



# FUNDAMENTOS DEL DISEÑO DE SOFTWARE

# OBJETIVOS

- Reconoce qué es el diseño de software y cuál es su importancia.
- Identifica los distintos factores que impactan en la arquitectura.
- Propone una arquitectura base para un proyecto de software.



1

# DISEÑO DE SOFTWARE

# ¿QUÉ ES EL DISEÑO DE SOFTWARE?

- Etapa en la que:
  - Se analizan los requerimientos del software.
  - Se obtiene una descripción de la estructura interna del software.
    - Se utiliza como base para la construcción del software.

# TIPOS DE DISEÑO DE SOFTWARE

- Diseño Arquitectónico (Diseño de alto nivel):
  - Define la estructura general del Software.
  - Especifica el contexto de la solución (plataforma – entorno).
  - Identifica los componentes del software a implementar: su estructura y organización.
- Diseño detallado:
  - Especifica el detalle de cada componente.
  - Especifica cómo se va a implementar cada componente.

# ¿POR QUÉ ES IMPORTANTE ESTA ETAPA?

- Análisis y evaluación de los modelos ("planos") elaborados.
  - ¿Nos ayudan a cubrir los requerimientos?
- Búsqueda de alternativas de solución.
- Análisis de ventajas y desventajas.
- Negociación de requerimientos.
- Insumo de entrada para la implementación y pruebas.



# 2

# ARQUITECTURA DE SOFTWARE

Una introducción ...



# Planos del Sistema





Hoja de ruta para la  
implementación del Sistema.



Conjunto de estructuras necesarias para idear el sistema. Comprende **elementos de software, sus relaciones y propiedades.**

-- Clements et al., 2010



La materialización de los conceptos o propiedades fundamentales de un sistema en sus **elementos, relaciones y principios de su diseño y evolución.**

-- ISO/IEC/IEEE 42010:2011

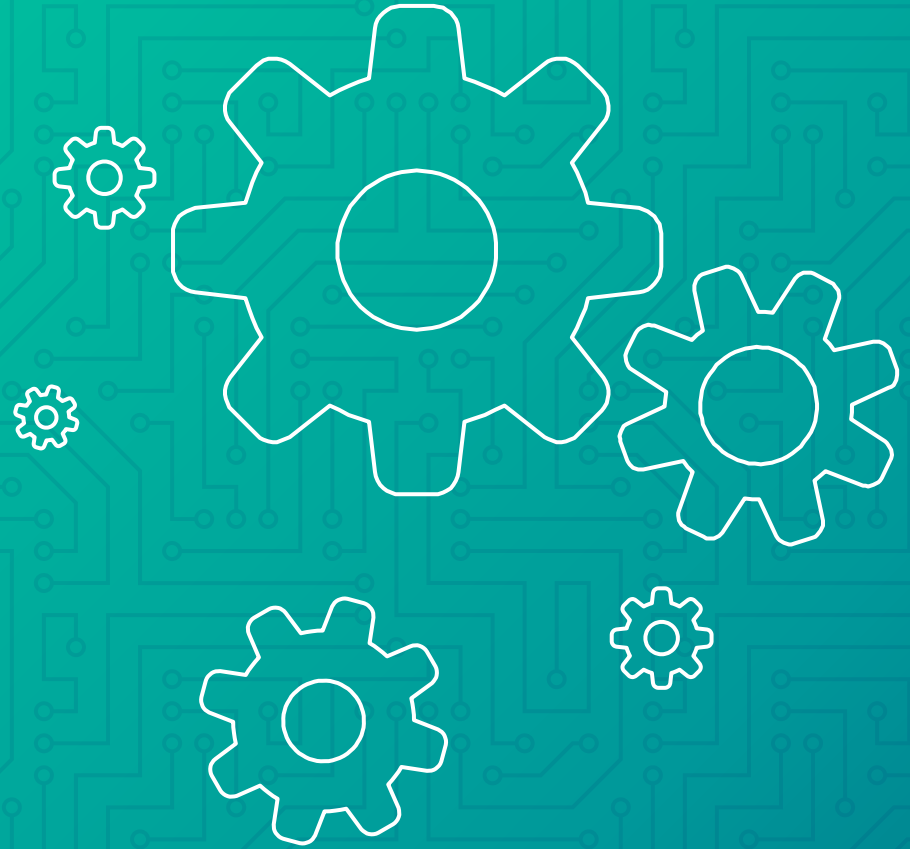


La Arquitectura de Software consiste en la **estructura del sistema**; combinada con las **características arquitectónicas** que el sistema debe soportar, **las decisiones arquitectónicas** y los **principios de diseño**.

-- Neal Ford

# ETAPAS INICIALES

La Arquitectura de Software debe definirse en las etapas iniciales de un proyecto de software.



# 3

## INFLUENCIAS EN LA ARQUITECTURA

Factores que impactan en el diseño de la arquitectura

# FACTORES

- Relacionados al Producto
  - Requerimientos funcionales
  - Requerimientos de calidad (Requerimientos no Funcionales)
- Restricciones
  - Tecnológicas (Leng. programación, librerías, infraestructura)
  - Organizacionales (cultura, estructura, socios, recursos, capacidades)
  - Legales / Regulatorias (leyes, normas, protección de datos, responsabilidades)

# REQUERIMIENTOS DE CALIDAD



Más información: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>



# 4

## VISTAS ARQUITECTÓNICAS

¿Qué son y cuántas elaborar?



Representa un **aspecto parcial** de una arquitectura de software que muestra **propiedades específicas** de un sistema software.

