

# Calidad, Validación y Verificación del Software

Técnicas Basadas en la Especificación:  
Clases de Equivalencia (Parte 1)

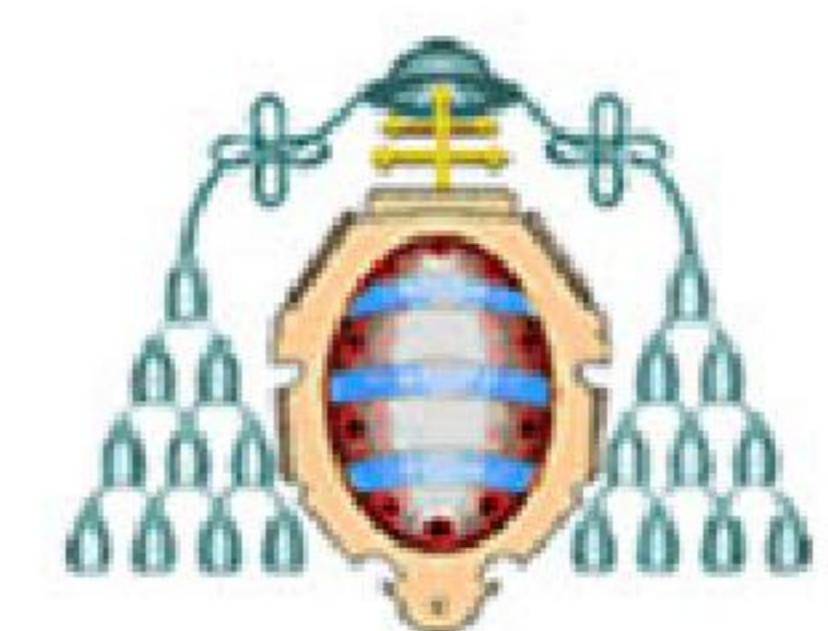
**Grado en Ingeniería Informática del Software**

Javier Tuya

Grupo de Investigación en Ingeniería del Software

<http://giis.uniovi.es>

Curso 2021-2022



# Contenidos

## ■ Parte 1 (Conceptos básicos):

- Técnica Básica: Partición en Clases de Equivalencia
- Técnica Complementaria: Análisis de Valores Límite
- Estrategia de combinación de clases – Jerarquía

## ■ Parte 2 (Diseño, implementación y automatización)

- Unitarias: Sin interfaz de usuario ni base de datos
- Unitarias: Con base de datos
- Integración con el interfaz de usuario
- (+Automatización Java/Swing y Spring Boot)
- Resumen

## ■ Parte 3 (Otras):

- Otras técnicas. Explosión combinatoria
  - Tablas de Decisión
  - Árbol de Clasificación
  - Técnicas Combinatorias
- Validaciones de Datos

# Conceptos iniciales



- Ejemplo:
  - Office, a partir de 2013 presenta dos modos para facilitar el uso con tablets: modo *Toque* (si está activo, muestra con mayor tamaño diferentes elementos de la interfaz) y modo *Mouse*.
- En teoría, deberíamos probar este programa en todas las circunstancias o situaciones posibles (infinitas).
  - Con esta especificación, al menos hay dos **situaciones** que se deben probar (modo mouse y modo toque).
  - Haremos pruebas en cada uno de los modos. Es decir, **cubriremos** ambos modos con nuestros **casos de prueba**
  - Para determinarlas nos hemos fijado en el modo. Esta es la **condición de prueba** (test condition).
  - Cada uno de ellos es una **situación a cubrir** (test coverage ítem)

# Partición en Clases de Equivalencia

## ■ Clase de equivalencia

- Representa un **conjunto de datos** para los que se supone que el programa tiene un **comportamiento similar**

- Los dominios se dividen en clases de equivalencia **disjuntas**

## ■ Identificación de clases de equivalencia

- Se examina cada **condición de entrada** (derivadas de las entradas)

- Cada condición de entrada **se divide en clases** de equivalencia

- Enumeraciones
- Rangos
- Valores lógicos

- Si hay razones para creer que los elementos de una clase no se tratarán de la misma forma, **dividir** la clase en otra más pequeña (jerarquía de clases)

# Partición en Clases de Equivalencia

- Problema 1: un sistema determina el tipo de interés aplicable a un crédito en función del importe del principal. Para valores menores de 10.000 euros se aplica el 4%, para valores mayores de 50.000 euros se aplica el 1%, en el resto de casos se aplica el 2%

