

DISEÑO DE PRUEBAS DE SOFTWARE

OBJETIVOS

- ① Determinar el enfoque de pruebas que mejor aplica en un contexto determinado.
- ① Diseñar pruebas utilizando las clases equivalentes



1

ENFOQUES AL PLANIFICAR PRUEBAS

Enfoques al planificar pruebas

Se puede elegir entre distintos enfoques para definir la estrategia a utilizar al planificar y/o estimar las pruebas:

- Analítico
- Basado en modelos
- Metodológico
- Por Cumplimiento de Estándares o Procesos
- Dinámico
- Dirigido
- Regresionista

Enfoques al planificar pruebas

- **Analítico**

- Riesgos, requerimientos u otros elementos para analizar

- **Basado en modelos**

- Creación/selección de modelos (matemáticos) para comportamientos críticos.

- **Metodológico**

- Procedimiento definido por experiencia, in-house.

- **Por Cumplimiento de Estándares o Procesos**

- Estándar IEE829, o XP (ágil) o cualquiera desarrollado por terceros.

Enfoques al planificar pruebas

● Dinámico

- Adaptarse a errores encontrados en la ejecución(pruebas exploratorias o basadas en ataques)

● Dirigido

- Foco en lo que quiere el usuario

● Regresionista

- Mantener lo probado una vez, funcionando.
- La automatización es fundamental.

Elementos para definir el enfoque

¿Cómo elegir cuál usar (o combinar)?

Debemos considerar:

- ⦿ Riesgos
- ⦿ Habilidades
- ⦿ Objetivos
- ⦿ Regulaciones
- ⦿ Producto
- ⦿ Negocio



2

CASOS DE PRUEBA Y CLASES DE EQUIVALENCIA

CASO DE PRUEBA

Un caso de prueba es el elemento fundamental del proceso de pruebas.
Se deben elegir casos efectivos, es decir:

Mostrar que el sistema funciona como se esperaba (cuando se usa como se indica).



Caso(s) ideal(es)

Si hay "errores", que sean visibles.

