

Técnicas avanzadas de diseño de casos de prueba

Calidad de Software - CSY4111



CONTENIDO

01
CASOS DE PRUEBA

02
DISEÑO CASO DE
PRUEBA

03
TECNICAS DE DISEÑO

Refrescando conocimiento

DuocUC[®]



REFRESCANDO CONOCIMIENTOS

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.

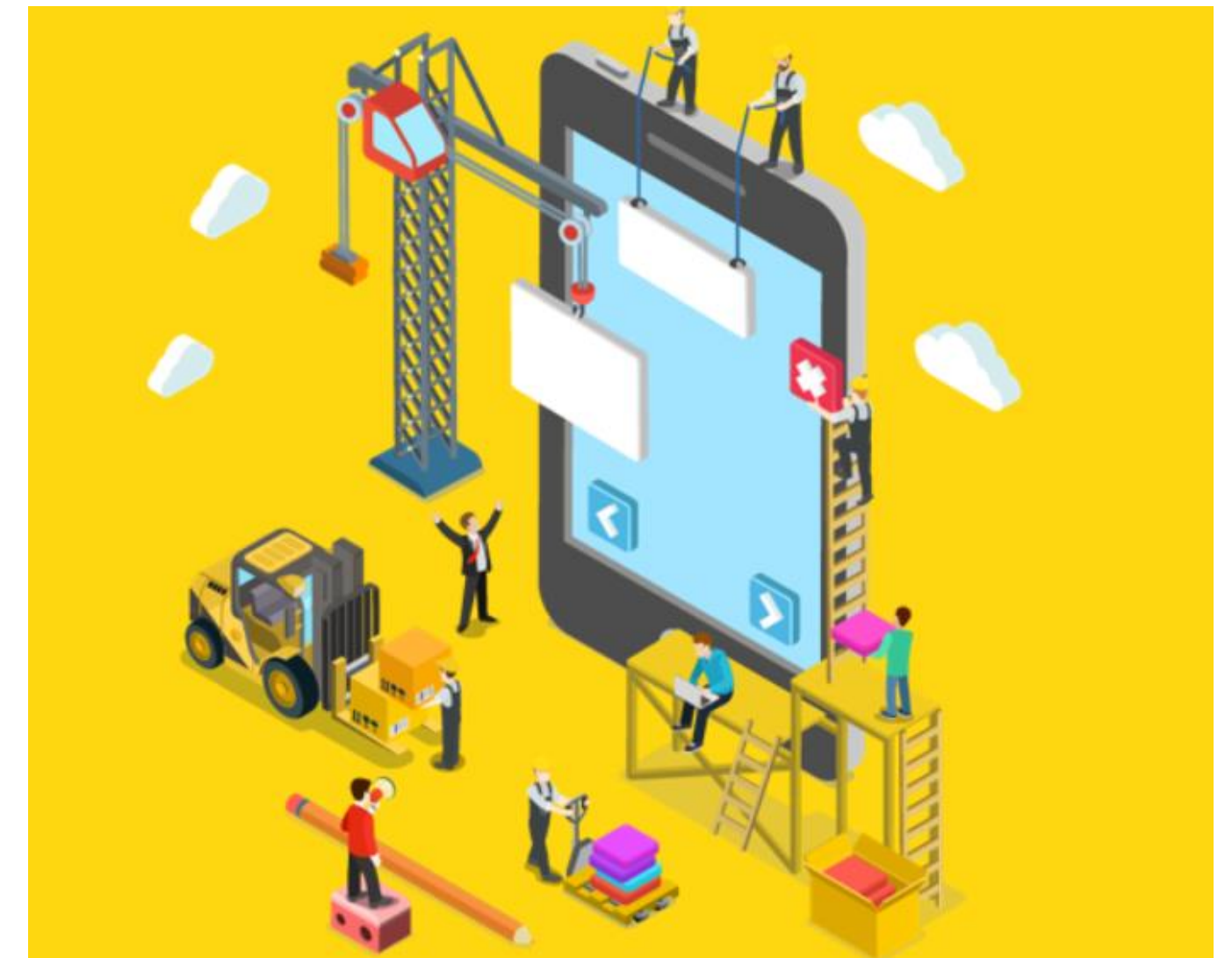
A continuación, exploraremos brevemente cada uno de estos enfoques y cómo se aplican en el diseño de casos de prueba.

REFRESCANDO CONOCIMIENTOS

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



01

CASOS DE PRUEBA

CASOS DE PRUEBA

Un caso de prueba es un conjunto de condiciones bajo las cuales un analista determinará si una aplicación, es parcial o completamente satisfactoria



Se pueden crear muchos casos de prueba para determinar que un requisito es completamente satisfactorio.

Debe haber al menos un caso de prueba para cada requisito.

