DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

Pruebas funcionales y Pruebas no funcionales

Calidad de Software - CSY4111

CONTENIDO

DuocUC®

Refrescando conocimiento

DuocUC®

REFRESCANDO CONOCIMIENTO

Está directamente relacionado al ciclo de vida del software.
No solo se realizan pruebas al software, sino también a los requerimientos, especificaciones de diseño, documentos de operación, material de capacitación, etc.

- Es estático y dinámico.
- Se planifica.
- Se prepara.
- Se evalúa.

Está relacionado al ciclo de vida del software, se ha observado que muchos defectos se encuentran en la toma de requerimientos, diseño, etc. no solo en la programación del SW.

REFRESCANDO CONOCIMIENTO

Caso de prueba:

- Descripción de lo que se va a probar.
- Es un proceso creativo.

Datos de prueba:

- Conjunto de datos necesario para ejecutar un caso de prueba.
- Se detectan y se pueden crear (es más costoso).

01

TÉCNICAS DE DISEÑO

DuocUC®

TÉCNICAS DE DISEÑO

Al diseñar casos de prueba, es importante considerar diferentes enfoques que nos permitan abordar de manera efectiva la evaluación del software.

En este sentido, existen tres enfoques principales que nos proporcionan diferentes perspectivas para diseñar casos de prueba:

- Enfoque estructural o de caja blanca,
- Enfoque funcional o de caja negra,
- Enfoque aleatorio.

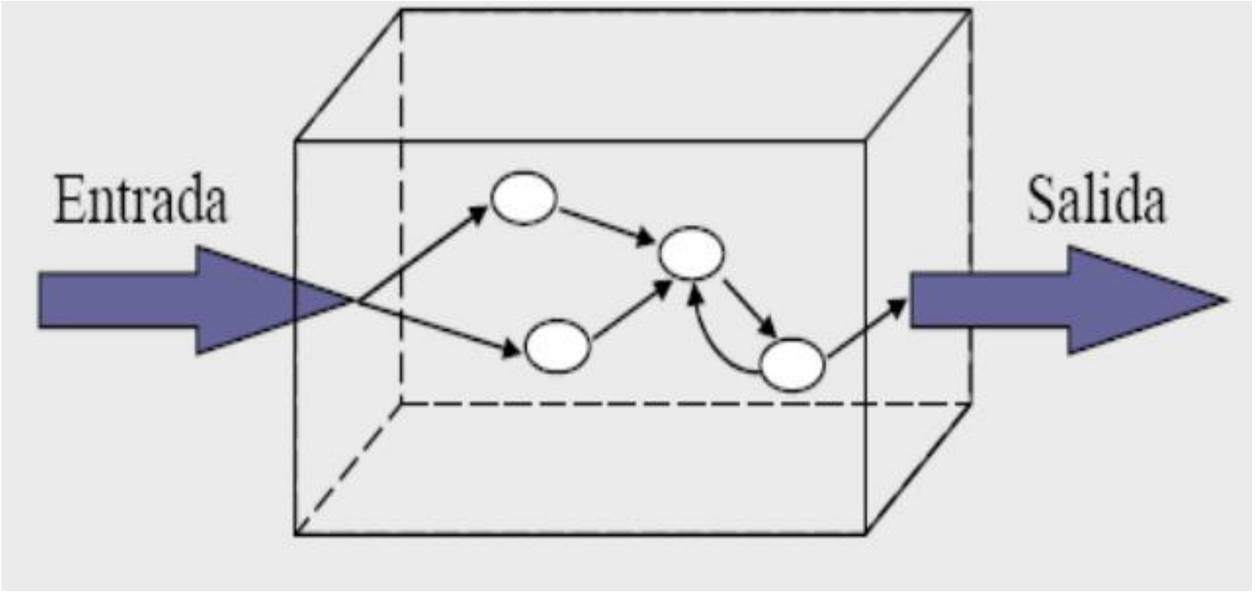
A continuación, exploraremos brevemente cada uno de estos enfoques y cómo se aplican en el diseño de casos de prueba.