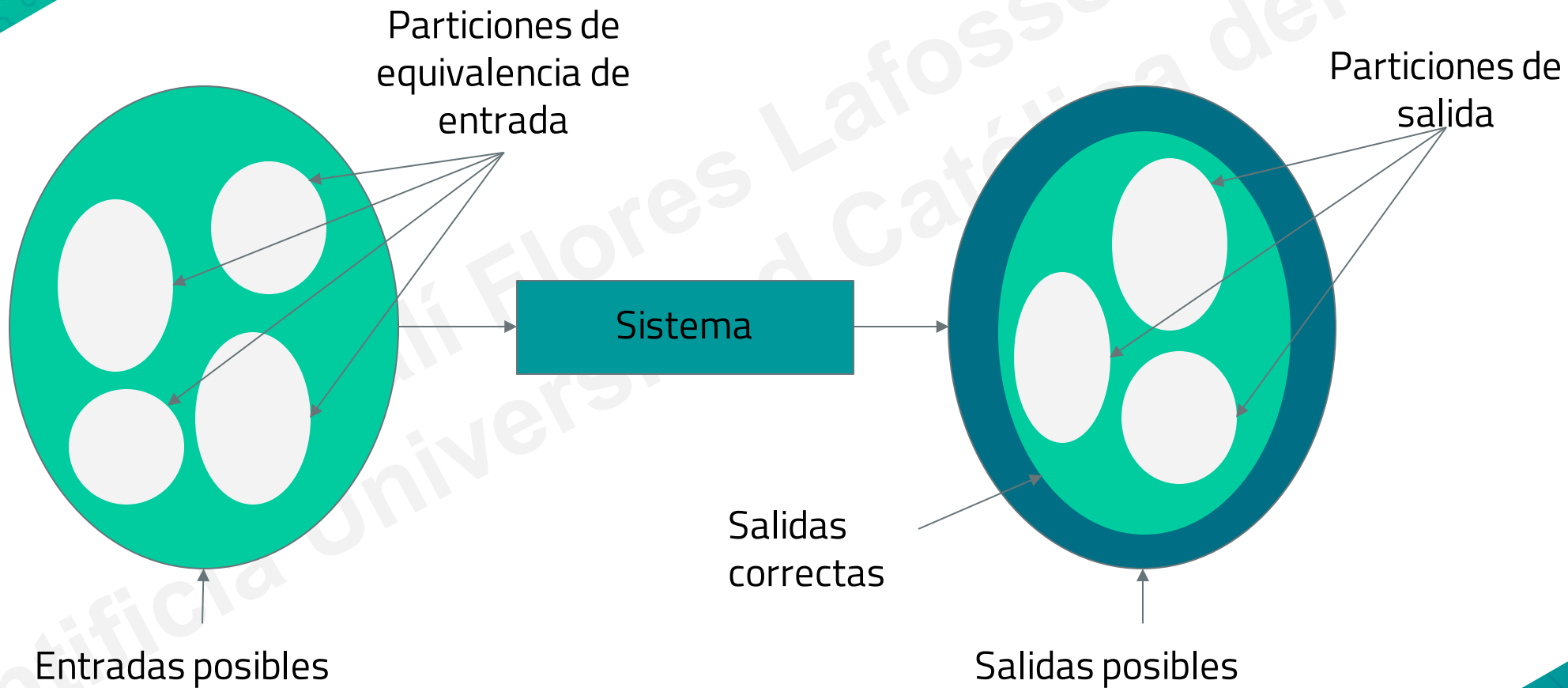


Caso(s) con problemas comunes

Partición de datos de entrada (para casos de prueba)

- Datos de entrada agrupados por características similares.
- Estos grupos se llaman “clases de equivalencia”.
- Definir (para cada una) un caso de prueba.
- Valores límites por separado para ver comportamientos no esperados
- Son pruebas de **caja negra**.

CLASES DE EQUIVALENCIA



CLASES DE EQUIVALENCIA

- Rango de valores: una clase válida y dos clases no válidas.
- Elementos de una clase que no se tratan igual: clases menores.
- Conjunto de valores admitidos: una clase válida por cada valor.
- Condición booleana (p.e. "el primer caracter debe ser una letra"): una clase válida ("es una letra") y una no válida ("no es una letra").

CLASES DE EQUIVALENCIA (Ejemplo 1)

Para calcular el Impuesto a la Renta se debe tomar en cuenta que las tasas varían según la Remuneración Neta Anual (RNA).

| RNA en S/. | Tasa |
|------------------------|------|
| Desde 5 UIT (25 750) | 14% |
| Desde 20 UIT (103 000) | 17% |
| Desde 35 UIT(180 250) | 20% |
| Desde 45 UIT (231 750) | 30% |

Armar las clases de equivalencia que se necesitaría para probar el ingreso como dato de entrada del RNA en un sistema.

Nota: UIT en 2024 es 5150 soles
Sólo se cobra IR desde 5 UIT

CLASES DE EQUIVALENCIA (Ejemplo 1)

| Condición de Entrada | Clases Válidas | ClasesNo Válidas |
|--|---|--|
| <i>Campo</i> Remuneración Neta Anual | 1. Valores mayores a 0 y menor o igual a 25 750 | 6. Vacío |
| | 2. Valores mayores a 25 750 y menor o igual a 103 000 | 7. Valor menor o igual a 0 |
| | 3. Valores mayores a 103 000 y menor o igual a 180 250 | 8. Valor mayor al máximo admitido por el tipo de dato. |
| | 4. Valores mayores a 180 250 y menor o igual a 231 750 | 9. Datos no numéricos |
| | 5. Valores mayores a 231 750 y menores al máximo admitido por el tipo de dato | |

CLASES DE EQUIVALENCIA (Ejemplo 2)

Para registrarse en un sistema, el usuario debe ingresar una contraseña que cumpla las siguientes condiciones:

- ❖ Debe incluir al menos una mayúscula
- ❖ Debe incluir al menos una minúscula
- ❖ Debe incluir al menos un número
- ❖ Debe tener una longitud mínima de 6 caracteres
- ❖ Debe tener una longitud máxima de 14 caracteres

Armar las clases de equivalencia que se necesitaría para probar que la contraseña es válida.
Asumir sólo caracteres alfanuméricos.

CLASES DE EQUIVALENCIA (Ejemplo 2)

| Condición de Entrada | Clases Válidas | ClasesNo Válidas |
|----------------------------|---|---------------------------------|
| <i>Campo</i> Contraseña | 1. Entre 6 y 14 caracteres, con 1 o más mayúsculas, con 1 o más minúsculas, con 1 o más caracteres numéricos. | 2. Vacío (0 caracteres) |
| | | 3. Sin caracteres en mayúscula. |
| | | 4. Sin caracteres en minúscula. |
| | | 5. Sin caracteres numéricos |
| | | 6. Entre 1 y 5 caracteres |
| | | 7. Más de 14 caracteres |
| | | |

Especificación de Caso de Prueba

Una vez diseñada la prueba, se debe crear su especificación.
Debe contener por lo menos:

- Identificador
- Objetivo
- Precondición
- Descripción de la prueba
- Resultados esperados

Ojo: Dependiendo del proyecto o institución, pueden requerirse otros elementos.

