Ingeniería de Software

Introducción a la Ingeniería de Software

¿Qué entendemos por ingeniería?

Del latín *ingenium* → producir

Conjunto de conocimientos y técnicas científicas, empíricas y prácticas aplicadas a la invención, el diseño, el desarrollo, la construcción, el mantenimiento y el perfeccionamiento de tecnologías, estructuras, máquinas, herramientas, sistemas, materiales y procesos para la resolución de problemas prácticos.

Profesión en la que el conocimiento de las ciencias naturales y matemáticas, ganado con estudio, experiencia y práctica, es aplicado con buen juicio para desarrollar formas de utilizar, económicamente, los materiales y las fuerzas de la naturaleza para el beneficio del género humano (*Acreditaction Board for Engineering and Technology*, 1996)

¿Qué entendemos por software?

¿Qué entendemos por software?

Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora (RAE, 2001)

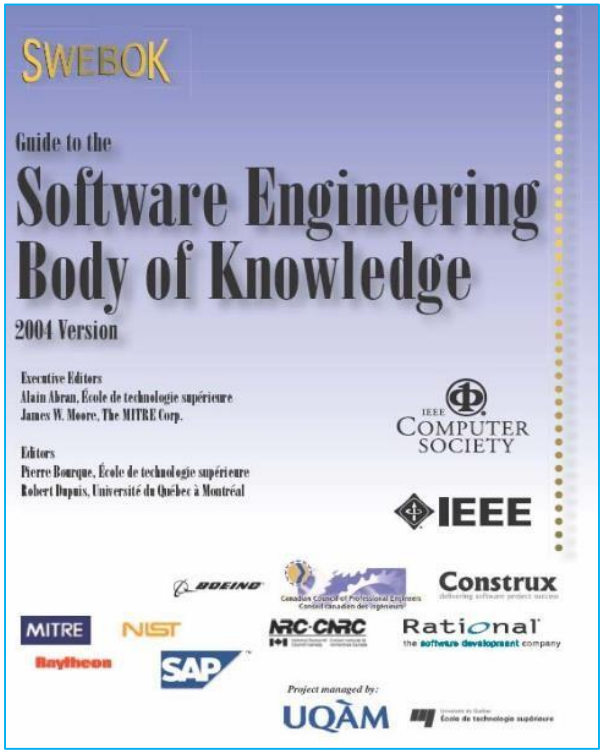
Conjunto de elementos lógicos que hacen posible que las tareas para las que se han diseñado, operen de forma correcta (Sommerville, 2005)

Conjunto de programas, procedimientos, reglas, datos y documentación que provocan el funcionamiento de un sistema informático (Lewis, 1994)

