

Introducción

Análisis y Diseño de Software / Fundamentos de
Ingeniería de Software

A series of horizontal lines of varying lengths and colors (teal, light blue, and white) extending from the left edge of the slide towards the right, positioned below the main title.

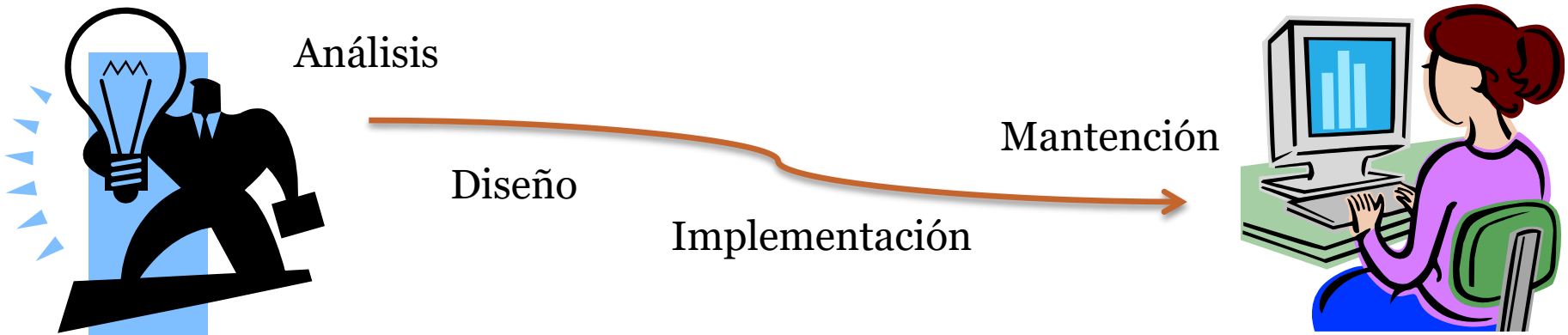
Pablo Cruz Navea
Departamento de Informática
Universidad Técnica Federico Santa María

Caracterizando al Software

- Software: término genérico que describe “programas computacionales”
- Pero hoy, software es mucho más que un algoritmo, una tarea o un programa sencillo
- Responde a una o más necesidades
- Complejidad viene dada por:
 - Tamaño
 - Interacción de componentes
 - Requiere equipos para su construcción (típicamente)
 - Usuario quiere versiones intermedias (antes del software completo)

¿Por qué es complejo construir software? [1]

- La construcción de software parte con una idea
- Existe una gran distancia entre la idea y el software en ejecución (correctamente)



¿Por qué es complejo construir software? [2]

- Ideas no son *generación espontánea*
 - Proviene de necesidades de personas
- **Problema:** las personas saben que tienen necesidades, pero no saben con precisión cuáles son esas necesidades
- Diversas presiones:
 - Monetarias (presupuestos deben cumplirse)
 - Tiempos (software debe llegar en el momento preciso)
 - Utilidad (el software debe servir al propósito)

¿Por qué es complejo construir software? [3]

- Muchos usuarios, muchos requerimientos
 - Pero sólo un “software”
- Cliente se confunde con usuario
 - Ambos involucrados, pero no siempre el que *paga* es el que *usa*
- Representantes de usuarios dicen lo que ellos quieren, no necesariamente lo que facilita el trabajo de los usuarios finales

Ingeniería de Software [1]

- Disciplina que se ocupa de la construcción:
 - Sistemática
 - Eficaz
 - Eficiente
- de sistemas de software:
 - Eficaces
 - Eficientes
- Distinción clave: Proceso v/s producto

Ingeniería de Software [2]

- **Proceso:**
 - Sistemático: técnicas y criterios probados y reproducibles
 - Eficaz: orientado al logro de objetivos del cliente
 - Eficiente: minimizando uso de recursos del desarrollador
- **Producto:**
 - Eficaz: hace lo que se desea
 - Eficiente: ocupa pocos recursos de máquina y de usuarios

Ingeniería de Software: conceptos clave

- Personas (y equipos de personas)
- Herramientas (para desarrollo y apoyo)
- Productos de trabajo (código, documentos, resultados)
- Procesos: “pegamento” para las personas, herramientas y productos de trabajo
 - Una vez definidos:
 - Permiten razonar sobre ellos y sobre lo que producen
 - Permiten escalar en cantidad y tamaño de proyectos

FIN