



# TEMA 1. Introducción

## Ingeniería del Software

Raquel Martínez España

Grado en Ingeniería Informática

# Ingeniería del software de un vistazo

□ En la asignatura cubriremos las siguientes partes:

1. Introducción a la Ingeniería del software => Definiciones fundamentales, Motivación, etc.
2. Planteamiento de un proyecto de desarrollo software
  - Existe un problema -> ¿Puede ser resuelto por computación?
  - Preguntas clave:
    - ¿Viable?
    - ¿Qué requisitos de funcionamiento tiene?
  - Necesitamos un **análisis técnico** simplificando el problema en **partes** mas **elementales** y sencillas de resolver
3. Etapas y técnicas para desarrolla un sistema informática que resuelva el problema planteado:
  - Diseño de una solución
  - Cómo construirla
  - Cómo ponerla en funcionamiento
  - ¿Satisface al cliente?

# Ingeniería del software de un vistazo

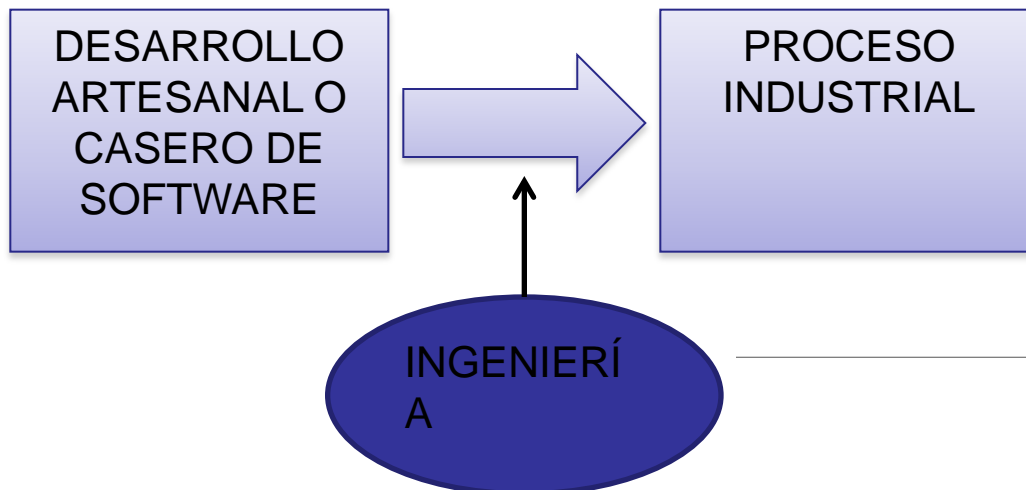
## (ii)

❑Clave: ¿Existe algún modo de articular un **proceso claramente definido** y **repetible** para abordar la resolución informática de problemas?

▪Sí:

- Podemos desarrollar cualquier proyecto por complejo que sea.
- Además garantizado la calidad

▪Muchas técnicas => Nos centramos con sw estructurado



# Sistemas de Información

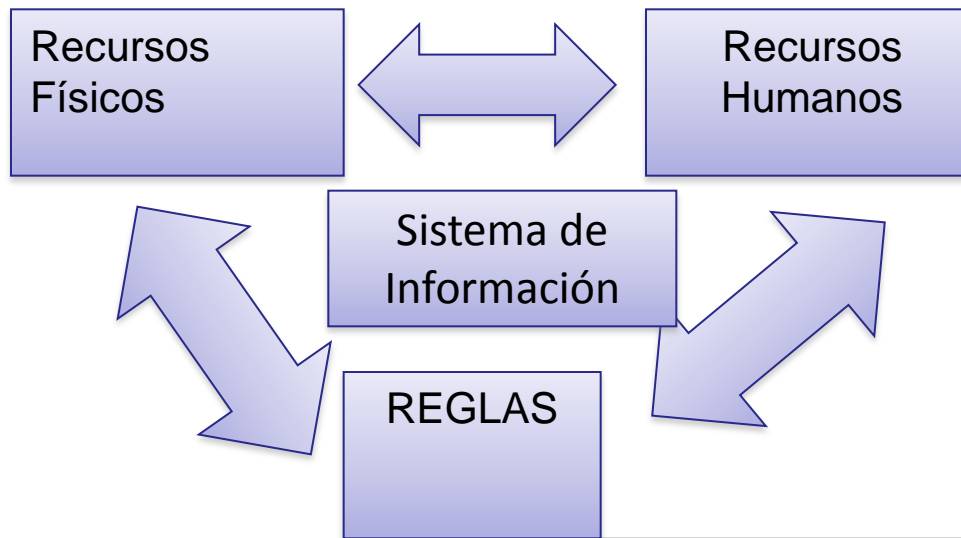
## ❑ Concepto de Sistema de información (SI):

Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) para desempeñar su actividad de acuerdo a su estrategia de negocio.