Especificación de Caso de Prueba (Ejemplo)

| Prueba MU-25 | |
|--------------------------|--|
| Objetivo | Dar de alta a un usuario dejando vacío el nombre de usuario. |
| Precondición | Se ingresó al sistema como "registrador" |
| Descripción de la prueba | <p>En la interfaz de entrada introducir</p> <ul style="list-style-type: none">- Username: usuario1- Primer Apellido: apellido- Segundo Apellido: apellido- Tipo de usuario: estándar <p>Dejar vacío el campo "Nombre"</p> |
| Resultados esperados | Se muestra el mensaje "El nombre de usuario es obligatorio" |

Casos de Prueba y Clases de Equivalencia

- ⦿ Cada combinatoria de clases válidas genera un caso de prueba.
- ⦿ Cada clase inválida genera un caso de prueba.

Casos de Prueba y Clases de Equivalencia

| Condición de Entrada | Clases Válidas | Clases No Válidas |
|---------------------------|------------------------------------|--------------------------------------|
| <i>Campo código</i> | 1. Cadena numérica de 5 caracteres | 2. Vacío (0 caracteres) |
| | | 3. Entre 1 y 4 caracteres |
| | | 4. Más de 5 caracteres |
| | | 5. Cadena de caracteres no numéricos |
| <i>Campo Tipo Usuario</i> | 6. Estándar | |
| | 7. Administrativo | |

● Casos de Prueba Válidos: (1,6) (1,7)

● Casos de Prueba Inválidos: (2,6) (3,6) (4,7) (5,7)

Casos de Prueba y Clases de Equivalencia

¡No olvidar derivar también casos de valor límite!



3

OTRAS FORMAS DE DERIVAR PRUEBAS

Pruebas basadas en Lineamientos

Se debe tomar en cuenta la experiencia del equipo y de la industria para probar errores recurrentes.

Algunos generales:

- ❖ Entradas que fuercen todos los mensajes de error
- ❖ Entradas que desborden el buffer
- ❖ Repetir las mismas acciones una y otra vez
- ❖ Forzar salidas inválidas
- ❖ Forzar cálculos muy grandes o muy pequeños

Otras formas de derivar pruebas

- Diseño de pruebas por ejecutables
- Diseño de pruebas por ramas
- Diseño de pruebas por cambio de estados
- Diseño de pruebas por valores aleatorios

.... Y muchas más

A large, bold, green number '4' is positioned in the upper left corner. The background is a teal gradient with a faint circuit board pattern. A diagonal line separates the teal background from a darker blue area on the right.

4

MINI-CASOS

Aplicando

Mini-caso 01

La consultora de Software STEIN ha conseguido el contrato con una empresa vendedora de celulares, para desarrollar su tienda web y aplicativo móvil. En la primera reunión, los representantes de la empresa han dejado en claro que lo más importante para ellos es que los clientes puedan comparar los diferentes productos, así que no debe haber fallas en dicha funcionalidad. Al seguir esta directiva, la consultora STEIN está siguiendo un enfoque de pruebas: ?

- ☒ A: Metodológico
- ☐ B: Dirigido
- ☐ C: Dinámico
- ☐ D: Regresionista

Mini-caso 02

Liste las clases de equivalencia para diseñar los casos de prueba según las siguientes consideraciones:

- ❖ Todos los campos son obligatorios.
- ❖ El nombre del curso es de máximo 20 caracteres alfanuméricos.
- ❖ El Código curso es de 6 caracteres alfanuméricos. Los 3 últimos deben ser números.
- ❖ La especialidad puede ser "Ingeniería Informática", "Ingeniería Industrial" o "Ingeniería Electrónica".
- ❖ El año de Inicio de Matrícula tiene que ser mayor a 2000 y máximo 2024.
- ❖ Cuando el año es menor a 2019, el registro se guardará como "Histórico".

The screenshot shows a mobile application interface for 'Gestión de Cursos'. At the top, there is a back arrow and the title 'Gestión de Cursos'. Below this is a section titled 'Nuevo Curso'. The form contains four input fields: 'Nombre del Curso' (with placeholder text 'Obligatorio'), 'Código de Curso' (with placeholder text 'Obligatorio'), 'Especialidad' (a dropdown menu showing 'Ingeniería Informática'), and 'Año de Inicio de Matrícula' (with placeholder text 'Obligatorio'). At the bottom of the form is a button labeled 'REGISTRAR'.

5

REFERENCIAS

BIBLIOGRAFÍA

- Sommerville, I. (2011). Software engineering (ed.). America: Pearson Education Inc.
- ISO/IEC/IEEE 29119-1:2022 Software and systems engineering <https://www.iso.org/standard/81291.html>

Créditos:

- Plantilla de la presentación por [SlidesCarnival](#)
- Diseño del fondo [Hero Patterns](#)