

## Introducción

“La mayoría de expertos está de acuerdo en que la causa más probable de destrucción del mundo sería por accidente; y aquí es donde entramos nosotros: somos profesionales de la informática, causamos accidentes”

Nathaniel Borenstein

# ¿Dónde hay software?



# Tipos de Software

- Software de sistemas
  - Conjunto de programas que han sido escritos para servir a otros programas (Compiladores, editores, utilidades de manejo de periféricos...)
- Software de tiempo real
  - Coordina/analiza/controla sucesos del mundo real conforme ocurren
- Software de gestión
  - Tratamiento de la información comercial (nóminas, inventarios...)
- Software de ingeniería y científico
  - Algoritmos de manejo de números (biología molecular, dinámica orbital...)

# Tipos de Software

- Software empujado
  - Reside en memoria de sólo lectura (control de teclas de un microondas, control de la gasolina...)
- Software de ordenadores personales
  - Procesamiento de textos, hojas de cálculo, gráficos, entretenimiento...
- Software de inteligencia artificial
  - Algoritmos no numéricos para resolver problemas complejos (redes neuronales, sistemas expertos...)
- Software basado en web
  - Software accedido a través de un navegador Web

# Reflexiona un momento

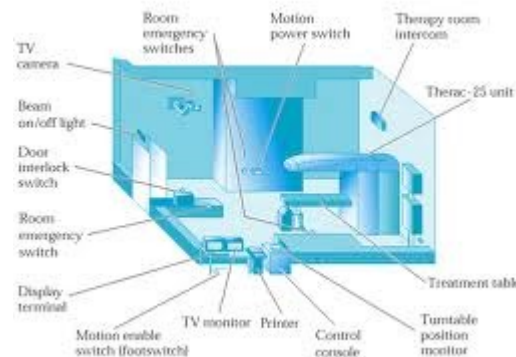
- ¿Viajarías en un avión cuyo software ha sido desarrollado por ti?
- ¿Confiarías tus transacciones bancarias a un sistema diseñado por ti?
- ¿Viajarías en un tren de alta velocidad cuyo software de control ha sido desarrollado por ti?

**¿TIENES DUDAS?**

**Veamos algunos hechos históricos**

# Desastres en la historia

- El **Therac-25** era una máquina utilizada en **terapia de radiación**, producida por *Atomic Energy of Canada Limited*. **Varias muertes** de pacientes de cáncer entre 1985-1987 se debieron a una **sobredosis de radiación** debida a un **problema en las tareas concurrentes en el software**
  - [Leveson y Turner, 1993], [Leveson, 1995]





# Desastres en la historia

- 2 Billones de dólares perdidos al no poder poner en marcha el aeropuerto de Denver (USA) por culpa del **software de control del sistema de traslado de equipajes** (fecha prevista apertura 1-noviembre-93; abrió el 28-febrero-95, retraso de 16 meses).
  - *Computer, Febrero, 1995; (Glass 98)*
- Diez minutos antes de su aterrizaje previsto en Marte, se perdió el contacto con la nave Mars Polar Lander. Un **error de software hizo creer a uno de los brazos de la sonda que ya había tocado suelo**, cuando aún se encontraba 40m de altura.
  - *(3 de diciembre 1999)*



# Desastres en la historia

- **Un sensor mal programado por Francia, destruyó el supercohete europeo Ariane 5**
  - *(El País, 23 de junio de 1996, nº 33-1996)*
- Error ocurrido en la conversión de datos de coma flotante, 64 bits, a valor entero con signo, de 16 bits. El valor real a ser convertido tenía un valor mayor de lo que podía representarse con el entero de 16 bits
- Ese valor demasiado grande ocurrió como resultado de mantener un requisito del software del Ariane 4, no necesaria para el Ariane 5, relacionado con la velocidad horizontal y ángulo de ataque detectado por un sensor de vuelo





# Desastres en la historia

- El 11 de diciembre de **2000**, un **helicóptero MV-22 Osprey del ejército de EEUU** se estrelló cerca de Tucson. Murieron 4 soldados. Las investigaciones posteriores revelaron que hubo un **error informático en el sistema de alarmas de fallos hidráulicos**. No se activó la alarma que debía advertir de un incidente en este sistema, y el piloto no obtuvo información fiable para volar



