

# Ingeniería del Software I

---

## Índice

---

- Tema 1
  - Definiciones
  - La Crisis del Software
  - Calidad del Software
  - Tipos de Productos Software
  - Conceptos Básicos de la Ingeniería del Software
  - Método de Ingeniería
  - Modelo Conceptual de la Ingeniería del Software
    - Conceptos
  - Proceso Software
    - Características
  - Ciclo de Vida del Software
  - Modelo de Proceso Software
  - Metodologías
    - Metodologías Estructurales
    - Metodologías Orientadas a Procesos
    - Metodologías Orientadas a Datos
    - Metodologías Orientadas a Objetos
    - Metodologías Ágiles
  - Tecnología CASE
- Tema 2
  - Sistema
  - Sistema de Información
    - Estructura de los Sistemas de Información
      - Niveles
      - Flujos de Información
      - Información
    - Clasificación de los Sistemas de Información
  - Principios Generales de Sistemas
  - Ingeniería de Sistemas
    - Adquisición del Sistema
- Tema 3
  - Modelos Tradicionales
    - Modelo Primitivo
    - Modelos Lineales o Secuenciales
    - Modelos Basados en Prototipos
    - Desarrollo Rápido de Aplicaciones
  - Modelos Evolutivos
    - Modelo Incremental
    - Modelo Iterativo

- Modelo en Espiral
  - Modelos para Sistemas Orientados a Objetos
    - Modelo de Agrupamiento
    - Proceso Unificado
  - Modelos Basados en Reutilización
    - Ingeniería del Software Basada en Componentes
  - Procesos Ágiles
    - SCRUM
    - Desarrollo de Software Adaptativo
    - Proceso Unificado Ágil
  - Modelos para la Ingeniería Web
    - Modelo IWEB
    - UWE
- Tema 4
  - Ingeniería de Requisitos
    - Definición
    - Proceso de Ingeniería de Requisitos
  - Requisitos
    - Problemas con los Requisitos
    - Requisitos No Funcionales
  - Especificación de Requisitos del Software
  - MDB: Una Metodología de Elicitación de Requisitos
  - Vista de Casos de Uso
    - Actores
    - Casos de Uso
  - Requisitos en el Proceso Unificado
    - Identificación de las Necesidades del Usuario
    - Actividades
    - Construcción del Modelo de Casos de Uso
- Tema 5
  - La Vida del Proceso Unificado
  - El Producto
  - El Proceso
- Tema 6
  - Requisitos en el Proceso Unificado
  - Modelo de Requisitos - Modelo de Casos de Uso
  - Análisis en el Proceso Unificado
  - Diseño en el Proceso Unificado
- Tema 7
  - Análisis
  - Análisis Orientado a Objetos
  - Modelo del Dominio
  - Identificación de Superclases y Subclases
  - Identificación de Relaciones Todo-Parte
- Tema 8
  - UML

- [Vista Estática](#)
- [Relaciones](#)

# Tema 1

---

El software es un producto de consumo con gran peso en la economía mundial.

## Definiciones

- Programas, procedimientos, reglas y la posible documentación asociada y datos que pertenecen a la explotación de un sistema de ordenador.
- Colección organizada de programas de ordenador, procedimientos, documentación asociada y datos referidos a un ordenador.
- Conjunto de programas, procedimientos y documentación asociada a la operación de un sistema informático.

[↑ Volver a índice](#)

## La Crisis del Software

Dificultad inherente, gran complejidad (número de estados elevado, conexión entre entidades y complejidad arbitraria), sujeto a continuos cambios, especificación de requisitos y comunicación de equipo.

Algunas causas de los problemas del software: responsables no cualificados, falta de comunicación, desconocimiento, falta de reconocimiento informático.

[↑ Volver a índice](#)

## Calidad del Software

Los ingenieros del software deben asegurar que sus productos sean de utilidad y calidad aceptable. La calidad se percibe desde diferentes puntos de vista:

- **Factores externos:** Facilidad de mantenimiento, confiabilidad, eficacia, usabilidad, reusabilidad, portabilidad, etc.
- **Factores internos:** Modularidad, tolerancia a fallos, legibilidad, eficiencia de ejecución, facilidad de expansión, autodescripción, etc.

