

# Casos de Uso

Fundamentos de Ingeniería de Software/Análisis y Diseño  
de Software

Pablo Cruz Navea – Gastón Márquez  
Departamento de Informática  
Universidad Técnica Federico Santa María



# Motivación

- Hemos visto qué son los requerimientos y la importancia de ellos
- Pero necesitamos ayuda para descubrir y registrar estos requerimientos
- Los ***casos de uso*** son un mecanismo ampliamente utilizado para descubrir y registrar requerimientos (principalmente funcionales)
- Juegan un papel importante en la problemática comunicacional del desarrollo de software

# Definición informal

- Historias que cuentan cómo los usuarios usan un sistema para alcanzar sus objetivos
- Concepto clave: **valor al usuario**
- Ejemplo (caso de uso breve):
  - Un **cliente** llega a una de las cajas con productos para pagar. El **cajero** recibe los productos y utiliza el **sistema** para registrar los productos y obtener un total de compra. El **cliente** entrega el dinero y el **cajero** registra el pago. El **cliente** se retira con los productos comprados.

# Caracterización formal

- **Actor:** alguien o algo que tiene comportamiento. Puede ser una persona (rol), un sistema o una organización
  - Esencial: interactúa con el sistema en forma directa, independiente de si su interés es primario o secundario
- **Escenario:** secuencia específica de acciones e interacciones entre actores y el sistema en discusión. También llamado “instancia de caso de uso”
  - Ej: la historia que describe la compra exitosa de productos
- **Caso de uso:** colección de escenarios (exitosos y fallidos) que describen actores usando un sistema para alcanzar un objetivo
  - Si bien veremos una notación gráfica, los casos de uso son documentos escritos

# Escenario exitoso

- 1) Cliente llega con productos a una de las cajas
- 2) Cajero inicia nueva venta
- 3) Cajero ingresa un ítem (producto)
- 4) SISTEMA registra el producto y entrega el total de compra hasta el momento  
*Cajero repite pasos 3 a 4 hasta ingresar todos los productos*
- 5) SISTEMA muestra el total de compra
- 6) Cajero entrega el total al cliente y solicita el pago
- 7) Cliente entrega el dinero
- 8) Cajero registra el pago
- 9) SISTEMA imprime boleta
- 10) Cajero entrega boleta al cliente
- 11) Cliente se retira con boleta y productos comprados



## Escenario no exitoso (fallido)

- 3.a) Producto no existe (sistema no lo conoce)
  - 1. Sistema rechaza ingreso, muestra error
- 3.b) Sistema no puede contactar al sistema de inventario
  - 1) Sistema se reinicia automáticamente
    - 1.a) Sistema no logra conexión
      - 1) Sistema muestra error
      - 2) Cajero anota en “libro” los productos a llevar

# Escenario alternativo

3-5.a) Cliente solicita a cajero retirar productos (se arrepiente de llevar uno)

1. Cajero ingresa identificador del producto a retirar

2. Sistema muestra total de compra actualizado

7.a) Cliente paga con tarjeta de crédito

En general, todos los cursos alternativos (errores, medios de pagos, etc.) se incluyen en una sola gran sección “**escenarios alternativos**”

# Uniendo escenarios

- **Supuesto clave:** al final de cada escenario alternativo, la “*ejecución*” del caso de uso continúa en su curso normal hasta que finaliza...
- ...salvo que
  - Se especifique lo contrario
    - Por ejemplo: el sistema se detiene y no puede continuar (una especie de BSOD)
- Se sugiere que, en caso de detención del sistema, se especifique qué estados son importantes de almacenar (para posterior recuperación)
  - Por ejemplo: venta rechazada



# Pre condiciones

- **Pre condiciones:** estados que siempre deben ser **verdaderos** antes del comienzo de un escenario
  - Ej: Cajero se encuentra identificado y autenticado
- Las pre condiciones no se prueban (*testean*) en el caso de uso; son requeridas como verdaderas antes del inicio del caso de uso
- Cuidado con precondiciones como:
  - **El sistema debe estar conectado a la corriente**
    - Precondición verdadera; ¡pero su escritura no agrega valor!

