



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS  
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**CURSO:**  
**Métodos Formales de Pruebas - Testing de Software**

**Tema:**  
**Diseño de casos de prueba**  
**Semana 6**

**Docente: Mg. Wilder Inga**

# CICLO DE VIDA GENÉRICO DEL SOFTWARE

PLANIFICACIÓN

DISEÑO

CODIFICACIÓN

PRUEBAS

LIBERACIÓN  
DEL  
PRODUCTO

CIERRE

ESTRATEGIA  
DE PRUEBAS

DISEÑO DE  
PRUEBAS

Pruebas  
unitarias

Pruebas  
sistema

GESTIÓN DE  
PUESTA EN  
MARCHA

Pruebas  
integración

UAT

PRUEBAS DE RENDIMIENTO

PRUEBAS DE SEGURIDAD



# Diseño de casos de prueba

- Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados, desarrollado para conseguir un objetivo particular o condición de prueba. Ejemplo: verificar el cumplimiento de un requisito específico
- En la ingeniería de software, mediante el caso de prueba o test case el analista determinará, si una aplicación o una característica de esta es parcial o completamente satisfactoria.

# Diseño de casos de prueba

- Para desarrollar software de calidad y libre de errores, el plan de pruebas y los casos de prueba son muy importantes.
- Un caso de prueba bien diseñado tiene gran posibilidad de llegar a resultados más fiables y eficientes, mejorar el rendimiento del sistema, y reducir los costos en tres categorías:
  - ✓ 1) productividad — menos tiempo para escribir y mantener los casos—;
  - ✓ 2) capacidad de prueba —menos tiempo para ejecutarlos—; y
  - ✓ 3) programar la fiabilidad —estimaciones más fiables y efectivas— (Perry, 1995).

# Diseño de casos de prueba

- El proceso de escribir casos de prueba y establecer su estándar es un logro especial muy dinámico, y es necesario que se enseñe, aplique, controle, mida y mejore continuamente.
- Para llevar a cabo un caso de prueba, es necesario definir los siguientes pasos:
  - ✓ Definir escenarios
  - ✓ Identificar condiciones de entrada
  - ✓ Definir valores de entrada
  - ✓ Generar casos de prueba

# Diseño de casos de prueba

- Para llevar a cabo un caso de prueba, es necesario definir los siguientes pasos:
  - Definir escenarios
  - Identificar condiciones de entrada
  - Definir valores de entrada
  - Generar casos de prueba

# Definir escenarios

- Consiste en identificar todos los escenarios (caminos) a probar de un caso de uso: flujo básico, sub flujos y flujos alternativos.

# Identificar condiciones de entrada

Las condiciones de entrada son parte del dominio de valores de entrada. Se pueden identificar condiciones de entrada con estados válidos (V) y no válidas (NV); asimismo se consideran condiciones de entrada con el estado que no se aplica (N/A) para un determinado escenario.

Existen los siguientes tipos de condiciones de entrada:

- Miembro de un conjunto
- Lógico
- Valor
- Rango



# Identificar condiciones de entrada

Ejemplo: Considérese una aplicación bancaria, donde el usuario puede conectarse al banco por Internet y realizar una serie de operaciones bancarias. Una vez accedido al banco con las siguientes medidas de seguridad (clave de acceso), la información de entrada del procedimiento que gestiona las operaciones concretas a realizar por el usuario requiere las siguientes entradas:

- **Código de banco:** En blanco o número de tres dígitos el primer debe ser mayor que 1.
- **Código de sucursal:** Número de cuatro dígitos, el primero de ellos mayor a 0.
- **Número de cuenta:** Número de cinco dígitos.
- **Clave personal:** Valor alfanumérico de cinco posiciones.
- **Orden:** Este valor se seleccionará de una lista desplegable, según la orden que se desee realizar. Puede estar en "Seleccione Orden" o una de las dos cadenas siguientes: "Talónario" o "Movimientos"
- En el caso "Talónario" el usuario recibirá un talonario de cheques, mientras que en "Movimientos" recibirá los movimientos del mes en curso. Si no se especifica este dato, el usuario recibirá los dos documentos.

