Telefónica

NTT DATA

Capacitación Pruebas Unitarias (xUnit)

9 Noviembre 2022 – Sesión 3

FUTURE AT HEART

Contenido

Modulo I – Conceptos Básicos

- ¿Qué es una prueba unitaria?
- Otros niveles de prueba
- Conociendo las pruebas unitarias
- Tecnicas de diseño de casos de prueba
- Principios FIRST

Modulo II – Test Driven Development

- Definición
- Desarrollo Ágil – Características
- Ciclo de desarrollo TDD
- Como escribir código que se pueda probar
- Cobertura de código
- Ventajas/Desventajas

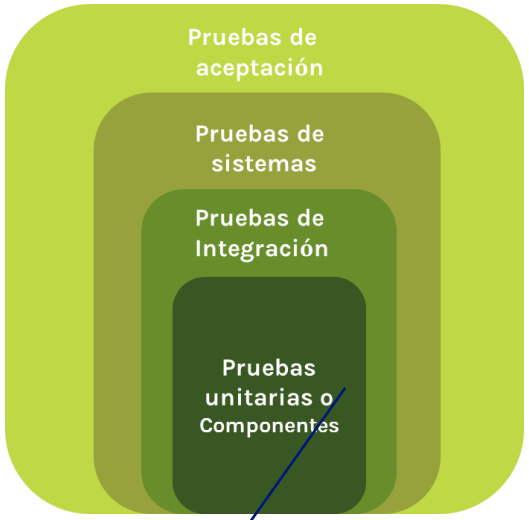
Modulo III – xUnit

- ¿Qué es xUnit?
- ¿Por qué xUnit?
- ¿Cómo funciona xUnit?
- Conceptos básicos
- Escenarios con xUnit
- Patrones de nombramiento de los escenarios
- Simulaciones
- Asersiones

Modulo IV – Taller

- Preparación del Entorno
- Ejemplo práctico
- Creando casos de prueba
- Ejecución de casos
- Mutation Testing
- Conclusiones/Recomendaciones

En capítulos anteriores...




```java
new *
@Test
@DisplayName("Prueba 1: 1 + 1=2 ")
void addsTwoNumbersV2() {
    Calculator calculator = new Calculator();//Arrange
    int result=calculator.add( a: 1,  b: 1);//Act
    assertEquals( expected: 2,result ,  message: "1 + 1 should equal 2");//Asssert
}

new *
@Test
@DisplayName("Prueba 1: -1 + (-1)= -2 ")
void addsTwoNumbersV3() {
    Calculator calculator = new Calculator();//Arrange
    int result=calculator.add( a: -1,  b: -1);//Act
    assertEquals( expected: -2,result ,  message: "1 + 1 should equal 2");//Asssert
}
```

Frameworks de Pruebas unitarias - Junit

JUnit 5
TestNG
Java

unittest
Python

Unit Test
nunit
MSTest
.Net

PHPUnit
Php

Jest
Node

https://github.com/oacarrillo/curso-xUnit

Técnicas de Diseño de Casos

https://www.mindmeister.com/es/map/2385619540/istqb-ctfl-2018?fullscreen=1&v=public

Técnicas de Diseño de Casos

Condiciones		
Ingreso de 800 a 2,000	→	TC Dorada
Ingreso de 2,000 a 3,500	→	TDC Platinum
Ingreso de 3,501 a N..	→	TC Negra

¿Cuantas pruebas se generarían?

Partición no válida	Partición válida para TDC Dorada	Partición válida para TDC Platinum	Partición válida para TDC Negra
Valores menores a 800$	800$ .. 2000$	2001$ .. 3500$	3501$ .. n

Técnicas de Diseño de Casos – Partición de Equivalencia

Lista de tipos de datos primitivos en JAVA			
Tipo	**Tamaño**	**Valor mínimo**	**Valor máximo**
byte	8 bits	-128	127
short	16 bits	-32768	32767
int	32 bits	-2147483648	2147483647
long	64 bits	-9223372036854775808	9223372036854775807
float	32 bits	-3.402823e38	3.402823e38
double	64 bits	-1.79769313486232e308	1.79769313486232e308
chart	16 bits	'\u000'	'\ufff'

Entrada Caracteres = “a”

