

CASO DE PRUEBA

Es un conjunto de acciones que se ejecutan para verificar una característica o funcionalidad particular de una aplicación de software.

Se presenta por escrito con información de qué y como probar y deben incluir todas las funciones del software.

- **Testing positivo:** sucede cuando se valida el flujo normal de un sistema bajo prueba.

- **Testing negativo:** validan flujos no contemplados dentro de los requisitos de un sistema bajo prueba.

CASOS DE USO:

Antes de la realización del diseño de los casos de prueba, es el análisis de los documentos que aseguran los requisitos del cliente. Estos requisitos por lo general son casos de uso.

Un caso de uso cuenta la historia de como un usuario interactúa con un sistema de software para lograr o abandonar un objetivo. Puede contener varias rutas que el usuario sigue, las cuales son denominadas escenarios de casos de uso.

PRUEBAS DE CASOS DE USO:

Es la combinación de las dos técnicas anteriores, realizando diferentes casos de prueba sobre un determinado caso de uso.

Es una técnica de caja negra donde se busca saber si la ruta utilizada por el usuario está funcionando según lo esperado o no.

Cobertura: Es la medida que utiliza el testeo para verificar que ha probado todos los casos de prueba derivados de la técnica de prueba.

$$\text{Cobertura} = \frac{\text{Rutas del caso de uso probadas}}{\text{Rutas del caso de uso Totales}}$$

CICLO DE VIDA DE LAS PRUEBAS DE SOFTWARE

Planeación

Se definen objetivos y el enfoque



→ Doc de salida: Plan de Prueba General y/o x nivel

Seguimiento y control

Se brinde información y se reevalúan las actividades de la prueba. Se pueden aplicar medidas correctivas.



→ Doc de salida: Informe de avance de la prueba

Análisis

Se determina "qué probar"



→ Doc de salida: Controles de prueba que contienen condiciones de prueba

Diseño

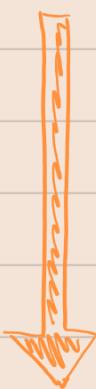
Se determina "cómo probar"



→ Doc de salida: Casos de prueba de alto nivel

Implementación

Se completan los "productos" de prueba incluyendo la secuenciación de casos de prueba en procedimientos de prueba



→ Doc de salida:

- Procedimientos y datos de prueba
- Calendario de ejecución
- Test suites

Ejecución

Se ejecutan los casos de prueba



→ Doc de salida:

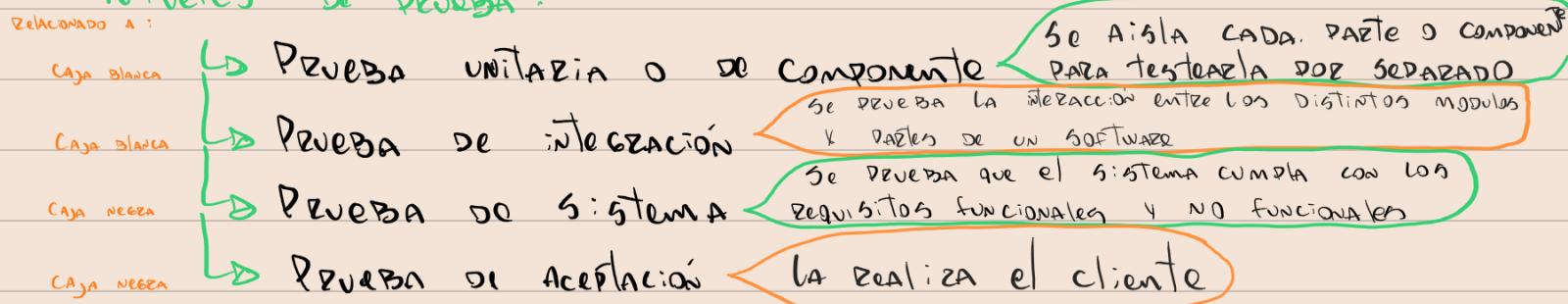
- Reporte de defectos
- Informe de ejecución de prueba

Conclusion

Actividades tendientes a organizar tras la finalización de las pruebas, como ser corroborar informes, guardar y almacenar entornos de prueba, etc.