TÉCNICAS DE DISEÑO

Enfoque estructural o de caja blanca:

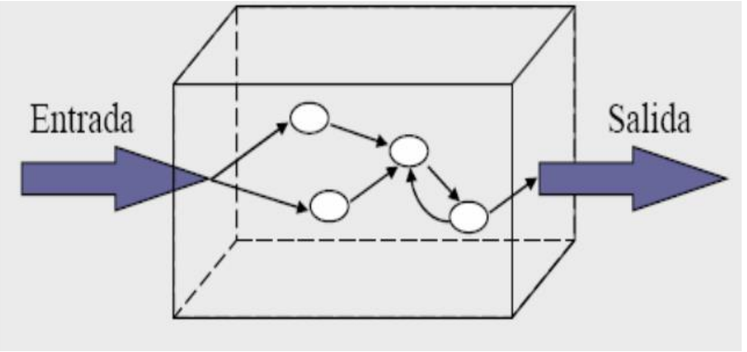
Se enfoca en comprender la estructura interna del programa y analizar los caminos de ejecución.

Se examina el código fuente y se identifican las condiciones y ramificaciones lógicas para diseñar casos de prueba que cubran diferentes rutas y aseguren una cobertura exhaustiva.



DuocUC®

TÉCNICAS DE DISEÑO

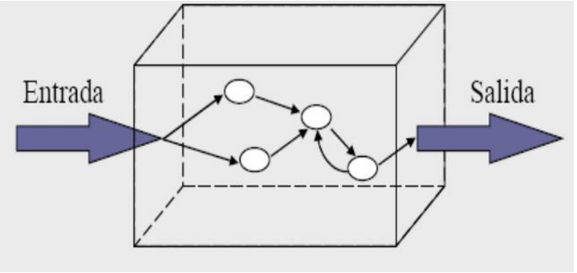


Técnica de cobertura de sentencia (caja blanca)

La cobertura de sentencia es una técnica cuyo objetivo principal es asegurarse de que todas las sentencias de código del programa han sido ejecutadas al menos una vez durante las pruebas.

Cuando hablamos de una sentencia de código nos referimos a cualquier instrucción que el programa puede ejecutar, como declaraciones de variable, asignaciones, llamadas de función, bucles y condicionales**.**

TÉCNICAS DE DISEÑO



Técnica de cobertura de sentencia (caja blanca)

La prueba de cobertura de sentencia es importante porque ayuda a identificar las sentencias que no han sido ejecutadas.

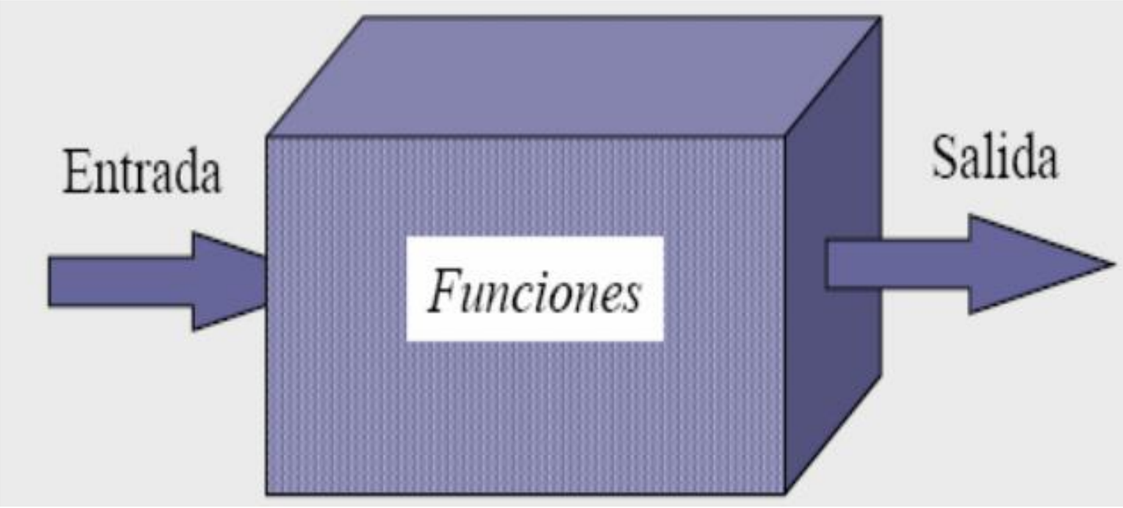
Las sentencias no ejecutadas pueden ser un signo de código muerto o de secciones de código que no se utilizan.

