



## ARQUITECTURA DE SOFTWARE

ARQUITECTURA – ASY4131

# **1.2.4**

## **Estilos y Patrones de Arquitectura**

# Estilos y Patrones

***“ Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y entonces se describe el núcleo de la solución a dicho problema de tal forma que se puede usar esta solución un millón de veces sin hacerlo dos veces de la misma forma” (Christopher Alexander).***

# Analicemos



*¿ Es importante Reutilizar ?*

*¿ Existe posibilidad de ahorros utilizando patrones de diseño ?*

# Estilos Arquitectónicos

Un **estilo arquitectónico** describe una categoría del sistema que contiene:

- Un conjunto de componentes, que realiza una función requerida por el sistema.
- Un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes; Restricciones que definen como se puede integrar los componentes que forman el sistema y,
- Modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema para analizar las propiedades conocidas de sus partes constituyentes.



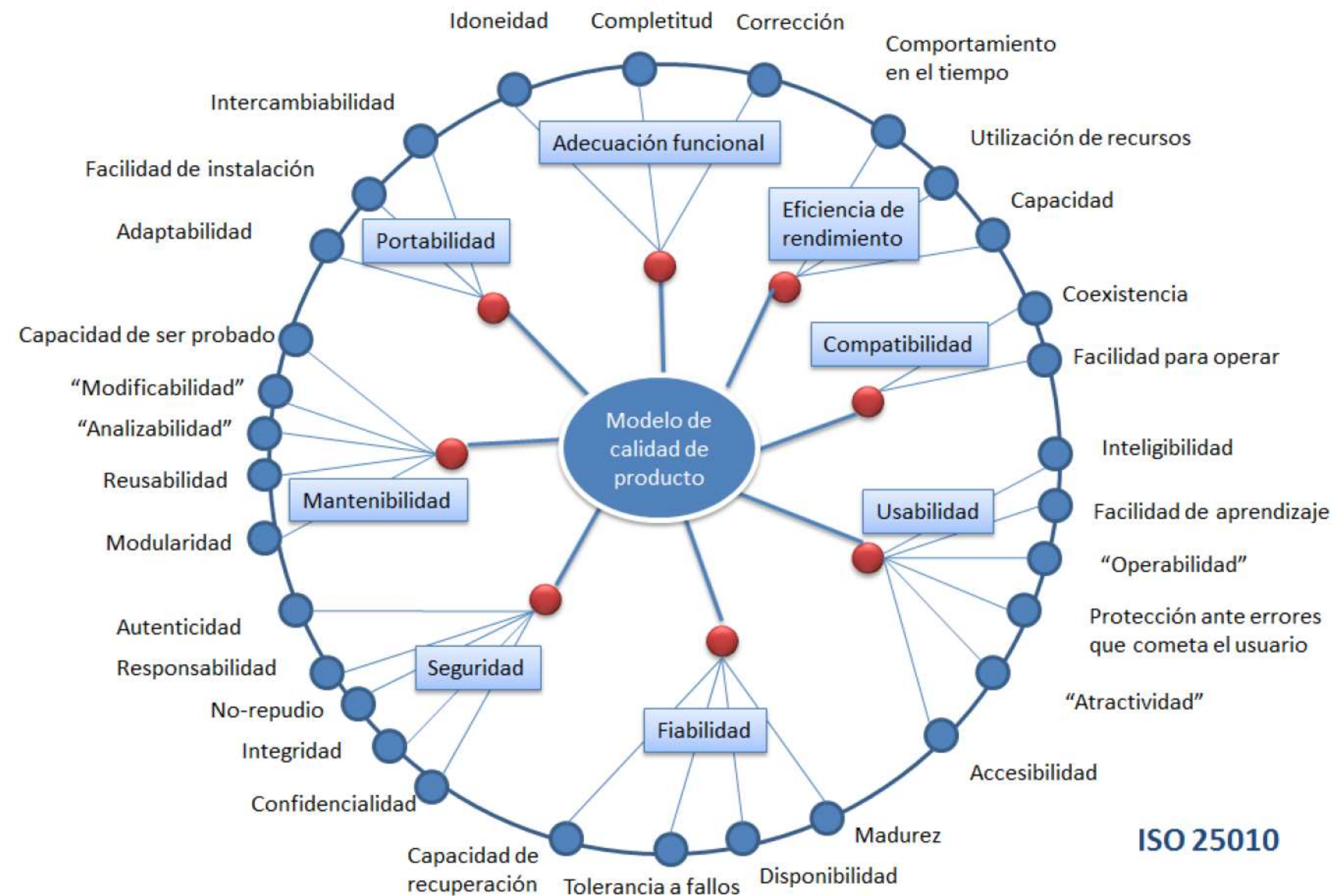
Un **estilo arquitectónico** describe:

