



Parte III – Desarrollo Dirigido por Pruebas

Tema 4: Pruebas funcionales




1



Agenda

- Repasando...
- Clasificación de las técnicas de prueba.
Comparativa pruebas estructurales-funcionales
- Pruebas funcionales
 - Clases de Equivalencia
 - Análisis de Valores Límite
 - Análisis Sintáctico (siguiente semana)



2

2

Tipos de Errores

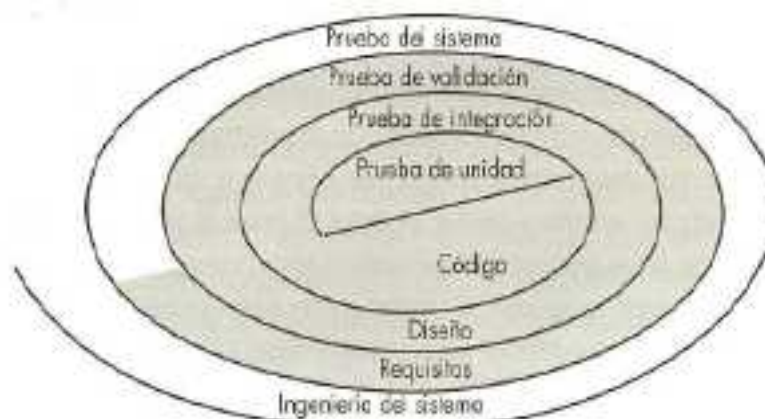
- Cálculo: La inclusión de la instrucción $x:=x+2$; en lugar de $x:=y+2$;
- Lógica: Definición incorrecta de una condición. "IF (a>b) THEN" en lugar de "IF (a<b) THEN"
- Entrada/Salida : Descripción incorrecta, mala conversión o formato inadecuado de entradas y salidas
- Transformación de datos: Incorrecto acceso o transformación de datos
- Interfaz: Comunicación incorrecta con otros componentes internos o externos
- Definición de Datos



3

3

Niveles de Pruebas de Software



4

4

Clasificación de las técnicas de prueba

- Con el propósito de aplicar las técnicas de prueba de software es necesario acceder a:
 - Las especificaciones del componente
 - El código fuente
 - El código objeto
- La clasificación se realiza atendiendo a todas las posibles combinaciones entre los elementos requeridos para las pruebas



5

5

Pruebas Unitarias

- Verifican la unidad más pequeña de software: el método

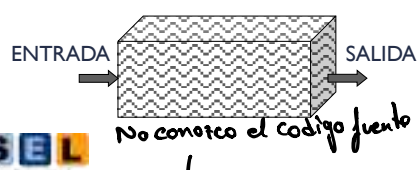
Pruebas de unidad

Funcionales o caja negra

Estructurales o caja blanca

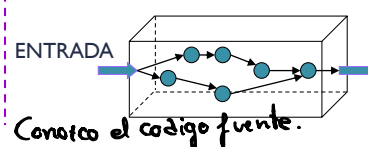
FUNCIONALES

- Se centra en las entradas y las salidas, NO en la estructura interna
- Es imposible probar todas las entradas y salidas posibles. Se seleccionarán las más relevantes



ESTRUCTURALES

- Se centra en la estructura interna
- Consiste en probar todos los caminos de ejecución




6

6

Hemos escrito las pruebas,
pero todavía no hemos
escrito el código.

Clasificación de las técnicas de pruebas en función de los elementos disponibles

Tipos	Elementos Requeridos			Ejemplos de Técnicas
	Especificaciones	Código Fuente	Código Objeto	
1				Algunas pruebas aleatorias
2				Análisis estático del flujo de datos. Revisiones de Código. Cálculo de Complejidad
3				Análisis dinámico del flujo de datos
4				Análisis formal de la coherencia de las especificaciones
5				Prueba de valores límites, Grafos Causa-Efecto, Análisis Particional
6				Ejecución Simbólica, Pruebas Formales
7				Métodos de cobertura, Pruebas de Mutación

7

7

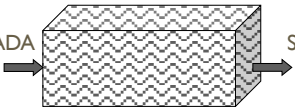
Un ejemplo:

Ejemplo:

Pruebas Funcionales: Conducir un coche para comprobar si la velocidad obtenida es la indicada en las especificaciones o hacer pruebas de frenado con el coche en un piso mojado para analizar el rendimiento de los frenos