# Componentes de un sistemas de Información

## □ Componentes principales de un SI:

- **Recursos físicos:** Documentos, archivos, HW, etc.
- **Recursos humanos** que participan en el SI.
- **Reglas:** Protocolos, normas, métodos que establecen cómo debe hacerse la transmisión de información



# Sistema Informático

- ❑ Sistema de Información que se realiza con **ordenadores**
  - > La gran mayoría en la actualidad
- ❑ Compuestos por
  - Documentos almacenados en **bases de datos**, hw, etc
  - Personal del **departamento de informática** y sistemas.
  - **Normas**, métodos y protocolos -> SSOO, Red, etc.
- ❑ Tipos de Sistemas Informáticos:
  - ❑ Transaccionales -> tareas automáticas del día a día
  - ❑ Gestión -> Datos que maneja la empresa
  - ❑ Ayuda a la decisión: Bussiness Intelligent -> Oportunidad de trabajo

# Sistemas de gestión

- ❑ Engloban los 3 sistemas informáticos anteriores
- ❑ Son sistemas muy complejos de desarrollar, para desarrollarlos
  - ❑ Tiempo y recursos materiales -> evaluación y planificación
  - ❑ Equipo Informáticos -> jefes de proyecto, analistas, programadores
  - ❑ Métodos y técnicas de resolución de problemas



Modelo SAP

# Sistemas de Información

## Jerarquía de sistemas:

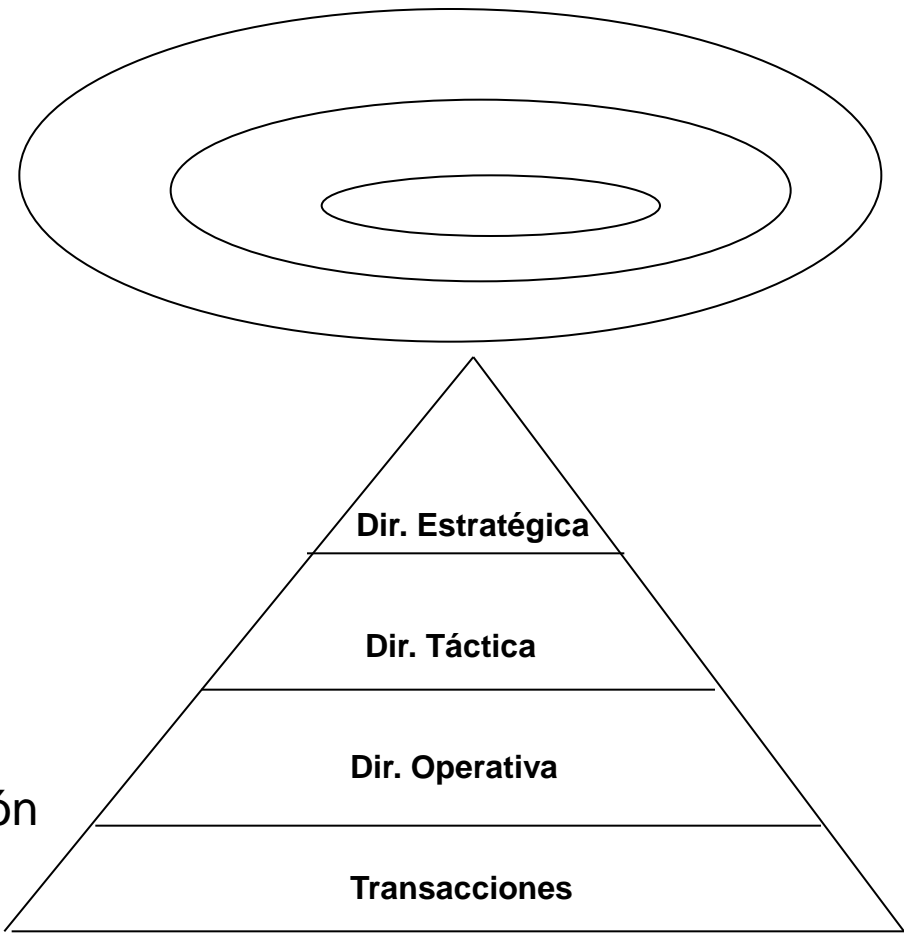
- Sistema Objeto (organización)
- Sistema de Información
- Sistema Informático

## Estructura de un Sistema de Información:

- Subsistema Decisional
  - Nivel Estratégico (Largo Plazo)
  - Nivel Táctico (Medio Plazo)
  - Nivel Operativo (Corto Plazo)
- Subsistema Transaccional

## Conceptos asociados:

- MIS: Sistemas de manejo de Información
- DSS: Sistemas de apoyo a la decisión
- EIS: Sistemas de información ejecutiva





# Sistemas de Información

¿ En que nivel del subsistema decisional se encontrarían las siguientes operaciones?

1. Decisión de aprovisionarse de determinados productos en base a la información del fichero de inventario.
2. Planteamiento de incorporación de nuevos servicios al cliente como banca electrónica a través de Internet, servicios de tarjeta-monederó electrónico
3. Establecimiento de necesidad de abrir una sucursal bancaria en un determinado barrio, en base a datos tales como volumen de negocio disponible en tal área, información sobre nº de cuentas abiertas en las oficinas adyacentes, presupuesto disponible para recursos humanos, etc...

