

Testing I

Profesor: Daniel Tejerina

Módulo 2: Diseño e Implementación de la Prueba

Sesión 4: Diseño de la Prueba

Octubre 26 de 2021

Caso de prueba

Un caso de prueba es un conjunto de acciones que se ejecutan para verificar una característica o funcionalidad particular de una aplicación de software. Todas las características de una aplicación de software van a ser representadas por uno o más casos de prueba.

Un caso de prueba es un documento que proporciona información escrita sobre qué y cómo probar.

Características de un buen caso de prueba

- Deben ser simples
- El título debe ser fuerte
- Tener en cuenta al usuario final
- No asumir
- Asegurar la mayor cobertura posible
- Autonomía
- Evitar la repetición de casos de prueba

¿Qué debe contener un caso de prueba?

- Identificador
- Nombre del caso de prueba (conciso)
- Descripción
- Precondición
- Pasos
- Resultados esperados

Testing positivo (+)

Son aquellos casos de prueba que validan el flujo normal de un sistema bajo prueba. Es decir, están asociados a los requisitos funcionales.

Testing negativo (-)

Son aquellos casos de prueba que validan flujos no contemplados dentro de los requisitos del sistema.

Happy path testing

Es el único camino con el que se prueba una aplicación a través de escenarios de prueba cuidadosamente diseñados, que deberían recorrer el mismo flujo

que realiza un usuario final cuando usa la aplicación de manera regular. Es una prueba positiva. Su propósito no es encontrar defectos.

Casos de uso y Casos de prueba

Casos de uso

Un caso de uso cuenta la historia de cómo un usuario interactúa con un sistema de software para lograr o abandonar un objetivo. Cada caso de uso puede contener múltiples rutas que el usuario sigue, estos caminos son denominados escenarios de caso de uso.

Casos de prueba

Un caso de prueba cubre el software con mayor detalle que el caso de uso. Estos incluyen todas las funciones que el programa es capaz de realizar y deben tener en cuenta el uso de todo tipo de datos de entrada/salida, cada comportamiento esperado y todos los elementos de diseño.

Se puede comenzar escribiendo casos de prueba para el "escenario principal" y luego para "escenarios alternativos". Es decir, se escribe uno o más casos de prueba por cada escenario de caso de uso.

Pruebas de casos de uso

Es una técnica de caja negra donde se verifica si la ruta utilizada por el usuario está funcionando según lo esperado o no. Se pueden crear casos de prueba para cada comportamiento detallado en los casos de uso: comportamiento básico o normal, excepcional o alternativo, y de tratamiento de errores.

Sincrónico

Id	Título / Nombre	Descripción	Precondición	Pasos		
				#	Acción	Resultado esperado
TC_Login_001	Login exitoso con credenciales validas	Login exitoso con credenciales validas, usuario con dominio gmail y contraseña alfanumerica.	TC_Register_001 - Registrar un usuario.	1.	ingresar url : https://playground.digitalhouse.com/login	se visualiza pagina de login de pg
				2.	completar campo mail con usuario valido, dominio gmail.	en el campo mail se visualiza el mail in
				3.	completar campo contraseña con contraseña alfanumerica.	en el campo contraseña no se visualiza ingresada, en su lugar se ve los puntos el largo de la contraseña ingresada.
				4.	click en boton 'Ingresar'	se visualiza pagina de pg desde perfil c