# Post condiciones

- **Post condiciones:** estados que deben ser **verdaderos** una vez que se ha “*ejecutado*” **correctamente** el caso de uso (sean escenarios exitosos o alternativos)
  - Ej: Venta es guardada (registrada), Bodega es actualizada, Comisión por venta es registrada. Boleta es generada.
- Similar al caso de las pre condiciones, hay post condiciones que no es necesario escribir (aunque sean verdaderas)
  - Ej: Sistema queda disponible

# Actores

- **Primarios:** son los principales beneficiados cuando se usan los servicios del sistema
  - Ej: cajero, administrador del sistema
- **Secundarios/sopORTE:** juegan un papel de soporte (ofrecen servicios)
  - Ej: sistema de inventario, bodeguero
- **Offstage:** no se benefician directamente ni tampoco ofrecen servicios, pero se interesan (hay interacción) en lo que ocurre en el caso de uso
  - Ej: gobierno (pago de IVA en cada venta)
- ¿Puede ser el **sistema en discusión** un actor?
  - Sí, pero sólo si el sistema *actúa* llamando servicios de otros sistemas
- ¿Puede ser un **cliente** (quien compra productos como en el ejemplo) un actor?
  - Sí, pero sólo si el cliente interactúa con el sistema
    - Ej: caso en supermercados: cliente pasa tarjeta a módulo Transbank y paga directamente

# ¿Y los componentes del software?

- Los casos de uso que hemos visto son del tipo “**caja negra**”
- Esencialmente funcionales
  - Aunque desde ellos podemos determinar algunos requerimientos no funcionales
- Existen tres tipos:
  - **Breve:** narración en un párrafo (ejemplo inicial)
  - **Casual:** narración en un párrafo con lenguaje informal
  - **Completo:** incluyen actores, pre condiciones, post condiciones, curso normal, cursos alternativos
- Por lo tanto, **no es recomendable incluir componentes de software**

# Recapitulando: Casos de uso completos

- Incluyen:
  - Nombre (ej: procesar venta)
    - Verbo + sustantivo (sugiere “acción”)
  - Actores
    - Primarios, secundarios y *offstage*
  - Pre condiciones
  - Post condiciones
  - Escenario normal (exitoso)
  - Escenarios alternativos
- Usualmente, se incluye al comienzo una lista con descripción de los principales *stakeholders*

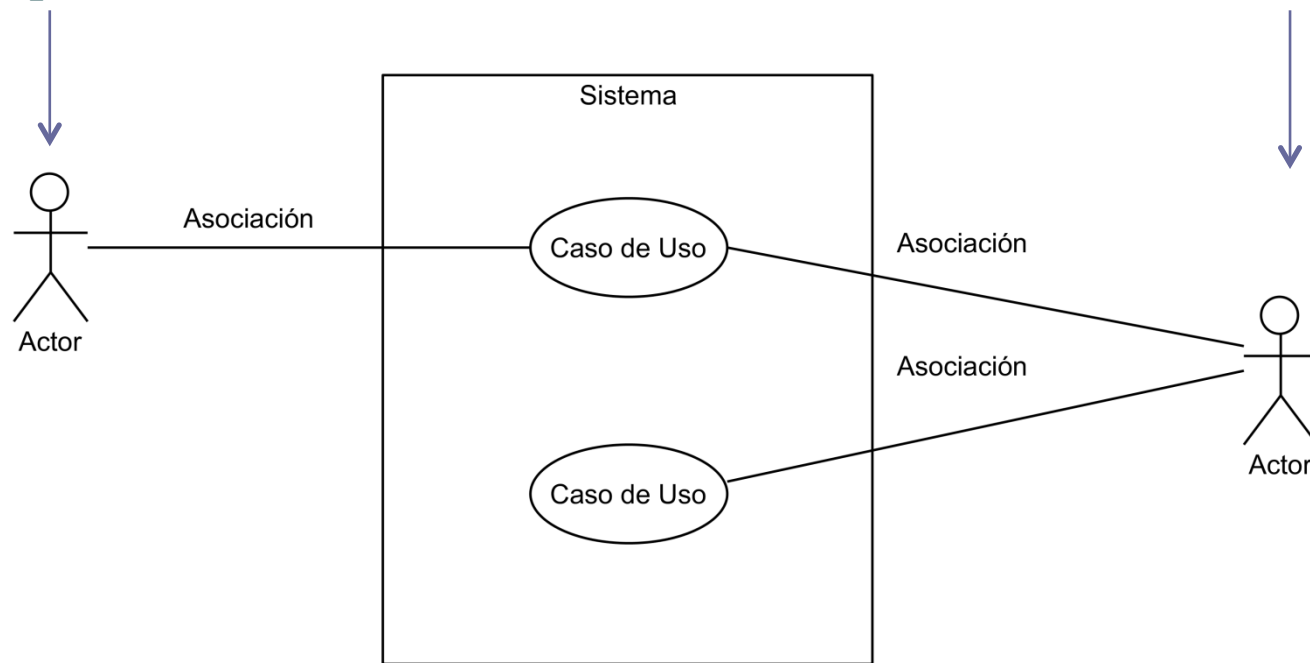
# UML para casos de uso [1]

- UML: Unified Modeling Language
- Lenguaje/especificación de la OMG
- Última versión **formal**: 2.0 (julio de 2005)
  - Última versión disponible: 2.4.1 (agosto de 2011)
- No olvidar:
  - UML es un lenguaje, NO una metodología
  - La notación gráfica para casos de uso es **secundaria**. La esencia de los casos de uso es la escritura de documentos.