# El Software

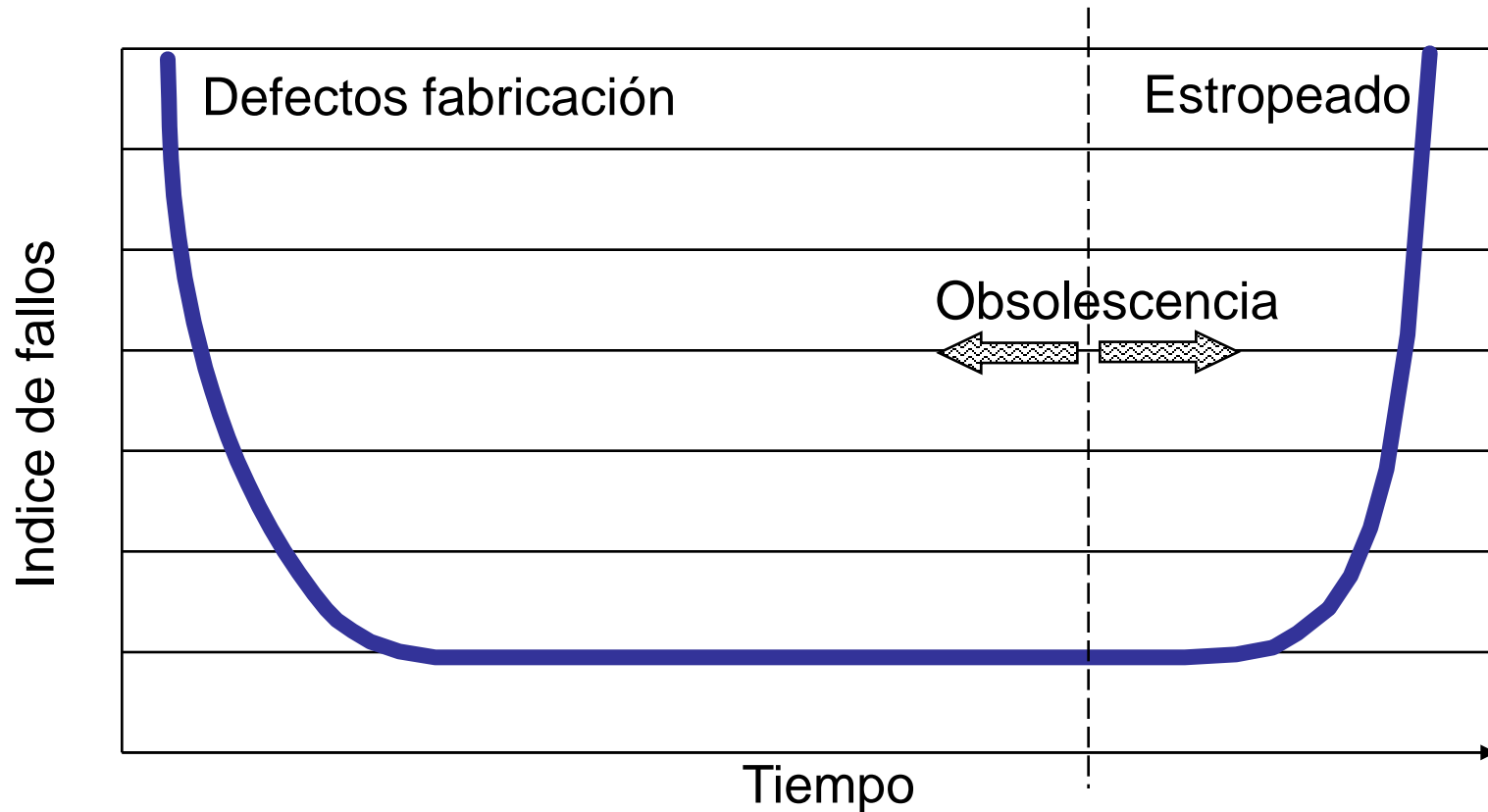
## Definición

❑ **Instrucciones** (programas de computadora) que cuando se ejecutan proporcionan la función y el comportamiento deseado, mas **estructuras de datos** que facilitan a los programas manipular adecuadamente la información y **documentos** que describen la operación y uso de los programas.