Técnicas de Diseño de Casos – PE – Cobertura de PE

Partición no válida	Partición válida para TDC Dorada	Partición válida para TDC Platinum	Partición válida para TDC Negra
Valores menores a 800$	800$ .. 2000$	2001$ .. 3500$	3501$ .. n

799$	801$	2500$	3700$

$$\frac{\text{\# particiones probadas (4)}}{\text{\# particiones identificadas (4)}} = 1 = 100\%$$

$$\frac{\text{\# particiones probadas (3)}}{\text{\# particiones identificadas (4)}} = 0{,}75 = 75\%$$

Técnicas de Diseño de Casos - Partición de Equivalencia

- Es una técnica basada en especificación (Requerimientos o HU)
- Puede ser aplicado a cualquier nivel de pruebas
- Se basa en dividir condiciones de prueba en grupos (Equivalentes o Iguales)
- Se prueba una condición para cada tipo de clase o grupo

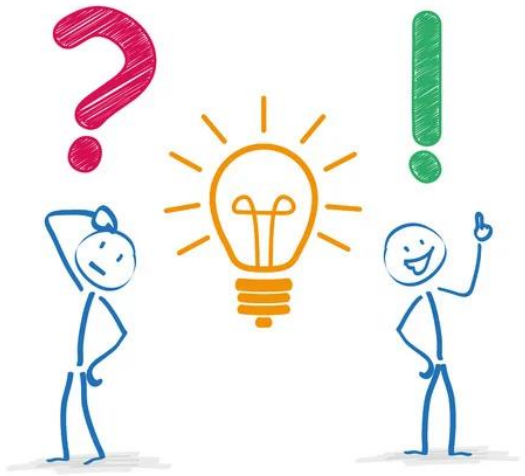
Ejercicio – Particiones de equivalencia

1- Uno de los campos de un formulario contiene un cuadro de texto que acepta valores numéricos en el rango de 18 a 25. Identifique la clase de equivalencia no válida.

a) 17
b) 19
c) 24
d) 21

Solución:

El cuadro de texto acepta valores numéricos en el rango de 18 a 25 (18 y 25 también forman parte de la clase). Entonces esta clase se convierte en nuestra clase válida. Pero la cuestión es identificar la clase de equivalencia no válida. Las clases serán las siguientes:
Clase I: valores clase inválida
Clase II: 18 a 25 => clase válida
Clase III: valores> 25 => clase no válida
17 entran en una clase inválida. 19, 24 y 21 entran en una clase válida. Entonces la respuesta es 'A'



Técnicas de Diseño de Casos - Valores Limite

Calificación	0	100

Un estudiante puede tener una calificación de 0 a 100
¿Cuantas pruebas se generarían?

Mínimo			Máximo		
-1	0	1	99	100	101

Técnicas de Diseño de Casos - Valores Limite

- Tiene como objetivo probar los limites de las particiones
- Tiene límites validos e inválidos
- Complementa la técnica de valores de equivalencia

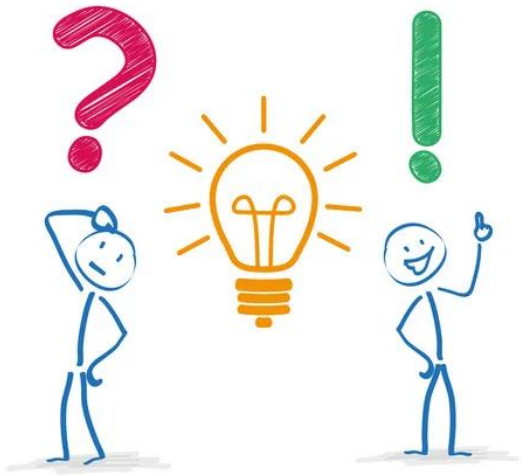
Ejercicio - Valores Limite

Un programa valida un campo numérico de la siguiente manera: se rechazan los valores inferiores a 10, se aceptan valores entre 10 y 21, se rechazan los valores mayores o iguales a 22. ¿Cuál de los siguientes cubre la MAYORÍA de los valores límite?

a. 9,10,11,22
b. 9,10,21,22
c. 10,11,21,22
d. 10,11,20,21

Solución:

Los límites se pueden identificar como 9, 10, 21 y 22. Estos cuatro valores están en la opción 'b'. **Entonces la respuesta es 'B'**



Técnicas de Diseño de Casos - Tabla de decisión

Es utilizada para probar combinaciones de entradas y salidas.