También puede ser un signo de que algunas funcionalidades del programa no se han probado adecuadamente.

TÉCNICAS DE DISEÑO

Enfoque estructural o de caja negra:

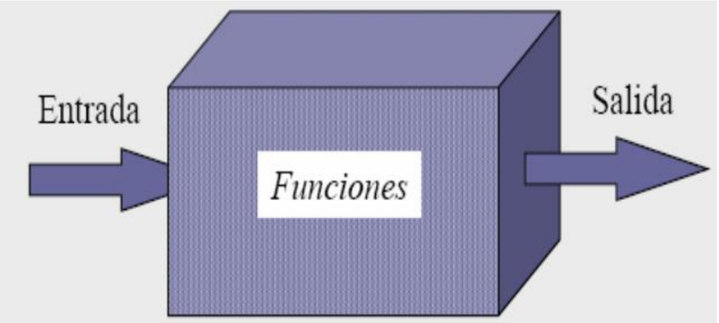
El enfoque funcional se centra en las funciones, entradas y salidas del software sin considerar su implementación interna.



Se diseñan casos de prueba basados en los requisitos y se evalúa el comportamiento funcional del sistema, sin necesidad de conocer los detalles de su implementación.

DuocUC®

TÉCNICAS DE DISEÑO



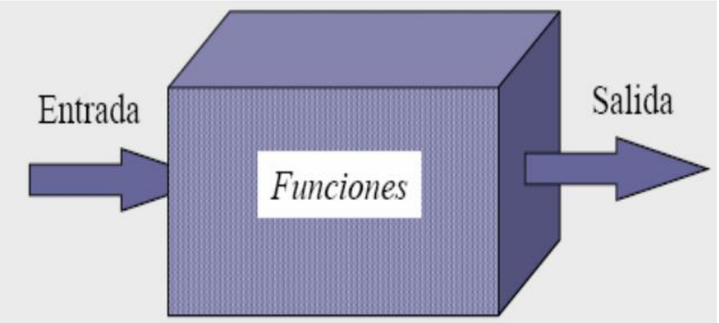
Técnica de particiones o Clases de Equivalencia (caja negra)

divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

Para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.

TÉCNICAS DE DISEÑO



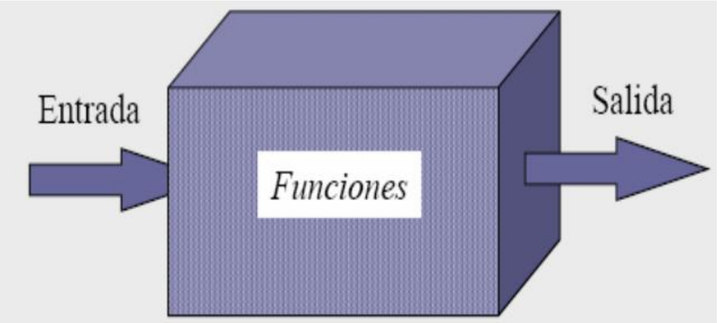
Técnica de particiones o Clases de Equivalencia (caja negra)

Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Si una condición de entrada especifica un rango delimitado por los valores a y b, se deben generar casos para a y b y casos no válidos justo por debajo y justo por encima de a y b, respectivamente.

Ej: si el programa requiere una entrada que sea un número real en el rango de 0.0 a 90.0, escribe casos de prueba para 0.0, 90.0, -0.0001 y 90.0001

TÉCNICAS DE DISEÑO



Técnica de particiones o Clases de Equivalencia (caja negra)

Si una condición de entrada especifica un número de valores (Ej: 2 a 5), se deben desarrollar casos de prueba que ejerciten los valores máximos (5) y mínimo (2), uno más el máximo (6) y uno menos el mínimo (1).

TÉCNICAS DE DISEÑO

Técnicas basadas en la experiencia:

Las técnicas de pruebas basadas en la experiencia son un conjunto de métodos y enfoques utilizados para diseñar y ejecutar pruebas de software, basados en el conocimiento y la experiencia acumulada por el equipo de pruebas y otros expertos en el campo.