# Partición en Clases de Equivalencia (Equivalence Partitioning)

- Condición de entrada: importe del principal
  - Clases de equivalencia
    - Hasta 10.000
    - Entre 10.000 y 50.000
    - Más de 50.000
  - Inclusión de **clases inválidas**
    - Importe negativo
- Discusión:
  - Comportamientos no especificados explícitamente
  - Recomendable otras situaciones singulares: valor cero ≠ negativo
  - Habilidad del tester:
    - Identificar las condiciones de entrada
    - Pensar en lo que no está explícito en la especificación

# Partición en Clases de Equivalencia

- Problema 2: un sistema determina el **tipo de interés** aplicable a una inversión en función del **saldo en la cuenta** corriente. Para valores menores de 10.000 euros se aplica el **1%**, para valores mayores de 50.000 euros se aplica el **4%**, en el resto de casos se aplica el **2%**

# Partición en Clases de Equivalencia

- Condición de entrada: saldo en cuenta
- Clases de equivalencia
  - Hasta 10.000
  - Entre 10.000 y 50.000
  - Más de 50.000
- Inclusión de clases inválidas?
  - Saldo negativo
- Discusión:
  - Saldo negativo es realmente una clase inválida?

# Partición en Clases de Equivalencia

- Problema 1b: un sistema determina el tipo de interés aplicable a un crédito en función del importe del principal. Para valores menores de 10.000 euros se aplica el 4%, para valores mayores de 50.000 euros se aplica el 1%, en el resto de casos se aplica el 2%. **Si el titular de la inversión es menor de 21 años se le añade medio punto porcentual adicional.**

# Partición en Clases de Equivalencia

- Condiciones de entrada: importe del principal y edad
- Clases de equivalencia (para cada condición de entrada)
  - Importe del principal
    - Hasta 10.000
    - Entre 10.000 y 50.000
    - Más de 50.000
    - Importe negativo (inválida)
  - Edad
    - Menor de 21
    - Mayor de 21
    - Negativa (inválida)
- Cuántos casos de prueba derivamos de estas clases?
  - Diversas formas de hacerlo

# Partición en Clases de Equivalencia

## Derivación de casos de prueba

- Estrategia típica: minimizada (minimized approach):
  - Crear **el menor número de casos** que cubran las clases válidas
  - Habitualmente **uno por cada una de las inválidas** (para evitar enmascaramiento de defectos) (one-to-one approach)
- En el ejercicio anterior, los casos de prueba son:

	Entradas		Salidas
Clases Válidas	Importe	Edad	Tipo interés
	Hasta 10.000	Menor de 21	4.5%
	Entre 10.000 y 50.000	Mayor de 21	2%
Clases inválidas	Más de 50.000	Cualquiera (p.e. Mayor 21)	1%
	Negativo	Cualquiera válido	error
	Cualquiera válido	Negativo	error

# Clases de equivalencia de las salidas

- Complementan el análisis respecto de las entradas

Problema 1 (interés crédito-importe ppal.)	Problema 2 (interés inversión-saldo cc)	Problema 1b (interés crédito-importe ppal.+edad)
1%, 2%, 4%, error	1%, 2%, 4%, 0%	1%, 1.5%, 2%, 2.5%, 4%, 4.5%, error

- En problema 1b, casos de prueba anteriores, no se han contemplado los valores de salida 4%, 2.5% y 1.5%. Añadirlas

- Discusión

- Las pruebas pueden realizarse con diferente intensidad
- Habilidad del tester: determinar la técnica/intensidad más adecuada al contexto del problema (coste/beneficio)
- Cómo lo haríamos para el problema del triángulo?

# Las condiciones de entrada no son necesariamente los parámetros de un programa

- Las condiciones de entrada, y por tanto, las clases de equivalencia vienen muchas veces determinadas por situaciones derivadas de **relaciones entre variables o parámetros**
- Ejemplo: Determinar si la posición de un objeto (dado por sus coordenadas) está dentro de un círculo determinado por su radio y centro (coordenadas)
  - Condiciones de entrada (para clases válidas):
    - ...
  - Clases de equivalencia:
    - ...

# Las condiciones de entrada no son necesariamente los parámetros de un programa

- Las condiciones de entrada, y por tanto, las clases de equivalencia vienen muchas veces determinadas por situaciones derivadas de **relaciones entre variables o parámetros**
- Ejemplo: Determinar si la posición de un objeto (dado por sus coordenadas) está dentro de un círculo determinado por su radio y centro (coordenadas)
  - Condiciones de entrada (para clases válidas):
    - Distancia del objeto al centro del círculo
  - Clases de equivalencia (para esta cond. ent.):
    - Menor que el radio (interior)
    - Mayor que el radio (exterior)

Las condiciones de entrada no son necesariamente los parámetros de un programa

# Las condiciones de entrada no son necesariamente los parámetros de un programa

## ■ Ejemplo del triángulo

- Clases válidas para Isósceles
  - Posición del lado desigual
    - Principio
    - Medio
    - Final
- Clases inválidas
  - Algún lado con valor inválido
    - Valor cero
    - Valor negativo
    - No entero
    - ...
  - Triángulo degenera en una línea
    - ...
  - No cierra
    - ...

