

*Este material está basado en el curso preparado por A. Navarro, UCM  
(que a su vez sigue el texto del libro de Pressman)*

## El Producto: Software

---

*Ingeniería del Software de Gestión 1  
Facultad de Informática*



Juan Pavón Mestras  
Dep. Sistemas Informáticos y Programación  
Universidad Complutense Madrid

<http://www.fdi.ucm.es/profesor/jpavon>

## Objetivos

---

- Entender lo que es el software y los problemas que conlleva su desarrollo
- Motivar la necesidad de IS

## Evolución del software

---

- Hace dos décadas (años 80):
  - Centrados en el hardware, factor principal en el presupuesto
    - Ingeniería del hardware, pero no del software
  - Poca difusión software
  - Software de venta específico para el hardware
  - No se vendían sistemas operativos sin hardware
  
- Actualmente:
  - El software es el factor principal en el presupuesto
  - Software con alto tiempo de desarrollo, incluso fuera de plazo → Costes elevados
  - Software entregado a clientes con errores (defectos)
  - Gran difusión del software
    - Los sistemas operativos se anuncian en prensa y televisión
  - ¿Quiénes son los mayores millonarios del mundo?

## Evolución del software

---

- Evolución del software ligada al hardware
  - Mayor complejidad hardware → mayor complejidad software → mayor complejidad en el desarrollo de software
  - Al principio, software añadido a computadora
  
- Evolución del desarrollo de software
  - Programar arte, debido a la falta de métodos sistemáticos
  - Desarrollo de software sin planificación → proyectos sin control → costes imprevisibles
  - Baja movilidad de programadores → falta de documentación sistematizada.

## Evolución del software

---

- Los primeros años (1950-1965 aprox.)
  - Proceso por lotes (*batch*)
  - Distribución limitada
  - Software a medida
  
- La segunda era (1965-1975 aprox.)
  - Sistema multiusuario
    - Sistemas Interactivos  
HCI: Human Computer Interaction (<http://www.hcibib.org/hci-sites/>)
  - Tiempo real
  - Bases de Datos
  - Productos software independientes del hardware
    - mantenimiento del software con versiones

## Evolución del software

---

- La tercera era (1975-1985 aprox.)
  - Sistemas distribuidos
  - Incorporación de "inteligencia"
  - Hardware de bajo coste (microprocesador)
  - Impacto en el consumo
  
- La cuarta era (1985-2000 aprox.)
  - Sistemas personales potentes
  - Tecnologías orientadas a objetos
  - Redes de computadoras
  - Computación en paralelo
  - Técnicas de inteligencia artificial

## Evolución del software

---

- Etapa actual (principios del tercer milenio)
  - Componentes y arquitecturas software reutilizables
  - Web semántica
  - Computación ubicua
  - Interfaces multi-modales

## Evolución del software

---

- Problemas persistentes en la evolución:
  - El software nunca explota las posibilidades plenas del hardware
  - El desarrollo del software no es tan rápido como su demanda
  - Sociedad dependiente de las computadoras → necesitamos software fiable
  - Los programas no son escalables ni mantenibles por culpa de diseños pobres y recursos inadecuados
- Cuestión: ¿Aumentan estos problemas?

## Perspectiva industrial

---

- Cuestiones
  - ¿Por qué lleva tanto tiempo terminar los programas?
  - ¿Por qué es tan elevado el coste?
  - ¿Por qué no podemos encontrar todos los errores antes de entregar el producto a los clientes?
  - ¿Por qué nos resulta difícil constatar el progreso conforme se desarrolla el software?
  - ¿Renovar software o reemplazarlo por nuevo?
  
