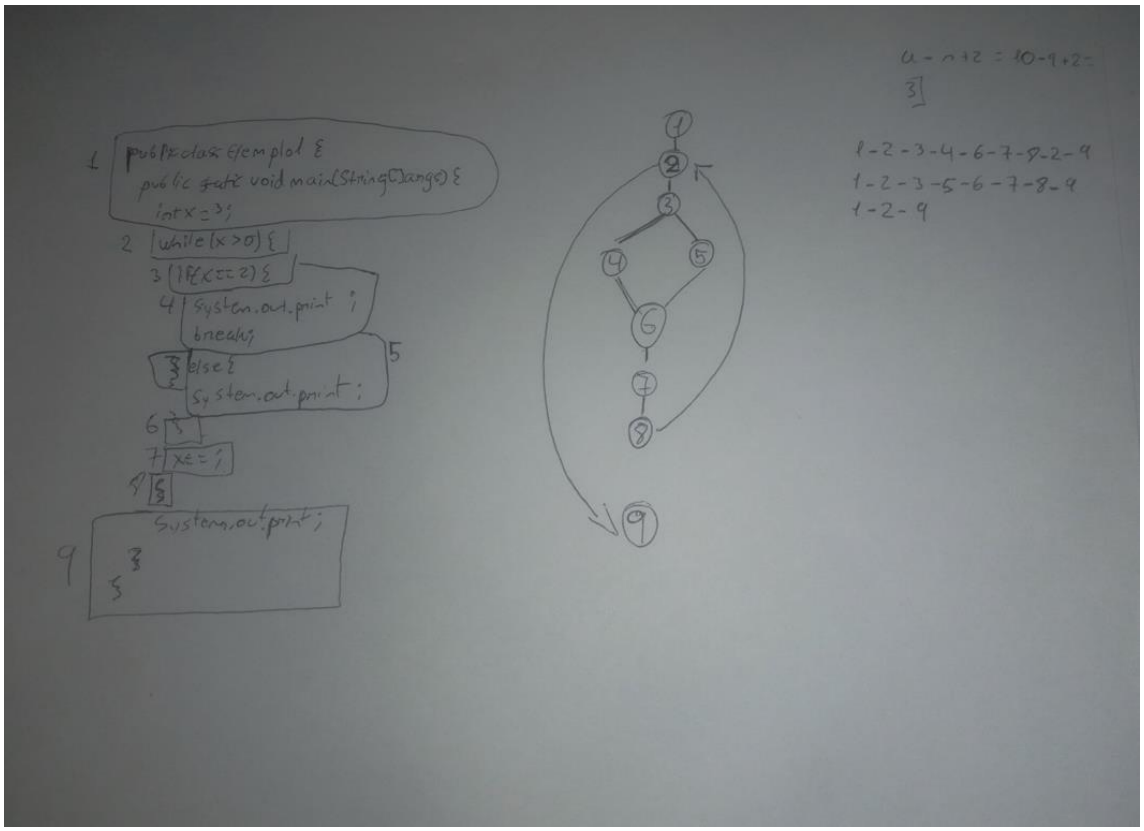
Diseño de pruebas

Pruebas estructurales (Caja Blanca)

Caso 1:

Se practica el detrimento del valor de una variable hasta que llegue a ser menor que 0, esto es útil para un software que por ejemplo tenga una cuenta regresiva controlada y deba mostrar los datos alojados en la variable, por ejemplo, para mostrar una cuenta atrás de fin de año en una web.

