

# Generación de casos de prueba a partir de especificaciones UML

Luis Fernández Sanz  
*Universidad Europea de Madrid*

## Introducción

- Pruebas como técnica de VyV  
“Ejecución de software para detectar defectos”
- Caso de prueba:  
“Conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular”
- Capacidad de detección:
  - Dsicrepancia entre salida esperada y obtenida
  - En general, referencia de una buena especificación
    - C.Kaner: lo que espera el usuario
  - Especificación revisada:
    - Completa, coherente, exacta, de calidad

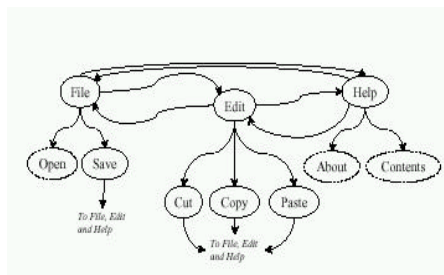
## Filosofías de pruebas

- Caja negra y caja blanca
- Inviabilidad de pruebas exhaustivas
- Diseño de pruebas:
  - Selección de casos: en el universo de posibilidades
  - Criterios: cobertura (funciones o estructura)
- Recomendable realizar diseño de pruebas en paralelo al desarrollo:
  - “No se hace justo antes de probar”
  - “Si hay especificación, hay diseño de pruebas”



## Panorama de pruebas

- Complejidad y sofisticación tecnológica
  - Interfaces gráficas, *event-driven*
- Especificaciones de interfaces gráficas:
  - Desde hace mucho (80s), diagramas de estados
  - Problema:
    - Peculiaridad notación
    - Reducida implantación
- Nuestra propuesta:
  - Notación UML
  - Insistencia en especificación



## Propuesta de especificación

- Uso de diagramas UML:
  - Casos de uso: muy conocido
    - “Unidad coherente de funcionalidad [...] manifestada por secuencias de mensajes intercambiados por el sistema y uno o más actores externos”
    - “Buena expresión de interacción de actores con sistema para obtener resultado observable y valioso”
- Relación conocida con las pruebas:
  - V JICS (Fernández): con grafos causa-efecto
  - Choi y von Mayrhauser: grafos de estado
  - Memon et al.: diagramas de estados (GUITAR)

## Casos de uso

### CASO DE USO: BAJA DE USUARIO

**Objetivo:** Borrar completamente los datos de un usuario de la base de datos de la aplicación.

**Flujo normal de acontecimientos:**



Actor	Sistema
1 Solicita Baja Usuario	2 Solicita ID Usuario
3 Introduce ID Usuario	4 Muestra datos de usuario y pide confirmación de baja
5 Confirma Baja Usuario	6 Muestra confirmación de Baja Usuario

**Flujos alternativos:**

En el 4, si el DNI no existe, entonces muestra un mensaje y vuelve al paso 3.

En cualquier caso, si el actor selecciona cancelar, entonces no se realiza ninguna opción y se vuelve al menú principal de la aplicación.

**Precondiciones**

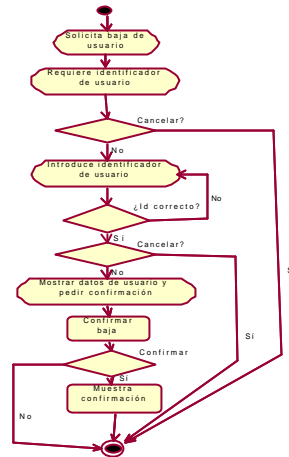
Debe existir el usuario que se desea dar de baja.

**Postcondiciones**

Eliminación de los datos del usuario de la base de datos de la aplicación.