- Los **componentes** incluyendo sus **responsabilidades**
- Un conjunto de **conectores** entre componentes (comunicación, coordinación, cooperación, etc.)
- **Restricciones** que definen cómo se **integran los componentes** para formar el sistema
- Los **modelos** que permiten comprender las propiedades de un sistema general en función de las propiedades conocidas de las partes que lo integran



# Estilo Arquitectónico

Los estilos impactan en las propiedades del sistema basados en la ISO 9126-25000

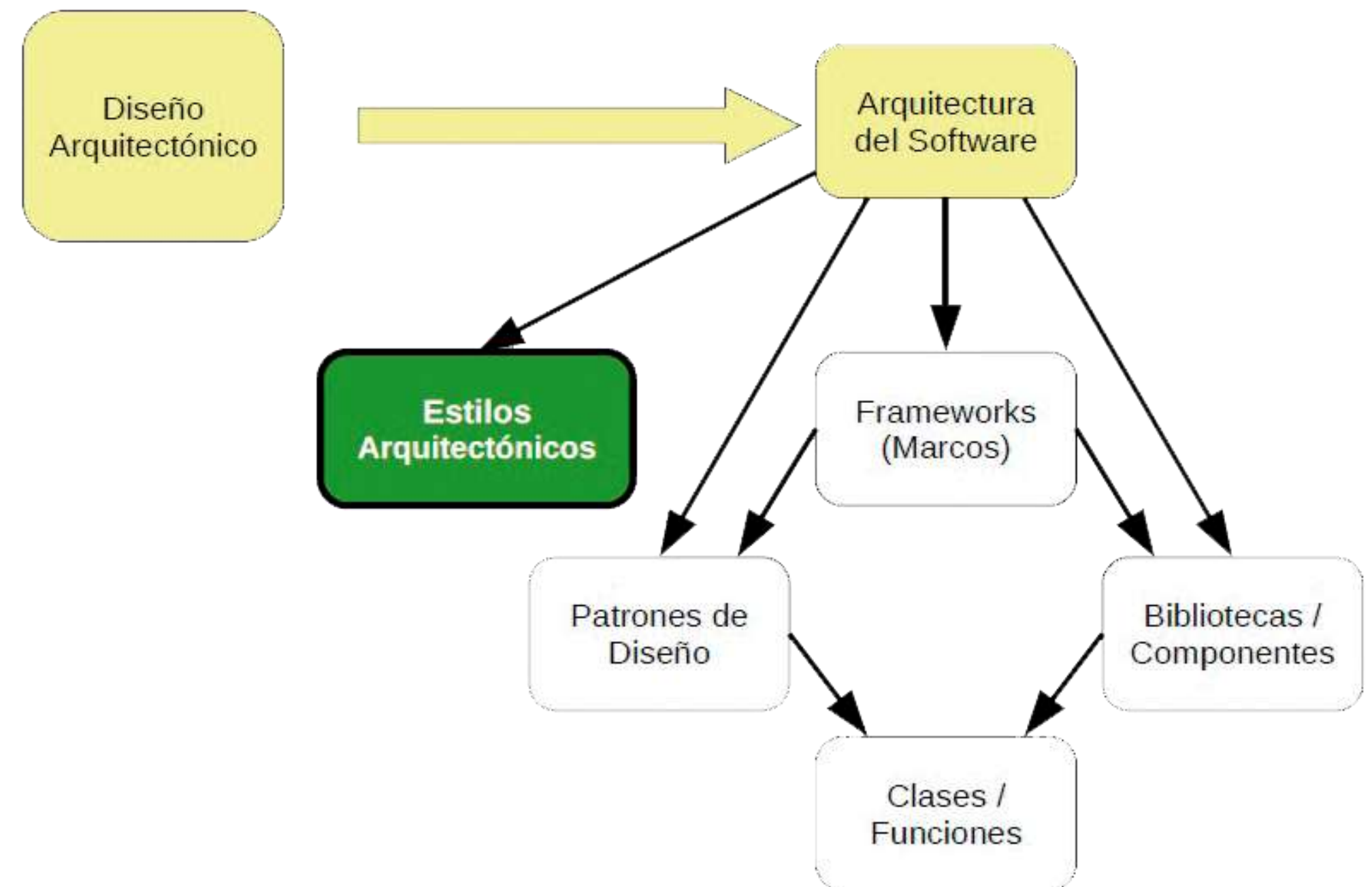


ISO 25010