Si eres un nuevo cliente y solicitas una tarjeta de crédito, obtienes un 15% de descuento en tus compras de hoy, si eres un cliente existente obtienes 10% de descuento. Si tienes un cupón obtienes 20% de descuento (Un cupón no puede ser usado por un cliente nuevo)

Condiciones	Regla1	Regla2	Regla3	Regla4	Regla5	Regla6	Regla7	Regla8
Nuevo Cliente (15%)	V	V	V	V	F	F	F	F
Cliente (10%)	V	V	F	F	V	V	F	F
Cupón (20%)	V	F	V	F	V	F	V	F
Resultado								
% Descuento	X	X	20	15	30	10	20	0

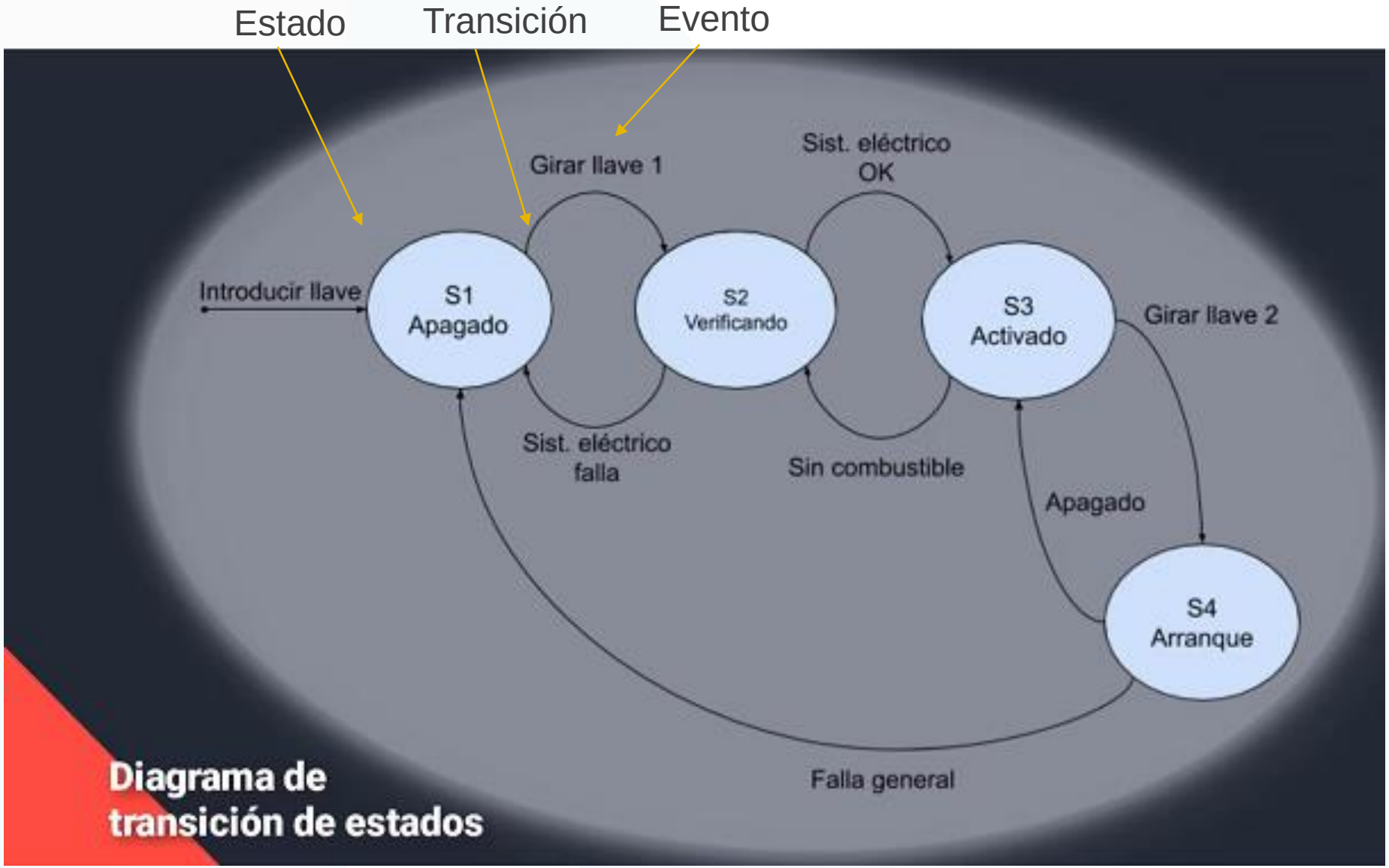