Estas técnicas se basan en la intuición, el juicio y el conocimiento adquirido a lo largo del tiempo, y se aplican para identificar posibles problemas, defectos o riesgos en el software, sin seguir un enfoque estructurado o sistemático específico.

TÉCNICAS DE DISEÑO

Enfoque	Ventajas	Desventajas
Caja Blanca	Cobertura detallada del código, identificación precisa de errores, mayor capacidad para probar casos extremos.	Requiere conocimiento profundo del código, no garantiza cobertura exhaustiva, cambios en la estructura pueden afectar los casos existentes.
Caja Negra	Enfoque centrado en las funcionalidades, independiente de la estructura interna, mejor representación de casos de uso reales.	No revela problemas internos, dependencia de requisitos precisos, cobertura limitada de rutas posibles.
Experiencia	Generación rápida de casos de prueba, cobertura diversa de escenarios, identificación de errores inesperados.	Poca representatividad de casos de uso reales, necesidad de gran cantidad de casos para cobertura adecuada, dificultad en la interpretación de resultados.

02

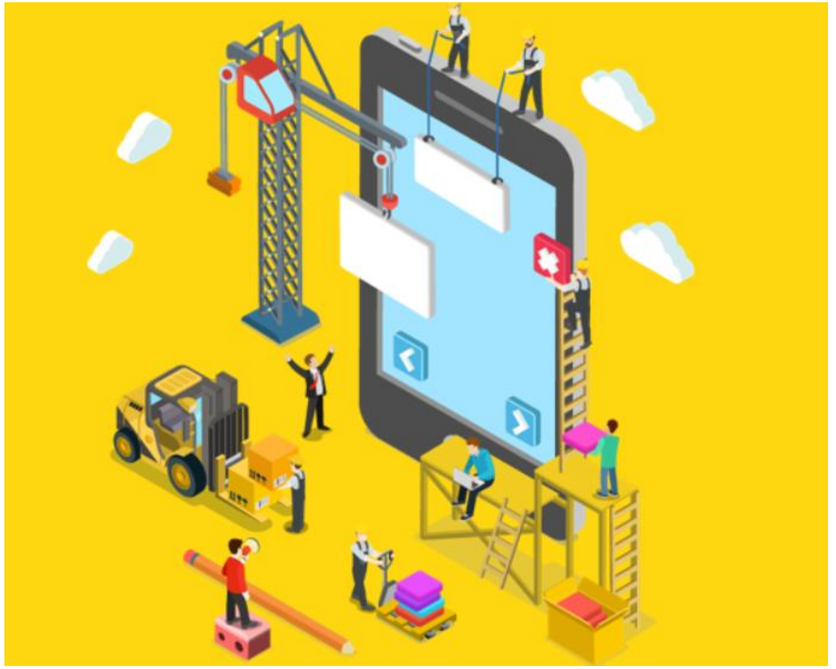
ESCENARIOS DE PRUEBAS

DuocUC®

ESCENARIOS DE PRUEBA

Es una secuencia de pasos o eventos que representan una situación o caso de uso específico en el sistema o software que se va a probar.

Un escenario de pruebas describe una interacción entre el usuario o sistema y el software, con el objetivo de evaluar su comportamiento y verificar si cumple con los requisitos establecidos.



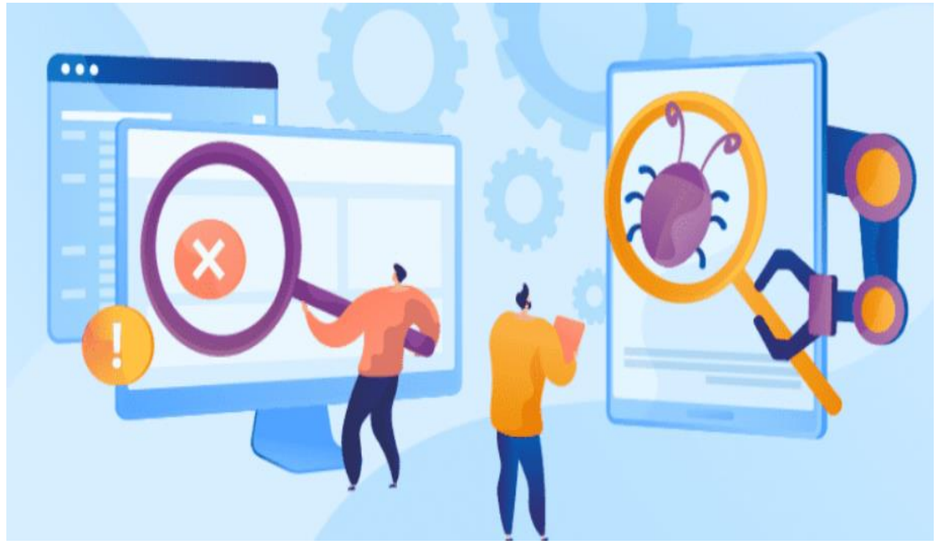
Los escenarios de pruebas pueden ser diseñados para probar diferentes funcionalidades, características o casos de uso del software

