



CURSO

# SOFTWARE TESTING

UNIDAD 3 – Diseño de Casos de Prueba (TC)

# PRESENTACIÓN

---

En esta unidad vamos aprender sobre como diseñar casos de prueba (Test Cases)

# TEMARIO

1. – Principios del Testing
2. – Características de un Test Case
3. – Diseño de Test Cases.
4. – Ejemplos.

# Principios del Testing

- Principio 1 - La prueba muestra la presencia de defectos
  - La prueba puede mostrar que los defectos están presentes, pero no puede probar que no hay defectos. La prueba reduce la probabilidad de los defectos no descubiertos restantes en el software pero, incluso si no se encuentran defectos, no es una prueba de corrección.
- Principio 2 - La prueba exhaustiva es imposible
  - Probar todo (todas las combinaciones de entradas y de precondiciones) no es factible a excepción de casos triviales. En vez de la prueba exhaustiva, usamos el riesgo y las prioridades para enfocar los esfuerzos de prueba.
- Principio 3 - Prueba temprana
  - Las actividades de prueba deberían comenzar tan pronto como sea posible en el ciclo de vida del desarrollo del software o del sistema y deberían estar enfocados en los objetivos definidos.

# Principios del Testing

- Principio 4 - Agrupamiento de defectos
  - Un número pequeño de módulos contiene la mayoría de los defectos.
- Principio 5 - Paradoja del pesticida (eficacia de las pruebas)
  - Las mismas pruebas se repiten una y otra vez, eventualmente el mismo conjunto de casos de prueba no encontrará más cualquier nuevo bug. Para superar esta "paradoja del pesticida", los casos de prueba necesitan ser repasados y revisados regularmente, y nuevos y diversos necesitan ser escritos al ejercicio diferente partes del software o del sistema potencialmente para encontrar más defectos.

# Principios del Testing

- Principio 6 - La prueba es dependiente del contexto
  - La prueba se hace diferentemente en diversos contextos. Por ejemplo, se prueba el software de seguridad crítica diferentemente de un sitio de comercio.
- Principio 7 - Falacia de la ausencia de errores
  - Encontrar y arreglar defectos no ayuda si el sistema construido es inutilizable y no cumple las necesidades y expectativas de los usuarios.

## Caso de Prueba - Ejemplo

Release									
Fecha									
Analista/Tester									
Caso Uso Referencia									
Proposito:	probar el circuito de login del sitio								
Precondiciones:	tener instalado google chrome								
Num. Acción	Usuario	Escenario	Acciones	Precondicio	Resultado Esperado	Resultado Obtenido	Observacion	Estado	Id. Ticket Defec
1			tippear en el navegador <b>cine.ar</b> y dar enter		aparece la pantalla del sitio	permite visualizar		ejecuta ok	
2			ingresar a la opcion de menu <b>cine.ar play</b>		se visualiza la pantalla de bienvenida	permite visualizar		ejecuta ok	
3			seleccionar el boton <b>iniciar sesion</b>		aparece pantalla de login, para ingresar email y password, y opciones para registrarse y recuperar contraseña	permite visualizar		ejecuta ok	
		Registrar nuevo usuario							
4			Ingresar a la opcion <b>Registrate Ahora</b> , Completar el formulario de registro con datos validos, Activar la cuenta desde el link del mail, seleccionar el boton iniciar sesion y loguearse con el usuario y password registrados		Envia correo y permite confirmarlo, luego permite inicio de sesion y propone administrar perfiles				
5			Ingresar a la opcion <b>Registrate Ahora</b> , Completar el formulario de registro con datos invalidos 1) cargando distintos tipos de datos a los esperados, 2) no tildando opciones obligatorias, 3) no utilizando el captcha, poniendo passwords distintas, 4) escribiendo mail sin		Debe mostrar el error correspondiente y no permitir la confirmación del usuario				
		Ingresar al sistema (Login)							
6			Ingresar a la opcion <b>Iniciar sesion</b> , cargar mail y password correctos y elegir <b>Ingresar</b>		Regresa a la pagina principal y aparece la sesion iniciada en el borde superior derecho de la pantalla				
7			Ingresar a la opcion <b>Iniciar sesion</b> , 1) cargar mail y password incorrectos, 2) cargar mail y passwords correctos sin haber confirmado el mail, 3) poner email correcto pero password incorrecta 4) presionar tecla escape y caracteres		No debe permitir el login del usuario				
		Recuperar contraseña							



## Test Cases de un sistema de facturación

- Que permita identificar al cliente
- Que permita cargar artículos, cantidad y precio
- Que permita ver el importe total de la operación con sus impuestos
- Que permita guardar el comprobante



## Características de los TC

### **Testeable:**

Debe ser independiente del autor y del ejecutor, debe ser claro, no hay que leer otro documento para entenderlo.