- El desarrollo de software pasa de ser arte a ser un proceso industrial
  - La fábrica de software (*Software Factory*)
  - El software es una empresa muy competitiva
  - Subcontratación de terceros

## Software

---

- Definición (Pressman):
  - Instrucciones (programas de computadora) que cuando se ejecutan proporcionan la función y el rendimiento deseados
  - Estructuras de datos que permiten a los programas manipular adecuadamente la información
  - Documentos que describen la construcción y uso de programas

## Características del software

---

- El software se desarrolla, no se fabrica
  - Los costes se centran en ingeniería, no en fabricación
  - Los proyectos software no se pueden gestionar como procesos de fabricación
- El software no se estropea

## Características del software

---

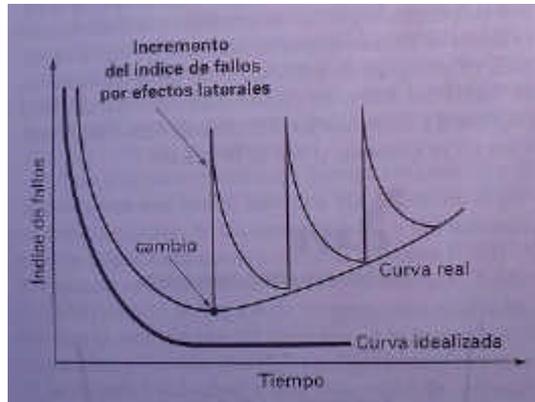
- Curvas de fallos



Curva de fallos del hardware (bañera)

## Características del software

---



Curvas de fallos software

## Características del software

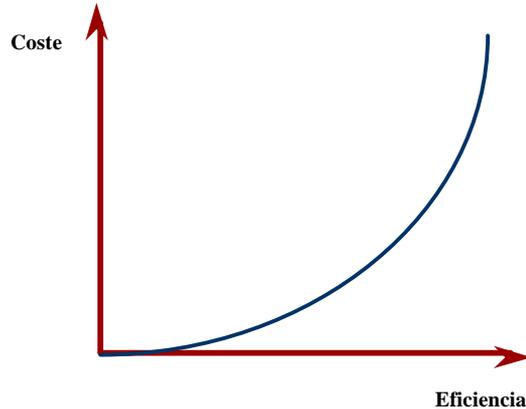
---

- Reparación del software
  - El software deteriorado no se puede reparar
    - ¿revisar miles de líneas de código?
  - Muchas veces las reparaciones dañan más al software
  - El software debe estar *bien diseñado* para facilitar su evolución

## Características del software

---

### Coste de la eficiencia del software



## Software bien diseñado

---

- Software bien diseñado
  - Ingeniería: creación y mantenimiento de una serie de componentes estándar con el fin de no reinventar la rueda
  - Software bien diseñado debe favorecer la reutilización de código
  - Las tecnologías OO y de componentes software reutilizables favorecen dicha reutilización

## Software bien diseñado

---

- Atributos del software bien diseñado
  - Mantenible
    - Capaz de evolucionar según las necesidades de cambio de los clientes
  - Seguro
    - Robusto, que no produce daños incluso bajo un fallo del sistema
  - Eficiente
    - No desperdicia los recursos del sistema (memoria, procesador, disco)
  - Amigoso
    - Buena interfaz
  - Bien documentado

Atributos en tensión: Su importancia depende del sistema y del entorno en el que será utilizado

El coste tiende a ser alto si se exige un alto nivel de alguna características

## Software

---

- Podremos construir una aplicación software cuando se haya definido previamente un conjunto específico de pasos procedimentales, es decir, un algoritmo
  - Discusión: *¿es esta una condición necesaria o suficiente?*

## Tipos de software

---

- Por su estructura
  - Funcional
  - Orientado a listas
  - Orientado a objetos
  - Orientado a componentes
- Por su función
  - Producto. Programas que gestionan y transforman información
    - Programas o Sistemas de Usuario
    - CAD
    - Procesadores de Texto
    - Aplicaciones Web
    - Etc.
  - Herramientas para desarrollar y ejecutar productos
    - Sistemas operativos
    - Middleware
    - Compiladores
    - Bases de datos
    - Librerías
      - Interfaces Hombre-Maquina

## Tipos de software

---

- Por su plataforma de ejecución
  - Sistemas embebidos
  - Sistemas de computación distribuida
  - Sistemas de computación paralela
  - Sistemas de tiempo real
  - Sistemas basados en Chips
  - Sistemas de computo ubicuos
  - *Wearable computing systems*