CASOS DE PRUEBA

Algunas políticas de empresa determinan crear al menos dos casos de prueba para cada requisito, con el objetivo de cumplir el resultado correcto como el incorrecto.

En un caso de prueba tenemos una entrada conocida más una precondición y una salida junto a una postcondición.



Ejemplo: “Caso de prueba para validar un Rut”

Entrada: números de Rut.

Precondición: El campo debe permitir el ingreso de letras puntos, guion medio y números

Salida: Correcto o incorrecto

Postcondición: En caso de error muestra un mensaje, en caso avanza a la siguiente fase.

CASOS DE PRUEBA

Los casos de prueba incluyen una descripción de la funcionalidad que se probará, esta funcionalidad se toma de los requisitos o de los casos de uso.

También incluyen una preparación requerida en caso de que lo amerite.

Estos procesos individuales a su vez están compuestos de condiciones y/o variables que son definidas en base a los requerimientos iniciales del proyecto, es decir, conocer con claridad cual es la finalidad del software.



CASOS DE PRUEBA

El diseño de casos de prueba tiene como objetivo conseguir la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.



Podemos diseñar casos de prueba conociendo la función para la que el software fue diseñado o conociendo el funcionamiento del producto.

CASOS DE PRUEBA

En resumen, necesitamos diseñar una serie de casos de pruebas para conseguir una alta probabilidad de encontrar tantos errores como nos sea posible antes de entregar el software al cliente.

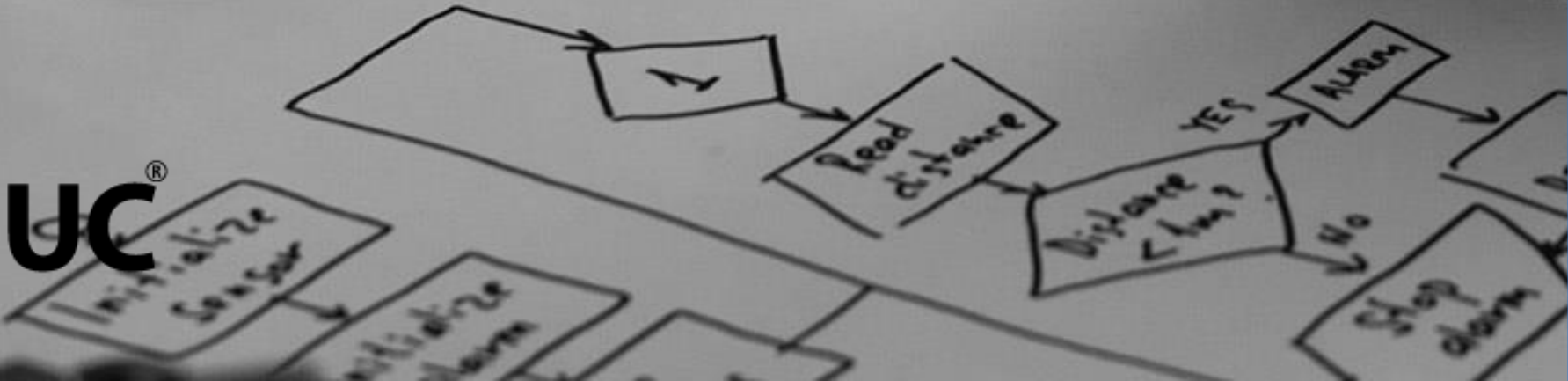


Tenemos que considerar que para la entrada tenemos una precondición y para la salida una postcondición.



02

DISEÑO CASO DE PRUEBA



DISEÑO CASO DE PRUEBA

El diseño de pruebas es un proceso creativo de un conjunto de casos de prueba efectivos y eficientes que cubran diferentes escenarios y condiciones del software a ser probado.

El objetivo del diseño de pruebas es garantizar una cobertura adecuada de las funcionalidades, características y requisitos del software, de modo que se puedan identificar y resolver los posibles defectos o problemas antes de su implementación.

Los casos de pruebas buscan verificar que el software haga lo que se supone que debe hacer.

DISEÑO CASO DE PRUEBA

Estructura de pruebas:

La estructura de pruebas se refiere a la organización y ordenamiento de los casos de prueba en un plan de pruebas coherente y estructurado.

Implica la clasificación y agrupación de los casos de prueba según diferentes criterios, como la funcionalidad, la prioridad, la criticidad o las características específicas que se desean probar.

Se refiere a la organización y disposición de los casos de prueba en un plan coherente y estructurado.

DISEÑO CASO DE PRUEBA

Estructura de pruebas:

La estructura de pruebas proporciona una visión general del alcance y la cobertura de las pruebas, permitiendo una mejor planificación y ejecución de las mismas.

También facilita la identificación de los casos de prueba relevantes para un determinado objetivo de prueba y ayuda a optimizar el tiempo y los recursos utilizados durante el proceso de prueba.