# UML para casos de uso [2]

Actores primarios

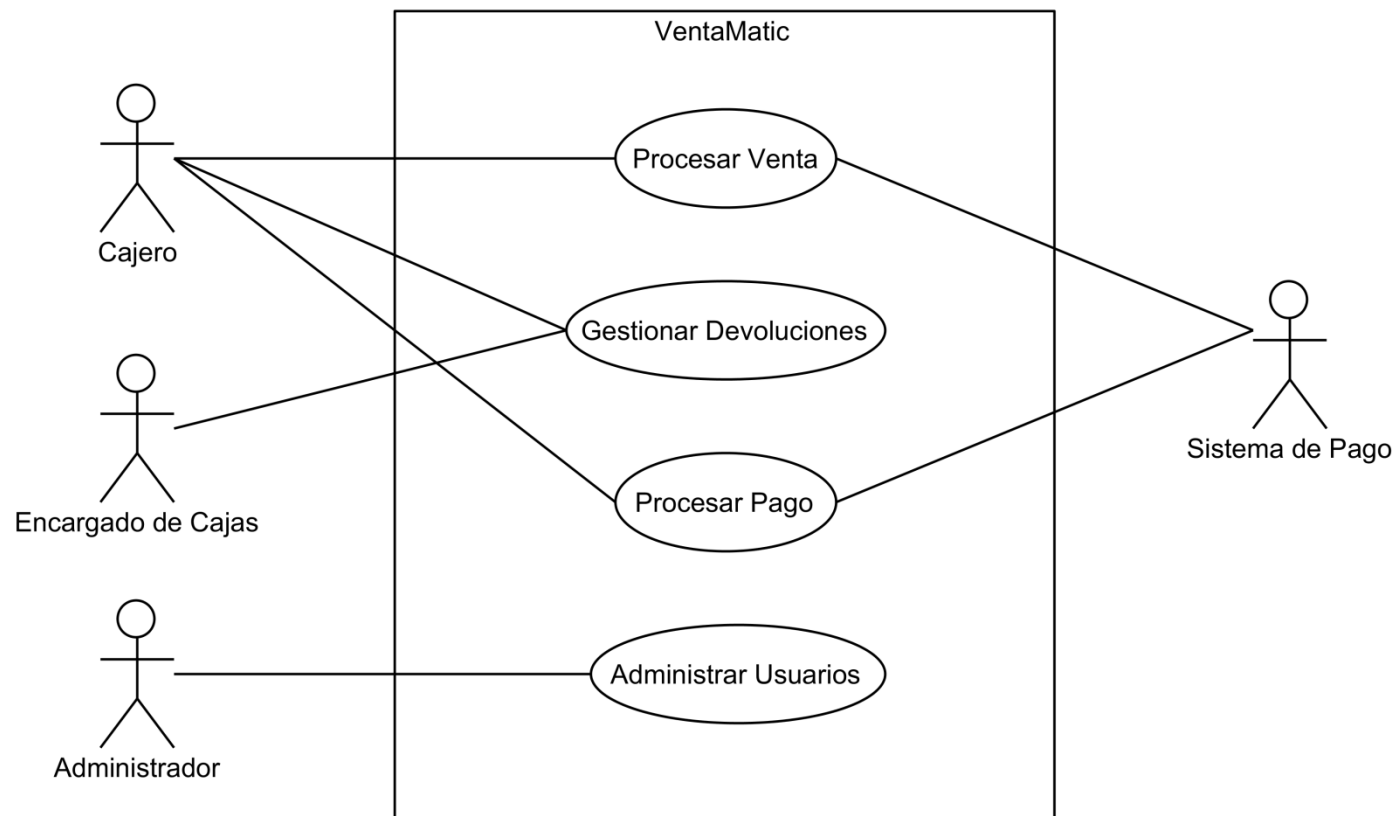
Actores secundarios



Frontera del sistema



# Ejemplo de notación con UML







FIN