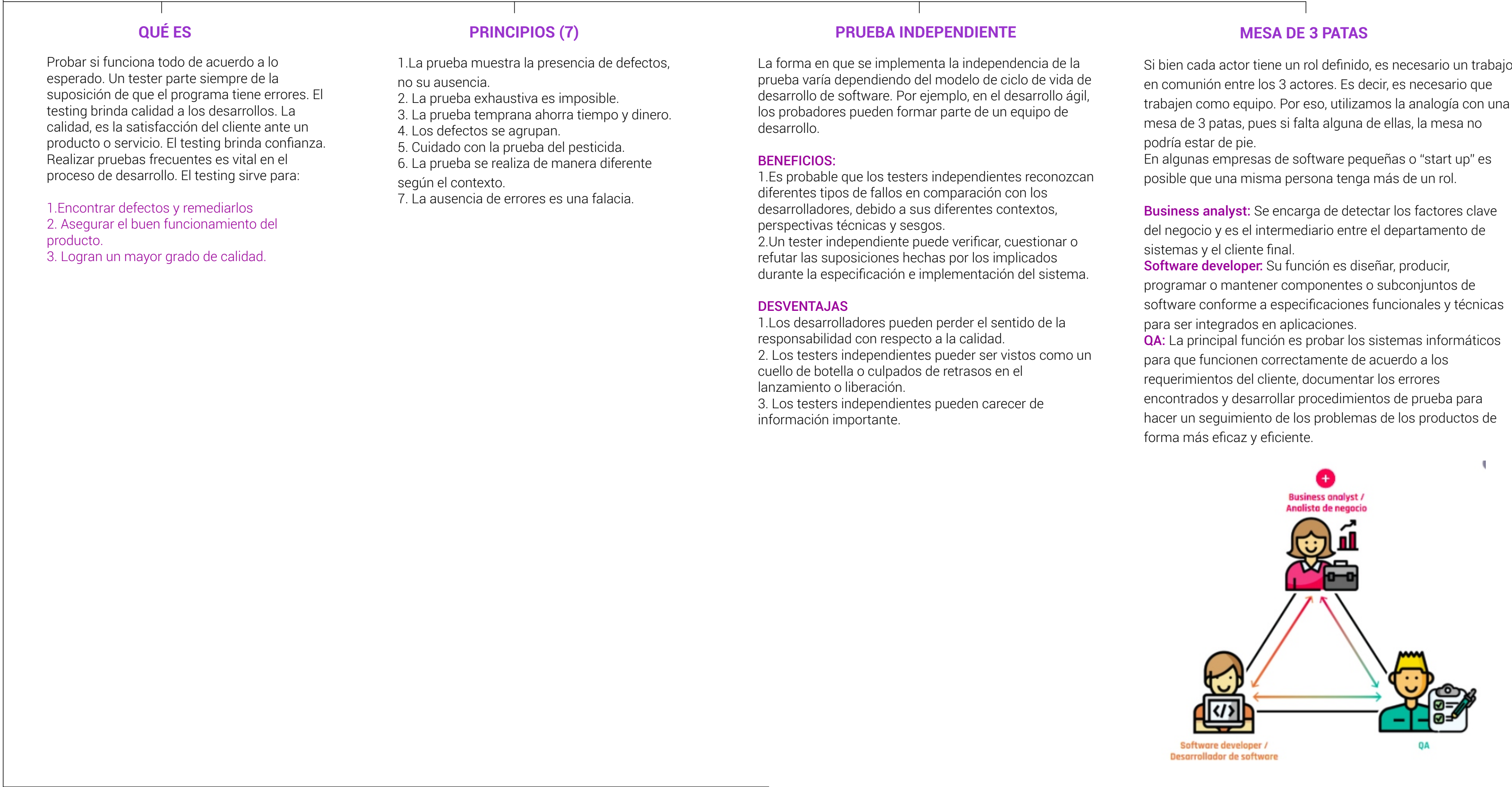


TESTING



DEFECTOS



VALIDACIÓN VS VERIFICACIÓN

Verificación es testear que el proyecto se adecue a las especificaciones y requerimientos establecidos. Construí lo que dije que iba a hacer?

Validación es testear que tan bien se adecua a las necesidades de negocio y del usuario que derivaron en la redacción de los requerimientos.

¿Estamos construyendo el producto **correctamente**?

¿Estamos construyendo el producto **correcto**?