2^3 = 8 Combinaciones

$8/2 = 4$ $4/2 = 2$ $2/2 = 1$

Técnicas de Diseño de Casos - Tabla de decisión

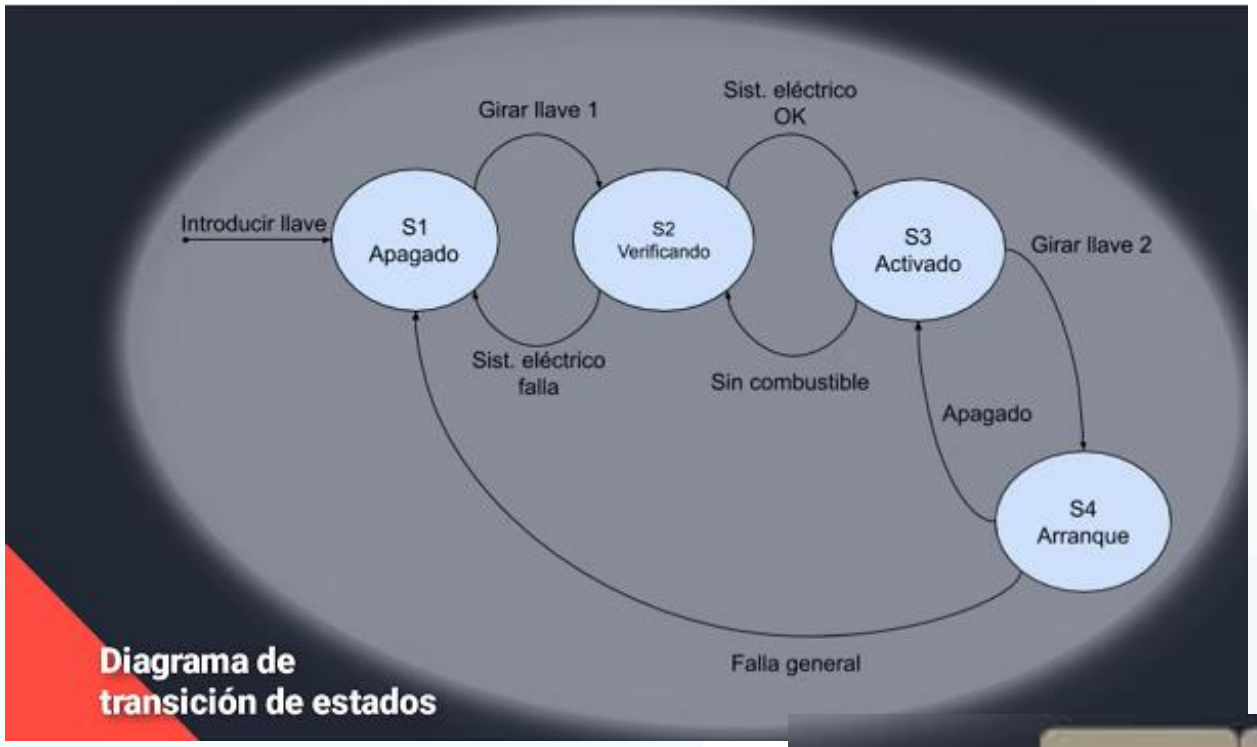
- Es una técnica basada en la especificación (Requerimientos)
- Esta enfocada en lógica y reglas de negocio
- Es una buena manera de generar combinaciones
- Proporciona una forma sistémica de validar reglas de negocio complejas
- Es una técnica para descubrir omisiones o ambigüedades (Pruebas de requisitos)
- No es viable para muchas combinaciones de entrada