# Patrón Arquitectónico

Un **patrón arquitectónico** expresa un esquema de **organización estructural** esencial para un sistema de software, que consta de **subsistemas**, sus **responsabilidades** e **interrelaciones**.

Un **patrón arquitectónico** es un conjunto de **decisiones de diseño** que se **repiten** en la práctica, que tienen **características bien definidas** y que pueden **reutilizarse**, describiendo así características de una **arquitectura bien diferenciada**.





# Patrones Arquitectónicos

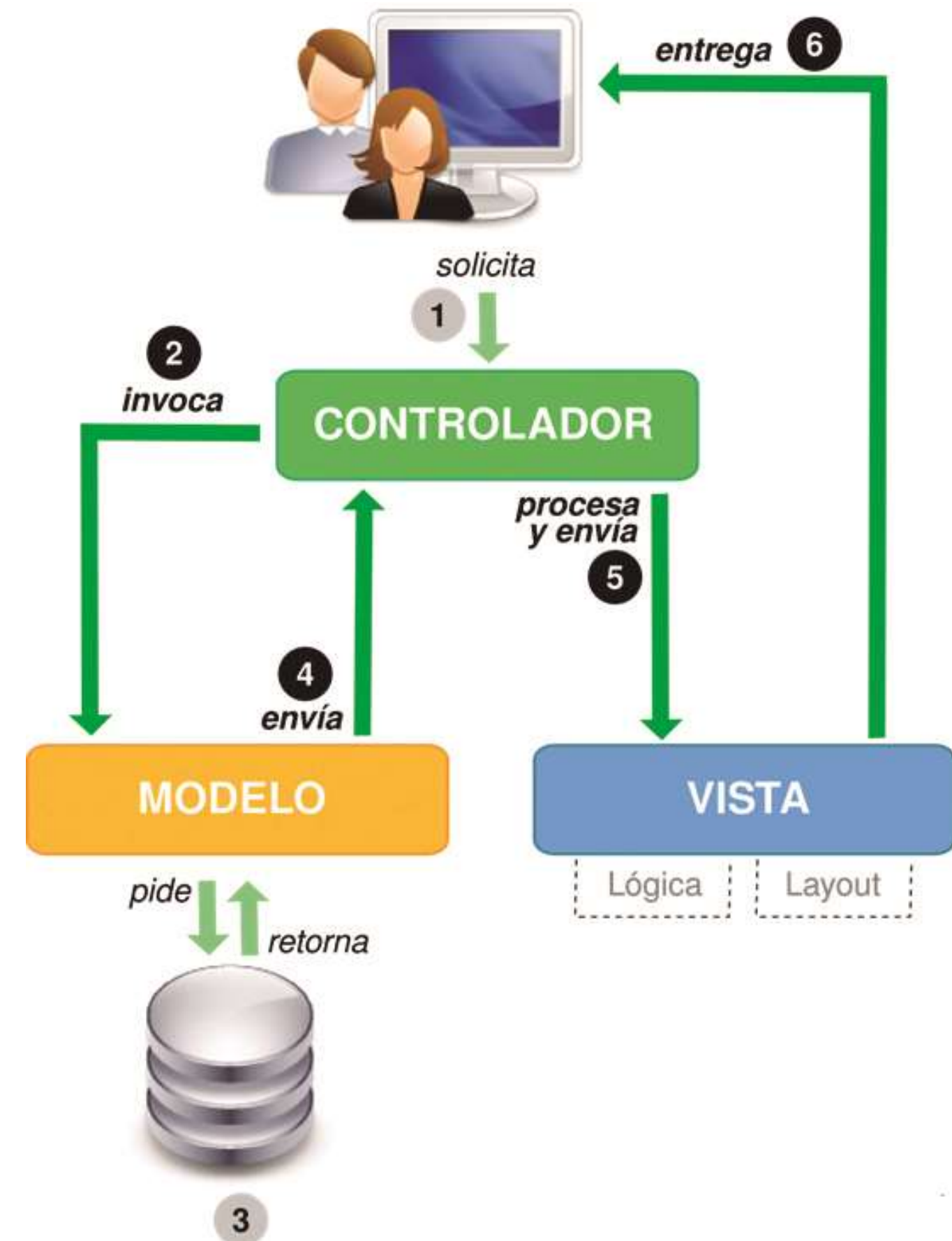
## Algunos ejemplos de patrones arquitectónicos:

- Programación por N capas
- Tres niveles
- Pipeline (Tuberías)
- Arquitectura en pizarra
- Arquitectura dirigida por eventos, Presentación-abstracción-control
- Peer-to-peer (de igual a igual)
- **Arquitectura orientada a servicios (SOA)**
- Aplicaciones Orientadas a Servicios (SODA)
- **Cliente - Servidor**
- **Modelo Vista Controlador**
- **Microservicios**

# Patrón MVC

**Modelo-Vista-Controlador** es un patrón de arquitectura de las aplicaciones software que separa la lógica de negocio de la interfaz de usuario, además facilita la evolución por separado de ambos aspectos e incrementa la reutilización y flexibilidad.

Este patrón es uno de los más usados, en la actualidad se puede encontrar, tanto en pequeños desarrollos, como en sistemas robustos.



# Patrón SOA

La arquitectura **orientada a servicios** (SOA) es un marco de trabajo conceptual que permite a las organizaciones unir los objetivos de negocio con la infraestructura de TI integrando los datos y la lógica de negocio de sus sistemas separados. Desarrollada a final de los '90, SOA establece un marco de trabajo para servicios de red tareas comunes de negocios – para identificar el uno al otro y comunicarlo. (mayo 2007 , D. Marsili, de Sybase)