Plantilla Casos de uso



EJEMPLOS DE PLANTILLAS DE CASOS DE USO

RF- 01	Alta de socio																
Objetivos asociados	Obj-02 Gestionar las socios																
Requisitos asociados	RI-02 Información sobre socios																
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite su ingreso como socio																
Precondición	El solicitante no es un socio del video-club y tiene su documentación disponible																
Secuencia Normal	<table><tr><th>Paso</th><th>Acción</th></tr><tr><td>1</td><td>El empleado del video-club solicita al sistema comenzar el proceso de alta de un nuevo socio</td></tr><tr><td>2</td><td>El sistema solicita los siguientes datos del nuevo socio: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto</td></tr><tr><td>3</td><td>El empleado del video-club solicita los datos requeridos y la documentación al nuevo socio</td></tr><tr><td>4</td><td>El empleado del video-club comprueba que los datos del nuevo socio coinciden con los de la documentación aportada</td></tr><tr><td>5</td><td>El empleado del video-club proporciona los datos requeridos y solicita al sistema que los almacene</td></tr><tr><td>6</td><td>El sistema almacena los datos proporcionados, imprime el carnet de socio e informa al empleado del video club de que el proceso ha terminado con éxito</td></tr><tr><td>7</td><td>El empleado del video-club entrega el carnet al nuevo socio</td></tr></table>	Paso	Acción	1	El empleado del video-club solicita al sistema comenzar el proceso de alta de un nuevo socio	2	El sistema solicita los siguientes datos del nuevo socio: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto	3	El empleado del video-club solicita los datos requeridos y la documentación al nuevo socio	4	El empleado del video-club comprueba que los datos del nuevo socio coinciden con los de la documentación aportada	5	El empleado del video-club proporciona los datos requeridos y solicita al sistema que los almacene	6	El sistema almacena los datos proporcionados, imprime el carnet de socio e informa al empleado del video club de que el proceso ha terminado con éxito	7	El empleado del video-club entrega el carnet al nuevo socio
Paso	Acción																
1	El empleado del video-club solicita al sistema comenzar el proceso de alta de un nuevo socio																
2	El sistema solicita los siguientes datos del nuevo socio: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto																
3	El empleado del video-club solicita los datos requeridos y la documentación al nuevo socio																
4	El empleado del video-club comprueba que los datos del nuevo socio coinciden con los de la documentación aportada																
5	El empleado del video-club proporciona los datos requeridos y solicita al sistema que los almacene																
6	El sistema almacena los datos proporcionados, imprime el carnet de socio e informa al empleado del video club de que el proceso ha terminado con éxito																
7	El empleado del video-club entrega el carnet al nuevo socio																
Postcondición	El solicitante es socio del video-club y el saldo de su cuenta es 0																
Excepciones	<table><tr><th>Paso</th><th>Acción</th></tr><tr><td>4</td><td>Si la documentación aportada no es correcta, el empleado del video-club cancela la operación, a continuación este caso de uso termina</td></tr></table>	Paso	Acción	4	Si la documentación aportada no es correcta, el empleado del video-club cancela la operación, a continuación este caso de uso termina												
Paso	Acción																
4	Si la documentación aportada no es correcta, el empleado del video-club cancela la operación, a continuación este caso de uso termina																

Sesión 5: Niveles y tipos de prueba

Octubre 28 de 2021

Ciclo de vida de las pruebas de software (STLC)

- Planificación

Se definen los objetivos y el enfoque de la prueba dentro de las restricciones impuestas por el contexto. Documento de salida: Plan de prueba, general o por nivel de prueba.

- Seguimiento y control

El objetivo de esta actividad es reunir información y proporcionar retroalimentación y visibilidad sobre las actividades de prueba. Documento de salida: Informe de avance de la prueba.

- Análisis

Durante esta actividad se determina **qué probar**. Documento de salida: Contratos de prueba que contienen las condiciones de prueba.

- Diseño

Durante esta actividad se determina **cómo probar**. Documento de salida: Casos de prueba de alto nivel diseñados y priorizados.

- Implementación

Se complementan los productos necesarios para la ejecución de la prueba, incluyendo la secuenciación de los casos en procedimientos de prueba. Documentos de salida: Procedimientos y datos de prueba, Calendario de Ejecución, Test Suite.

- Ejecución

Documento de salida: Reporte de defectos, Informe de ejecución de pruebas.

- Conclusión

Documento de salida: Informe resumen de prueba, Lecciones aprendidas.

Niveles de pruebas

Prueba unitaria o de componente

Objetos de prueba:

- Componentes, unidades o módulos.
- Código y estructuras de datos
- Clases
- Módulos de base de datos

En general, el desarrollador que escribió el código, realiza la prueba de componente.

Pruebas de integración

Se centra en las interacciones entre componentes o sistemas.

Objetos de prueba:

- Subsistemas
- Bases de datos
- Infraestructura
- Interfaces
- Interfaces de Programación de Aplicaciones (API).
- Microservicios

En general, es responsabilidad de los testers.

Pruebas de sistema

Objeto de prueba:

- Aplicaciones
- Sistemas hardware/software
- Sistemas operativos
- Sistema sujeto a prueba (SSP)
- Configuración del sistema y datos de configuración

Los testers independientes, en general, llevan a cabo la prueba de sistema.