[↑ Volver a índice](#)

## Tipos de Productos Software

Un producto software es un sistema software acompañado de la documentación para su instalación y uso.

- **Tipos de mercados:** productos genéricos y productos personalizados.
- **Áreas de aplicación:** software de sistemas, de tiempo real, empujado, de Inteligencia Artificial, de gestión de computadores personales y aplicaciones web.

**Sistemas de software intensivo:** combinan tecnologías emergentes de sistemas empujados, pero están inmersos en sistemas globales de cómputo. Son sistemas programables (son dinámicos y evolucionan,

comportamiento adaptativo y anticipatorio, procesan conocimiento).

Son productos complejos, gran funcionalidad, objetivos diferentes, intervienen un gran número de personas, elevado tamaño y cambios continuos.

[↑ Volver a índice](#)

## Conceptos Básicos de la Ingeniería del Software

1. Aplicación del conocimiento científico y técnico, métodos y experiencia para el diseño, implementación, prueba y documentación del software.
2. Aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software.

[↑ Volver a índice](#)

## Método de Ingeniería

- Recolección y análisis de requisitos.
- Diseño del sistema.
- Implementación.

[↑ Volver a índice](#)

## Modelo Conceptual de la Ingeniería del Software

Un sistema software es parte de un sistema mayor. La ingeniería del Software es solo una parte del diseño del sistema.

### Conceptos

- **Proceso:** Define el marco de trabajo y permite un desarrollo racional y oportuno.
- **Método:** Cómo construir el software. Técnicas de modelado y otras técnicas descriptivas.
- **Herramientas:** Proporciona soporte automático o semiautomático.
- **Notación:** Reglas gráficas o textuales para la representación de un modelo.
- **Metodologías:** Descompone el proceso en actividades y proporciona métodos para desarrollarlas.

[↑ Volver a índice](#)

## Proceso Software

Marco de trabajo de las tareas que se requieren para construir software de alta calidad.

### Características

- Establecer todas las actividades principales.
- Utiliza recursos, sujeto a restricciones.
- Compuesto de subprocesos.
- Cada actividad tiene criterios de entrada y salida.
- Se organizan en secuencia.

Otras características: comprensión, visibilidad, soporte, aceptación, confianza, robustez, rapidez, adaptación y capacidad de mantenimiento.

Un proceso software debe especificar: la secuencia de actividades, los productos que deben crearse, la asignación de tareas a cada miembro, los criterios para controlar el proceso y las posibles heurísticas.

[↑ Volver a índice](#)

## Ciclo de Vida del Software

Cuando un proceso implica la construcción de un producto, se refiere a éste como un ciclo de vida.

Los proyectos software se desarrollan en diferentes fases que pueden ser temporales o lógicas.

- **Ciclo de vida:** consiste en las fases de análisis de requisitos, diseño, implementación, prueba y mantenimiento.
- **Ciclo de desarrollo software:** periodo de tiempo que comienza con la decisión de desarrollar un producto software y finaliza cuando se entrega.

[↑ Volver a índice](#)

## Modelo de Proceso Software

Representación abstracta de un proceso software. Cada modelo representa un proceso desde una perspectiva particular. Ayuda al equipo de desarrollo a encontrar inconsistencias, redundancias y omisiones.

**Modelo general:** especificación, diseño, fabricación, prueba, instalación y mantenimiento.

**Tres fases genéricas:** definición, desarrollo y mantenimiento.

[↑ Volver a índice](#)

## Metodologías

Proceso para producir software de forma organizada, empleando una colección de técnicas y convenciones de notación predefinida.

**Objetivos:** establecer requisitos de forma acertada, proporcionar un método sistemático de desarrollo, construir un sistema software dentro de un tiempo y costes aceptables, bien documentado y fácil de mantener, etc.

Una metodología debe cubrir: proceso de vida completo, colección de reglas y guías, técnicas probadas, métricas, etc.

### Metodologías Estructurales

Proponen la creación de modelos del sistema que representan los procesos, flujos y estructura de datos descendente. Nivel de abstracción sencillo.

### Metodologías Orientadas a Procesos

Utilizan un enfoque de descomposición descendente para evaluar los procesos del espacio del problema y los flujos de datos conectados.