## Características

- ❑ Se desarrolla
- ❑ No se estropea
- ❑ Se construye a medida

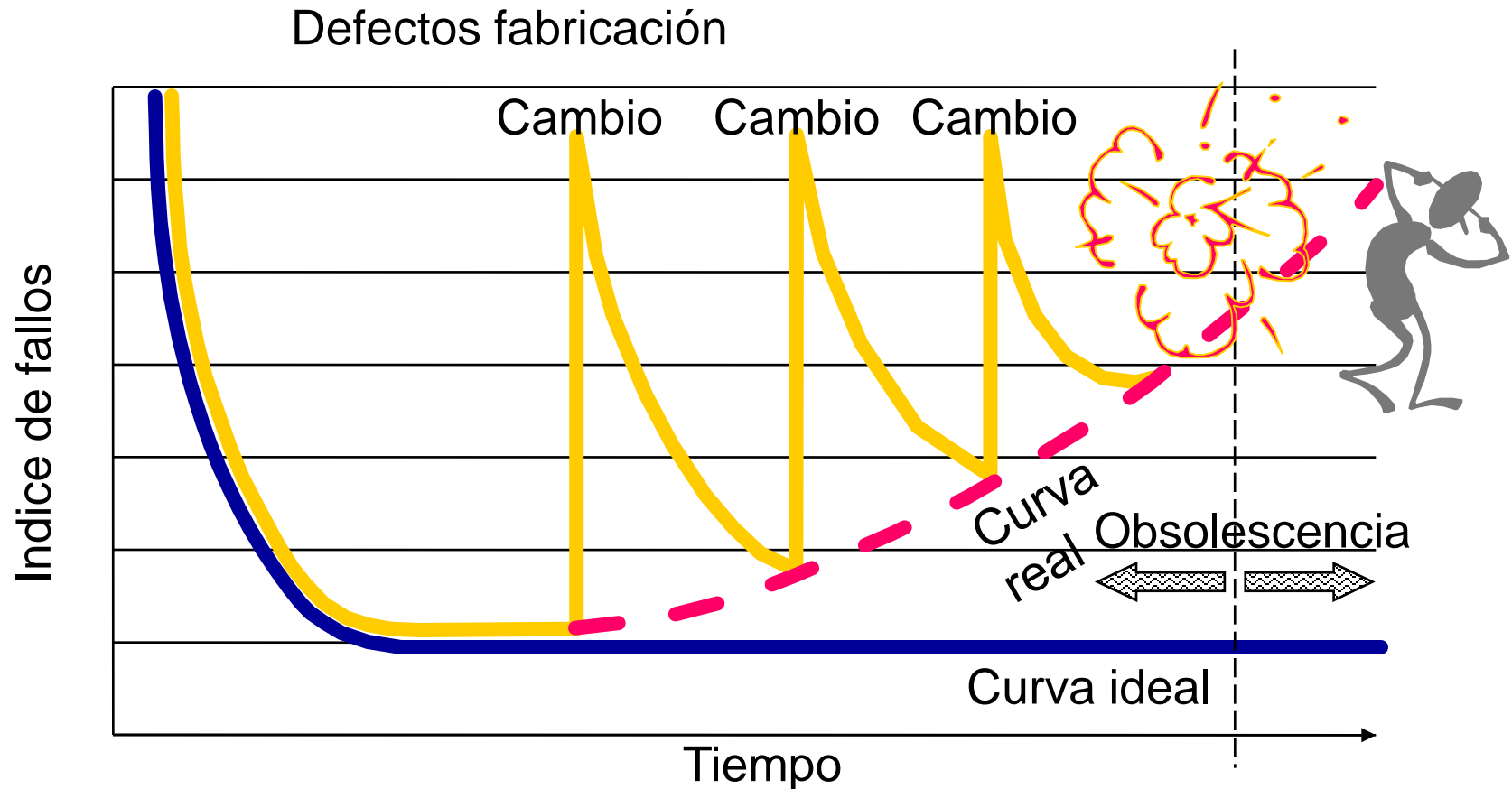
# Curva de fallos del Hardware.



# Curva ideal de fallos del Software.



# Curva real de fallos del Software.



# El Software

- Tipos de software
  - ☐ Software de Sistemas
  - ☐ Software de Tiempo Real
  - ☐ Software de Gestión
  - ☐ Software de Ingeniería y científico
  - ☐ Software Empotrado
  - ☐ Software de Inteligencia Artificial.

# El Software

- 1ª Era
  - ☐ Sistemas Batch
  - ☐ Desarrollos a Medida
- 2ª Era
  - ☐ Sistemas Multiusuario
  - ☐ Tiempo Real
  - ☐ Software como producto
- 3ª Era
  - ☐ Sistemas distribuidos
  - ☐ Impacto en el consumo
- 4ª Era
  - ☐ Potentes sistemas sobremesa
  - ☐ Sistemas Expertos
  - ☐ Tecnologías orientadas a objetos
  - ☐ Computación Paralela
- 5ª Era ??????

# El Software

- Problemática del Software
  - ❑ Desarrollos largos
  - ❑ Desarrollos costosos
  - ❑ No se pueden encontrar todos los errores antes de entregar al cliente
  - ❑ Resulta difícil de constatar el progreso conforme se desarrolla
- La crisis del software. Una aflicción crónica
  - ❑ Aflicción => “Algo que causa pena o desastre”
  - ❑ Crónica => "muy duradero o que vuelve a aparecer con frecuencia; continuando indefinidamente".
  - ❑ Existen mitos que han hecho mucho daño en el desarrollo de software.



# Mitos del software

- Creencias erróneas desde los primeros días. Mitos del software:
  - ❑ Mitos de **administración** o gerentes -> Presupuestos, plazos y calidad
    - ¿Basta con disponer de un conjunto de estándares y procedimientos ? -> ¿Quién los conoce y aplica? ¿Son reales?
    - ¿Basta con disponer de tecnología avanzada? -> Subcontratar. Si tu no lo entiendes internamente, tendrás problemas cuando otra compañía lo elabore
    - ¿Es beneficiosa la incorporación cuantitativa de personal a un proyecto? -> No es un proceso mecánico.