### Discusión

Las condiciones de entrada no son necesariamente los parámetros de un programa

Las condiciones de entrada no son necesariamente los parámetros de un programa

Las condiciones de entrada no son necesariamente los parámetros de un programa

# Análisis de Valores Límite (Boundary Value Analysis)

## Clases equivalencia

Hasta 10.000

Entre 10.000 y 50.000

Más de 50.000

Importe negativo o cero

- En los problemas anteriores:
  - Qué significa **hasta 10.000**?
    - El 10.000 está incluido en la primera clase o en la segunda?
  - La tercera significa **más de 50.000** ó **50.000 o más**?
- Debemos probar valores correspondientes a estos puntos en la frontera de las clases para verificar si la implementación clasifica estos valores correctamente

# Análisis de Valores Límite (Boundary Value Analysis)

- Utilizando valores límite se sitúan las pruebas en los extremos de las clases de equivalencia (2-way/2-value)

Clases equivalencia	Valores límite (2 way)
Hasta 10.000	0,01; 9.999,99
Entre 10.000 y 50.000	10.000,00; 49.999,99
Más de 50.000/50.000 o más?	50.000,00
Importe negativo o cero	0,00; -0,01

- Opcional (3-way/3-value): establece tres valores (el que marca la frontera, uno más y uno menos)
  - Ejemplo: 49.999,99; 10.000,00; 10.000,01
- Se pueden incluir además valores típicos (no en los extremos)

# Estrategia de Combinaciones de Clases

- Ejemplo:
  - En Office 2013 y posteriores, además de tener en cuenta el modo (Toque, Mouse), se tiene en cuenta el tipo de dispositivo de entrada. Si tiene panel táctil por defecto se usa modo toque, y si no, modo mouse:
- Condiciones de entrada y clases de equivalencia:
  - Modo
    - Toque
    - Mouse
  - Tipo de dispositivo de entrada
    - Con panel táctil
    - Sin panel táctil
- Sería suficiente probar con dos casos de prueba utilizando la estrategia minimizada (minimized approach)?

# Estrategia de Combinaciones de Clases (estrategia minimizada). Trazabilidad

## ■ Clases de equivalencia:

- Modo
  - Toque
  - Mouse
- Tipo de dispositivo de entrada
  - Con panel táctil
  - Sin panel Táctil

- | Casos de prueba (estrategia minimizada) |
|---|
| Caso 1                                  |
| Caso 2                                  |
- Los casos de prueba CUBREN las diferentes clases
  - Cómo hacemos si estos casos no son suficientes para nuestra prueba?
    - Estrategia combinada

# Estrategia de Combinaciones de Clases (estrategia combinada). Proceso

1. Diseño “plano” inicial:
    - Modo
      - Toque
      - Mouse
    - Tipo de dispositivo de entrada
      - Con panel táctil
      - Sin panel Táctil
  2. Definimos que para probar esto habrá que hacer las combinaciones de modo y tipo
  3. Transformamos en un diseño “jerárquico”
    - Combinación tipo Disp. Entrada / Modo
      - Con panel táctil
        - Modo Toque
        - Modo Mouse
      - Sin panel táctil
        - Modo Toque
        - Modo Mouse
- 

# Estrategia de Combinaciones de Clases (estrategia combinada). Trazabilidad

## 3. Clases de equivalencia combinadas

- Combinación tipo Disp. Entrada / Modo
  - Con panel táctil
    - Modo Toque
    - Modo Mouse
  - Sin panel táctil
    - Modo Toque
    - Modo Mouse

### Casos de prueba (estrategia combinada)

- Caso 1 (táctil y toque)
- Caso 2 (táctil y mouse)
- Caso 3 (no táctil y toque)
- Caso 4 (no táctil y mouse)

**Discusión:**  
Cuáles son las test conditions y test coverage items?

# Más...

- Cómo se diseña e implementa:
  - Cuando hay base de datos?
  - Cuando hay un interfaz de usuario?
  
- Cómo se ejecutan los casos de prueba:
  - Manualmente?
  - De forma automática?