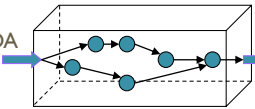
Estructurales: Examinar el motor del coche (encendido y apagado)

ENTRADA




SALIDA

ENTRADA




SALIDA

8

8

Efectividad de las técnicas de prueba


Tipo de Error	Técnica	
	Estructural	Funcional
Calculo	Alta	Media
Logico	Alta	Media
Entrada/Salida	Media	Alta
Procesado de datos	Limitada	Alta
Interfaces de componente	Alta	Media
Definición de datos	Limitada	Media

9

9

Pruebas de Caja Gris

- Las pruebas de caja gris suponen que se tiene acceso a la estructura interna de datos y algoritmos con el propósito de definir los casos de prueba pero siguiendo una aproximación similar a las pruebas de caja negra
- Son útiles para identificar clases de equivalencia y valores límite para los resultados y parámetros de salida
- También son útiles durante la fase de integración

10

No está el código pero tenemos idea de cómo va a funcionar.

10

Pruebas Funcionales o de Caja Negra

- Objetivo: reducir el número de casos de prueba a un nivel fácil de gestionar mientras se mantiene una cobertura razonable de las pruebas
 - Identificación de las clases de equivalencia
 - Análisis de Valores Límite
 - Análisis Sintáctico



11

11

Necesidad de clases de equivalencia

0-16	Don't hire
16-18	Can hire on a part-time basis only
18-55	Can hire as a full-time employee
55-99	Don't hire*

```

if (applicantAge == 14) hireStatus="NO";
if (applicantAge == 15) hireStatus="NO";
if (applicantAge == 16) hireStatus="PART";
if (applicantAge == 17) hireStatus="PART";
if (applicantAge == 18) hireStatus="FULL";
if (applicantAge == 19) hireStatus="FULL";
...
if (applicantAge == 53) hireStatus="FULL";
if (applicantAge == 54) hireStatus="FULL";
if (applicantAge == 55) hireStatus="NO";
if (applicantAge == 56) hireStatus="NO";
...
if (applicantAge == 99) hireStatus="NO";
if (applicantAge == 99) hireStatus="NO";
    
```

Tenemos varios valores que se procesan de la misma manera.

Varios valores se procesan de igual manera



Clases de Equivalencia



12

12

Definición de clases de equivalencia

-1

- Clase de equivalencia:
 - Consiste en un **conjunto de datos que se tratan de igual manera por un método o deberían proporcionar el mismo resultado**
 - Cualquier valor dentro de una clase es equivalente, en términos de prueba, a otro valor de la misma clase
 - Si un caso de prueba en una clase de equivalencia detecta un defecto, **todos** los otros casos en la misma clase de equivalencia deben detectar el **mismo defecto**
 - Si un caso de prueba en una clase de equivalencia NO detecta un defecto, todos los otros casos en la misma clase de equivalencia deben NO detectar ese defecto



13

13

Definición de clases de equivalencia

-2

- Un grupo de pruebas forma una clase de equivalencia si crees que:
 - **Todos ellos prueban el mismo procesamiento**
 - Si una **prueba** identifica un **error**, los **otros** probablemente **también lo harán**
 - Si una prueba no identifica un error, los otros probablemente tampoco lo harán



14

14

Análisis de Clases de Equivalencia -

Procedimiento

- Identificar las Clases de Equivalencia
 - Diferentes tipos de entregas requieren distintos tipos de clases de equivalencia

- Para cada tipo de clase de equivalencia, es necesario considerar:

- Clases Válidas (Diferentes casos de procesamiento normal del método)
- Clases Inválidas (Diferentes casos relativos a situaciones de error)

Que dan lugar a lo mismo.

- Definir los Casos de Prueba

- Un único caso de prueba puede considerar varias clases de equivalencia válidas – Si es posible, esta es la mejor alternativa
- Cada clase de equivalencia inválida tiene que ser considerada en un caso de prueba diferente

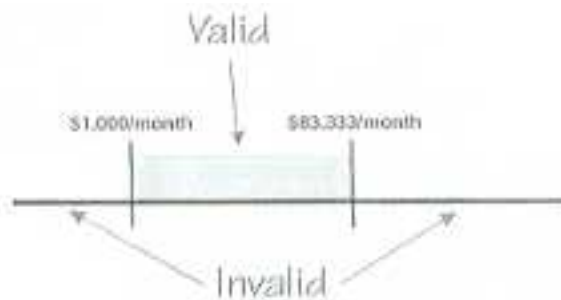


15

15

Reglas para identificar clases de equivalencia - I

- Rangos de valores continuos
 - Identificar el límite inferior
 - Identificar el límite superior
 - Identificar N particiones válidas



16

16

Type Of Service
2 classes validas

Cada clase de Eq. Inválida
en un único caso de prueba.