### **Completo:**

No debe asumir cosas que debe hacer el ejecutor del caso de prueba, tampoco debe tener información innecesaria

### **Reutilizable:**

Los casos de prueba deberían poder ser reutilizables para la prueba de otra funcionalidad

## Características de los TC

Formalmente los casos de prueba escritos consisten principalmente en tres partes con subdivisiones:

- a. **Visión general:** contiene información general acerca de los Casos de Prueba.
- b. **Actividades** (de los casos de prueba)
- c. **Resultados**

## Diseño de los TC – Visión General

1. **Identificador (ID):** es un identificador único para futuras referencias, por ejemplo, mientras se describe un defecto encontrado.
2. **Caso de prueba dueño/creador** es el nombre del analista o diseñador de pruebas, quien ha diseñado pruebas o es responsable de su diseño.
3. **Versión** la actual definición del caso de prueba (ya que podría modificarse en el futuro).
4. **Nombre**
  - Debe tener un título entendible por cualquier persona
  - Debe contener las palabras clave que faciliten su búsqueda
  - No deben existir dos casos de prueba con la misma descripción

## Diseño de los TC – Visión General

5. **Identificador de requerimientos** el cuál está incluido por el caso de prueba. También aquí puede ser identificador de [casos de uso](#) o [especificación funcional](#). (REQUISITOS)
6. **Propósito** contiene una breve descripción del propósito de la prueba, y la funcionalidad que chequea.
  - Ejemplo: Permitir login de usuario

Funcionalidad a chequear:

  - De un usuario existente pedir ingreso de password
  - Validar la coincidencia de la password ingresada y la asignada al usuario.
7. **Dependencias:** si el caso de test depende de la ejecución anterior de otro caso de prueba o proceso, aquí se indica el mismo.

## Diseño de los TC – Visión General

**8. Prioridad:** indica qué tan importante es la realización del caso de prueba. La clasificación se realiza luego del análisis de riesgo.

- ALTA: Son los que nos aseguran que la aplicación se puede usar. Generalmente imposibilitan realizar otras pruebas. Ejemplo:
  - Desc.: Login. Usuario correcto. Contraseña correcta. Ingresar
  - Resul. Esperado: Se ingresa al sistema y se muestra la pantalla de usuarios
- MEDIA: Condiciones alternativas. Ejemplo:
  - Desc.: Login. Usuario correcto. Contraseña incorrecta. Ingresar
  - Resul. Esperado: No se puede ingresar. El sistema muestra un mensaje indicando que el usuario o la contraseña son incorrectos.
- BAJA: Agregan calidad al producto. No son bloqueantes. Ejemplo:
  - Desc.: Login. Ortografía y gramática
  - Resul. Esperado: La ortografía y gramática de la página de ingreso es correcta.

## Diseño de los TC – Actividades

- 1. Ambiente de prueba/configuración** contiene información acerca de la configuración del hardware o software en el cuál se ejecutará el caso de prueba.  
Ejemplo: La aplicación se ejecuta sobre un browser (Chrome, Edge, Opera, etc), la ruta de acceso es <http://192.168.1.190/index>  
Luego utilizar el botón de LOGIN que lleva a la ruta <http://192.168.1.190/login>
- 2. Inicialización** (PRECONDICIONES) describe acciones, que deben ser ejecutadas antes de que los casos de prueba se hayan inicializado. Por ejemplo, debemos abrir algún archivo.  
Ejemplo: Conexión VPN con el server, utilizar un usuario con permisos de acceso al server.  
Es necesario tener instalado en el equipo un browser.  
Debemos conocer datos preexistente ya cargados de usuarios y sus claves de acceso.