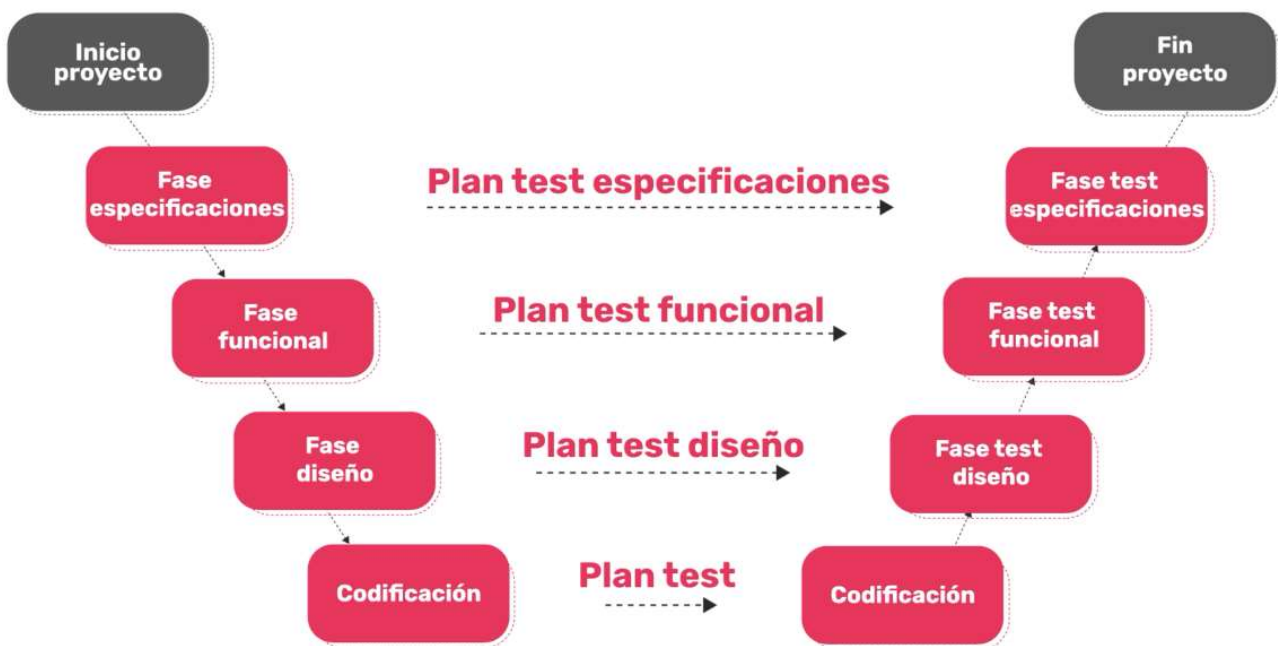
Prueba de aceptación

Objeto de prueba:

- Sistema sujeto a prueba
- Configuración del sistema y datos de configuración
- Procesos de negocio para un sistema totalmente integrado
- Sistemas de recuperación y sitios críticos -para pruebas de continuidad del negocio y recuperación de desastres.
- Procesos operativos y de mantenimiento
- Formularios
- Informes
- Datos de producción existentes y transformados

A menudo es responsabilidad de los clientes, usuarios de negocio, propietarios de producto u operadores de un sistema, y otros implicados.

Modelo V



Verificación

¿Se está construyendo el producto correctamente?

Validación

¿Se está construyendo el producto correcto?

Agile Testing

1. Iteraciones cortas definidas como sprints.
2. Pruebas dinámicas y continuas. Se crean escenarios de prueba usando un enfoque de Desarrollo Guiado por Comportamiento (BDD).
3. Se dividen las features en User Stories y se modelan escenarios de prueba para estas.
4. Se escriben los Unit Tests y luego se codifica la solución con un enfoque de Desarrollo Guiado por Pruebas (TDD).
5. Documentación ligera que prioriza la entrega de valor al cliente.
6. Se ejecutan pruebas continuamente y desde etapas tempranas, aplicando Integración Continua y Distribución Continua (CI/CD).
7. Toman protagonismo las Pruebas Automatizadas.
8. La Integración Continua permite tener una retroalimentación activa.

Tipos de pruebas

Pruebas funcionales

Incluyen pruebas que evalúan las funciones que el sistema debe realizar. Las funciones describen **qué hace** el sistema.

Aplica para todos los niveles de prueba.

Pruebas no funcionales

Prueban **qué tan bien** se comporta el sistema. Evalúan características como la usabilidad, la eficiencia del desempeño o la seguridad.

Aplica para todos los niveles de prueba.

