

# Técnicas – Basadas en condiciones

## Pruebas y Despliegue del Software

José García Fanjul ([jgfanjul@uniovi.es](mailto:jgfanjul@uniovi.es))

Grupo de Investigación en Ingeniería del Software - <http://giis.uniovi.es>

Curso 2022-2023



# Técnicas basadas en condiciones

- Utilizan **decisiones lógicas** implícitas o explícitas generalmente en la especificación como fuente para la obtención de pruebas
- Decisiones y condiciones
  - ☐ Condición, expresión lógica sin operadores relaciones
  - ☐ Decisión, expresión lógica formada por varias condiciones utilizando operadores relacionales
- Varios criterios
  - ☐ Decisión
  - ☐ Condición
  - ☐ Condición/Decisión
  - ☐ Múltiple Condición
  - ☐ Condición/Decisión Modificada

# Ejemplo

- Ejemplo – Cuota renovación en un club social
  - El sistema determinará el descuento sobre la cuota vigente que se aplicará en el momento de renovación. Se aplicará solamente a los socios con una antigüedad de al menos 5 años a mayores de 55 años y a los socios que han generado ingresos totales (por inscripción en actividades) por un valor de al menos 1000 euros en los tres últimos años
- Identificar en la especificación
  - Condiciones
  - Decisiones

# Pruebas de decisiones

## ■ Decisión

- Antigüedad  $\geq 5$  **AND** (Edad  $> 55$  **OR** Ingresos  $\geq 1000$ )

## ■ Objetivo

- Diseñar pruebas de forma que cada decisión tome los valores cierto y falso

| A $\geq 5$ | E $> 55$ | I $\geq 1000$ | Salida | A | E  | I    |
|------------|----------|---------------|--------|---|----|------|
| C          | C        | C             | C      | 5 | 56 | 1000 |
| F          | C        | C             | F      | 4 | 56 | 1000 |

- Qué pasa si en la condición de la edad se ha implementado por error un límite como por ejemplo, 45?

# Pruebas de condiciones

## ■ Decisión

- Antigüedad  $\geq 5$  **AND** (Edad  $> 55$  **OR** Ingresos  $\geq 1000$ )

## ■ Objetivo

- Diseñar pruebas de forma que cada condición tome los valores cierto y falso

| A $\geq 5$ | E $> 55$ | I $\geq 1000$ | Salida | A | E  | I    |
|------------|----------|---------------|--------|---|----|------|
| C          | F        | F             | F      | 5 | 55 | 999  |
| F          | C        | C             | F      | 4 | 56 | 1000 |

- Qué pasa si en la decisión relativa a la edad e ingresos se ha implementado un AND en vez de un OR?

# Pruebas de condición/decisión

## ■ Decisión

- Antigüedad  $\geq 5$  **AND** (Edad  $> 55$  **OR** Ingresos  $\geq 1000$ )

## ■ Objetivo

- Diseñar pruebas de forma que cada condición y decisión tomen los valores cierto y falso

| A $\geq 5$ | E $> 55$ | I $\geq 1000$ | Salida | A | E  | I    |
|------------|----------|---------------|--------|---|----|------|
| C          | C        | C             | C      | 5 | 56 | 1000 |
| F          | F        | F             | F      | 4 | 55 | 999  |

- Qué pasa si por error se ha implementado en el primer operador lógico OR en vez de AND?

# Pruebas de múltiple condición

## ■ Decisión

- Antigüedad  $\geq 5$  **AND** (Edad  $> 55$  **OR** Ingresos  $\geq 1000$ )

## ■ Objetivo

- Diseñar pruebas de forma que se ejerciten todas las posibles combinaciones lógicas de las condiciones

## ■ Problemas

- Ejemplo -> 8 posibles combinaciones
- Crecimiento en el número de casos exponencial con el número de condiciones
  - 4 condiciones -> 16 combinaciones
  - 5 condiciones -> 32 combinaciones
  - 6 condiciones -> 64 combinaciones
  - ...

# Pruebas de condición/decisión modificada (MCDC)

## ■ Objetivo

- Mantener efectividad en detección de defectos similar a Múltiple Condición
- Reduciendo el número de casos de prueba necesarios
- **Derivar pruebas de forma que cada condición afecta de forma independiente al resultado de la decisión**

## ■ Decisión

- Antigüedad  $\geq 5$  **AND** (Edad  $> 55$  **OR** Ingresos  $\geq 1000$ )
- En general para  $n$  condiciones,  $n + 1$  combinaciones.

| A $\geq 5$ | E $> 55$ | I $\geq 1000$ | Salida | A | E  | I    |
|------------|----------|---------------|--------|---|----|------|
| C          | F        | C             | C      | 5 | 55 | 1000 |
| C          | F        | F             | F      | 5 | 55 | 999  |
| C          | C        | F             | C      | 5 | 56 | 999  |
| F          | C        | F             | F      | 4 | 56 | 999  |



# Discusión/Resumen

- Dos grandes grupos de técnicas de prueba
  - Basadas en la especificación (por ejemplo, clases de equivalencia)
  - Basadas en la estructura,
    - La especificación se modela en base a decisiones y condiciones lógicas
    - Se pueden aplicar también a código fuente?
- Importante: Combinación de técnicas
  - Por ejemplo, análisis valores límite (especificación) en las condiciones (estructura)
- **En general se utiliza la especificación para derivar las pruebas (independientemente de las técnica)**
- Denominación clásica
  - Basadas en la estructura: pruebas de caja blanca
  - Basadas en la especificación: pruebas de caja negra

# Discusión/Resumen

## ■ También **Pruebas Negativas**

- ☐ Probar si hace lo que no debería hacer
- ☐ Ejercitar el software en situaciones de fallo, protección a errores de usuario, excepciones, etc...
- ☐ También en valores que NO deberían influir en las salidas
  - En el ejemplo, se utilizar el ingreso total (en los últimos tres años) para determinar el descuento
  - Añadir situaciones a probar donde no se utilice el ingreso total sino el ingreso del año en curso.

# Resumen/Discusión

- La prueba se realiza ejecutando casos de prueba (**test cases**)
- Para ello hay que determinar qué es lo que hay que probar (**test conditions**) y qué situaciones se probarán (**test coverage items**) apoyándose en el uso de técnicas de prueba
- La cobertura es el porcentaje de situaciones (**test coverage items**) cubiertas por los casos de prueba