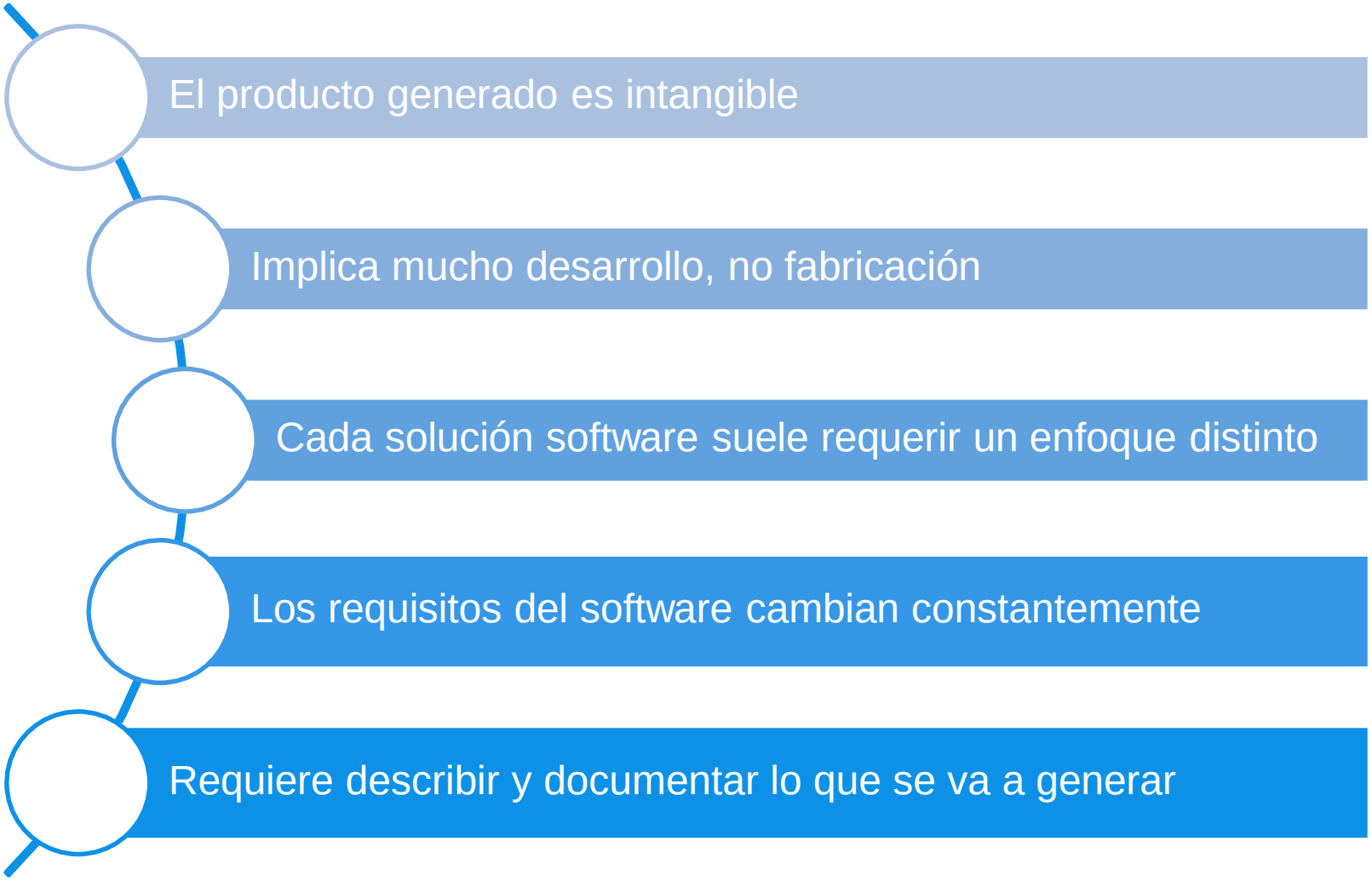
¿Qué es la ingeniería de software?

¿Qué es la ingeniería de software?

Disciplina de la ingeniería que cubre todos los aspectos para la producción de software (Sommerville, 2005)



Características particulares de la ingeniería de software



Propiedades inherentes de diferentes ingenierías

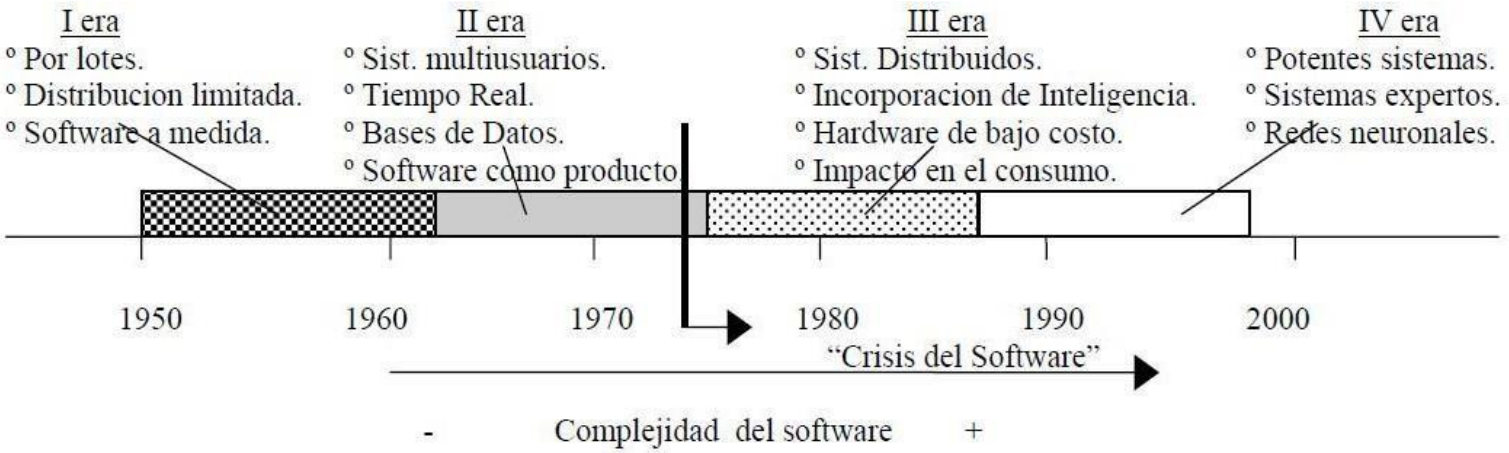
Ingeniería	Producto	Teoría	Propiedades
Civil	Puente	Mecánica clásica	Carga máxima, resistencia lateral...
Aeronáutica	Avión	Dinámica de fluidos	Relación de planeo, turbulencias...
Software	Software	¿?	¿?