# Identificar condiciones de entrada

A continuación, se muestra una tabla con estados de las condiciones de entrada por cada resultado esperado.

ID CP	Escenario	CONDICIONES DE ENTRADA					Resultado esperado
		Código de banco	Código de sucursal	Número de cuenta	Clave personal	Orden	
CP1	Escenario 1	V	V	V	V	V	Mensaje "Envío de talonarios"
CP2	Escenario 1	V	V	V	V	V	Mensaje "Envío de movimientos "
CP3	Escenario 1	V	V	V	V	V	Mensaje "Envío de talonarios y movimientos"
CP4	Escenario 2	NV	V	V	V	V	Mensaje "Código de banco incorrecto"
CP5	Escenario 3	V	NV	V	V	V	Mensaje "Código de sucursal incorrecto"
CP6	Escenario 4	V	V	NV	V	V	Mensaje "Número de cuenta incorrecto"
CP7	Escenario 5	V	V	V	NV	V	Mensaje "Clave incorrecta"

# Definir valores de entrada

Pueden usarse varias técnicas para identificar los valores de los datos de entrada, la técnica de particiones o clases de equivalencias es una de ellas.

Las clases de equivalencia se identifican examinando cada condición de entrada (normalmente una frase en la especificación) y dividiéndola en dos o más grupos.

Se definen dos tipos de clases de equivalencia:

- Clases Válidas: Entradas válidas al programa.
- Clases no Válidas: Valores de entrada erróneos.

# Definir valores de entrada - ejemplo

- A continuación, se muestra las clases de equivalencia para el caso de gestión bancaria anterior.

Sec.	Condición de Entrada	Tipo	Clases Válidas		Clases No Válidas	
			Entrada	Código	Entrada	Código
1	Código de banco	Lógico (puede estar o no)	En blanco	CEV<01>	Un valor no numérico	CENV<01>
		Si está, es Rango	200 <= Código de banco <= 999	CEV<02>	Código de banco < 200	CENV<02>
					Código de banco > 999	CENV<03>
2	Código de sucursal	Rango	1000 <= Código de sucursal <=9999	CEV<03>	Código de sucursal < 1000	CENV<04>
					Código de sucursal > 9999	CENV<05>
3	Número de cuenta	Valor	Cualquier número de 5 dígitos	CEV<04>	Número de más de cinco dígitos	CENV<06>
					Número de menos de cinco dígitos	CENV<07>
4	Clave personal	Valor	Cualquier cadena de caracteres alfanuméricos de 5 posiciones	CEV<05>	Cadena de más de cinco posiciones	CENV<08>
					Cadena de menos de cinco posiciones	CENV<09>
5	Orden	Miembro de un conjunto, con comportamiento distinto	Orden = "Seleccione Orden"	CEV<06>		
			Orden = "Talónario"	CEV<07>		
			Orden = "Movimientos"	CEV<08>		

# Generar casos de prueba

En esta última etapa, se generan los casos de pruebas. Para ello, se considera como referencia la tabla de condiciones de entrada, indicando en cada caso de prueba las clases de equivalencia creadas.