## Metodologías Orientadas a Datos

Se centran en la parte de entrada/salida.

## Metodologías Orientadas a Objetos

Integración de datos y procesos, un sistema se concibe como un conjunto de objetos que se comunican mediante mensajes, acortan la distancia entre espacio de conceptos y espacio de diseño e implementación.

## Metodologías Ágiles

Los sistemas web caracterizados por su presión temporal priorizan el *cuándo* sobre el *qué*. La reducción del ciclo de desarrollo software sin perder calidad ni capacidades de evolución y mantenimiento.

Los procesos software que intentan dar solución a estas circunstancias se conocen como procesos ágiles o procesos ligeros.

La agilidad no significa poner en el mercado o en explotación los productos software más rápidamente.

[↑ Volver a índice](#)

## Tecnología CASE

Software que se utiliza para dar soporte a las actividades del proceso, ofrece soporte al proceso software automatizando actividades y da información sobre el software.

**Diferentes perspectivas:** funcional, de proceso y de integración.

[↑ Volver a índice](#)

---

# Tema 2

---

## Sistema

Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto.

**Elementos principales:** componentes, relaciones entre componentes y objetivo.

**Elementos secundarios:** entorno, límites y realimentación.

[↑ Volver a índice](#)

## Sistema de Información

Combinación organizada de personas, mecanismos físicos, procedimientos e instrucciones de procesamiento de información, canales de comunicación y; datos almacenados que reúne, transforma y disemina información en una organización.

Cuenta con un sistema informático de soporte.

**Elementos:** procedimientos, información, personas o usuarios y equipo de soporte.

## Estructura de los Sistemas de Información

### Niveles

- Operaciones y transacciones.
- Nivel operativo.
- Nivel táctico.
- Nivel estratégico.

### Flujos de Información

- Vertical ascendente.
- Vertical descendente.
- Horizontal.

### Información

La información que se maneja en los niveles bajos de la pirámide es de naturaleza descriptiva y en los altos, de naturaleza comparativa.

## Clasificación de los Sistemas de Información

Los sistemas de información tienen diferentes propósitos dependiendo de las necesidades del negocio u organización.

Existen muchos tipos y diferentes clasificaciones, una de las más comunes es la siguiente:

- Procesamiento de transacciones.
- Automatización de oficina.
- Manejo del conocimiento.
- Información administrativa o gerencial.
- Apoyo a las decisiones.
- Apoyo a las decisiones de grupo.
- Apoyo a ejecutivos o de planificación estratégica.

[↑ Volver a índice](#)

## Principios Generales de Sistemas

El estudio de las características comunes de los sistemas se conoce como teoría general de sistemas. Los principios de esta teoría se puede aplicar a los sistemas automatizados.

El análisis o estudio de los sistemas se realiza siguiendo el enfoque sistemático u holístico. Proceso general para analizar y explicar elementos correlacionados que constituyen sistemas.

[↑ Volver a índice](#)

# Ingeniería de Sistemas

Actividad de especificar, diseñar, implementar, distribuir y mantener sistemas como un todo.

**Se debe considerar:** Hardware, Software e interacciones con usuario y entorno.

**Fases:** definición de requisitos, diseño, desarrollo de subsistemas, instalación del sistema, evolución del sistema y desmantelamiento del sistema.

El proceso es iterativo entre fases y dentro de cada fase.

## Adquisición del Sistema

Se puede comprar como un todo, como partes separadas que deben integrarse o como un sistema diseñado y desarrollado.

Antes de la adquisición se debe realizar una especificación de alto nivel y un diseño arquitectónico.

[↑ Volver a índice](#)

---

## Tema 3

---

### Modelos Tradicionales

Formados por un conjunto de fases o actividades en las que no tienen en cuenta la naturaleza evolutiva del software.

#### Modelo Primitivo

Proceso de las primeras experiencias de programación, supone una iteración de fases codificación-depuración sin ninguna planificación ni diseños previos.

#### Modelos Lineales o Secuenciales

- **Modelo clásico:** en cascada, compuesto por una serie de fases que se ejecutan secuencialmente, se pasa de fase al conseguir objetivos, el final de una fase puede suponer un punto de revisión. La linealidad no se corresponde con la realidad.
- **Modelo en V:** variación del modelo en cascada, presenta implantación ascendente, el desarrollo de las pruebas se efectúa de manera síncrona con el desarrollo del programa, centra su atención en la actividad y exactitud.