## Diagrama de actividad

- Caso especial de diagrama de estados:
  - Detalle de comportamiento de un caso de uso, de un objeto o de un método en un objeto
  - Pueden incluir *swimlanes*
- En caso de uso:
  - Ayuda a comprender y validar
  - Complementan la descripción
- Especificación:
  - Casos de uso con d. actividad



## Criterio de selección

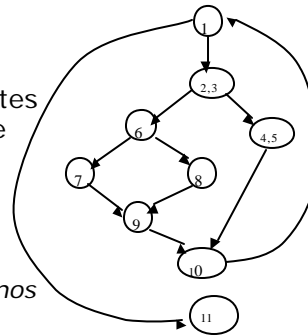
- Exploración de opciones de caso de uso:
  - Escenarios definidos por caminos en el diagrama de actividad
- Selección de escenarios:
  - Número de caminos del grafo muy elevado
  - Dentro de cada camino distintos escenarios de prueba basados en selección de valores de entrada
    - Que provocan distintas acciones de la aplicación
- Primer criterio de selección:
  - *Traza de caminos a través del diagrama de actividad*

## Criterio 1 de selección

- Similar a la cobertura de McCabe:
  - Parecido al de GUITAR
  - Número de caminos independientes que cubra el grafo ("cobertura de decisiones")

- Criterio 1:

*"Un caso de uso, a través de su diagrama de actividad, estará suficientemente probado si los caminos utilizados (tipos de escenarios) permiten pasar por todas las flechas o transiciones al menos una vez"*



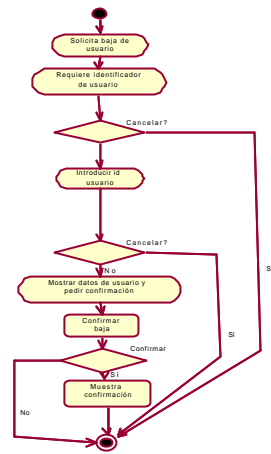
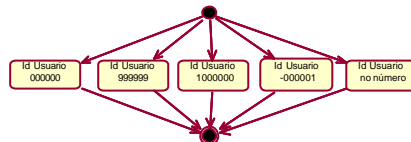
## Criterios 2 de selección

- En cada camino del diagrama:
  - Demasiados escenarios distintos
  - Aplicación de los criterios tradicionales de caja negra
  - Concepto de clases de equivalencia
- Segundo criterio de selección:
 

*Para las entradas de datos en un caso de uso, debemos lograr que todas las clases de equivalencia válidas sean ejercitadas por un caso de pruebas y que cada clase no válida sea comprobada en solitario en un caso específico, ejercitando todos los valores límite*

## Transformación de diagrama

- Creación de jerarquía:
  - Estado de actividad con entrada de datos
  - Genera un subdiagrama de acciones de las clases de equivalencia
- Permite evaluar cobertura en grafo extendido



## Método de generación

1. Generar para cada caso de uso un diagrama de actividad primario donde se marcan los caminos de flujos de eventos
2. Identificar en cada flujo la introducción de datos y analizar sus clases de equivalencia, valores límites e , incluso, tratamiento de combinaciones
3. Transformar cada acción de entrada de datos en un estado de actividad con un subdiagrama de clases de entradas y de tratamiento de las mismas:
  - Se genera el diagrama definitivo
4. Sobre el diagrama definitivo, cumplir con los dos criterios de cobertura: todas las flechas o aristas del mismo ejercitadas, al menos, una vez.

## Posibilidades de mejora

- Existencia de algoritmo de recorrido de grafo y de cálculo de caminos primarios
  - Posibilidad de exportar en XMI los grafos para su tratamiento a partir de las CASE
- Precauciones:
  - Control de caminos imposibles
    - Aplicación de estereotipos a acciones de diagrama
  - Incorporación de entradas y de salida esperada
    - Consideración de pre y poscondiciones
    - Incluir nuevos caminos o restricciones UML
  - Integración con herramientas de requisitos

## Conclusiones

- Buena especificación:
  - Según buenas prácticas
  - Según práctica diaria de organizaciones
- Ventajas de la filosofía de casos de uso
- ¿Pruebas = repetir análisis de aplicación?
  - Retorno de inversión del esfuerzo de análisis
  - Simultanear la especificación con el estudio de las pruebas asociadas
- Prometedor:
  - Vías de automatización: generación/ejecución