DISEÑO CASO DE PRUEBA

Estructura de un caso de prueba

Nombre del proyecto: Nombre de la aplicación

Ambiente de pruebas: Versión web o de escritorio

Autor del caso de prueba: Nombre del analista que diseña el caso de prueba.

ID: caso de prueba: Número que identifica el caso de prueba

ID Historia de usuario: Número con el que identifica la historia de usuario que hace referencia al caso.

Propósito: Cual es la finalidad del caso de prueba.

#: Nro. De la acción.

Los casos de pruebas buscan verificar que el software haga lo que se supone que debe hacer.

DISEÑO CASO DE PRUEBA

Estructura de un caso de prueba (Continuación...)

Acciones: pasos a seguir para la verificación del caso.

Salida Esperada: Resultado esperado según los requerimientos iniciales.

Salida obtenida: Resultados obtenidos.

Resultado: Aprobado/En Seguimiento/Rechazado

Seguimiento: Código de la acción donde el resultado no fue el esperado.

Severidad: El grado de la falla.

Evidencia: Soporte de que los resultados son correctos.

Aprobado por: Nombre del analista que realizó la Prueba.

Fecha de Revisión: Fecha en la que se realizaron las pruebas.



03 TECNICAS DE DISEÑO

DuocUC[®]

TECNICAS DE DISEÑO

Técnica de Partición de Equivalencia:

Divide el conjunto de datos de entrada en clases de equivalencia o grupos, asumiendo que, si un caso de prueba dentro de una clase es válido o inválido, los demás casos de la misma clase tendrán un comportamiento similar.

Selecciona casos de prueba representativos de cada clase, minimizando la redundancia y maximizando la cobertura.



TECNICAS DE DISEÑO

Técnica de Análisis de Valores Límites:

Selecciona los valores límites y los puntos de transición dentro de los rangos de entrada.

Diseña casos de prueba que evalúen el comportamiento del software en los límites y puntos de transición, donde es más probable que ocurran errores.





















TECNICAS DE DISEÑO

Técnica de Tabla de Decisiones:

Identifica las condiciones lógicas y las acciones asociadas en forma de tabla de decisión.

Diseña casos de prueba que cubran todas las combinaciones posibles de condiciones, garantizando una cobertura exhaustiva.