Pruebas estructurales

Están basadas en la estructura interna del sistema o en su implementación. La estructura interna puede incluir código, arquitectura, flujos de trabajo y/o flujos de datos dentro del sistema.

Se pueden realizar en el nivel de componente y de integración.

Pruebas asociadas a los cambios

- **Pruebas de confirmación.** El objetivo es confirmar que el defecto original se ha solucionado de forma satisfactoria.

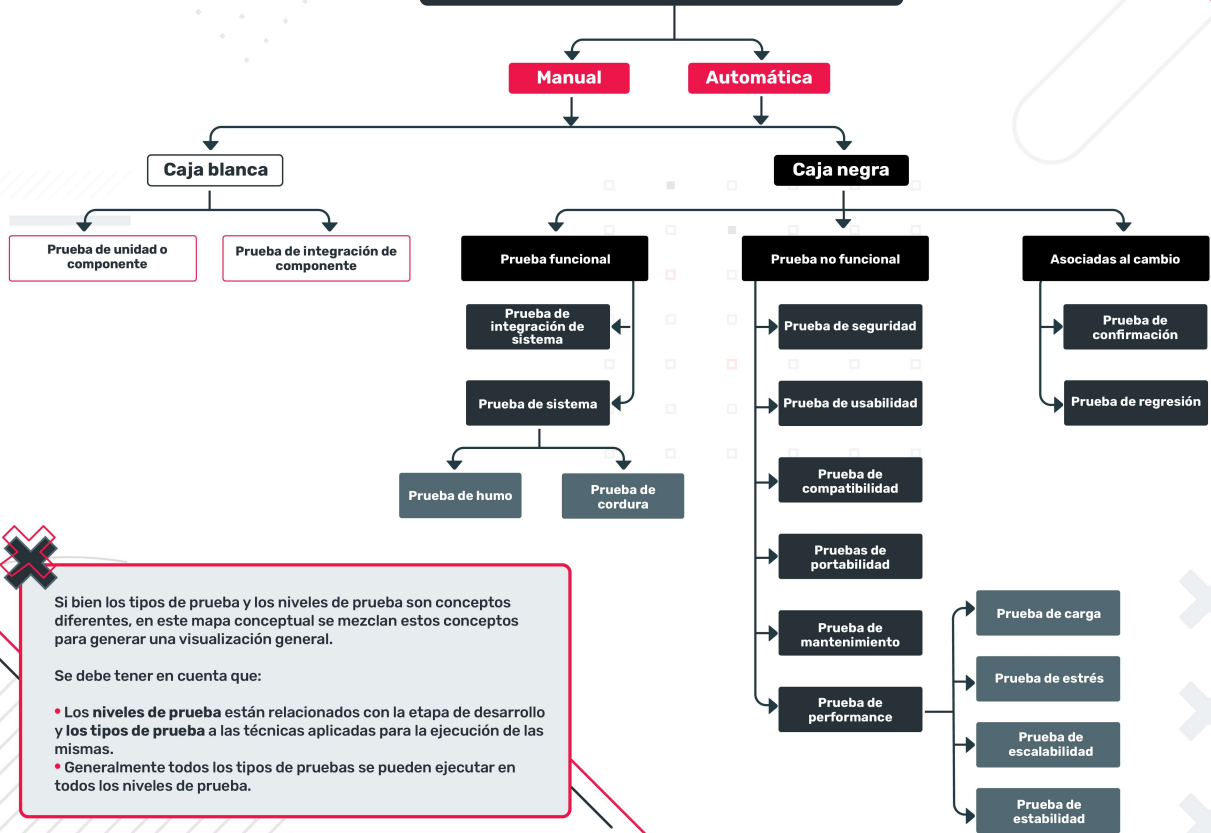
Se realizan en todos los niveles de prueba.

- **Pruebas de regresión.** implica la realización de pruebas para detectar posibles afectaciones accidentales al comportamiento de otras partes del código. Las pruebas de regresión son fuertes candidatos para la automatización.

Se realizan en todos los niveles de prueba.