#### Modelos Basados en Prototipos

Modelo experimental de un sistema o componente.

Enfoques de desarrollo:

- **Desechable:** versión rudimentaria del sistema que posteriormente se desecha.
- **Evolutivo:** el prototipo debe convertirse en el sistema final usado.



- **Mixto.**

## Desarrollo Rápido de Aplicaciones

Surgió como respuesta al modelo formal y al ciclo en espiral, enfatiza un ciclo de desarrollo extremadamente corto, no es un modelo bien definido.

[↑ Volver a índice](#)

## Modelos Evolutivos

Se adaptan a la evolución que sufren los requisitos del sistema en función del tiempo.

### Modelo Incremental

Se divide el subsistema de acuerdo a su funcionalidad, las versiones se definen comenzando con un subsistema funcional pequeño y agregando funcionalidad con cada nueva versión.

### Modelo Iterativo

Entrega un sistema completo desde el inicio, para posteriormente cambiar la funcionalidad de cada subsistema con cada iteración. Minicascada. Deben definirse criterios de evaluación de las iteraciones.

### Modelo en Espiral

Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. El avance se realiza desde el centro de la espiral hacia el exterior.

Se divide en un número de actividades estructurales (regiones de tareas): planificación, análisis de riesgos, ingeniería y evaluación del cliente.

[↑ Volver a índice](#)

## Modelos para Sistemas Orientados a Objetos

Modelos con un alto grado de iteratividad y solapamiento entre fases.

**Características principales:** eliminación de las fronteras entre fases, uso de componentes reutilizables, alto grado de iteratividad y solapamiento, desarrollo incremental.

### Modelo de Agrupamiento

Unidad organizativa clave, grupo de clases relacionadas. Tienen un componente secuencial y un componente concurrente, existencia de diferentes subciclos de vida. Enfoque ascendente.

### Proceso Unificado

Conducido por casos de uso, centrado en la arquitectura, iterativo e incremental. Se repite a lo largo de una serie de ciclos, cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición.

Cada fase se puede descomponer en iteraciones. Cada fase termina con un hito. Las iteraciones discurren a lo largo de los flujos de trabajo.

[↑ Volver a índice](#)

## Modelos Basados en Reutilización

Tienen en cuenta la reutilización sistemática del software, existencia de un número significativo de elementos reutilizables. Tiende a estructurarse en dos subprocesos distintos y separados.

Las unidades software reutilizables pueden ser de distintos tamaños: sistemas de aplicaciones, componentes y funciones.

Familias de aplicaciones, existen varios tipos de especialización: plataforma, configuración y funcionalidad.

### Ingeniería del Software Basada en Componentes

El objetivo es identificar, construir, catalogar y diseminar un conjunto de componentes de software.

- **Actividades de la ingeniería del dominio:** análisis del dominio, modelo del dominio y modelado estructural.
- **Actividades del desarrollo basado en componentes:** cualificación, adaptación, composición, ingeniería y actualización de componentes.

[↑ Volver a índice](#)

## Procesos Ágiles

Nuevo enfoque en el desarrollo del software, características: menor énfasis en diseño, análisis y documentación, equipos pequeños, desarrollo incremental, programación en cajas de tiempo y supervivencia en un entorno caótico.

### SCRUM

Dentro de cada actividad, las tareas se organizan con un patrón de proceso (sprint), se adapta al problema y se define y modifica en tiempo real por el equipo SCRUM.

### Desarrollo de Software Adaptativo

Basado en la colaboración y orientado al desarrollo de sistemas complejos.

Fases del ciclo de vida: especulación, colaboración y aprendizaje.

### Proceso Unificado Ágil

Versión simplificada del proceso unificado.

Disciplinas: modelo, puesta en práctica, prueba, despliegue, gestión de la configuración, gestión del proyecto y ambiente.

Lanzamiento de versiones incrementales a lo largo del tiempo.

[↑ Volver a índice](#)

## Modelos para la Ingeniería Web

Creados específicamente para el desarrollo de aplicaciones web.

