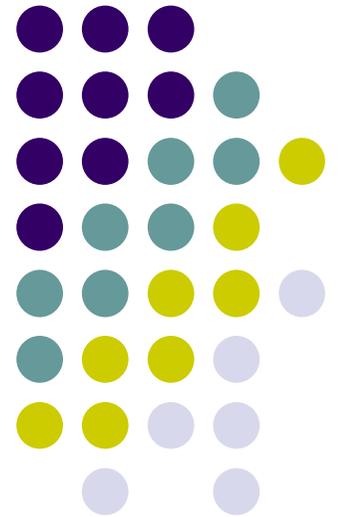


# Metodologías de Diseño y Programación

---

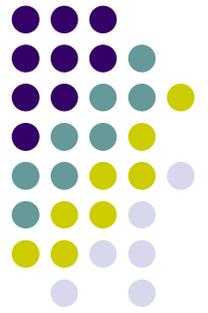
## Requisitos de Software





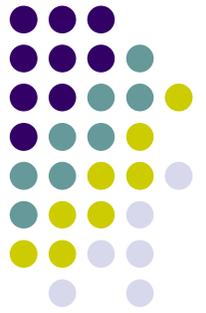
# Contenido

- Especificación de requisitos
- Tipos de requisitos
- Requisitos funcionales
- Casos de uso



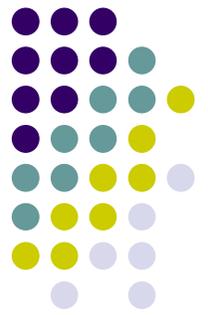
# Especificación de Requisitos

- La Especificación de Requisitos es un insumo fundamental en el desarrollo de software
  - Es la principal fuente de información a partir de la cual se diseña, implementa y prueba el sistema
  - Representa un contrato con el usuario
  - Es uno de los aspectos más delicados de un proyecto
    - Es algo complejo de obtener
    - De su correctitud depende el éxito del proyecto
- No se genera por completo al inicio del proyecto sino incrementalmente
- Suele presentarse como la agregación de diferentes artefactos



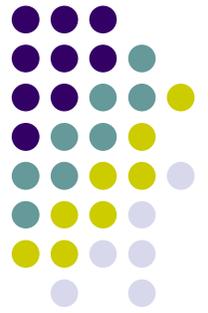
# Requisitos

- Un requisito de software es una condición o capacidad que un sistema debe cumplir
- Requisito No Funcional
  - Expresa una propiedad o cualidad que el sistema debe presentar
  - También restricciones físicas sobre los funcionales
- Requisito Funcional
  - Expresa una acción que debe ser capaz de realizar el sistema
  - Especifica comportamiento de entrada/salida



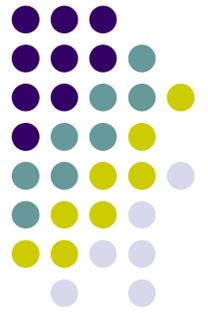
# Requisitos No Funcionales

- Los requisitos no funcionales suelen referir a
  - **Usabilidad:** factores humanos, ayuda, documentación
  - **Confiabilidad:** frecuencia de fallas, tiempo de recuperación
  - **Performance:** tiempo de respuesta, tasa de procesamiento, precisión, capacidad de carga
  - **Soportabilidad:** adaptabilidad, mantenibilidad, configurabilidad, internacionalización



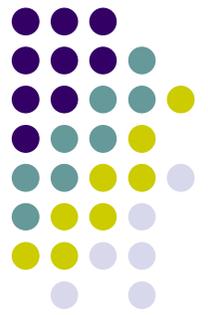
# Requisitos No Funcionales

- y más
  - **Interfaces:** restricciones en la comunicación con sistemas externos
  - **Restricciones:** en el uso de
    - Sistemas o paquetes existentes
    - Plataformas
    - Lenguajes de programación
    - Ambientes de desarrollo
    - Herramientas (sistemas de bases de datos, middleware, etc.)