# Mitos del software

## ❑ Mitos del cliente

- Para comenzar a escribir programas, es suficiente el **enunciado general de los objetivos**->detalles más adelante
  - Un planteamiento ambiguo de los objetivos puede ser desastroso. Necesitamos tener las cosas claras antes de empezar a desarrollar
- Los requisitos cambian continuamente. El **software** los acomodará porque es **flexible**
  - Cuanto mas tiempo pase, introducir cambios en el software puede ser muy tedioso.

# Mitos del software

## ❑ Mitos de los desarrolladores o del profesional

- El software acaba cuando el programa se acaba y funciona
  - Datos: Entre el 60% y el 80% de todo el esfuerzo dedicado al sw se produce una vez entregado
- No se puede comprobar la calidad hasta que el producto está terminado
  - Revisión de la calidad. Pruebas de caja blanca, negra, etc.
- Lo único que se ha de entregar al terminar es el programa funcionando
  - Necesitamos apoyo para el software.
- La Ingeniería del sw hará que generemos documentación voluminosa e innecesaria.
  - El objetivo es un sw de calidad, no generar documentación

# El Software

## La Ingeniería del software

- Definición
  - ☐ Incrementar la productividad
  - ☐ Incrementar la calidad
- Elementos Clave
  - ☐ Métodos
  - ☐ Herramientas
  - ☐ Procedimientos

- Elementos de los métodos
  - ☐ Estructura
  - ☐ Ciclo de desarrollo
  - ☐ Fases
  - ☐ Módulos
  - ☐ Etapas
  - ☐ Resultados
    - Consistencia Interna (verificación)
    - Consistencia externa (validación)
  - ☐ Sistema deseado
  - ☐ Técnicas

# El Software

## Principios generales básicos de la ingeniería del software

- ❑ El software existe para dar valor a sus usuarios.
- ❑ El diseño debe ser tan simple como sea posible, pero no más.
- ❑ Una visión clara es esencial para el éxito de un proyecto de software.
- ❑ Otros consumirán el producto que se desarrolla.
- ❑ Un sistema de larga vida útil tiene más valor.
- ❑ La reutilización ahorra tiempo y esfuerzo.
- ❑ Pensar ... !!!
  - en todo con claridad antes de emprender la acción.

# Índice

- ❑ Organización de la asignatura
  - ✓ Temario
  - ✓ Resultados del aprendizaje
  - ✓ Metodología
  - ✓ Bibliografía
  - ✓ Evaluación
  - ✓ Calendario
- ❑ Introducción
  - ✓ Tema 1. Sistemas de información
  - ✓ Tema 2. El software
  - ✓ **Tema 3. Ciclo de vida de los sistemas informáticos**
- ❑ Orientaciones del trabajo

# Ciclo de vida

## Bibliografía básica:

- Pressman, R. “Ingeniería del Software: Un enfoque práctico”. 7ª edición. Madrid: McGraw Hill, 2010.
  - ❑Capítulo 2. Modelos del proceso
- Braude, E. “Ingeniería de Software. Una perspectiva orientada a objetos”. Ra-Ma, 2003.
  - ❑Capítulo 1. Proceso

# Ciclo de vida

- Definición:

❑ El proceso que se sigue para construir, entregar y hacer evolucionar el software, **desde la concepción** de una idea hasta la entrega y el retiro del sistema.

❑ **Según IEEE:** “una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software”.

❑ **Según ISO:** “un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”.

**¿Ciclo de vida = ciclo de desarrollo?**



# ISO 12207. Estándar para los procesos de ciclo de vida del software

TRABAJO



# Procesos Principales ISO 12207

- **Proceso de adquisición.** En este proceso se define **las tareas que tiene que realizar el comprador**, cliente cuando adquiere un producto o servicio software. Algunas de las tareas son: **preparación de ofertas, elección del distribuidor del software**, gestión de la adquisición, etc....
- **Proceso de suministro.** En este proceso se define las actividades que el **suministrador** realiza, desde que prepara el **presupuesto** de una petición de compra hasta la entrega del software al cliente.
- **Proceso de desarrollo.** Este proceso engloba las actividades de **análisis, diseño, codificación, integración, pruebas e instalación y aceptación.**
- **Proceso de operación.** En este proceso se incluyen la **explotación del software y servicios** de soporte para los usuarios del sistema.
- **Proceso de mantenimiento.** En este proceso aparecen reflejadas las actividades de modificación del software, debido a **errores a deficiencias**, necesidades de mejora, etc. En este proceso se incluye la migración y retirada del producto software.