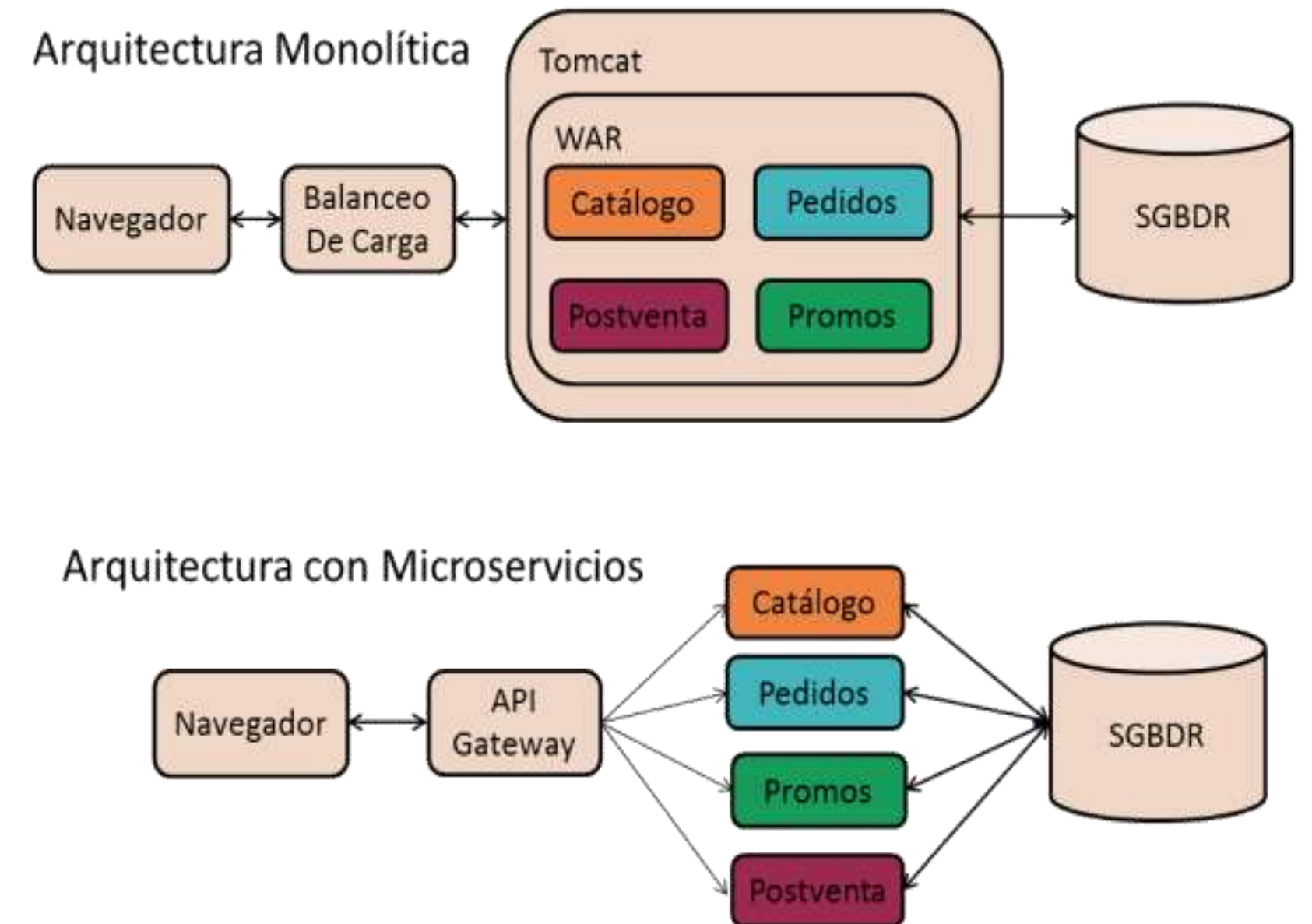


# Patrón de Servicios

La arquitectura de **microservicios** es un enfoque para desarrollar una aplicación de software, como una serie de pequeños servicios, cada uno ejecutándose de forma autónoma y comunicándose entre sí, por ejemplo, a través de peticiones HTTP a sus API.

## Beneficios:

- Cada microservicio se puede desplegar de forma independiente: Por ejemplo, un cambio en el módulo de pedidos, no afectará a los demás, solo actualizando ese módulo.
- Es fácil de entender, ya que la lógica de negocio está bien separada.
- Facilita la gestión de equipos multifuncionales y autónomos.