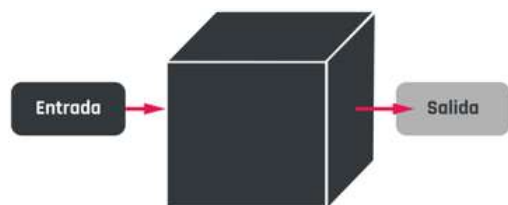
Niveles y tipos de prueba:

Niveles y tipos de prueba



Sesión 7: Técnicas de pruebas

Noviembre 2 de 2021

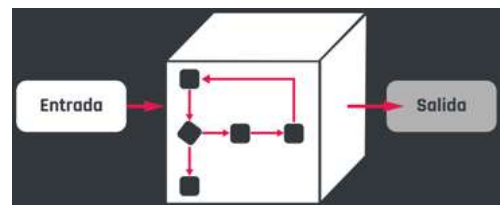


Caja negra

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 la implementación de los requisitos, 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 blanca

Las condiciones de la prueba, los casos de prueba y los datos de prueba se deducen de una base de prueba que puede incluir código, la arquitectura de software, el diseño detallado o cualquier otra fuente de información relacionada a la estructura de software.

La cobertura se mide en base a los elementos probados dentro de la estructura seleccionada (por ejemplo, el código o las interfaces).

Las especificaciones se utilizan a menudo como una fuente adicional de información para detectar el resultado esperado de los casos de prueba.

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.

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.

Técnicas basadas en la experiencia

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

Técnicas de caja negra

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.

Cobertura = Particiones probadas / Particiones identificadas

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.

Ej: Si una condición para un dato de entrada i especifica un rango de valores definido como $n \leq i \leq m$, los casos de prueba resultantes de aplicar el análisis de valores límite corresponden a $n-1$, n , $n+1$, $m-1$, m y $m+1$.

Cobertura = Valores límites probados / Valores límites identificados

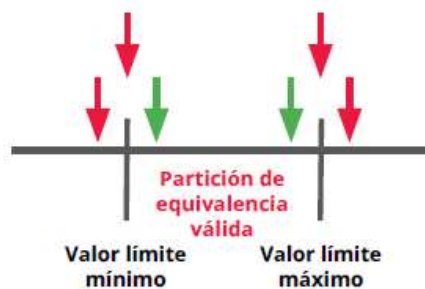


Tabla de decisión

Esta técnica se utiliza para pruebas combinatorias, formadas por reglas de negocio complejas que un sistema debe implementar. Esta técnica ayuda a identificar todas las combinaciones importantes de condiciones y a encontrar cualquier desfase en los requisitos.

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			

Cobertura = Reglas de decisión probadas / Reglas de decisión totales

Ejemplo:

Condiciones	1	2	3	4	5	6	7	8
Cliente de la empresa	S	S	N	N	S	N	-	N
Plazo de pago <= 30 días	S	N	N	-	S		N	-
Cantidad solicitada <= cantidad en stock	S	S	S	S	N	N	N	N
Acciones	1	2	3	4	5	6	7	8
Facturar con descuento	X	-	-	-	X	-	-	-
Facturar sin descuento	-	X	X	X	-	X	X	X
Enviar mercadería solicitada	X	X	X	X	-	-	-	-
Enviar cantidad en stock	-	-	-	-	X	X	X	X
Dejar pendiente cantidad solicitada-stock	-	-	-	-	X	X	X	X

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. 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.

Cobertura = Estados, transiciones probados / Estados, transiciones identificados

Técnicas basadas en la experiencia

Predicción de errores

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

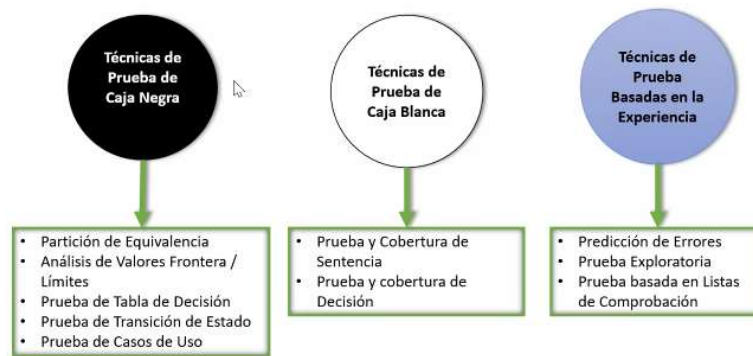
Prueba exploratoria

Se diseñan, ejecutan, registran y evalúan de forma dinámica pruebas informales durante la ejecución de la prueba. Generalmente se utilizan para complementar otras técnicas formales.

Prueba basada en listas de comprobación

Se diseñan, implementan y ejecutan casos de prueba que cubren las condiciones que se encuentran en una lista de comprobación definida.