# Procesos Apoyo o Soporte ISO 12207

- **Proceso de documentación.** Define las actividades para el registro de **la información producida** por un proceso o actividad del ciclo de vida.
- **Proceso de gestión de la configuración.** Define las actividades para identificar y establecer las líneas bases fundamentales para el desarrollo de elementos software, **gestión de versiones**, variante, en general actividades para el control del cambio.
- **Proceso de aseguramiento de la calidad.** Define las actividades para verificar que los productos software **cumplen con los requisitos especificados** por el usuario y se ajustan a los planes establecidos. El aseguramiento de la calidad puede realizarse utilizando el resultado obtenido en otros procesos como el de apoyo, verificación, validación, revisiones conjuntas, auditorías, etc.
- **Proceso de verificación.** Define las actividades **de verificación de los requisitos** en cuanto a que éstos sean **completos y correctos** y además que cumplan con las condiciones establecidas en fases previas.
- **Proceso de validación.** En este proceso se definen las actividades para asegurar que el software final contempla todos los requisitos previos para su uso.
- **Proceso de revisiones conjuntas.** Define las actividades para evaluar el estado y productos de una actividad.
- **Proceso de auditoría.** Define las actividades para determinar el cumplimiento de los requisitos, planes y contrato. Este proceso puede ser empleado por dos partes cualesquiera, donde una parte (la auditora) audita los productos software o actividades de otra parte (la auditada).
- **Proceso de solución de problemas.** Define un proceso **para analizar y eliminar los problemas** (incluyendo las no conformidades) que sean descubiertos durante la ejecución del proceso de desarrollo, operación, mantenimiento u otros procesos, de esta manera se asegura que todos los problemas que surgen se solucionan.

# Procesos Organizativos ISO 12207

- **Proceso de gestión.** Define las **actividades básicas de gestión** de los procesos durante el ciclo de vida.
- **Proceso de infraestructura.** Define las actividades básicas para establecer la **infraestructura necesaria para los procesos**: hardware, software, instalaciones, normas, etc.
- **Proceso de mejora.** Define las actividades básicas para controlar, valorar, medir los procesos del ciclo de vida.
- **Proceso de formación.** Define las actividades para **mantener al personal formado**.

# Etapas principales del ciclo de desarrollo

## □ Especificación de requisitos:

- Conocimientos del problema a resolver -> Características, detalles, limitaciones
  - Técnicas: Entrevistas, análisis de costes y beneficios -> estudio de viabilidad.

## □ Análisis:

- Se descompone el problema en partes.
- Subproblemas pequeños y sencillos.
- Nos centramos en el **QUÉ** y no en el **CÓMO**.
  - Identificar las funciones que debe desarrollar sistema pero no implementar el algoritmo.
  - Identificar datos y eventos
- Técnicas para el análisis estructurado de procesos y datos: DFD, E/R, Historia de vida de Entidades, etc

# Etapas principales del ciclo de desarrollo

## □ Diseño:

- Cómo resolver cada uno de los subproblemas identificados por el análisis.
- Cómo integrarlos en una solución.
- Importante modularidad (acoplamiento y cohesión) e Interfaces (interacción con el usuario)
- Nos centramos en el **CÓMO** y no en el **QUÉ**.
  - Escritura de algoritmos.
- Técnicas *diagramas de cuadros de constantine* para los procesos y paso a tablas del modelo relacional para los datos.

# Etapas principales del ciclo de desarrollo

## ❑ Implementación:

- Programación con lenguaje de alto nivel.
- Generación de tablas y consultas.

## ❑ Pruebas:

- Verificación de la solución: correcto funcionamiento del código programado.
- Validación: producto correcto.
- Técnicas: Pruebas unitarias y de integración, caja blanca, negra, etc.

## ❑ Instalación y mantenimiento

- Entrega y puesta en funcionamiento en su destino final
- Mantenimiento-> seguir trabajando con el cliente.

# Etapas principales del ciclo de desarrollo

## □ Procesos integrales de gestión del software:

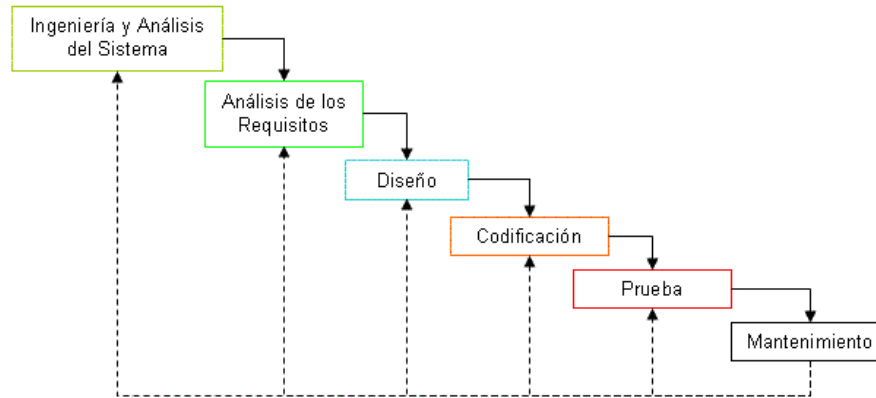
- Garantizar la integridad y coherencia de todo el desarrollo -> Producto final sea fiable y se utiliza al máximo de sus capacidades:
  - **Gestión de cambios:** Propagar a todas las etapas.
  - **Gestión de configuraciones:** Hardware y software necesario.
  - **Gestión de la documentación:** Planificar, diseñar, editar, producir, distribuir y mantener los documentos para los desarrolladores y usuarios => **TRABAJO**
  - **Gestión de la calidad:** Asegurar que se cumple lo pactado.