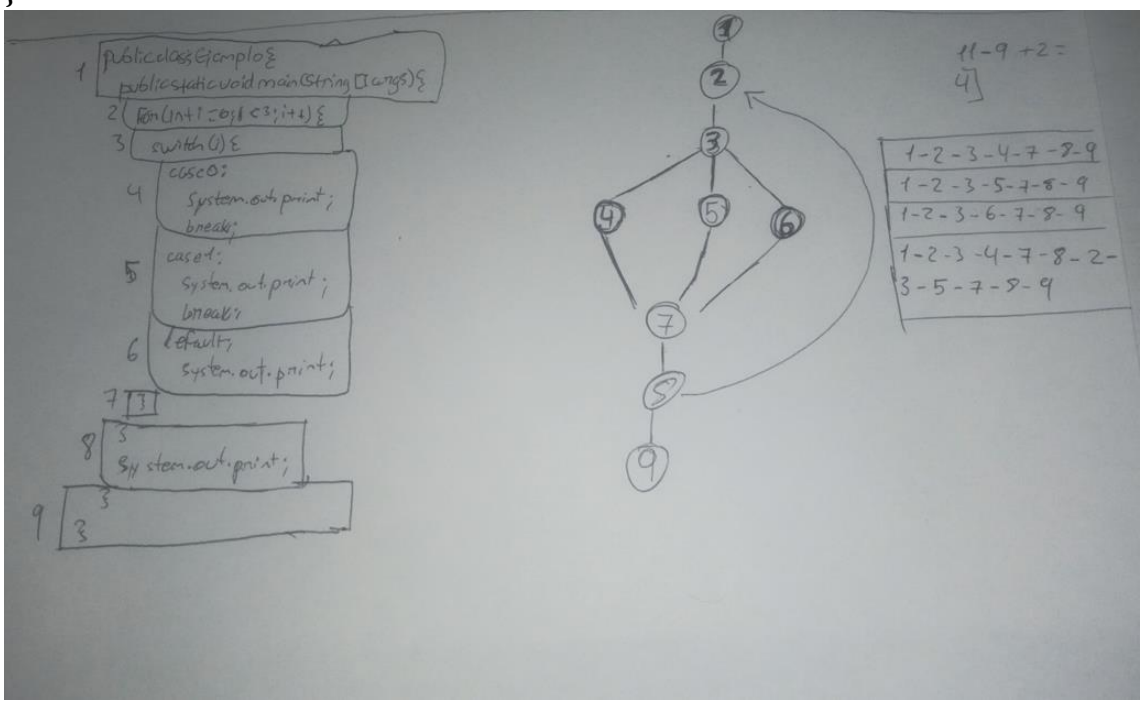
```
public class EjemploCaja1 {

 public static void main(String[] args) {

int x = 3;

while (x > 0) {
        if (x == 2) {
           System.out.println("Valor especial: 2");
           break;
        } else {
           System.out.println("Valor de x: " + x);
        }
        x--;
     }
     System.out.println("Fin del programa.");
  }
}
```


a - n + 2 = 10 - 9 + 2 =
3]
1 public class Ejemplo1 {
public static void main(String[]args) {
int x = 3;
2 while(x>0) {
3 if(x==2) {
4 System.out.print ;
break;
}else{
5
System.out.print;
6 }
7 x=;
8 }
System.out.print;
9 }
1-2-3-4-6-7-8-2-9
1-2-3-5-6-7-8-9
1-2-9

Caso 2:

Esto sería útil para un software que desencadenase una serie de eventos de naturaleza distinta según el caso una vez se usara alguna función, por ejemplo, imprimiendo los datos de un usuario si se presionara algún botón, utilizando funciones distintas, que corresponden a un case u otro, pero imprimiéndolo uno debajo de otro, de tal forma que no se note que tienen naturalezas distintas a nivel de usuario.

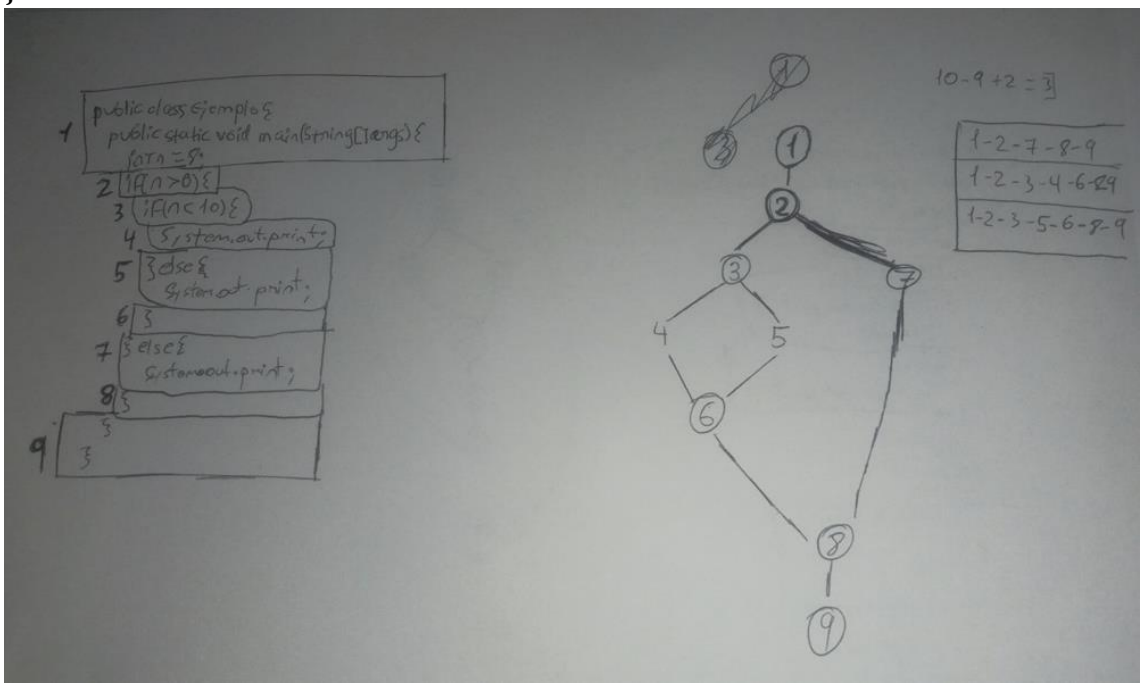
```
public class EjemploCaja2 {
   public static void main(String[] args) {
      for (int i = 0; i < 3; i++) {
         switch (i) {
            case 0:
               System.out.println("Cero");
               break;
            case 1:
               System.out.println("Uno");
               break;
            default:
               System.out.println("Otro valor");
         }
      }
      System.out.println("Terminado.");
   }
}
```