DOC DE SALIDA: • Informe resumen de PRUEBA
• LECCIONES APRENDIDAS

Niveles de Prueba:



Típos de Prueba

- ↳ funcional: Son aquellas que evalúan las funciones que el sistema debe realizar. Describen "que hace" el sistema.
Los requerimientos funcionales suelen estar detallados en:
Requisitos del negocio; epicas; historias de usuario y casos de uso.
- ↳ No funcional: Prueba el "cómo se hace" se compone el sistema y suelté evaluar características como la USABILIDAD, eficiencia de desempeño o la SEGURIDAD.

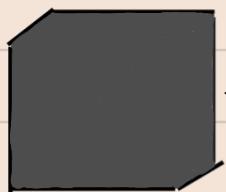
• Por qué la PRUEBA NO FUNCIONA
C NO es considerada de Caja

BLANCA? PARA VERIFICAR SEGURIDAD
NO NECESITO VER EL CODIGO?

Técnicas de Prueba:

CAJA NEGRA

Entrada →

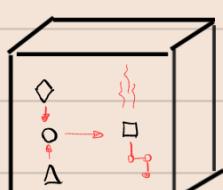


→ Salida

Se basan en el comportamiento extraído del análisis de los documentos que son base de prueba (DOC de Requerimientos formales, casos de uso, historias de usuario, etc). Son aplicables para pruebas funcionales y no funcionales. Se concentran en las entradas y en las salidas sin tener en cuenta la estructura interna.

CAJA BLANCA

Entrada →



→ Salida

Se basan en la estructura extraída de los documentos de ARQUITECTURA, Diseño detallado, estructura interna o código del sistema. Se concentran en el procesamiento dentro del objeto de prueba.

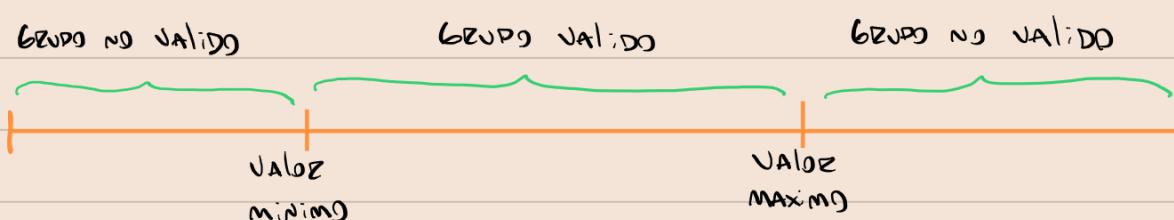
Técnicas de Prueba de Caja Negra:

↳ Partición de Equivalencias: Se busca separar los datos posibles a probar en aquellos que podrían darme el mismo resultado. Estos grupos son las particiones.

Una partición podrá ser **VALIDA** cuando se compone de datos admitidos por el sistema, y será **INVALIDA** cuando los datos deben ser rechazados por el sistema.

$$\text{COBERTURA} = \frac{\text{Particiones Probadas}}{\text{Particiones Identificadas}}$$

↳ Analisis de Valores Límites: Es una variante de la partición de equivalencias pero solo es posible usarla cuando la partición es secuencial y está ordenada. Se identifican los valores maximo y minimo de la secuencia y se establecen los grupos validos y no validos.



$$\text{COBERTURA} = \frac{\text{Valores Límites Identificados}}{\text{Valores Límites Probados}}$$

↳ Tabla de Decisión: Es una técnica utilizada para evaluar el comportamiento de un sistema cuando existen varias combinaciones posibles de datos de entrada. Estas combinaciones se grafican en forma de tabla habitualmente con valores verdaderos y falsos.

Si bien su requisito son las combinaciones de entrada, pueden existir varias salidas.