# Requisitos Funcionales

- Los requisitos funcionales se expresaban en términos de “funciones del sistema”
- Una función del sistema es algo puntual que el sistema debe hacer
- Técnica básica: Si X es una función del sistema, la frase “El sistema debe hacer X” tiene que tener sentido



# Requisitos Funcionales

- Esta herramienta funcionó bien mientras
  - Las funcionalidades de los sistemas eran sencillas
  - El modelo de uso de los usuarios se limitaba a una interacción simple y puntual con el sistema
- Perdió su efectividad cuando las interacciones a soportar se volvieron más complejas tomando períodos de tiempo más prolongados
- Nuevo enfoque: Casos de Uso



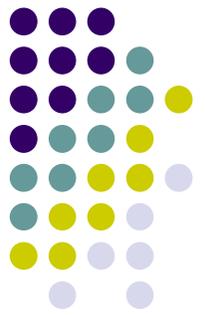
# Casos de Uso

- El enfoque de casos de uso está basado en la noción de actor
- Un actor es un agente externo (humano o no) que interactúa directamente con el sistema
- Un caso de uso narra la historia completa (junto a todas sus variantes) de un conjunto de actores mientras usan el sistema
  - La historia termina cuando uno de los actores (el principal) logra su objetivo y obtiene un resultado de valor



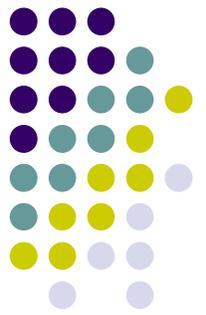
# Casos de Uso (2)

- Los casos de uso son la herramienta más aplicada para la especificación de requisitos funcionales
- Por ser expresados textualmente resultan simples de comprender (hasta para personal no-técnico)
- Por estar orientados a los objetivos de los actores (y al camino hacia su obtención)
  - Son intuitivos
  - Propician la completitud de especificación



# Casos de Uso (3)

- Los casos de uso no suelen especificarse con todo detalle de una sola vez
- Esto se realiza en forma gradual y posterior a la identificación de actores
  - Los actores son más fáciles de identificar y sus necesidades son las que dan lugar a los casos de uso
- Formas posibles de un caso de uso
  - Identificado o detectado: solo su nombre y actores participantes
  - Especificado en alto nivel: se incorpora una sinopsis
  - Especificado en forma expandida: se incorpora la “historia” de uso y sus variantes



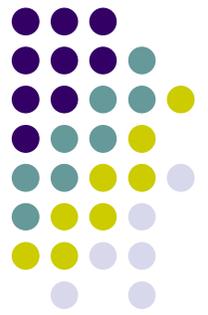
# Casos de Uso (4)

- Método básico (variable según el avance):
  1. Detectar actores
  2. Identificar algunos casos de uso  
(detectando objetivos y necesidades de actores)
  3. Especificarlos en alto nivel
  4. Examinarlos y expandir algunos de ellos
  5. ...
- A medida que se avanza en el desarrollo se detectan nuevos casos de uso y se especifican otros ya detectados



# Casos de Uso (5)

- Sobre el principio del proyecto se tiende a buscar y especificar los casos de uso más importantes
- Los casos de uso se usan además como criterio de partición del problema en un proceso iterativo e incremental
  - En una iteración se desarrolla “uno a la vez”
  - Los incrementos no refieren a “partes” físicas sino a conjuntos de funcionalidades



# Ejemplo (Formato Expandido)

**Caso de Uso:** Realizar una compra

**Actores:** Cajero

**Sinopsis:** Un cliente llega a la caja con artículos para comprar. El cajero registra los artículos y recibe el pago. Al finalizar, el Cliente se retira con los artículos.

**Escenario Típico:**

1. El Cliente llega a la caja con artículos para comprar.
2. El Cajero comienza un nueva venta.
3. El Cajero ingresa el identificador del artículo.
4. El Sistema registra el artículo y presenta su descripción, precio y subtotal.  
*El Cajero repite los pasos 3 y 4 hasta terminar los artículos.*
5. El Sistema presenta el total con los impuestos incluidos.
6. ...



# ¿Qué sigue después?

- Una vez detectado y especificado el conjunto inicial de casos de uso:

El equipo de desarrollo está listo para analizarlos, diseñar una solución para ellos e implementarlos

- Mientras tanto el equipo de analistas avanza en la detección y especificación de otros