DuocUC®

ESCENARIOS DE PRUEBA

Los escenarios de pruebas permiten simular situaciones reales en las que el software puede encontrarse durante su uso normal, y ayudan a identificar posibles problemas, defectos o comportamientos inesperados.

Al diseñar escenarios de pruebas, es importante cubrir una amplia gama de situaciones para garantizar una evaluación completa y exhaustiva del software.



Proporcionan un enfoque concreto y estructurado para realizar pruebas efectivas y garantizar un software sólido y libre de errores.

DuocUC®

ESCENARIOS DE PRUEBA

Algunos ejemplos:

Escenario de transferencia exitosa:

Paso 1: Iniciar sesión en la aplicación bancaria.
Paso 2: Seleccionar la opción de transferencia de fondos.
Paso 3: Ingresar el monto a transferir y las cuentas de origen y destino.
Paso 4: Confirmar los detalles de la transferencia.
Paso 5: Verificar que los fondos se hayan transferido correctamente a la cuenta de destino.
Paso 6: Verificar que se haya registrado un registro de transacción en el historial de transferencias.

ESCENARIOS DE PRUEBA

Escenario de transferencia con fondos insuficientes:

Paso 1: Iniciar sesión en la aplicación bancaria.
Paso 2: Seleccionar la opción de transferencia de fondos.
Paso 3: Ingresar un monto superior al saldo disponible en la cuenta de origen.
Paso 4: Confirmar los detalles de la transferencia.
Paso 5: Verificar que se muestre un mensaje de error indicando fondos insuficientes.
Paso 6: Verificar que no se realice la transferencia y que los saldos no se vean afectados.

ESCENARIOS DE PRUEBA

Escenario de transferencia con información incorrecta:

