

# Ingeniería de Software

Agustín J. González

EIO329: Diseño y Programación Orientados a Objeto

Adaptado de: material asignatura CS169, Software Engineering, UC Berkeley, entre otras fuentes.

# Definiciones de Ingeniería de Software

- (1993) La aplicación de mecanismos sistemáticos, disciplinados, y cuantificables para el desarrollo, operación y mantenimiento de software; esto es la aplicación de la ingeniería al software.
- Establecimiento y uso de principios con caracteres de ingeniería apropiados para obtener, eficientemente, software confiable, que opere eficaz y eficientemente en máquinas reales
- La aplicación del arte del desarrollo software junto con las ciencias matemáticas y computadores para **diseñar, construir, y mantener programas computacionales eficientes y económicos** que **logran sus objetivos**.
- Wikipedia: “Software engineering is a profession and field of study dedicated to **designing, implementing, and modifying** software so that it is of **higher quality, more affordable, maintainable, and faster to build.**”
- Se busca: Resolver el problema a costo y en tiempo controlados.

# Estado del arte en Ing. de Software

- ¿Es una ciencia rigurosa con fuertes fundamentos matemáticos?
- ¿Es un campo técnico bien desarrollado con mucho de disciplina de ingeniería?
- O está realmente en un estado primitivo...
  - A lo más una serie de “mejores prácticas”, desarrolladores de software construyen software y si éstos funcionan, entonces nosotros estudiamos cómo ellos lo hicieron.
  - Si éstos funcionan por un largo tiempo, entonces estudiamos sus procesos de software aun más cuidadosamente.

# Construcción de una casa para “fido”



La puede hacer una sola persona

Requiere:

Modelado mínimo

Proceso simple

Herramientas simples

# Construcción de una casa

Construida eficientemente y en un tiempo razonable por un equipo

Requiere:

- Modelado

- Proceso bien definido

- Herramientas más sofisticadas



# Construcción de un rascacielos



- Moraleja: No es lo mismo hacer una tarea en programación 1er. Año que desarrollar un software tipo Eclipse, Jgrasp, Matlab, Matemática, u otro.
- ¿Cómo debe cambiar el proceso de desarrollo?

# Claves en Desarrollo de IS

## Notación (UML)



**Herramientas**  
(Ej: Rational Rose,  
Umbrello, IDEs)

**Proceso**  
(Metodologías  
Ej: ITIL, Extreme Programming,  
RUP: Rational Unified Process,  
Personal Software Process)

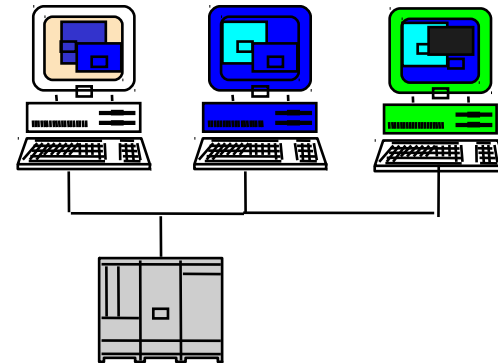
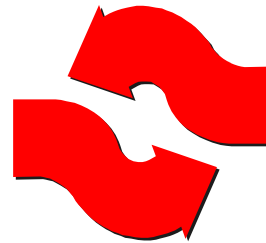
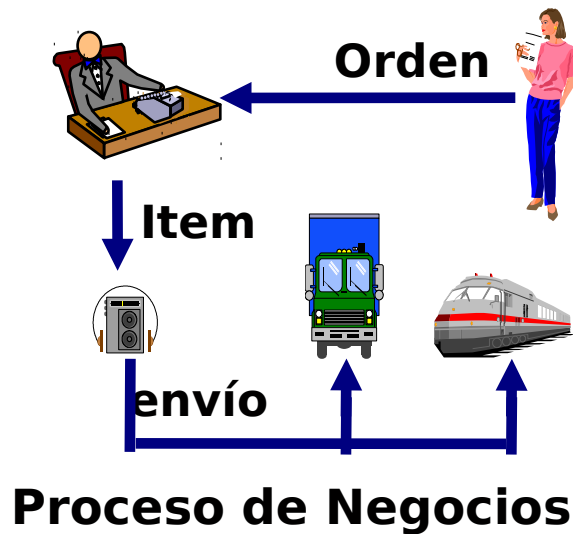


# Abstracción - Modelado Visual (MV)

*“El modelado captura las partes esenciales del sistema”*

*En otras disciplinas se conoce el beneficio de tener representaciones visuales del modelo.*