Crisis del software

Crisis del software

- Dificultad que existía para implementar programas, que no solo funcionaran correctamente, sino que además fueran comprensibles, verificables, adaptables a la cambiante necesidad de los usuarios y que fueran menos costosos.
- Problemas detectados:
  - No acababan en la fecha establecida
  - Superaban de manera sustancial el presupuesto inicial
  - No se cumplían con las especificaciones establecidas
  - Se desarrollaba software de baja calidad

Crisis del software



- Muchos cambios en hardware.
- Poca evolución del software.
- Desarrollo de software sin planeacion, y sin documentación.

- Tecnicas Interactivas.
- Control en Tiempo Real.
- Mejora de los dispositivos de almacenamiento.
- Primeras casas de software.
- Problemas por el mantenimiento.

- Aparecen las PC's.
- Cias. de Software venden miles de dolares.
- Hardware standar, el software marca la diferencia.

- Software complejo.

Crisis del software

¿Por qué es tan difícil desarrollar software?

Crisis del software – Errores informáticos de gravedad en España

Año	Entidad	Descripción	Nivel de gravedad
2015	Avión A400M	Un fallo en el *software* del ordenador que controlaba los motores fue la causa del accidente de que un avión A400M se estrellara en Sevilla y que provocara la muerte de cuatro tripulantes y heridas a otros dos.	Alta
2013	Juzgados de Madrid	Un fallo informático colapsa 46 juzgados de Madrid durante 12 días.	Media
2012	Gobierno de Canarias	Como consecuencia de un error informático, el gobierno canario pierde datos fundamentales acerca de la erupción volcánica de 'El Hierro'.	Media
2012	Gobierno de Navarra	Un fallo en el sistema informático de la red corporativa del Gobierno de Navarra interrumpe los servicios de Salud, Hacienda y Desarrollo Rural.	Media
2012	Aeropuerto de Málaga	Un fallo informático deja a oscuras la torre de control del aeropuerto de Málaga.	Alta
2011	Periódico 'El País'	El periódico 'El País' casi no sale a la calle un domingo por un fallo en el sistema informático y la ausencia de técnicos para poder solventarlo.	Media
2003	Vodafone España	La red de telefonía móvil de Vodafone en España sufre una caída de servicio dejando a más de 8 millones de usuarios sin cobertura.	Alta

Crisis del software – Errores informáticos más costosos de la historia

Año	Entidad	Descripción	Nivel de gravedad
2010	Toyota	Toyota tuvo que retirar más de 400.000 de sus vehículos híbridos, debido a un problema *software* que provocaba un retraso en el sistema anti-bloqueo de frenos. Se estima que, entre sustituciones y demandas, el error le costó a Toyota unos 2,8 billones de Euros.	Alta
1999	NASA	Los ingenieros de la NASA perdieron el contacto con la sonda *Mars Climate Orbiter* en un intento que orbitase en Marte. La causa fue debida a un programa que calculaba la distancia en unidades inglesas, mientras que otro programa utilizaba unidades métricas.	Alta
1996	Agencia Espacial Europea (ESA)	El cohete Ariane 5 explotó debido a que un número real de 64 bits en coma flotante, relacionado con la velocidad, se convirtió en un entero de 16 bits. Las pérdidas se estiman en 400 millones de Euros.	Alta
1994	Intel	Un profesor de Matemáticas descubrió e informó acerca de un fallo en el procesador Pentium de Intel. La sustitución de chips costó a Intel uno 443 millones de Euros.	Alta
1988	Internet	Un estudiante de posgrado, Robert Tappan Morris, fue condenado por el primer ataque con 'gusanos' a gran escala en Internet. El coste de limpiar el desastre ocasionado por Robert se cifra en unos 100 millones de dólares.	Alta
1978	Hartford Coliseum	Apenas unas horas después de que miles de aficionados abandonaran el estadio Hartford Coliseum (Estados Unidos), el techo se derrumbó por el peso de la nieve. La causa fue debida al cálculo incorrecto que se había introducido en el *software* CAD utilizado para diseñar el estadio.	Alta
1962	NASA	La sonda espacial Mariner I se desvió de la trayectoria de vuelo prevista con destino a Venus poco después de su lanzamiento. Desde control se destruyó la sonda a los 293 segundos del despegue. La causa de la desviación fue una fórmula manuscrita que se programó de manera incorrecta. Las pérdidas económicas se estimaron en unos 15 millones de Euros.	Alta