-- Clemments et al. 2010

# PUNTOS DE VISTA

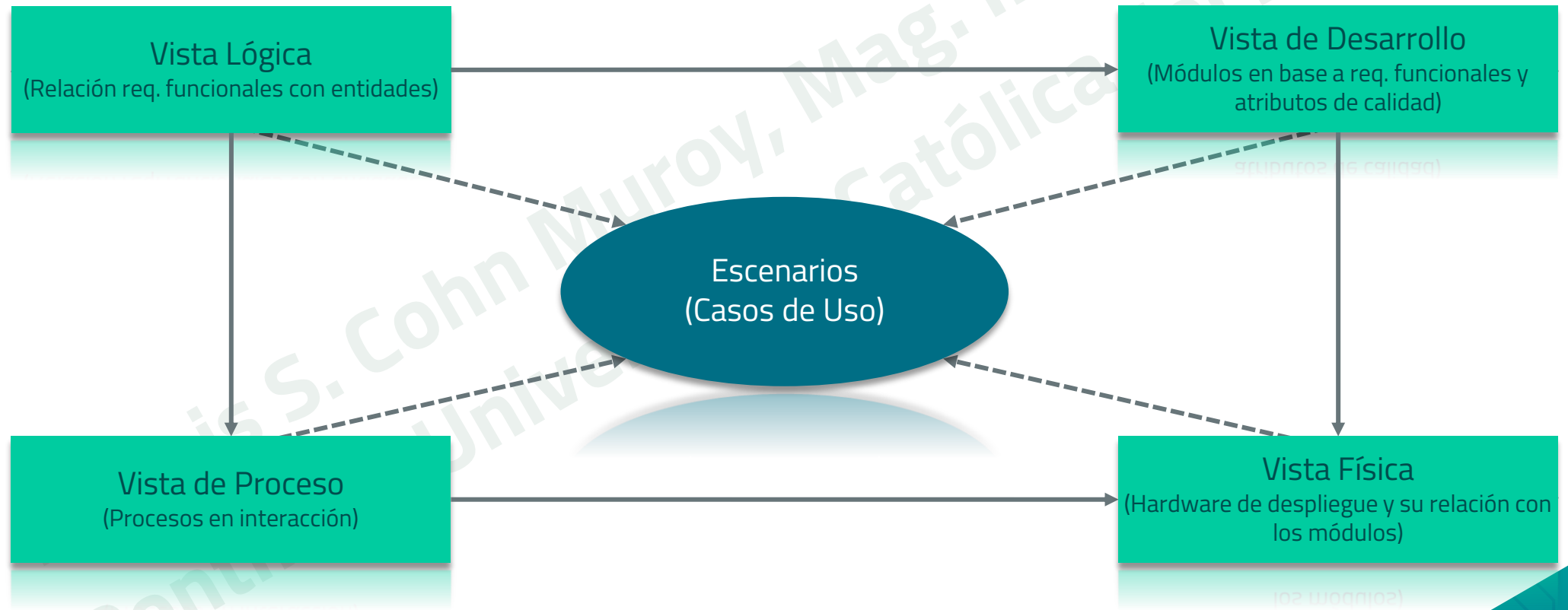
- Describen un comportamiento particular del sistema de acuerdo al punto de vista de los **interesados**.
- Se deben escoger en base a:
  - Escenarios clave.
  - Preocupaciones.

# MATRIZ DE INTERESADOS

Interesado (*)	Descripción	Interés	Vistas / Puntos de Vista
Nombre del interesado o el rol que tiene.	Descripción del interesado.	Requerimientos, escenarios clave y preocupaciones.	Vista o Puntos de Vista que cubren el requerimiento, interés o la preocupación.
...	...	...	

(\*) Interesado es cualquier persona o actor que se verá impactado de forma positiva o negativa por el proyecto.

# MODELO 4+1



# MODELO C4

## Contexto

- Sistema como una caja negra.
- Relaciones con actores y sistemas externos.

## Contenedor

- Descomposición del sistema en bloques de construcción y sus relaciones.
- Cada elemento es un módulo autocontenido bajo ejecución independiente.
- Ejemplos: sistema web, aplicativo móvil, base de datos, servidor de archivos, etc.

## Componente

- Descomposición de cada bloque de construcción.
- Los componentes pueden definirse en base al contexto del dominio (seguridad, analítica, ventas, etc.) o por una taxonomía técnica (controlador, modelo, vista, servicio, etc.).

## Código

- Detalle de los componentes.
- Diagramas de clase de diseño, diagramas entidad relación, diagramas de interacción.



**5**

**IDENTIFICANDO  
COMPONENTES**

# PASOS PARA IDENTIFICAR COMPONENTES

- Identificar componentes iniciales:
  - Por dominio.
  - Por tecnología (capacidades, taxonomía).
- Asignar requerimientos funcionales a los componentes.
- Analizar roles y responsabilidades de los requerimientos.
- Analizar atributos de calidad.
- Reestructurar componentes en base a la retroalimentación.



# MATRIZ DE TRAZABILIDAD

Componente	Descripción	Roles & Responsabilidades	Requerimientos Funcionales	Atributos de Calidad
Nombre del componente	Descripción del componente. Justificar su necesidad.	Actores que interactuarán con el componente	Requerimientos funcionales principales que asumirá el componente.	Atributos de Calidad que el componente garantiza.
...	...		...	

En caso hubiera requerimientos funcionales o atributos de calidad no asignados a componentes, puede significar que ha omitido alguno.

# 6

## REFERENCIAS

# BIBLIOGRAFÍA

- Sommerville, I. (2011). Software engineering (ed.). America: Pearson Education Inc.
- Starke, G., & Lorz, A. (2021). Software Architecture Foundation: CPSA Foundation® Exam Preparation (2nd ed.). Van Haren.

## Créditos:

- Plantilla de la presentación por [SlidesCarnival](#)
- Fotografías por [Unsplash](#)
- Diseño del fondo [Hero Patterns](#)