Las válidas se engloban en 1.

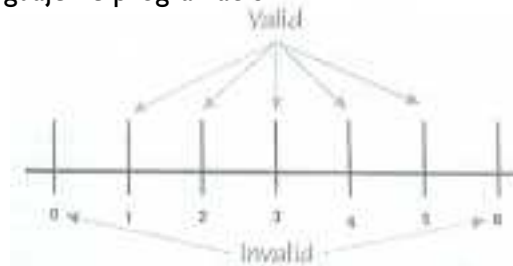
Comentar Casos en la práctica.

No es una prueba.

Lo proporcionan la misma salida.

Reglas para identificar clases de equivalencia -2

- Valores discretos dentro de un rango de valores permisibles
 - Una clase de equivalencia válida (uno de los valores permitidos)
 - Dos clases de equivalencia no válidas (Posibilidades inferiores y superiores)
 - Los valores fraccionarios o decimales, dependiendo del lenguaje de programación



17

17

Reglas para identificar clases de equivalencia -3

- Si el dato NO es un intervalo numérico:
 - Clase válida para cada valor válido
 - Clase no válida para los demás
- Ejemplo: Supongamos que tenemos un formulario y queremos probar uno de los campos de entrada: Estado Civil, dónde puede adoptar cualquiera de los siguientes valores: Soltero (S), Casado (C), Divorciado (D) o Viudo (V).
 - 4 clases válidas con valores tratados de forma diferente: S, C, D, V.
 - Clase no válida: cualquier valor diferente de S, C, D, V; por ejemplo M.



18

18

Reglas para identificar clases de equivalencia -4

- Número de valores de entrada. Si el parámetro está compuesto por un número determinado de valores:
 - Identificar el número de valores mínimo
 - Identificar el número de valores máximo
 - Identificar 1 clase válida y 2 clases inválidas
- Ejemplo: Asumir que se tiene que probar un método y uno de sus parámetros de entrada es la Lista de Puntuación:
 - Número mínimo de valores: 1
 - Número máximo de valores: 10 (hasta 10 puntuaciones distintas)
 - Clase Válida: {25, 29}
 - Clases Inválidas: {Vacía}; {23,24,25,26,34,65,76,87,98,90,32,65,43}



19

19

Reglas para identificar clases de equivalencia -5

- Otra aproximación para utilizar clases de equivalencia consiste en considerar las salidas en vez de las entradas
- Dividir las salidas en clases de equivalencia, entonces determinar qué entradas podrían causar cada una de esas salidas
- Solo es apropiado cuando es factible poder realizar fácilmente la generación de las distintas posibles salidas



20

20

Definición de casos de prueba

- Los elementos para definir correctamente casos de prueba son:
 - Identificador
 - Valores de Entrada
 - Para cada parámetro de entrada, indicar el valor y la clase de equivalencia a la que se pertenece
 - Resultados esperados



21

21

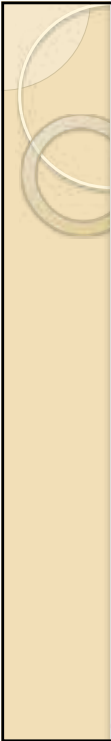
Aplicabilidad y Limitaciones

- El análisis de clases de equivalencia puede reducir significativamente el número de casos de prueba que se pueden crear y ejecutar
- Es más apropiado para sistemas en los que sus datos tienen valores incluidos en rangos o en conjuntos preestablecidos
- Requiere que las entradas o las salidas puedan ser particionadas de acuerdo a requisitos o precondiciones
- El análisis de clases de equivalencia se aplica igualmente a pruebas unitarias, de integración y sistema




22

22



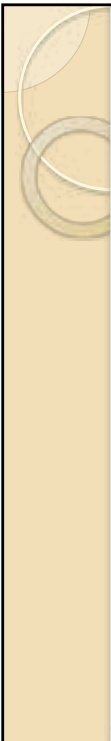
Pruebas Funcionales o de Caja Negra

- Objetivo: reducir el número de casos de prueba a un nivel fácil de gestionar mientras se mantiene una cobertura razonable de las pruebas
 - Identificación de las clases de equivalencia
 - Análisis de Valores Límite
 - Análisis Sintáctico



23

23




Necesidad de analizar los valores en los límites

Es muy importante este tipo de pruebas.