# Desastres en la historia

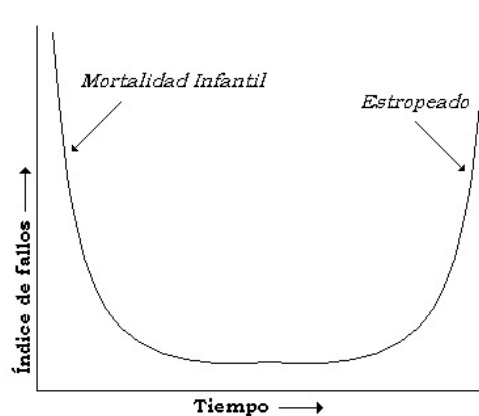
- Problema del año 2000 (Y2K)
  - Error de software causado por utilizar sólo dos dígitos para representar el año. Después del 31 de diciembre de 1999, sería el 1 de enero de 1900 en vez de 1 de enero de 2000.
  - Efectos:
    - Si una persona nació en 1977, la edad de esta persona en 2000 sería:  $00-77 = -77$  años.
    - El problema del año 1900, común en programas que cuentan el año utilizando los dos últimos dígitos y muestran los dos primeros dígitos del año como 19. Al intentar mostrar el año después del 1999, el programa muestra 19 y luego el número que sigue el 99, o sea: 19**100**

# Problemas en un futuro

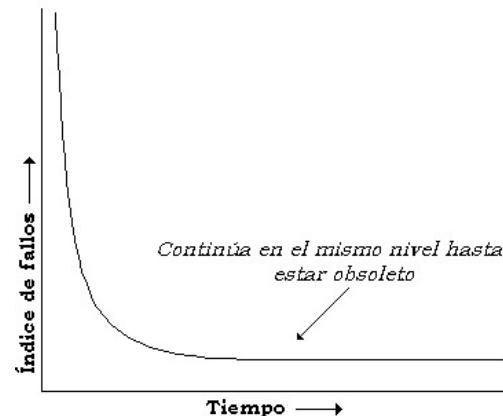
- El **problema del año 2038** podría causar que una parte del software falle en ese año. El problema afecta a los programas que usen la representación del tiempo basada en el sistema POSIX, que se basa en contar el número de segundos transcurridos desde el 1 de enero de 1970 a las 00:00:00
- En la mayoría de sistemas de 32 bits, el tipo de dato `time_t` usado para guardar el contador de segundos es un entero de 32 bits con signo
- El último segundo representable con este formato será a las 03:14:07 del 19 de enero de 2038

# Características del software

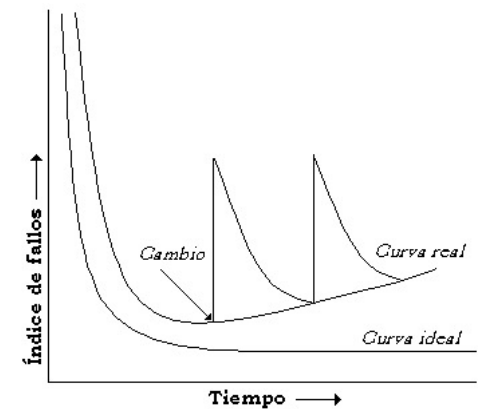
- El software se desarrolla, no se fabrica
  - Los costes del software se concentran en la ingeniería
- El software no se estropea, se deteriora
  - Cambios en la fase de mantenimiento



Curva de fallos del hardware



Curva ideal de fallos del software



Curva real de fallos del software

- A pesar de que la industria tiene una tendencia hacia la construcción por componentes, la mayoría del software aún se construye a medida

# Desarrollar SW vs construir edificios

- Si comparamos saber desarrollar software con saber construir edificios, lo que uno sabe hacer básicamente es “pegar ladrillos”
- Aunque los ladrillos son la base principal del producto final, existen una serie de roles sin los cuales un edificio no podría ser construido
  - Maestros de obra, arquitectos, ingenieros civiles...
- Un programador es sólo un rol del conjunto de roles implicados en el proceso de desarrollo del software

# Factores de calidad del SW

## Correcto

- Se ajusta a las especificaciones dadas por el usuario

## Fiable

- Capacidad de ofrecer los mismos resultados bajo las mismas condiciones

## No erróneo

- No existe diferencia entre los valores reales y los calculados

## Eficiente

- Utilización óptima de los recursos de la máquina

## Robusto

- No poseer un comportamiento catastrófico ante situaciones excepcionales (tolerante a fallos)