**Características:** intensivas de red, controladas por contenido, evolución continua, inmediatez, estética, etc.

**Ciclo de desarrollo:** definición y análisis, diseño y pruebas.

## Modelo IWEB

Toma como base el modelo de ciclo de vida en espiral.

**Etapas:** formulación, planificación, análisis, ingeniería, generación de páginas y pruebas y evaluación con el cliente.

Propuso una modificación con las siguientes actividades: comunicación con el cliente, planificación, modelado, construcción y despliegue.

## UWE

**Características:** desarrollo iterativo e incremental, uso de UML, centrado en la sistematización y automatización.

[↑ Volver a índice](#)

---

# Tema 4

---

## Ingeniería de Requisitos

Primera fase del ciclo de vida del software, deben obtenerse y documentarse los requisitos de información, funcionales, no funcionales y criterios para medir el grado de su consecución.

### Definición

Un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades.

### Proceso de Ingeniería de Requisitos

1. Obtención (elicitación) de requisitos.
2. Análisis de requisitos.
3. Verificación de requisitos.
4. Validación de requisitos.
5. Negociación de requisitos.
6. Gestión de requisitos.

[↑ Volver a índice](#)

## Requisitos

Una propiedad que debe exhibirse para solucionar algún problema del mundo real. Aparente simplicidad del concepto.

Para clarificar la situación es frecuente identificar dimensiones para clasificar los requisitos: ámbito, características que define, audiencia y representación.

## Problemas con los Requisitos

- No reflejan las necesidades reales del cliente.
- Inconsistentes o incompletos.
- El cambio de requisitos es muy costoso.
- Problemas de comunicación.

## Requisitos No Funcionales

No relacionados directamente con la funcionalidad del sistema, definen las cualidades globales que el sistema ha de exhibir, suelen ser más críticos y difíciles de verificar.

**Clasificación:** requisitos de producto, de organización y externos.

[↑ Volver a índice](#)

## Especificación de Requisitos del Software

Una especificación es un documento que define, de forma completa, precisa y verificable, los requisitos, el diseño, el comportamiento u otras características de un sistema o componente de un sistema.

Propiedades deseables de los requisitos: comprensible, no ambigua, completa, consistente, verificable, modificable, trazable, anotada con importancia y estabilidad, independiente del diseño y de la implementación.

## MDB: Una Metodología de Elicitación de Requisitos

Metodología para la obtención y documentación de los requisitos de sistemas de información.

**Técnicas más importantes:** casos de uso, plantillas y patrones lingüísticos.

[↑ Volver a índice](#)

## Vista de Casos de Uso

Un diagrama de casos de uso es un grafo de actores, un conjunto de casos de uso encerrados por los límites de un sistema, asociaciones entre los actores y los casos de uso y relaciones de generalización entre los actores.

### Actores

Idealización de una persona, proceso o entidad externa que interacciona con un sistema, subsistema o clase.

**Tipos de actores:** principales, de apoyo y pasivos.

Se representan en UML como un monigote. Puede definirse una jerarquía de generalización de actores:



## Casos de Uso

Unidad coherente de funcionalidad externamente visible proporcionada por un clasificador y expresada mediante secuencias de mensajes intercambiados por el sistema y uno o más actores de la unidad del sistema.

**Cuatro dimensiones:** propósito, contenidos, pluralidad y estructura.

Se representan en UML como una elipse conteniendo el nombre.

Pueden participar en varias relaciones (asociación, extensión, generalización e inclusión), además de poderse asociar con actores.

**<<include>>** las condiciones de destino se inician son descritas en el caso de uso iniciado, mientras que **<<extend>>** se inician en la extensión.

[↑ Volver a índice](#)

## Requisitos en el Proceso Unificado

Construye dos modelos principales: modelo de dominio y modelo de casos de uso. Se desarrollan de forma incremental.

### Identificación de las Necesidades del Usuario

Recogen información del dominio de la aplicación: investigación, observación, entrevistas y prototipado.

Distinguir las necesidades de los deseos.

### Actividades

Proceso iterativo: identificar actores, escenarios, casos de uso, refinar casos de uso, identificar relaciones entre casos de uso e identificar requisitos no funcionales.