## Aplicaciones

---

- Software de sistemas
  - Programas escritos para servir a otros programas
    - Compiladores, Sistemas Operativos (SOs), etc.
  - Características:
    - Fuerte interacción con el hardware de computadora
    - Múltiples usuarios
    - Operación concurrente
    - Compartición de recursos
    - Estructuras de datos complejas

## Aplicaciones

---

- Software de tiempo real
  - Mide, analiza y controla sucesos del mundo real conforme ocurren
    - Control de aviones, procesos de fabricación, etc.
    - Sistemas de tiempo real estricto (*hard real time*) o no (*soft real time*)
  - Componentes:
    - Adquisición de datos. Recolecta y da formato a la información recibida del entorno externo
    - Análisis. Transforma la información según lo requiere la aplicación
    - Control/salida. Responde al entorno externo
    - Monitorización. Coordina a los demás componentes para obtener una respuesta en tiempo real (de 1 milisegundo a 1 minuto)

## Aplicaciones

---

- Software de gestión
  - Proceso de información comercial, accediendo a Bases de datos que contienen dicha información
    - Gestión de nóminas, control de almacén, etc.
  
- Software de ingeniería y científico
  - Algoritmos numéricos
    - Programas CAD, predicción meteorológica, etc.

## Aplicaciones

---

- Software de PC
  - Se venden en la gran distribución
    - Procesadores de texto, hojas de cálculo, etc.
  
- Software empotrado (*embedded systems*)
  - Controla productos y sistemas de mercados industriales y de consumo
  - Control de fábricas, etc.
  - Reside en ROM
  - Relacionado con el tiempo real

## Aplicaciones

---

- Software de inteligencia artificial
  - Algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o análisis directo
    - Sistemas expertos
    - Reconocimiento de patrones (voz, imágenes, etc.)
    - Agentes software

## De vuelta con los mitos del software

---

- Mitos del gestor
  - Mito: Tenemos un manual de desarrollo de software, ¿qué más necesitamos?
    - Realidad: ¿Se entiende? ¿Se utiliza? ¿El personal tiene práctica en su aplicación?
  - Mito: Disponemos de las herramientas de desarrollo más avanzadas, ya que compramos siempre los mejores equipos
    - Realidad: ¿Se invierte en herramientas CASE? ¿Y en entornos de desarrollo? ¿Se forma al personal en el uso de estas herramientas?
  - Mito: Si fallamos en la planificación, podemos añadir más programadores y adelantar el tiempo perdido
    - Realidad: En el proceso de software añadir gente puede retrasar más el proyecto. La gente debe añadirse de forma planificada y ordenada. Además si sacamos a gente de otros proyectos, en último término retrasaremos otros proyectos

## De vuelta con los mitos del software

---

- Mitos del cliente
  - Mito: Una declaración general de objetivos es suficiente para comenzar a escribir los programas, y podemos dar los detalles más adelante
    - Realidad: Una mala definición inicial conlleva trabajo inútil
  - Mito: Los requisitos del proyecto cambian continuamente, pero los cambios pueden acomodarse fácilmente porque el software es flexible
    - Realidad: Es cierto que los requisitos cambian, pero el impacto del cambio varía en función del momento en que se introduzcan los cambios

## De vuelta con los mitos del software

---

<b>Impacto del cambio</b>	
<b>Momento</b>	<b>Coste del cambio</b>
Definición	1x
Desarrollo	1,5-6x
Después entrega	60-100x

## De vuelta con los mitos del software

---

- Mitos de los desarrolladores
  - Mito: Una vez que escribimos el programa y hagamos que funciones, nuestro trabajo ha terminado
    - Realidad: Entre el cincuenta y el setenta por ciento de todo el esfuerzo dedicado a un programa se realiza después de que se entregue al cliente por primera vez
  - Mito: Hasta que no tenga el programa ejecutándose, no tengo forma de medir su calidad
    - Realidad: Revisiones Técnicas Formales durante el desarrollo de software.
  - Mito: Lo último que se entrega al terminar el proyecto es el programa funcionando
    - Realidad: Software = programas + datos + documentos

## Conclusiones

---

- Software: producto de desarrollo (más que de fabricación)
- Software: rápida evolución
- Constante en la evolución: problemas que se pretende solucionar con IS
- ¡Cuidado con los mitos!