Atributo	Descripción	Ejemplo
ID	Abreviatura de identificador, un código único e irreplicable que puede ser número o letras.	001 - Test01
Título	El título debe ser corto y específico, que se entienda en este lo que queremos reportar. Cuando el desarrollador o el equipo vean el título pueden interpretar rápidamente qué es, dónde está y cuál importante es ese defecto.	Login - Ingresar con campos en blanco
Descripción	Describir un poco más sobre el error, es decir, desarrollar lo que dejamos afuera en el título lo podríamos explicar acá.	En la pantalla login si dejo vacío los campos nombre y password y apreté ingresar, me lleva a la página principal.
Fecha del informe del defecto	La fecha que detecté el defecto para saber posteriormente el tiempo en que se resolvió.	23/04/21
Autor	El nombre del tester que descubrió el defecto, por si el desarrollador tiene una duda, sabe a quién consultar.	Pepito Román
Identificación del elemento de prueba	Nombre de la aplicación o componente que estamos probando.	Carrito compras
Versión	Es un número que nos indica en qué versión está la aplicación.	1.0.0
Entorno	El entorno en el que probamos (desarrollo, QA, producción).	Desarrollo
Pasos a reproducir	Los pasos a seguir para llegar al defecto encontrado.	1) Ingresar a la aplicación. 2) Dejar en blanco el campo nombre. 3) Dejar en blanco el campo password. 4) Hacer click en el botón "Ingresar".
Resultado esperado	Es lo que esperamos que suceda o muestre la aplicación muchas veces según los requerimientos de la misma.	No debe ingresar a la aplicación sin un usuario y una contraseña validos.
Resultado obtenido o actual	Es lo que sucedió realmente o lo que nos mostró la aplicación. Puede coincidir o no con el resultado esperado, si no coincide, hemos detectado un error o bug.	Ingresa a la aplicación sin usuario y sin contraseña.
Severidad	Cuán grave es el defecto que hemos encontrado, puede ser: bloqueado, critico, alto, medio, bajo o trivial.	Critico
Prioridad	Con esto decimos qué tan rápido se debe solucionar el defecto, puede ser: alta, media, baja	Alta
Estado del defecto	Los estados pueden ser: nuevo, diferido, duplicado, rechazado, asignado, en progreso, corregido, en espera de verificación, en verificación, verificado, reabierto y cerrado.	Nuevo
Referencias	Link al caso de prueba con el cual encontramos el error.	https://repositorio.-com.ar/TC-001-User-Login
Imagen	Se puede adjuntar una captura de pantalla del error, esto nos permite demostrar que el error sucedió y al desarrollador lo ayuda a ubicar el error.	

TÉCNICAS DE PRUEBA

DEFINICIONES

El objetivo de una técnica de prueba es ayudar a identificar las condiciones, los casos y los datos de prueba.

Elección de una técnica de prueba

La elección de la técnica de prueba a utilizar depende de los siguientes factores:

- Tipo y complejidad del componente o sistema
- Estándares de regulación
- Requisitos del cliente o contractuales
- Clases y niveles de riesgo
- Objetivo de la prueba
- Documentación disponible
- Conocimientos y competencias del probador
- Modelo del ciclo de vida del software
- Tiempo y presupuesto

TÉCNICAS DE CAJA NEGRA

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

- Las condiciones de prueba, casos de prueba y datos de prueba, se deducen de una base de prueba que puede incluir **requisitos de software, especificaciones, casos de uso e historias de uso**.
- Los casos de prueba se pueden utilizar para detectar diferencias entre los requisitos y su implementación, así como desviaciones respecto a los requisitos.
- La cobertura se mide en función de los elementos probados en la base de prueba y de la técnica aplicada a la base de prueba



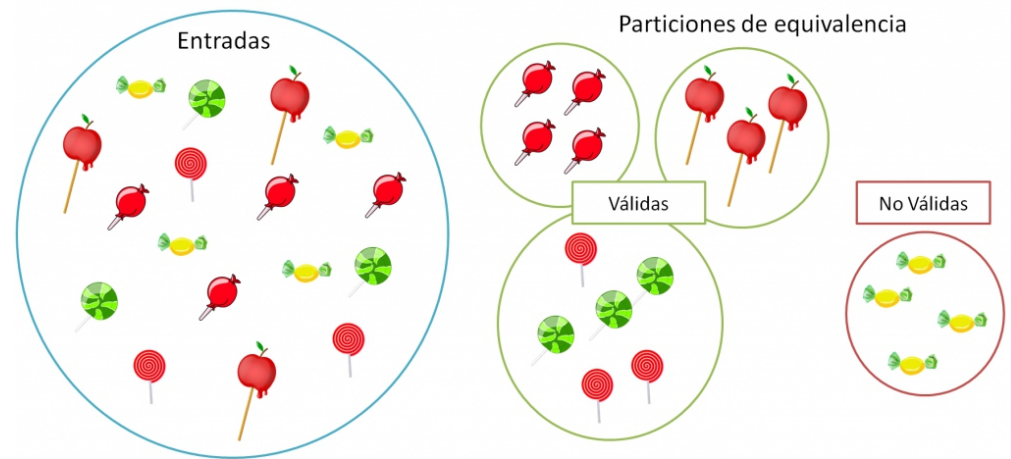
Caja negra

1 PARTICIÓN DE EQUIVALENCIA

En esta técnica se dividen los datos en particiones conocidas como **clases de equivalencia** donde cada miembro de estas clases o particiones es procesado de la misma manera. Las características de esta técnica son:

- 1.La "partición de equivalencia **válida**" contiene valores que son aceptados por el componente o sistema
- 2.La "partición de equivalencia **no válida**" contiene valores que son rechazados por el componente o sistema.
- 3.Se pueden dividir las particiones en subparticiones.
- 4.Cada valor pertenece a solo una partición de equivalencia.
- 5.Las particiones de equivalencia no válidas deben probarse en forma individual para evitar el enmascaramiento de fallos.
6. La cobertura se mide de la siguiente manera:

Cobertura = $\frac{\text{Particiones probadas}}{\text{Particiones identificadas}}$

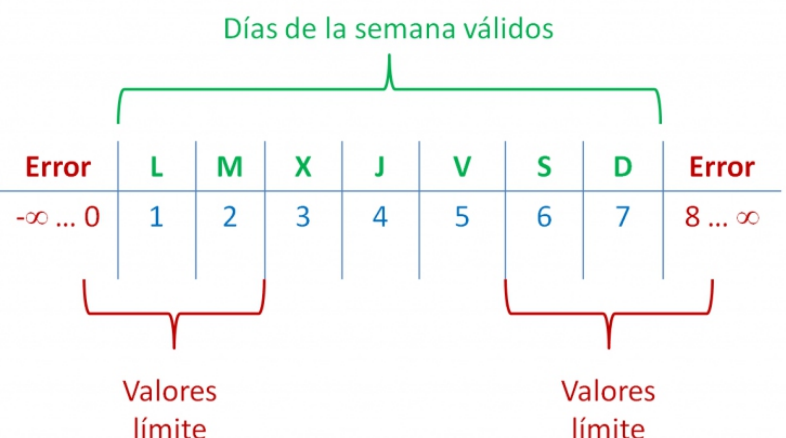


2 ANÁLISIS DE VALORES LÍMITES

Es una extensión de la técnica de partición de equivalencia que solo se puede usar cuando **la partición está ordenada**, y consiste en **datos numéricos o secuenciales**.

- Se deben identificar los valores límites mínimo y máximo (o valores inicial y final).
- Se pueden utilizar 2 o 3 valores límites.
- Para 2 valores límites se toma el valor que marca el límite (como valor que corresponde a la partición válida), y el valor anterior o posterior que corresponda a la partición de equivalencia inválida.
- Para 3 valores límites se toma el valor que marca el límite, un valor anterior y otro posterior a ese límite.
- La cobertura se mide de la siguiente manera:

Cobertura = $\frac{\text{Valores límites probados}}{\text{Valores límites identificados}}$



3 TABLA DE DECISIÓN

Esta técnica se utiliza para **pruebas combinatorias**, formadas por reglas de negocio complejas que un sistema debe implementar. Las características de esta técnica son:

- Se deben identificar las condiciones (entradas) y las acciones resultantes (salidas). Estas conforman las filas de la tabla.
- Las columnas de la tabla corresponden a reglas de decisión. Cada columna forma una combinación única de condiciones y la ejecución de acciones asociadas a esa regla.
- Los valores de las condiciones y acciones pueden ser valores booleanos, discretos, numéricos o intervalos de números.
- Ayuda a identificar todas las combinaciones importantes de condiciones y a encontrar cualquier desfase en los requisitos.
- La cobertura se mide de la siguiente manera:

Cobertura = $\frac{\text{Número de reglas de decisión probadas}}{\text{Número de reglas de decisión totales}}$

Condiciones	Reglas							
Condición 1	S	S	S	S	N	N	N	N
Condición 2	S	S	N	N	S	S	N	N
Condición 3	S	N	S	N	S	N	S	N
Acción 1	X	X						
Acción 2			X			X		X
Acción 3			X				X	
Acción 4				X				

4 TRANSICIÓN DE ESTADOS

Un diagrama de transición de estado muestra los **posibles estados del software**, así como la forma en que el software entra, sale y **realiza las transiciones** entre estados. Las características de esta técnica son:

- Una tabla de transición de estado muestra todas las transiciones válidas y las transiciones potencialmente inválidas entre estados, así como los eventos, las condiciones de guarda y las acciones resultantes para las transiciones válidas.
- Los diagramas de transición de estado, normalmente, sólo muestran las transiciones válidas y excluyen las transiciones no válidas.
- La prueba de transición de estado se utiliza para aplicaciones basadas en menús y es extensamente utilizada en la industria del software embebido. La técnica también es adecuada para modelar un escenario de negocio con estados específicos o para probar la navegación en pantalla.
- La cobertura se mide de la siguiente manera:

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

- Con esa información se valida que se pueda asignar a un cliente los distintos estados civiles como se muestran en el flujo, desde el inicio hasta el final (end-to-end).
- En este caso tendremos como mínimo seis casos de pruebas:

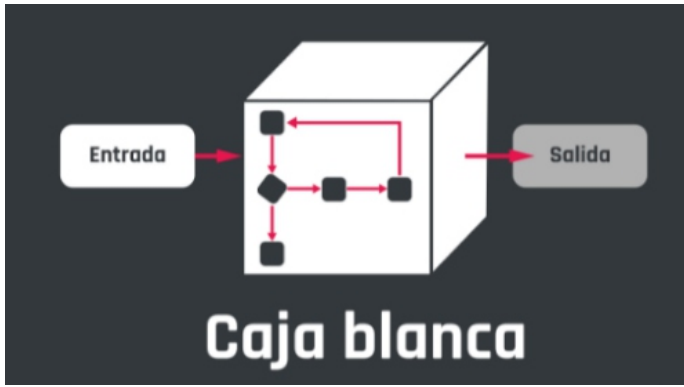
Estado 1	Estado 2	Estado 3	Estado 4	Estado 5	Estado Final
nacido	soltero	con hijo/s			con hijo/s
nacido	soltero	casado	con hijo/s		con hijo/s
nacido	soltero	casado	viudo	con hijo/s	con hijo/s
nacido	soltero	casado	viudo	casado	casado
nacido	soltero	casado	divorciado	con hijo/s	con hijo/s
nacido	soltero	casado	divorciado	casado	casado

CLASIFICACIÓN

TÉCNICAS DE CAJA BLANCA

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.

- Las condiciones de la prueba, casos y datos de prueba se deducen de una base de prueba que puede incluir código, arquitectura de software, **diseño detallado o cualquier otra fuente de info relacionada a la estructura de software**.
- La cobertura se mide en base a los elementos probados dentro de la estructura seleccionada (ej, el código o las interfaces).
- Las especificaciones se utilizan a menudo como fuente adicional de información para detectar el resultado esperado de los casos de prueba.



Caja blanca

1 PRUEBA Y COBERTURA DE SENTENCIA.

2 PRUEBA Y COBERTURA DE DECISIÓN.

TÉCNICAS BASADAS EN LA EXPERIENCIA

aprovechan el conocimiento de desarrolladores, probadores y usuarios para diseñar, implementar y ejecutar las pruebas.

1 PREDICCIÓN DE ERRORES

Esta técnica se utiliza para **anticipar la ocurrencia de equivocaciones, defectos y fallos** basados en el conocimiento del probador. Se crea una lista teniendo en cuenta:

- Cómo ha funcionado la aplicación en el pasado.
- Equivocaciones comunes en los desarrolladores.
- Fallos en aplicaciones relacionadas.

En base a esa lista se diseñan pruebas que expongan esos fallos y defectos.

2 PRUEBA EXPLORATORIA

En esta técnica se **diseñan, ejecutan, registran y evalúan de forma dinámica pruebas informales durante la ejecución de la prueba**. Los resultados de estas pruebas se utilizan para aprender más sobre el funcionamiento del componente o sistema. Generalmente se utilizan para complementar otras técnicas formales o cuando las especificaciones son escasas, inadecuadas o con restricciones de tiempo.

3 PRUEBAS BASADAS EN LISTAS DE COMPROBACIÓN

En esta técnica se diseñan, implementan y ejecutan **casos de prueba que cubren las condiciones que se encuentran en una lista de comprobación definida**. Se crean basadas en la experiencia y conocimiento de lo que el probador cree que es importante para el usuario y se utilizan debido a la falta de casos de prueba detallados. Durante la ejecución puede haber cierta variabilidad, dependiendo de quién ejecuta la prueba y condiciones del contexto. Esto da lugar a una mayor cobertura. Se utiliza tanto en pruebas funcionales como no funcionales.

CAJA NEGRA VS CAJA BLANCA

Criterio de Comparación	Caja Negra	Caja Blanca
Base De Prueba	<ul style="list-style-type: none">• Requisitos de software• Especificaciones,• Casos de uso• Historias de usuario.	<ul style="list-style-type: none">• Arquitectura de software• Diseño detallada de software• Cualquier otra fuente de información relacionada a la estructura de software.
Uso de Especificaciones de Software	Son una fuente principal de consulta para el diseño de estos casos de prueba.	Se utilizan a menudo como una fuente adicional de información para detectar el resultado esperado de los casos de prueba.
Cobertura	Se mide en función de los elementos probados en la base de prueba y de la técnica aplicada a la base de prueba.	Se mide en base a los elementos probados dentro de la estructura seleccionada (por ejemplo, el código o las interfaces).