## Diseño de los TC – Actividades

- 3. Finalización** describe acciones, que deben ser ejecutadas después de realizado el caso de prueba. Por ejemplo si el caso de prueba estropea la base de datos, el analista debe restaurarla antes de que otro caso de prueba sea ejecutado.
- 4. Acciones** pasos a realizar para completar la prueba (PASO A PASO). El primer paso debería identificar el usuario / rol con el que se ingresa al sistema, a menos que esto ya se encuentre definido como 'Pre condición'. Ejemplo:
- 1) Ingresar en la ruta de la pantalla de login
  - 2) Ingresar un usuario correcto y contraseña correcta
  - 3) Presionar el botón 'Ingresar'
- En cada paso también se puede incluir el resultado esperado de cada acción.



## Diseño de los TC – Actividades

5. Descripción de los **datos de entrada** (VALORES DE ENTRADA). Se especifica un dominio de datos con los que se debe / puede realizar la prueba.

Ejemplo:

Usuario [Cadena de caracteres]: juanito12

Contraseña [Cadena de caracteres]: pepe123

Si el dominio de datos es específico, es obligatorio indicar el lote de datos.

## Diseño de los TC – Resultados

- 1. Resultados esperados** contiene una descripción de lo que el analista debería ver tras haber completado todos los pasos de la prueba. Ejemplo:  
El sistema presenta un mensaje de bienvenida y lleva a otra pantalla que posee un menú para el usuario
- 2. Resultados reales** contienen una breve descripción de lo que el analista encuentra después de que los pasos de prueba se hayan completado. Esto se sustituye a menudo con un **Correcto/Fallido**. Si un caso de prueba falla, frecuentemente la referencia al defecto implicado se debe enumerar en esta columna.  
Ejemplo:  
OK: El sistema se comporta según lo esperado  
NO OK: El sistema arroja un error luego de presionar ‘ingresar’. Ver defecto 1122

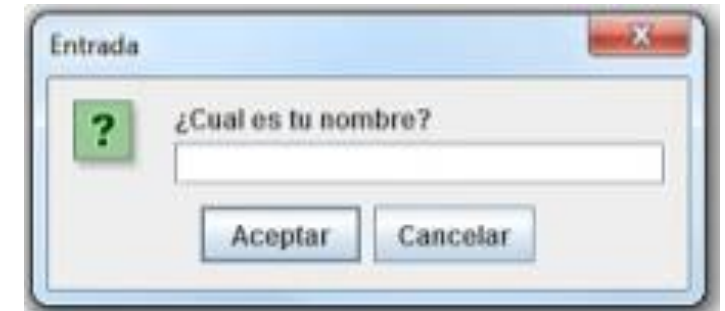
## Tipos de Entradas



- Cadenas
- Números
- Fechas
- Listas desplegables
- Radio Button
- Check Button
- Importar Archivos
- Desde Base de Datos
- Desde Web

## Tipos de Entradas - Cadenas

- Cadena vacía
- Cadena consistente únicamente de un espacio en blanco
- Cadena que empieza o termina con un espacio en blanco
- Cadena con carácter especiales como escape, numeral, mayor y menor, etc.
- Cadena con valores en mayúsculas y minúsculas
- Cadena con pocos caracteres
- Cadena con el máximo de caracteres permitidos
- Cadena con inyección de código SQL (por ej. " or 1=1 ")
- Cadena con caracteres extranjeros (tilde, ñ, etc)



## Tipos de Entradas - Números

- Campo vacío
- Cero
- Números fuera del rango
- Números negativos
- Comienzo con cero
- Valores decimales
- Sintácticamente inválidos (incluir letras)