Por ejemplo, para el caso bancario se tendría lo siguiente:

ID CP	Clases de equivalencia	CONDICIONES DE ENTRADA					Resultado esperado
		Código de banco	Código de sucursal	Número de cuenta	Clave personal	Orden	
CP1	CEV<02>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	200	1000	10000	Aaaaa	"Talonario"	Mensaje "Envío de talonarios"
CP2	CEV<01>, CEV<03>, CEV<04>, CEV<05>, CEV<08>	820	9999	99999	Zzzzz	"Movimientos"	Mensaje "Envío de movimientos "
CP3	CEV<02>, CEV<03>, CEV<04>, CEV<05>, CEV<06>	999	1001	12345	A1b2c	"Seleccione Orden"	Mensaje "Envío de talonarios y movimientos"
CP4	CENV<01>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	30A	1989	12345	1a2b3	"Seleccione Orden"	Mensaje "Código de banco incorrecto"
CP5	CENV<04>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	999	12345	1a2b3	"Seleccione Orden"	Mensaje "Código de sucursal incorrecto"
CP6	CENV<07>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	1989	123	1a2b3	"Seleccione Orden"	Mensaje "Número de cuenta incorrecto"
CP7	CENV<09>, CEV<03>, CEV<04>, CEV<05>, CEV<07>	210	1989	12345	---	"Seleccione Orden"	Mensaje "Clave incorrecta"

# Componentes de un caso de prueba

Un caso de prueba es un conjunto de acciones con resultados y salidas previstas basadas en los requisitos de especificación del sistema; sus componentes son:

1. Propósito: de la prueba o descripción del requisito que se está probando
2. Método: o forma como se probará
3. Versión: o configuración de la prueba, versión de la aplicación en prueba, el hardware, el software, el sistema operativo, los archivos de datos, entre otros
4. Resultados: acciones y resultados esperados o entradas y salidas
5. Documentación: de la prueba y sus anexos.

# Componentes de un caso de prueba

En cada nivel de la prueba, estos componentes deben probarse utilizando casos de prueba para pruebas de unidad, pruebas de integración, pruebas del sistema, pruebas Alpha y Beta,..., además, son útiles para las pruebas de rendimiento, pruebas funcionales y pruebas estructurales.

# Factores de calidad de los casos de prueba

Un caso de prueba debe cumplir con los siguientes factores de calidad:

1. Correcto. Ser apropiado para los probadores y el entorno. Si teóricamente es razonable, pero exige algo que ninguno de los probadores tiene, se caerá por su propio peso.
2. Exacto. Demostrar que su descripción se puede probar.
3. Económico. Tener sólo los pasos o los campos necesarios para su propósito.
4. Confiable y repetible. Ser un experimento controlado con el que se obtiene el mismo resultado cada vez que se ejecute, sin importa qué se pruebe.
5. Rastreadable. Saber qué requisitos del caso de uso se prueban.
6. Medible. Este es un ejercicio muy útil para quienes escriben pruebas, para verificar constantemente dónde están, si pierden alguno de los elementos, o si no se cumple un estándar.



# Formato de los casos de prueba

1. Paso a paso. Este formato se utiliza en:

- La mayoría de las reglas de procesamiento
- Casos de prueba únicos y diferentes Interfaces GUI
- Escenarios de movimiento en interfaces diferentes
- Entradas y salidas complicadas para representar en matrices.

# Formato de los casos de prueba

2. Matrices. Sus usos más productivos son:

- Formularios con información muy variada, mismos campos, valores y archivos de entrada diferentes
- Mismas entradas, diferentes plataformas, navegadores y configuraciones
- Pantallas basadas en caracteres
- Entradas y salidas con mejor representación matricial.

# Formato de los casos de prueba

- 3. Scripts automatizados. La decisión de utilizar software para automatizar las pruebas depende de la organización y del proyecto que se esté probando. Existen algunas cuestiones técnicas que deben concretarse y que varían de una herramienta a otra. El beneficio real de automatizar las pruebas se obtiene en la fase de mantenimiento del ciclo de vida del software; en ese momento, los scripts se pueden ejecutar repetidamente, incluso sin supervisión, y el ahorro en tiempo y dinero es sustancial (Moller and Paulish, 1993)

# Formato de los casos de prueba

- Un caso de prueba paso a paso tiende a ser más verbal, y el de matrices más numérico. A menudo, la ruta más productiva es utilizarlos todos. Los dos primeros se utilizan para las pruebas unitarias, de integración y del sistema; y el automatizado, para pruebas de regresión (Voas, 1993).