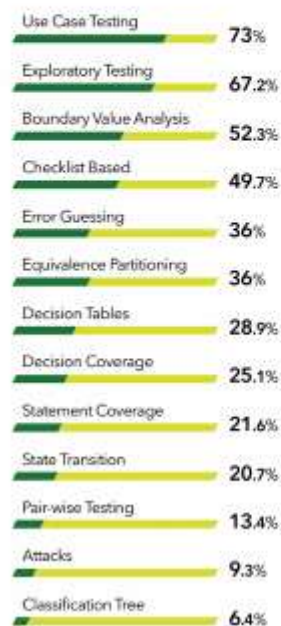
Sincrónico



Niveles y técnicas de testing en la industria

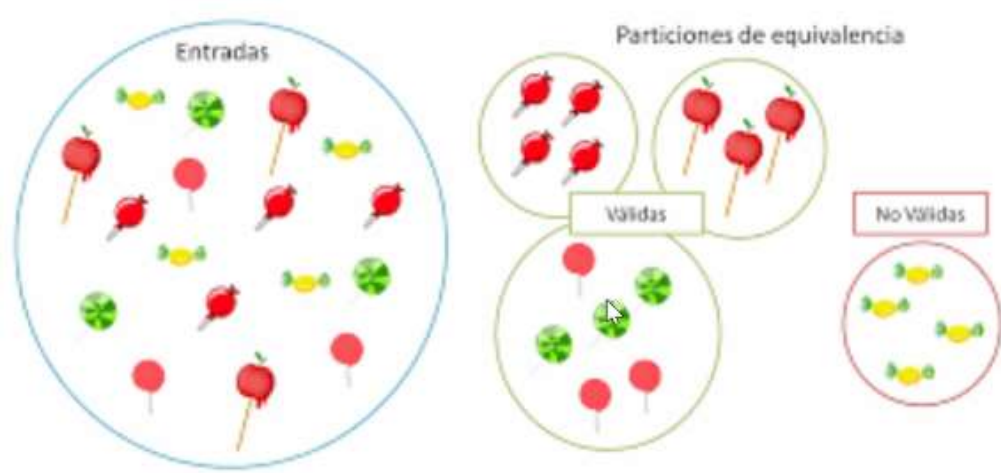
WHICH TEST TECHNIQUES ARE UTILIZED BY YOUR TESTING TEAM?

* Selecting multiple choices were available



The top five test techniques selected by survey respondents are the same as those selected by the survey respondents for the report conducted in 2015-2016. Beginning from the highest ranking technique, top five techniques are use case testing (73.0%), exploratory testing (67.2%), boundary value analysis (52.3%), checklist based (49.7%), and error guessing (36.0%). Compared to the previous report, only boundary value analysis and checklist based techniques switched their rankings. While the boundary value analysis technique moves up in the ranks and becomes the third ranked item, checklist based technique moves down in the ranks and becomes the fourth ranked item.

Partición de equivalencias



Sesión 8: Diseño e Implementación de la prueba

Noviembre 4 de 2021

Ejecución de las pruebas:

1. Registrar los objetos, herramientas y productos de prueba.
2. Ejecutar las pruebas
3. Comparar resultados obtenidos con resultados esperados.
4. Analizar las anomalías
5. Informar los defectos
6. Registrar el resultado de la ejecución
7. Repetir las actividades de prueba

Suites de casos de prueba

Pruebas de humo y pruebas de regresión

Las pruebas de humo se ejecutan para evaluar la **estabilidad** de las compilaciones de software iniciales o desarrolladas recientemente. Se ejecutan en las etapas iniciales del SDLC.

Las pruebas de regresión tienen la tarea de verificar y validar las funcionalidades existentes de la aplicación después de cada modificación o en la adición de nuevas funciones. En un proyecto de automatización, deberían ser las primeras candidatas.

Las pruebas de humo son previas a las de regresión.

Sesión 9.

Noviembre 5 de 2021

Sincrónico

Ejemplo de experiencia con pruebas de regresión:

Las pruebas de regresión manuales podían durar 3 semanas.

Las pruebas automatizadas duraron 4 horas.

Al tratar de realizar las pruebas en la nube, las pruebas automatizadas no funcionaron, y fue necesario hacerlas manualmente. El proyecto se atrasó 6 semanas (3 semanas de pruebas en AWS y 3 más en Azure)

Se logró armar un framework con Docker para lograr que la ejecución de los tests fueran independientes de la plataforma.