# Ciclo de vida

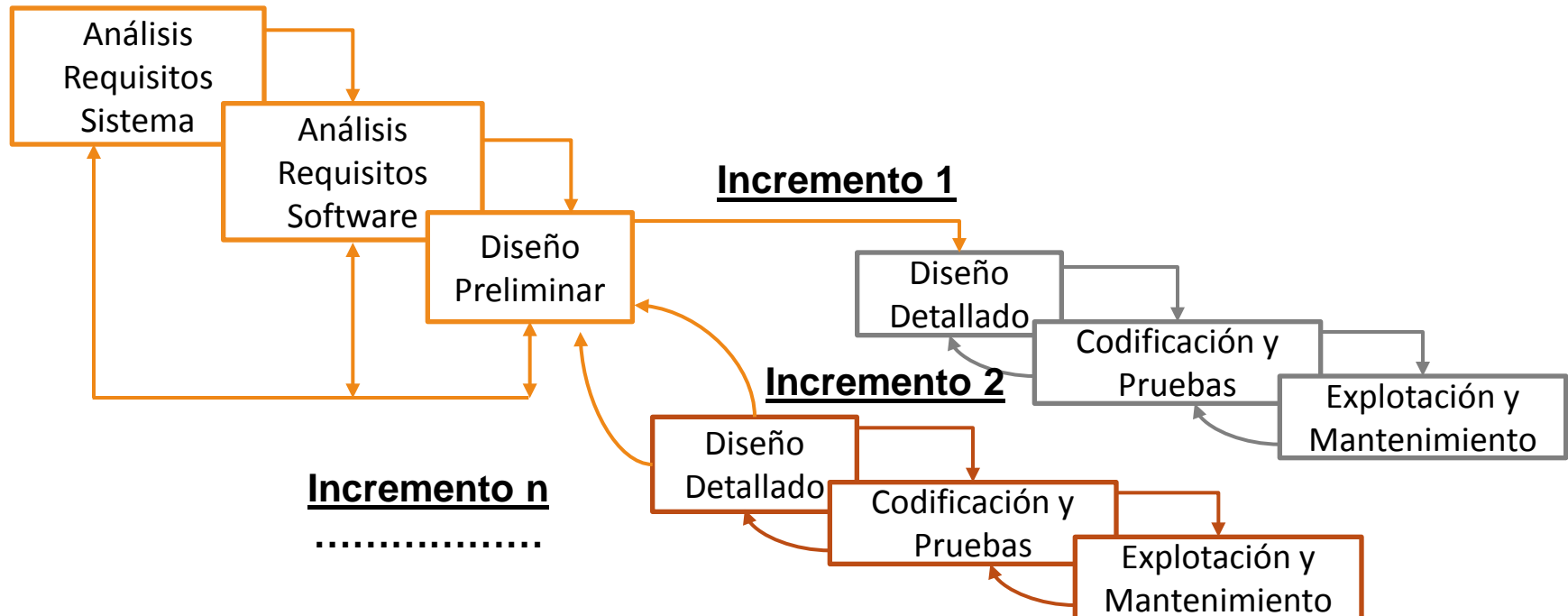
- El ciclo de vida incluye la etapas suelen ser las anteriores pero se pueden entender de distintas maneras según el proyecto:
- Las etapas se agrupan 3 grandes fases:
  - ❑ **Definición del problema** -> Incluye requisitos y análisis.
  - ❑ **Desarrollo**: Diseño, programación y pruebas
  - ❑ **Mantenimiento**.

# Ciclo de Vida en Cascada



- Cada fase empieza cuando se ha terminado la fase anterior.
- Para pasar de una fase a otra es necesario conseguir todos los objetivos de la anterior.
- Ayuda a prevenir que se incumplan los plazos de entrega.
- Al finalizar cada fase, se puede revisar el progreso del proyecto.
- En la realidad, hay iteraciones en el proceso -> **Ciclo de retroalimentación**
- Se tarda mucho en obtener un producto. Pueden existir fases “interminables”.
- El usuario debe tener paciencia.

# Ciclo de Vida en Cascada Incremental



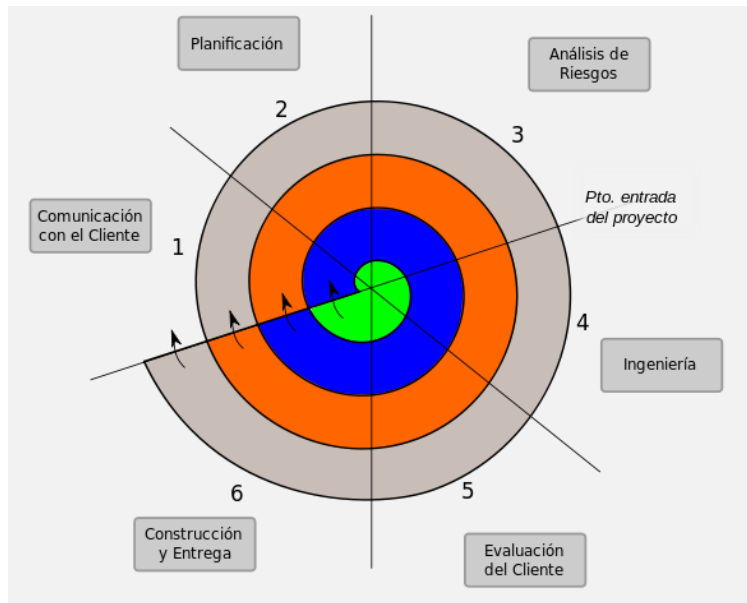
- El sistema se va creando añadiendo componentes funcionales (incrementos).
- El software es la integración de varios resultados sucesivos.
- Se ajusta a entornos de alta incertidumbre (e.g. no es necesario que se definan todos los requisitos al inicio).
- Los errores en la definición de los requisitos se detecta tarde. Corregirlos puede ser tan caro como en el modelo en cascada.