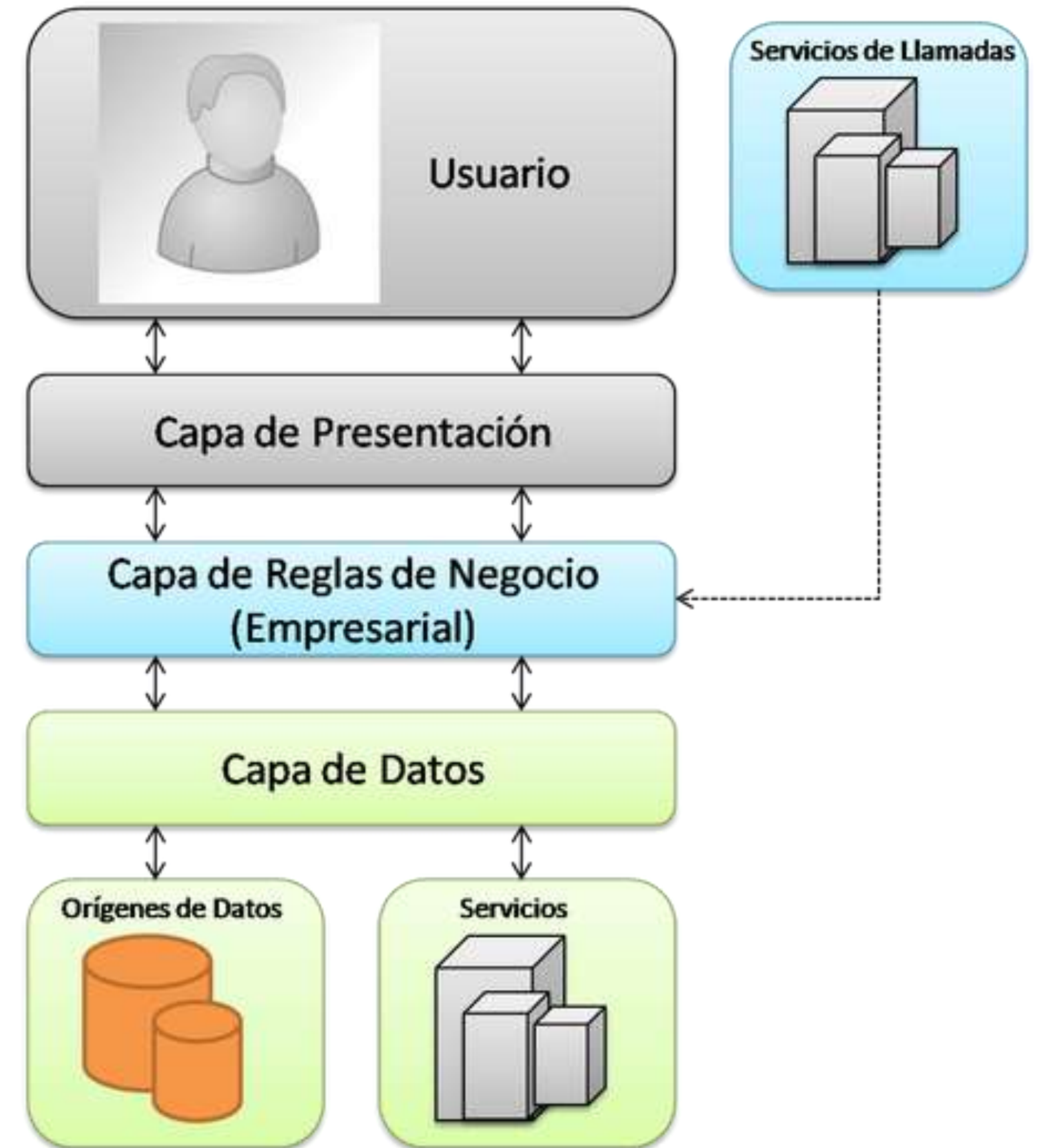


# Patrón de Capas

Este patrón se puede utilizar para estructurar programas que se pueden **descomponer en grupos de subtareas**, cada una de las cuales se encuentra en un **nivel** particular de abstracción. Cada capa proporciona servicios a la siguiente capa superior.

Las 4 capas más comúnmente encontradas de un sistema de información general son las siguientes.

- **Capa de presentación (capa UI )**
- **Capa de aplicación (capa de servicio )**
- **Capa de lógica de negocios (capa de dominio )**
- **Capa de acceso a datos (capa de persistencia )**







- Un Estilo Arquitectónico describe sus **componentes** y **responsabilidades**.
- Los **estilos impactan en las propiedades** del sistema.
- **Un Patrón arquitectónico se repite** y tiene **características** que pueden **reutilizarse**.
- Existen una **serie de Patrones**, como el **MVC**, **SOA** o el de **Capas**.

A black and white photograph of a man with a beard, wearing a white lab coat, working on a laptop in a server room. He is looking at the screen. The background shows server racks with cables. A large blue 'A' graphic is overlaid on the right side of the image. A semi-transparent dark rectangle is positioned over the man's chest, containing the word 'GRACIAS' in blue capital letters.

**GRACIAS**

**DuocUC<sup>®</sup>**