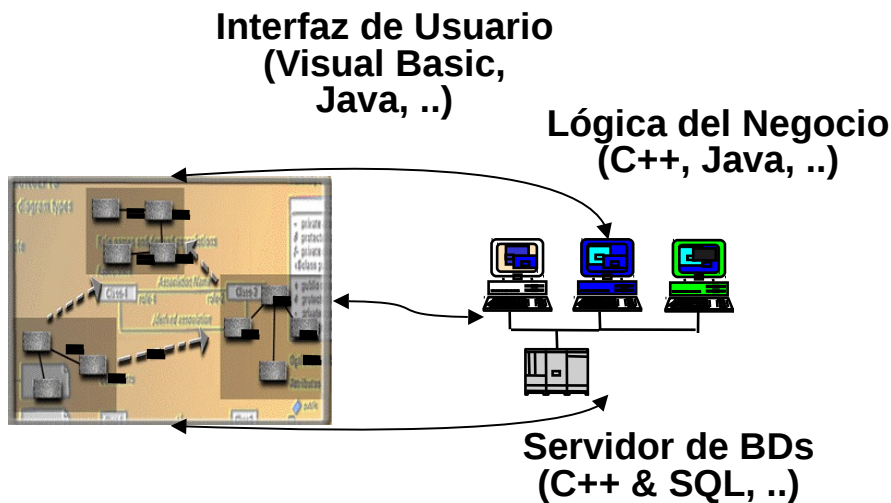
*Ej. Plano circuito, plano arquitectura, diagrama cuerpo libre*



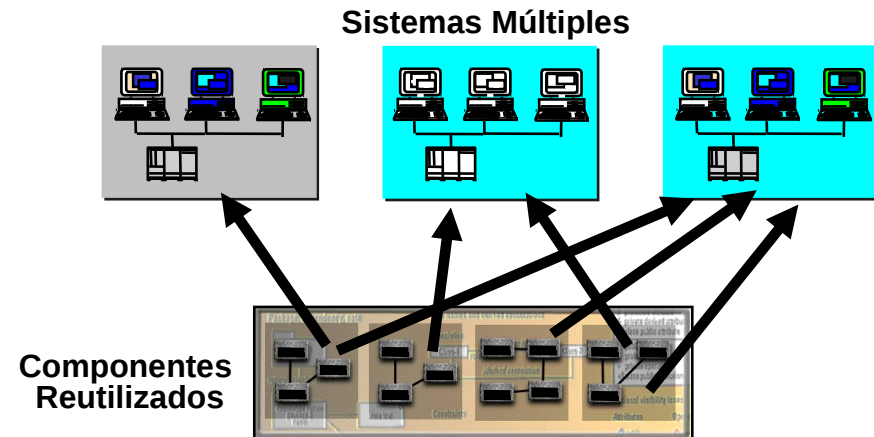
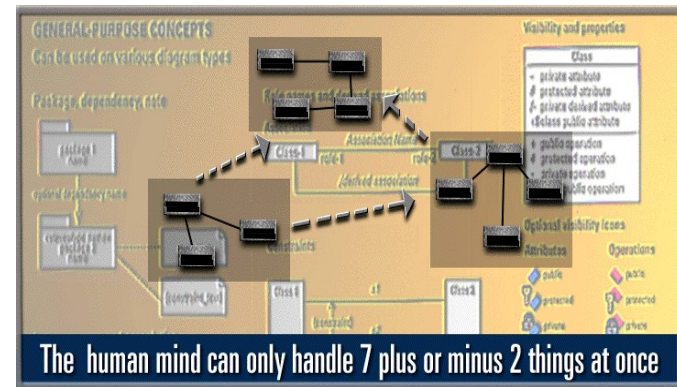
**Sistema Computacional**

# Notación (Visual) - Beneficios

## Manejar la complejidad



“Modelar el sistema independientemente del lenguaje de implementación”



Promover la Reutilización

# ¿Por qué la Orientación a Objetos?

- Por su proximidad a las entidades del mundo real que surgen del modelado.
  - Mejora la captura y validación de requisitos
  - Acerca el “espacio del problema” al “espacio de la solución”
- Modelado integrado de propiedades estáticas (estado de cada ente) y dinámicas (interacción entre entes) del ámbito del problema
  - Facilita construcción, mantenimiento y reutilización
- Su diseño facilita:
  - la creación de **Abstracciones** (Ignorar detalles)
  - la **Modularización** (separación en módulos)
  - **Ocultar información** (separar la implementación del uso)
- Podríamos dar muchas razones pero hay problemas.

# Problemas en OO

- Un objeto contiene datos y operaciones que manipulan los datos, pero ...
- Podemos distinguir dos tipos de objetos degenerados:
  - Un objeto sin datos (que sería lo mismo que una biblioteca de funciones). Si los métodos son estáticos, “peor” aún.
  - Un objeto sin “operaciones”, con sólo atributos públicos (que equivaldría a las estructuras de datos tradicionales)
- Un sistema construido con objetos degenerados no es un sistema verdaderamente orientado a objetos.

# Moraleja

- Las técnicas deben ser utilizadas para obtener de ellas su real beneficio.
- Estudiaremos algunas metodologías
- Qué viene ahora: Pongamos el desarrollo de software en perspectiva: Conozcamos cuál es el proceso de desarrollo de software.