### Construcción del Modelo de Casos de Uso

Cada caso de uso debe representar un comportamiento distinto e identificable del sistema o de una parte del mismo.

Para el análisis de requisitos hay que centrarse en los casos de uso al nivel de procesos del negocio elementales o EBPs.

Los casos de uso se suelen nombrar comenzando por un verbo.

[↑ Volver a índice](#)

---

## Tema 5

---

El proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Está basado en componentes y utiliza UML.

[↑ Volver a índice](#)

### La Vida del Proceso Unificado

Se repite a lo largo de una serie de ciclos de desarrollo que constituyen la vida de un sistema y concluye con una versión entregable. Tiene cuatro fases: inicio, elaboración, construcción y transición.

- **Etapas de ingeniería:** inicio y elaboración.
- **Etapas de producción:** construcción y transición.

Los hitos son puntos de control en los cuales los participantes en el proyecto revisan el progreso del proyecto.

Las disciplinas o flujos de trabajo organizan las actividades fundamentales de gestión y desarrollo del proyecto.

[↑ Volver a índice](#)

### El Producto

Es un sistema de software lo componen todos los artefactos necesarios. El artefacto más importante es el modelo.

**Modelos:** caso de uso, análisis y diseño, despliegue, implementación y pruebas.

Existen dependencias entre el modelo de casos de uso y los demás modelos.

[↑ Volver a índice](#)

### El Proceso

Es una definición de un conjunto completo de actividades necesarias para convertir los requisitos de usuario en un conjunto consistente de artefactos que conforman un producto software, y para convertir los cambios sobre esos requisitos en un nuevo conjunto consistente de artefactos.

**Características principales:** conducido por casos de uso, centrado en la arquitectura, iterativo e incremental.

[↑ Volver a índice](#)

---

## Tema 6

---

## Requisitos en el Proceso Unificado

Construye 2 modelos principales: dominio y casos de uso.

Se desarrollan de forma incremental, principalmente en las fases de inicio y elaboración: inicio, elaboración construcción y transición.

La captura y recogida de requisitos se centra en la visión que el usuario tiene del sistema.

Recoger información del dominio de la aplicación: investigación, observación, entrevistas y prototipado.

Distinguir las necesidades de los deseos.

[↑ Volver a índice](#)

## Modelo de Requisitos - Modelo de Casos de Uso

El objetivo es delimitar el sistema y definir qué funcionalidad tiene que afectar al sistema. El primer modelo a crear es el de casos de uso, son una representación orientada a la funcionalidad del sistema.

- **Escenario:** Hechos que describen un sistema existente y su entorno incluyendo el comportamiento de los agentes con la suficiente información de contexto para permitir el descubrimiento y la validación de los requisitos del sistema.

[↑ Volver a índice](#)

## Análisis en el Proceso Unificado

- **Modelo de análisis:** se representa mediante un sistema de análisis que denota el paquete de más alto nivel.
- **Clase de análisis:** abstracción de una o varias clases y/o subsistemas del diseño del sistema. Tres tipos de objetos: entidad, interfaz y control.
- **Realización de casos de uso - Análisis:** colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y cómo se ejecuta un caso de uso en término de las clases de análisis y de sus objetos.
- **Paquete de análisis:** medio para organizar los artefactos del modelo de análisis en piezas manejables.
- **Descripción de la arquitectura:** contiene una vista de la arquitectura del modelo de análisis, que muestra sus artefactos significativos para la arquitectura.

[↑ Volver a índice](#)

## Diseño en el Proceso Unificado

- **Modelo de diseño:** modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con tras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

- **Clase de diseño:** representa una abstracción de una o varias clases en la implementación del sistema.
- **Realización de casos de uso - Diseño:** colaboración en el modelo de diseño que describe cómo se realiza un caso de uso específico.
- **Subsistema de diseño:** forma de organizar los artefactos del modelo de diseño en piezas más manejables.
- **Interfaz:** especifica una colección de operaciones públicas, tipos y parámetros necesarios para acceder y usar las capacidades de una clase de diseño o subsistema.
- **Modelo de despliegue:** modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.
- **Descripción de la arquitectura:** contiene una vista de la arquitectura del modelo de diseño que muestra sus artefactos relevantes para la arquitectura.