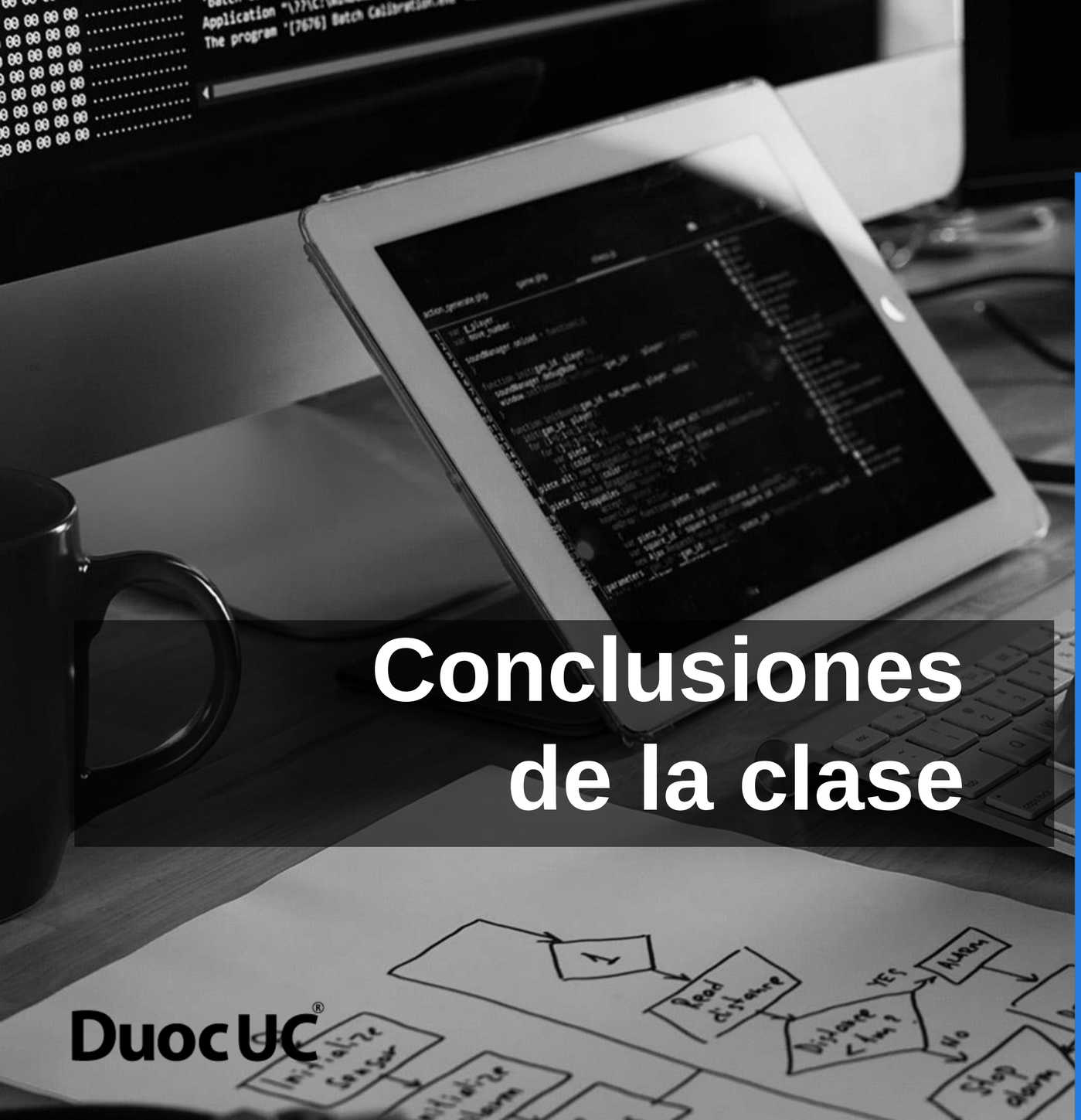
Paso 1: Iniciar sesión en la aplicación bancaria.
Paso 2: Seleccionar la opción de transferencia de fondos.
Paso 3: Ingresar un monto válido, pero ingresar una cuenta de destino inválida.
Paso 4: Confirmar los detalles de la transferencia.
Paso 5: Verificar que se muestre un mensaje de error indicando información de cuenta inválida.
Paso 6: Verificar que no se realice la transferencia y que los saldos no se vean afectados.

ESCENARIOS DE PRUEBA

Mas ejemplos para desarrollar:

Transferencia exitosa con saldo suficiente.
Transferencia fallida por saldo insuficiente.
Transferencia a una cuenta inexistente.
Transferencia con monto mínimo permitido.
Transferencia con monto máximo permitido.
Transferencia con datos de cuenta inválidos.
Transferencia programada para una fecha futura.
Transferencia entre cuentas del mismo titular.
Transferencia internacional con cambio de moneda.
Transferencia reversa de una transacción anterior.

Conclusiones de la clase



DuocUC®

Conclusiones

- ✓ Los escenarios de pruebas proporcionan un enfoque concreto y estructurado para realizar pruebas efectivas y garantizar un software sólido y libre de errores.

- ✓ El enfoque de caja negra se centra en las funciones, entradas y salidas del software sin considerar su implementación interna.

DuocUC®

Bibliografía

- ✓ Alex Andrade. (10 septiembre, 2021). Cómo desarrollar un plan de pruebas sólido pero simple. https://alexandrade.net Recuperado de https://alexandrade.net/blog-de-ingenieria-de-software/calidad-de-software/como-desarrollar-un-plan-de-pruebas-solido-pero-simple/

DuocUC®

DuocUC® INFORMÁTICA Y TELECOMUNICACIONES

¿Qué entendemos por calidad y Calidad de software?

Calidad de Software - CSY4111