Técnicas de Diseño de Casos – Transición de estados



https://www.youtube.com/watch?v=BlmfclxstyQ

Técnicas de Diseño de Casos – Transición de estados



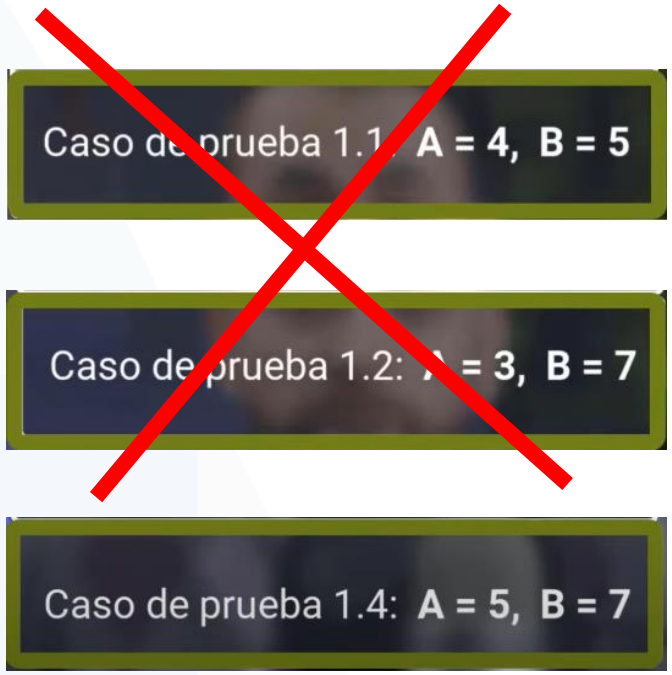
	Girar llave 1	Sistema eléctrico falla	Sistema eléctrico OK	Sin combustible	Girar llave 2	Apagado	Falla general
S1 : Apagado	S2	-	-	-	-	-	-
S2 : Verificar	-	S1	S3	-	-	-	-
S3 : Activo	-	-	-	S2	S4	-	-
S4 : Arranque	-	-	-	-	-	S3	S1

https://www.youtube.com/watch?v=BlmfclxstyQ

Tecnicas de Diseño de Casos – Transición de estados

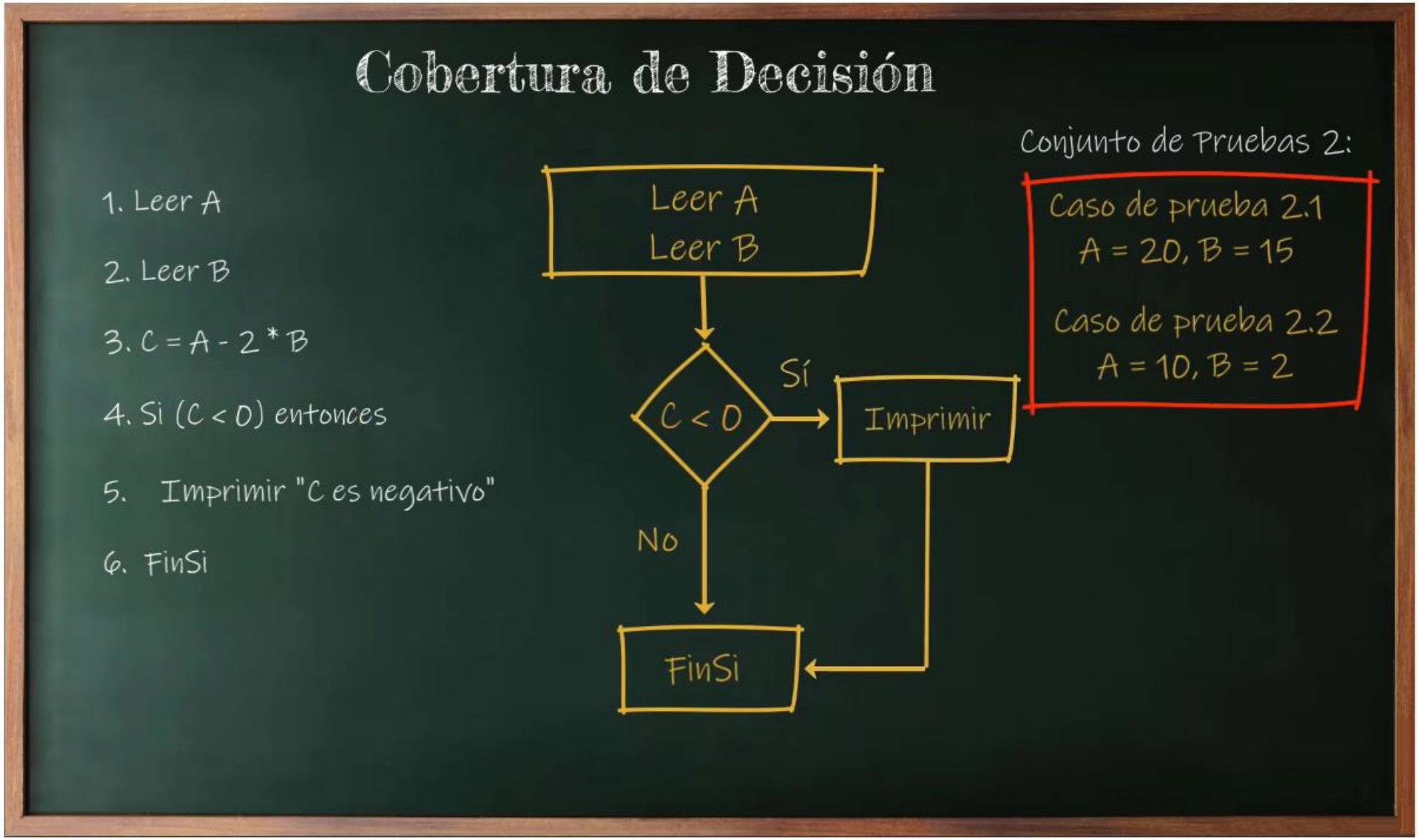
- Es una técnica basada en la especificación (Requerimientos)
- Los casos de prueba modelas la transición de un estado a otro
- Se puede validar si todas las transiciones son permitidas
- Cada estado es una condición de prueba
- Se puede medir la cobertura en base a las transiciones