Años	
0-16	Don't hire
16-18	Can hire on a part-time basis only
18-55	Can hire as a full-time employee
55-99	Don't hire

```
If (applicantAge >= 0 && applicantAge <=16)
    hireStatus="NO";
If (applicantAge >= 16 && applicantAge <=18)
    hireStatus="PART";
If (applicantAge >= 18 && applicantAge <=55)
    hireStatus="FULL";
If (applicantAge >= 55 && applicantAge <=99)
    hireStatus="NO";
```



24

24

Antes de análisis de valores límite se hace después de las clases de equivalencia.

Análisis de valores límite

- La probabilidad de que los defectos sean más frecuentes en los valores límite de las particiones de equivalencia es muy alta
- Se debe aplicar conjuntamente con la identificación de clases de equivalencia
- El análisis de valores límite considera valores en los límites del intervalo de cada clase de equivalencia (en, justo antes, justo después)
- Existe un conjunto de reglas para determinar los valores límite a analizar en cada caso



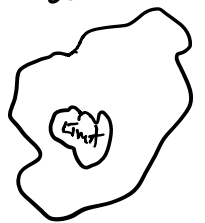
25

25

Aplicabilidad y limitaciones - I

- Dificultades para formalizar el concepto de valor marginal y límite
- Este tipo de análisis es más intuitivo y requiere heurística
- Falta de datos estadísticos suficientes sobre su eficacia

Son unos pocos de todos los posibles.



26

26

Aplicabilidad y limitaciones - 2

- El análisis de valores límite puede reducir significativamente el número de casos de prueba que se deben generar y ejecutar
- Está más dirigido a sistemas en los cuales muchos de los datos de entrada toman valores dentro de rangos o conjuntos
- Se requiere que las entradas o salidas se puedan particionar y los límites se puedan identificar con base en los requisitos
- El análisis de valores límite se puede aplicar en los niveles de pruebas unitarias, de integración, de sistema y de aceptación



27

27

Análisis de valores límite - Procedimiento

1. Identificar las clases de equivalencia
2. Identificar los límites de cada clase de equivalencia
3. Generar los casos de prueba para cada valor límite considerando las reglas que se presentan en las siguientes transparencias



28

28

Reglas para identificar los valores límite - I

- Regla 1: Valores límite para un rango continuo de entradas:
 - Si el parámetro corresponde con un intervalo de valores
 - Identificar el valor límite superior del intervalo, después considerar diferentes valores (en, justo antes, justo después)
 - Identificar el valor límite inferior del intervalo, después considerar diferentes valores (en, justo antes, justo después)
 - Ejemplo: Vamos a asumir que tenemos que probar un método y uno de sus parámetros de entrada es: Edad. Asimismo, los valores válidos de edades para que el método funcione deben estar entre:
 - 18 y 45 años:
 - Límite inferior: 18
 - Límite superior: 45
 - Valores a probar: 17, 18, 25, 45, 46



29

29

Reglas para identificar los valores límite - 2

- Regla 2: Valores límite para un rango discreto de entradas:
 - Si el parámetro se corresponde con un conjunto de valores:
 - Considerar el primero, el segundo, el penúltimo y el último
 - Ejemplo: Vamos a asumir que tenemos que probar un método y uno de sus parámetros de entrada es: 'opción', siendo las 'opciones' válidas para el método: {1,3,5,7,8,9}
 - Es necesario considerar: 1, 3, 8, 9



30

30

Reglas para identificar los valores límite - 3

- Regla 3: Valores límite de las salidas producidas
 - Usar la regla 2, pero analizando los valores de las salidas
 - Ejemplo: El máximo descuento aplicable a un seguro médico es 0,5 y el mínimo es 0,06
 - Valores a probar: 0,059 - 0,06 - 0,061 - 0,499 - 0,50 - 0,0501
 - Ejemplo: La aplicación es capaz de imprimir entre 1 y 5 copias del mismo documento
 - Valores a probar: 1, 2, 4 y 5 copias



31