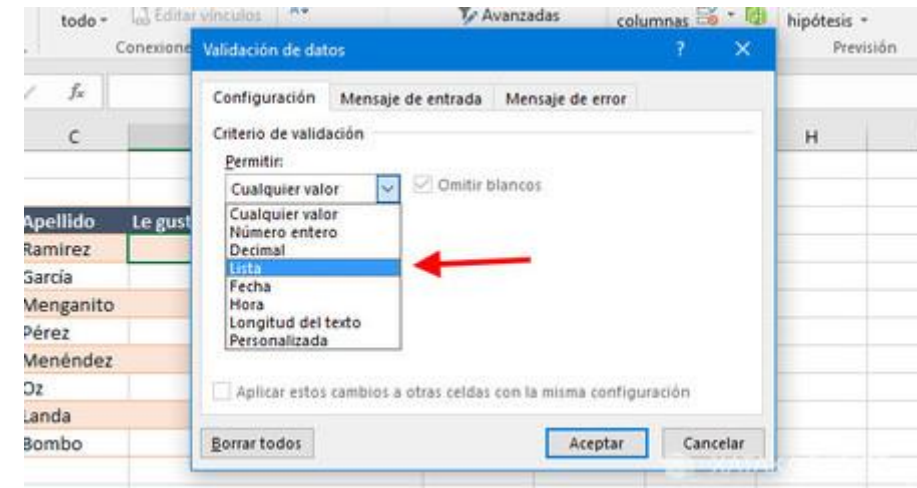
## Tipos de Entradas - Fechas

- Campo vacío
- Formato incorrecto
- Fechas inválidas

The image shows a date selection interface. At the top, there are two input fields: 'Desde' (From) containing '01/02/2014' and 'Hasta' (Until) which is empty. Below these is a calendar widget for February 2014. The calendar has a header with the month 'Feb' and the year '2014'. The days of the week are abbreviated as L, M, X, J, V, S, D. The dates 1 through 28 are displayed in a grid. The date '1' is highlighted. At the bottom of the calendar, there are two buttons: 'Fecha actual' (Current date) and 'Cerrar' (Close).

## Tipos de Entradas – Listas Desplegables

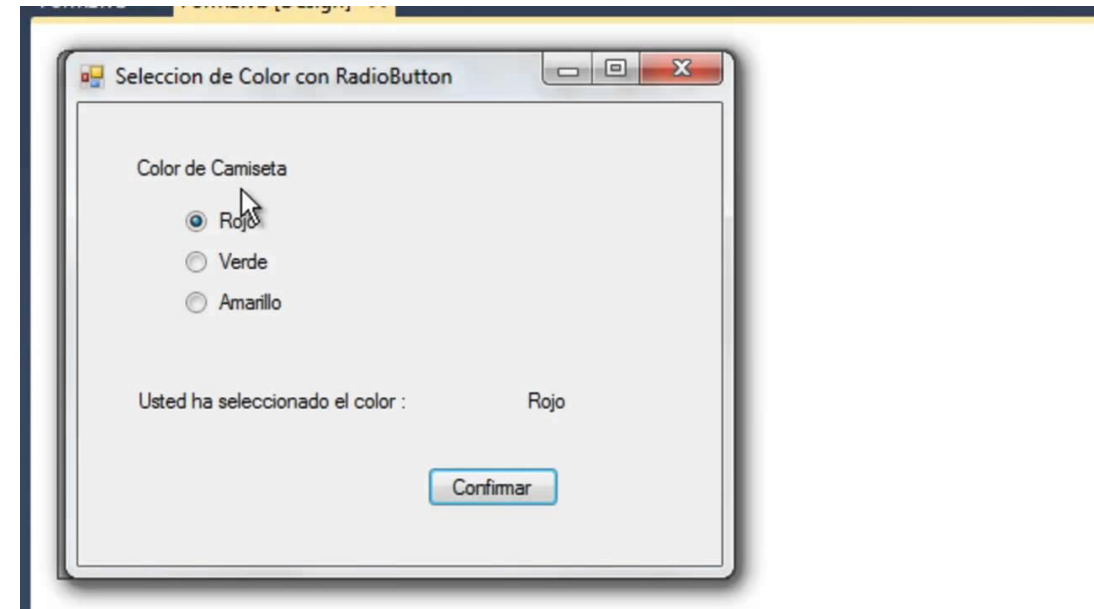
- Sin seleccionar
- Selección un elemento
- Selección varios elementos
- Típo de elemento existente
- Típo de elemento inexistente