Técnicas de Diseño de Casos - Cobertura de Sentencia



https://www.youtube.com/watch?v=BlmfclxstyQ

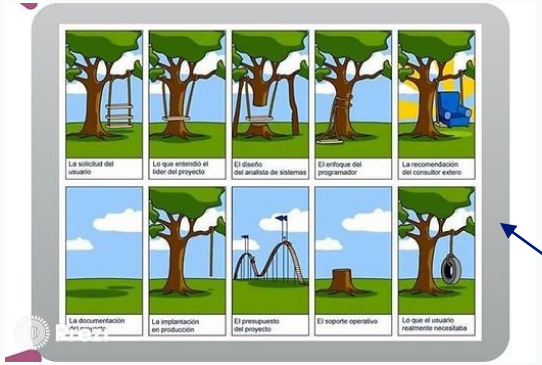
Técnicas de Diseño de Casos - Cobertura de Decisión



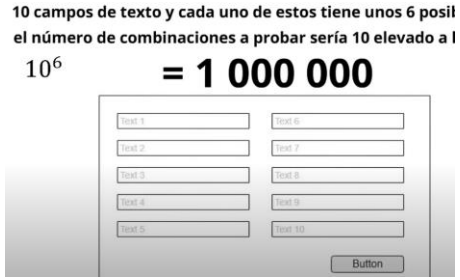
Tecnicas de Diseño de Casos – Basada en experiencia

- Predicción de error
- Pruebas exploratorias

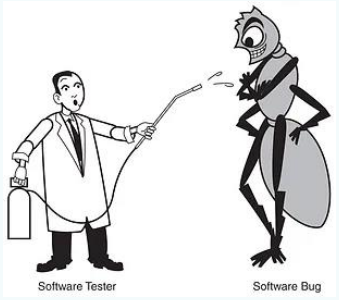
Principios del software testing

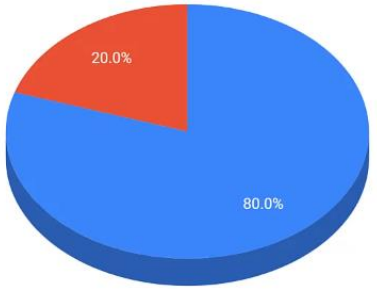


No su ausencia



Design and architecture	Implementation	Integration testing	Customer beta test	Postproduct release
1X*	5X	10X	15X	30X





https://cl.abstracta.us/wp-content/uploads/2022/04/7-principios-esenciales-testing-software-1024x1024.png


NTT DATA
GRACIAS
FUTURE
AT HEART