# Mitos y realidades de los casos de prueba

Mito	Los casos de prueba paso a paso toman mucho tiempo para escribirse. No lo podemos permitir.
Realidad	Puede o no que tomen más tiempo para escribirse, pero su detalle los hace resistentes y fáciles de mantener; además, son necesarios para probar adecuadamente algunas de las funciones.
Mito	Una matriz es siempre la mejor opción. Hagámosla trabajar.
Realidad	Un problema persistente es armar una matriz con la información adecuada de la configuración. Frecuentemente se omite dicha información, o peor aún, si las configuraciones o clases de entrada son diferentes no se pueden forzar dentro de una matriz como grupo similar, ya que no se han probado todos.
Mitos	La alta tecnología es la mejor. Si es posible automatizar los casos de prueba, se debe hacer.
Realidad	La decisión de utilizar pruebas automatizadas debe basarse en muchos factores.
Mito	No tenemos tiempo para escribir los casos de prueba manuales. Vamos a automatizarlos.
Realidad	Automatizar los casos de prueba toma más tiempo que los otros dos tipos.

# Plantilla para casos de prueba

Proyecto No.: Nombre del Proyecto:	Página No.:
Caso No.: Nombre del Caso:	Ejecución No.: Nombre: Estado de la prueba:
Marca/Subsistema/Módulo/Nivel/Función/Código de la Unidad bajo prueba:	Requisito No.: Nombre:
Escrito por: Fecha:	Ejecutado por: Fecha:
Descripción del caso de prueba (propósito y método):	
Configuración de la prueba para (H/W, S/W, N/W, datos, pre-requisitos de prueba, seguridad y tiempo):	

Paso	Acción	Resultados esperados	Pasado/Fallido
1			
2			
...			



# Checklist para la calidad de un caso de prueba

Atributo	Lista de chequeo	S/N
Calidad	Correcto. Es apropiado para los probadores y el entorno	
	Exacto. Su descripción se puede probar	
	Económico. Tiene sólo los pasos o los campos necesarios para su propósito	
	Confiable y repetible. Se obtiene el mismo resultado sin importa qué se pruebe	
	Rastreable. Se sabe qué requisito se prueba	
	Medible. Retorna al estado de la prueba sin valores en su estado	
Estructura y capacidad de prueba	Tiene nombre y número	
	Tiene un propósito declarado que incluye qué requisito se está probando	
	Tiene una descripción del método de prueba	
	Especifica la información de configuración –entorno, datos, pre-requisitos de prueba, seguridad–	
	Tiene acciones y resultados esperados	
	Guarda el estado de las pruebas, como informes o capturas de pantalla	
	Mantiene el entorno de pruebas limpio	
	No supera los 15 pasos	
	La matriz no demora más de 20 minutos para probarse	
	El <i>script</i> automatizado tiene propósitos, entradas y resultados esperados	
	La configuración ofrece alternativas a los pre-requisitos de la prueba cuando es posible	
	El escenario de aplicación se relaciona con otras pruebas	
Administración de la configuración	Emplea convenciones de nomenclatura y numeración	
	Guarda en formatos especificados los tipos de archivo	
	Su versión coincide con el software bajo prueba	
	Incluye objetos de prueba necesarios para el caso, tales como bases de datos	
	Almacena con acceso controlado	
	Realiza copias de seguridad en red	
	Archiva por fuera del sitio	



# Actividad - Individual

Diseño de pruebas Partiendo de los grupos establecidos. Realizar:

- Cada integrante escoge dos requisitos del proyectos del curso (definidos o por definir).
- Definir escenarios para los dos requisitos
- Identificar condiciones de entrada
- Definir datos de entrada usando Clases de Equivalencia (revise el video de apoyo en Classroom).
- Realizar casos de prueba. Mínimo 5 por cada requisito.