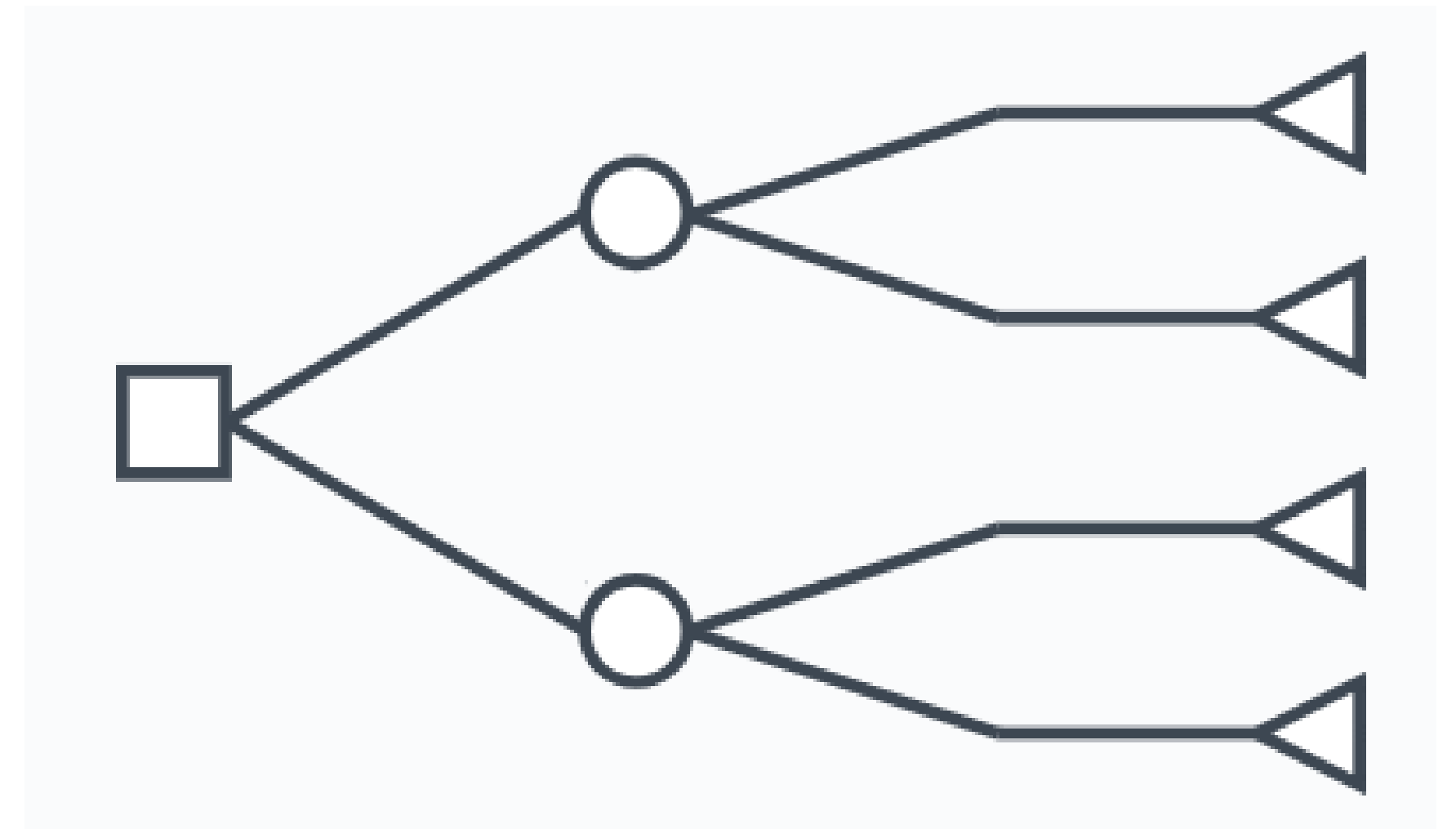
					
					
					

TECNICAS DE DISEÑO

Técnica de Ruta de Decisión:

Analiza el flujo de control del software y crea casos de prueba para cubrir todas las posibles rutas de ejecución.

Identifica las decisiones y condiciones lógicas en el software y diseña casos de prueba para evaluar su comportamiento.



TECNICAS DE DISEÑO

Técnica de Pruebas Basadas en Dominio:

Utiliza el conocimiento del dominio del negocio para diseñar casos de prueba relevantes.

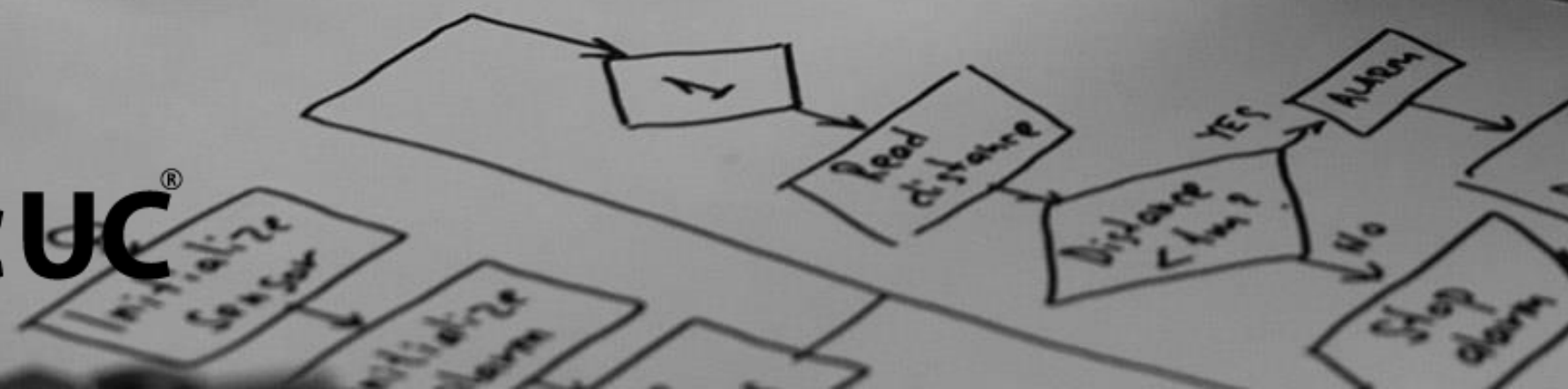
Se enfoca en los requisitos funcionales y no funcionales específicos del dominio del negocio, posteriormente se deben diseñar casos de prueba que cubran esos aspectos críticos.



Es importante combinar varias técnicas y enfoques para lograr una cobertura efectiva y exhaustiva durante el diseño de casos de prueba.

Conclusiones de la clase

DuocUC[®]



Conclusiones

- ✓ Es importante combinar varias técnicas y enfoques para lograr una cobertura efectiva y exhaustiva durante el diseño de casos de prueba.
- ✓ Los casos de pruebas buscan verificar que el software haga lo que se supone que debe hacer.

BIBLIOGRAFIA

- ✓ William Mc Mullin. (27 de mayo de 2021). Cómo escribir casos de prueba para software: ejemplos y tutorial. <https://es.parasoft.com> Recuperado de <https://es.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/>

Técnicas avanzadas de diseño de casos de prueba

Calidad de Software - CSY4111