Caso 3:

Este código sería útil por ejemplo para un software donde debamos validar una edad, en el caso de que el valor fuera menor que 18 imprimiendo “menor de edad” tras introducirlo en un formulario, o si fuera mayor, “mayor de edad”.

```
public class EjemploCaja3 {
   public static void main(String[] args) {
      int n = 8;
      if (n > 0) {
         if (n < 10) {
            System.out.println("Número entre 1 y 9");
         } else {
            System.out.println("Número mayor o igual a 10");
         }
      } else {
         System.out.println("Número no positivo");
      }
   }
}
```



Pruebas funcionales (Caja Negra)

- Caso de prueba funcional: Ingresar correo electrónico válido en formulario

- El correo electrónico ha de tener;
    - Entre 3 y 20 caracteres.
    - Sólo contener letras, números, y guión bajo "_".

        - El dominio debe tener entre 5 y 18 caracteres.
            - La extensión del dominio debe ser .com, .es, o .net

- La contraseña ha de tener entre 5 y 15 caracteres

- La contraseña ha de incluir una mayúscula, una minúscula, y un número

- TABLAS:

Clases de equivalencia	Clases válidas	Clases inválidas
Correo de 3 a 20 carac	Correo>=3 && Correo<=20 1	Correo <3 || Correo >20 7
Correo Solo puede contener letras, números y guion bajo	Correo =[a-z,A-Z,0-9,_] 2	Correo!=[a-z,A-Z,0-9,-] 8
Dom de 5 a 18 carac	Dom >=5 && Dom <=18 3	Dom <5 || Dom >18 9
La extensión del dominio debe ser ".com", ".es" o ".net"	Ext=[.com||.es||.net] 4	Ext!=[.com||.es||.net] 10
pass de 5 a 15 carac	pass>=5 && pass<=15 5	pass<5 || pass>15 11
pass minimo 1 mayus, 1 minus y 1 num	pass==[A_Z,a-z,0-9] 6	pass!==[A_Z,a-z,0-9] 12

Prueba	Clases que cumple	Resultado

Cortes512@hotmail.net Password; Ct43xl	1,2,3,4,5,6	OK
Cortes125?@gmail.com Password; Cortessxl6	1,8,3,4,5,6	ERROR
CortesL@mail.es Password; Op24koi	1,2,9,4,5,6	ERROR
CortesL@gmail.camion Password; Op24koi	1,2,3,10,5,6	ERROR
CortesL@gmail.camion Password; Ahnehnlaieuyui123412	1,2,3,4,11,6	ERROR
CortesL@gmail.camion Password; OpiKaxm	1,2,3,4,5,12	ERROR
C2@hotmail.net Password; Op24koi	7,2,3,4,5,6	ERROR

Enlace a GitHub

https://github.com/Cortes-cmd/Entornos-de-Desarrollo.git

Bibliografía

Clemente, I. S. (s/f). *Ejercicios ::* IES San Clemente. Recuperado el 8 de mayo de 2025, de https://manuais.pages.iessanclemente.net/plantillas/DUAL/cd/ud03/6.ejercicios/index.print.html

Ejercicios Caja Blanca. (s/f). Scribd. Recuperado el 8 de mayo de 2025, de https://es.scribd.com/document/543733801/ejercicios-caja-blanca

El modelo de caja. (s/f). MDN Web Docs. Recuperado el 8 de mayo de 2025, de https://developer.mozilla.org/es/docs/Learn_web_development/Core/Styling_basics/Box_model

Prueba caja blanca bucles. (s/f). Prezi.com. Recuperado el 8 de mayo de 2025, de https://prezi.com/bfu3exukjedk/prueba-caja-blanca-bucles/

¿Qué son las pruebas de caja blanca? - Software Check Point. (2022, marzo 28). Check Point Software. https://www.checkpoint.com/es/cyber-hub/cyber-security/what-is-white-box-testing/

Wikipedia contributors. (s/f). *Caja blanca (sistemas)*. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=Caja_blanca_(sistemas)&oldid=155259843