## Portable

- Capaz de integrarse en entornos distintos con el mínimo esfuerzo

## Adaptable (extensible)

- Modificar alguna función sin que afecte a sus actividades

## Inteligible

- Diseño claro, bien estructurado y documentado

## Reutilizable

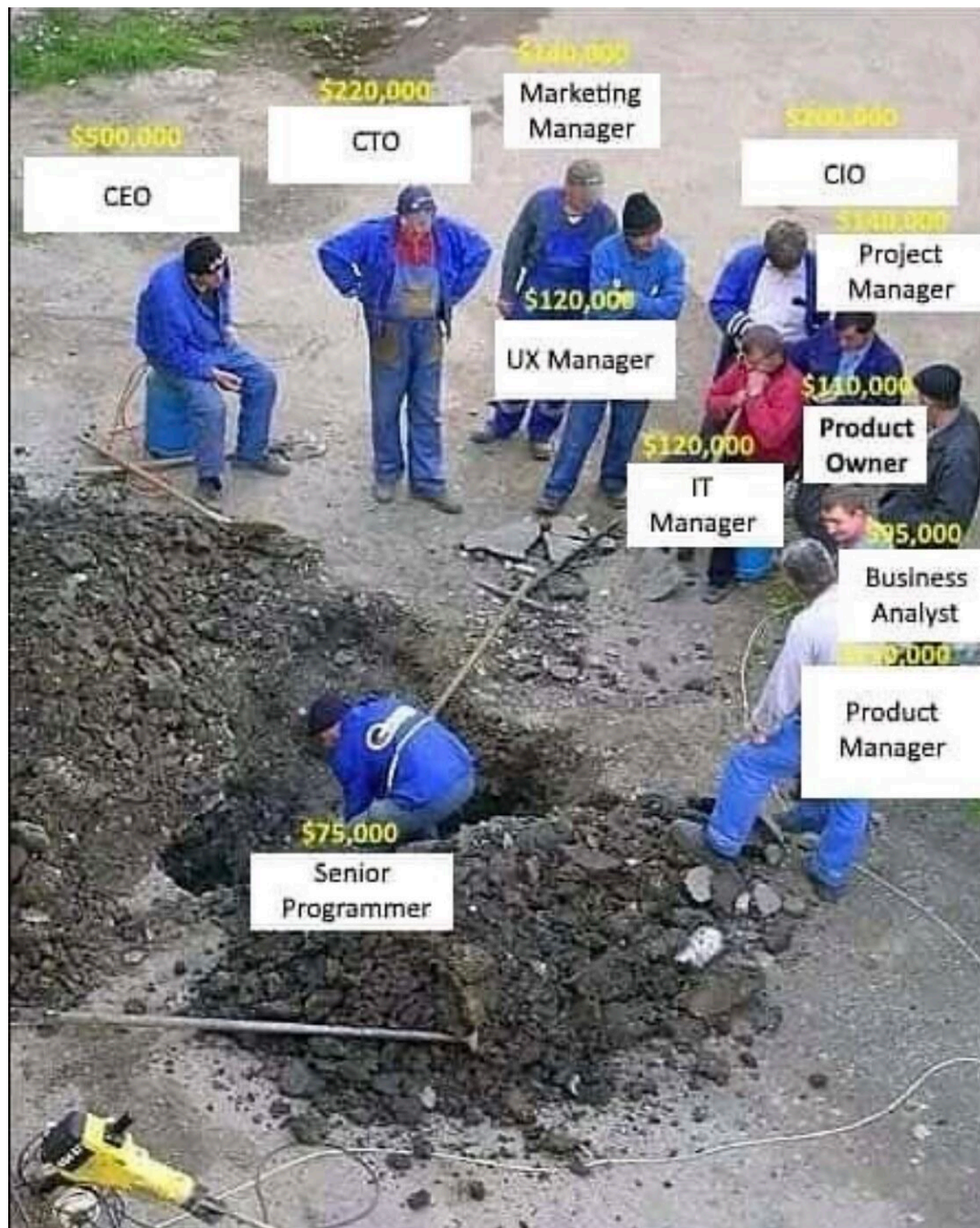
- El software puede ser usado con facilidad en nuevos desarrollos



# Resumiendo

- Un carpintero no es toda persona que tiene madera, clavos y martillo
- Un desarrollador de software no es toda persona que sabe programar





Quién es el programador ??