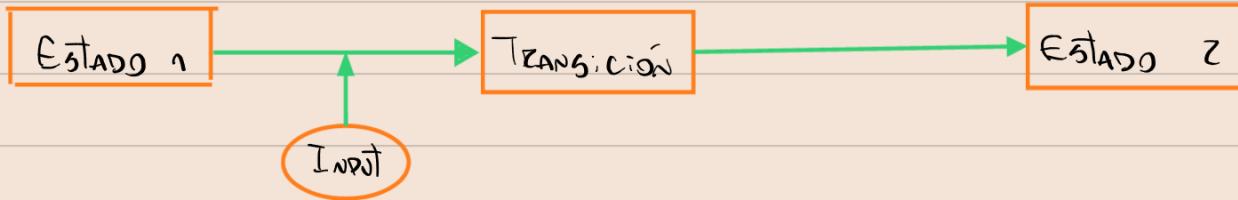
CONDICIONES	Regla 1	Regla 2	Regla 3	Regla 4
CONDICION 1	VERDADEO	VERDADEO	FALSO	FALSO

CONDICION 2	VERDADERO	FALSO	VERDADERO	FALSO
OUTPUT 1	VERDADERO	Falso	Falso	Falso
OUTPUT 2	Falso	VERDADERO	VERDADERO	VERDADERO

PARA SABER CUANTAS REGLAS USAREMOS SIMPLEMENTE ELEVAMOS EL NÚMERO 2 POR LA CANTIDAD DE CONDICIONES EXISTENTES. EN ESTA CASO EXISTEN 2, POR LO QUE EL CALCULO ES 2^2 DANDO 4 REGLAS.
SI EXISTIERA 3 SERÍA 2^3 DANDO 8 REGLAS.

↳ TRANSICIÓN DE ESTADOS: Muestra los posibles estados del software y las condiciones necesarias para su transición.

SE UTILIZA NORMALMENTE EN APLICACIONES BASADAS EN MENÚS, O DONDE UNA ENTRADA PUEDE PROVOCAR DIFERENTES SALIDAS SEGUN UNA CONDICIÓN ANTERIOR.



COBERTURA = $\frac{\text{Número de estados o transiciones identificados probados}}{\text{Número de estados o transiciones identificados en el objeto de prueba}}$

* Existen Técnicas Basadas en la Experiencia que no fueron desarrolladas en Clases, las cuales son: Predicción de errores, Prueba Exploratoria y Prueba Basada en lista de comprobación.

↳ LISTA DE COMPROBACIÓN: Se puede utilizar cuando existen varios requisitos para una misma condición (por ej: contraseñas con +6 caracteres, al menos 1 mayus, 1 num y 1 carácter especial) y son creadas basándose en la experiencia y conocimiento de lo que el probador cree que es importante para el usuario.

PRUEBAS DE HUMO Y RESECCIÓN

↳ PRUEBAS DE HUMO: Se ejecutan para evaluar la estabilidad de las compilaciones.



de software iniciales o desarrolladas recientemente. Su función es determinar si las funciones básicas del software se encuentran funcionando para saber si se continua con el desarrollo o no.

Si falla una funcionalidad principal, no tiene sentido seguir testeando. Es por ello que si existen 2000 casos de prueba, 10 serán parte de la suite de pruebas de humo.

Una vez nos encontramos en una versión estable donde todo el test de humo es probado exitosamente, pasamos a la siguiente etapa.

L **D** Pruebas de retroceso: Su función es verificar que un programa actualizado o cambio de código no ha afectado a las características existentes.



Se compone de una selección de casos de prueba previamente completados para garantizar que los elementos de prueba sigan funcionando.

Por lo general, se selecciona un grupo reducido del total de pruebas existentes (incluyendo todas las pruebas de humo). Los criterios de selección son, entre otros:

- Casos de prueba que suelen presentar defectos.
- Funciones que son más visibles para los usuarios.
- Casos de prueba que verifican características principales del objeto de prueba.