# Ciclo de vida basado en prototipos

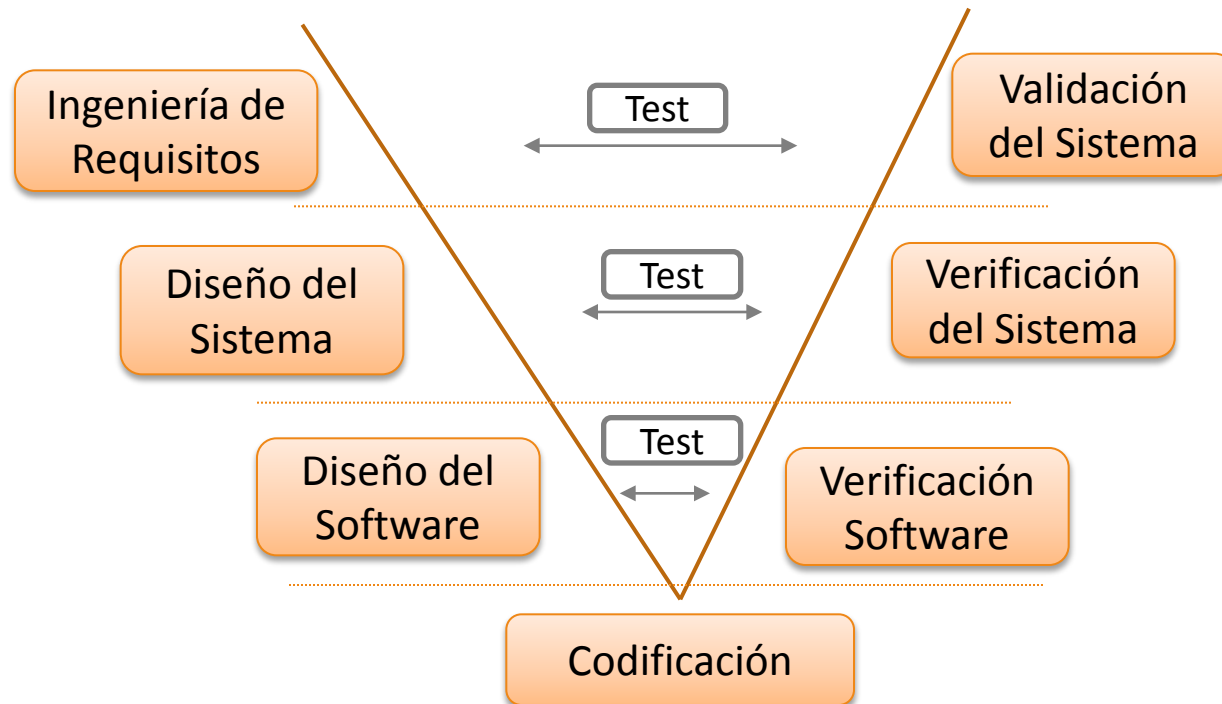
- ❑ **Problema principal:** Definición clara del problema -> El usuario no sabe lo que quiere -> No podemos desarrollar una solución completa sin correr el riesgo de que no sea la buscada.
- ❑ **Solución Prototipos:** Modelo evolutivo de la solución software final.
  - Modelo -> no es la solución final.
  - Evolutivo -> nos podemos adaptar a la variabilidad del cliente.
- ❑ **Tipos de Prototipos**
  - **Pantallas o maquetas:** Solo parte visual, no tiene funcionalidad
  - **Funcional evolutivo:** Incluye los requisitos entendidos hasta la fecha y se van iterando.
- ❑ Necesidad de interacción con el cliente.

# Ciclo de vida en espiral

- ❑ Mejorar los anteriores y **incorporar análisis de riesgos** (aspectos económicos)
- ❑ Cuatro grandes etapas: **planificación, análisis de riesgos, ingeniería y evaluación.**
- ❑ Ciclo de vida iterativo y creciente -> Cada etapa es mas compleja y cercana a la solución final.
- ❑ Proceso de **construcción del software** claramente **evolutivo** y en cada iteración es lineal (como cascada).



# Modelo en V



# Ciclo de vida

- Modelos de proceso especializados:
  - ❑ Desarrollo basado en componentes
  - ❑ Modelo de métodos formales
  - ❑ Desarrollo orientado a aspectos
- Proceso unificado
- Modelos del proceso personal y del equipo:
  - ❑ PPS: Proceso personal del software (también PSP, en inglés)
  - ❑ PES: Proceso del equipo de software (también TSP, en inglés)

Según: Pressman, R. "Ingeniería del Software: Un enfoque práctico". 7ª edición. Madrid: McGraw Hill, 2010.

# Distintas visiones de una realidad