[↑ Volver a índice](#)

---

## Tema 7

---

### Análisis

Se define como: La distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos.

[↑ Volver a índice](#)

### Análisis Orientado a Objetos

Proceso que modela el dominio del problema mediante la identificación y la especificación de un conjunto de objetos semánticos que interaccionan y se comportan de acuerdo a los requisitos del sistema.

Se centra en la elaboración de un modelo del sistema, el modelo de análisis: modelo funcional, modelo objeto análisis y modelo dinámico.

El modelo de análisis estructura el sistema independientemente del entorno actual de implementación.

[↑ Volver a índice](#)

### Modelo del Dominio

Representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades.

Para hacer un modelo de dominio:

- Listar las clases conceptuales candidatas.
- Representar las clases en un modelo de dominio.



- Añadir las asociaciones necesarias.
- Añadir los atributos necesarios.

[↑ Volver a índice](#)

## Identificación de Superclases y Subclases

Todos los miembros del conjunto de una subclase conceptual son miembros del conjunto de su superclase.

**Modelado de los cambios de estado:** no se debe modelar el estado de un concepto X como subclases de X, sino que se debe seguir una de estas estrategias:

- Definir jerarquía de estados y asociar los estados de X.
- Ignorar la representación de los estados de un concepto en el modelo de dominio; en lugar de esto, representar los estados en diagramas de estado.

[↑ Volver a índice](#)

## Identificación de Relaciones Todo-Parte

Tres tipos de configuraciones:

- Ensamblaje y sus partes: ordenador (placa base, disco, etc).
- Contenedor y sus contenidos: oficina (mesas, teléfonos, estanterías, etc).
- Grupo y sus miembros: asociación y sus miembros.

Dos tipos de relación:

- **Agregación:** Construido a partir de otros objetos, es mayor que la suma de sus partes, todas las interacciones a través de la interfaz del objeto agregado y los objetos componentes están ocultos o encapsulados dentro del agregado.
- **Composición:** Forma fuerte de agregación, el ciclo de vida de las partes dependen del ciclo de vida del agregado, las partes no existen fuera de su participación en el agregado y la pertenencia fuerte implica objetos físicos que se unen para formar el compuesto.

[↑ Volver a índice](#)

---

## Tema 8

---

### UML

Es un lenguaje de modelado para visualizar, especificar, construir y documentar partes de un sistema software desde distintos puntos de vista.

El objetivo de UML es la unificación de los métodos de modelado de objetos. La especificación UML se define usando el enfoque de metamodelo.

UML define varios modelos para la representación de los sistemas que pueden verse y manipularse mediante un conjunto de diagramas diferentes:

- Diagramas de estructura
- Diagramas de comportamiento

Una vista es un subconjunto de las construcciones de modelado de UML que representa un aspecto del sistema.

[↑ Volver a índice](#)

## Vista Estática

Modela conceptos del dominio de la aplicación, sus propiedades internas y sus relaciones.

Componentes principales: clases y relaciones.

Características de las clases:

- Tienen un nombre único dentro de su contenedor.
- Tiene una visibilidad con respecto a su contenedor.
- Se representa por rectángulos compartimentados.

[↑ Volver a índice](#)

## Relaciones

- **Asociaciones:** una instancia es un enlace, llevan la información sobre relaciones entre objetos, puede ser recursiva, binaria o n-aria, puede contar con: símbolo de agregación, indicador de navegación, multiplicidad, visibilidad, cadena de propiedades.

Clases asociación: se utilizan cuando cada enlace debe tener sus propios valores para los atributos, operaciones propias o sus propias referencias a objetos.

- **Agregación:** Forma de asociación que representa una relación todo-parte entre un agregado y las partes que los componen.
- **Agregación compartida:** Las partes pueden pertenecer a cualquiera de los agregados.
- **Agregación compuesta:** Asociación de agregación con las restricciones adicionales de que un objeto sólo puede ser parte de un compuesto a la vez y que el objeto compuesto es el único responsable de la disponibilidad de todas sus partes.
- **Generalizaciones:** Relación entre un elemento general y otro más específico que es plenamente consistente con el primer objeto y que le añade información adicional.

[↑ Volver a índice](#)