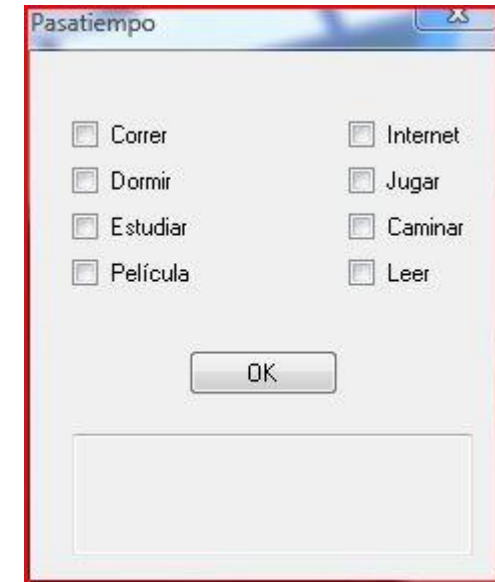
## Tipos de Entradas – Radio Button

- Sin seleccionar
- Un objeto seleccionado



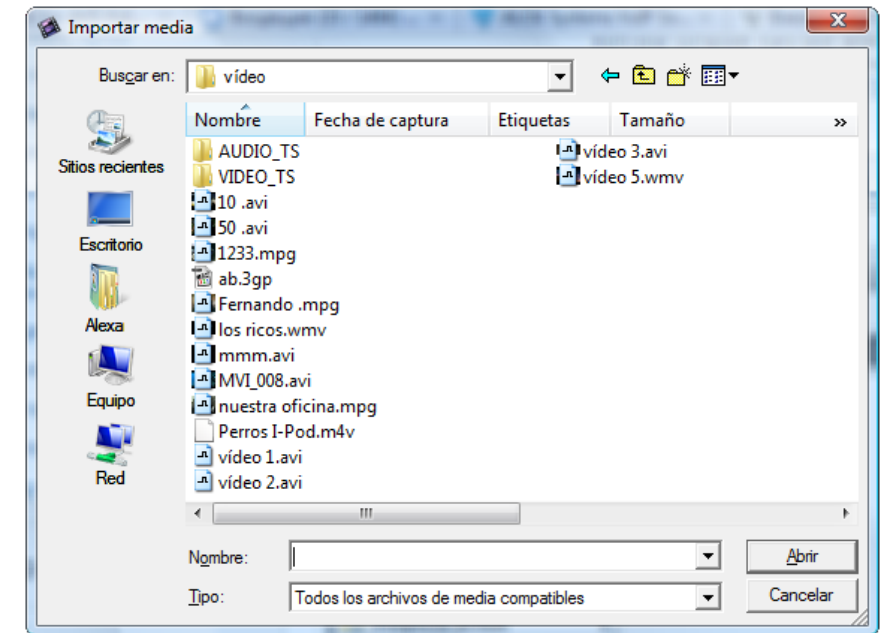
## Tipos de Entradas – Check Button

- Sin seleccionar
- Seleccionar varios



## Tipos de Entradas – Importar archivo

- En blanco
- De 0 bytes
- Archivo grande
- Archivo con nombre largo
- Archivo con nombre con espacios
- Archivo con nombre con caracteres no convencionales
- Archivo con datos inválidos



# Reporte de Fallos

- **ID de Fallo:** Un identificador único
- **Test Case Asociado:** Numero de TC
- **Reporter:** Tester que lo reportó. Esto facilitará la comunicación entre el que lo reportó y el que lo va a solucionar.
- **Software y Release:** Modulo en donde se detecto la falla
- **Ambiente:** Indica sobre que ambiente de pruebas ocurrió la falla.
- **Plataforma y Sistema Operativo:** Indica sobre las características de hard y sistema operativo de la maquina en donde se detectó la falla.
- **Navegador:** Indica el navegador que se estaba utilizando cuando se detectó la falla.

# Reporte de Fallos

- Prioridad:** Indica la urgencia.
- Severidad:** Indica el impacto de la falla sobre la aplicación (Bloqueante...)
- Resumen:** Descripción corta que describe la falla en forma simple y concreta. Describe el error exacto.
- Esperado:** Descripción de que debería ocurrir en lugar de la falla.
- Pasos:** Detalle de los pasos realizados para poder reproducir la falla, junto con los datos utilizados y la generación del escenario para que ocurra el bug.
- Adjuntos:** Cualquier material complementario que sirva para ayudar al desarrollador a solucionar la falla (screenshots, logs, documentos, videos, etc...)

# PREGUNTAS



# PRACTICA

---

Practica: Mejora de TC utilizados en el curso

Practica: Reporte de 2 fallos encontrados en el programa testing.exe.





Fin de a Unidad 3.