Crisis del software

- La dificultad para satisfacer las necesidades de los clientes, la complejidad del software y sus constantes cambios, hacen que desarrollar software sea un proceso arduo y muy costoso.
- Ince (1993) enumera 3 requisitos que ha de satisfacer un software para que sea viable y de calidad, adaptándose al cambio acelerado de la sociedad:
  - Utilización de diferentes técnicas de desarrollo que permitan minimizar la complejidad de un software.
  - Utilización de métodos y conceptos para poder valorar la naturaleza y validez de un software lo antes posible.
  - Utilización de técnicas de desarrollo que minimicen los efectos colaterales por modificaciones y evolución del software.

Ingeniería de software *vs.* la ciencia de la computación

- La ciencia de la computación se refiere a las teorías y métodos subyacentes a la computación y los sistemas de software.
- La ingeniería del software se refiere a los problemas prácticos para producir software.
- Ingeniero del software: Profesional de la industria informática que aplica los conocimientos disponibles y las teorías científicas para construir productos software.
- No confundir ingeniero del software con programador.

Ingeniería de software *vs.* la ciencia de la computación

- Responsabilidades de un ingeniero de software:
  - Conocer el dominio de aplicación del software
  - Participar en el diseño de la configuración de los sistemas
  - Analizar el rendimiento del diseño propuesto para asegurar que el sistema propuesto cumplirá con los requisitos del cliente.
  - Diseñar la estructura básica del software
  - Analizar la consistencia, disponibilidad y completitud de la estructura del software propuesta para implementar la solución
  - Implementar el producto software
  - Integrar la nueva solución, si es necesario, en otras aplicaciones
  - Llevar a cabo las pruebas necesarias para validar y verificar el software
  - Revisar y mejorar los sistemas software

Ética y responsabilidad profesional

"Trata a los demás como te gustaría que te trataran a ti"

- La ingeniería del software debe estar dentro de un marco legal y social que debe limitar la libertad de los ingenieros a la hora de desarrollar software (Sommerville, 2005).
- Un ingeniero del software siempre ha de partir de la premisa de construir software que respete y valore a las personas de acuerdo a una conducta ética (Génova et al, 2007).
- ACM y IEEE Computer Society crearon el código de ética y práctica profesional de ingeniería de software, que recoge unos principios morales en el campo de la ingeniería del software.

Ética y responsabilidad profesional

- Principios básicos que deben seguir los ingenieros del software:
  - Público: Deben actuar de manera consistente en bien de interés general.
  - Cliente y contratista: Actuar de modo que sea del mejor interés para sus clientes.
  - Producto: Garantizar que cumplen con la mayor número de estándares.
  - Juicio: Mantener la integridad e independencia de su valoración profesional
  - Gestión: Los gestores deben promover un enfoque ético.
  - Profesión: Realizar avances referidos a la integridad y reputación profesional.
  - Compañeros: Ser justos y apoyar a los compañeros de profesión.
  - Persona: Participar en el aprendizaje continuo de la práctica de la profesión.

Fábricas de software

- Según el artículo Inmunes a la crisis (El País, 2010), las fábricas de software dedicadas al desarrollo y mantenimiento de aplicaciones no se vieron afectadas por la crisis, ya que a las empresas les sale más barato que sus software lo desarrollen estas fábricas.
- Y según el post Ingeniería del software, una de las ramas de la ingeniería con más salidas profesionales (Sicilia, 2015), los ingenieros especializados en software tienen un gran abanico de posibilidades profesionales.

Dudas

Bibliografía

UNIR(2016) Metodologías, Desarrollo y Calidad en la Ingeniería de SW

Booch, G., Rumbaugh, J. y Jacobson, I. (2006). *UML 2.0 2ª Edición*. Pearson Addison-Wesley.

Larman, C. (2003). *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda edición*. Pearson Prentice-Hall.

Pressman, R. (2010). *Ingeniería del Software* (7ª ed.). Mcgraw-Hill.

Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.

http://www.javiergarzas.com/