

ESTRUCTURA DE UNA APLICACIÓN EMPRESARIAL Y PATRONES DE DISEÑO





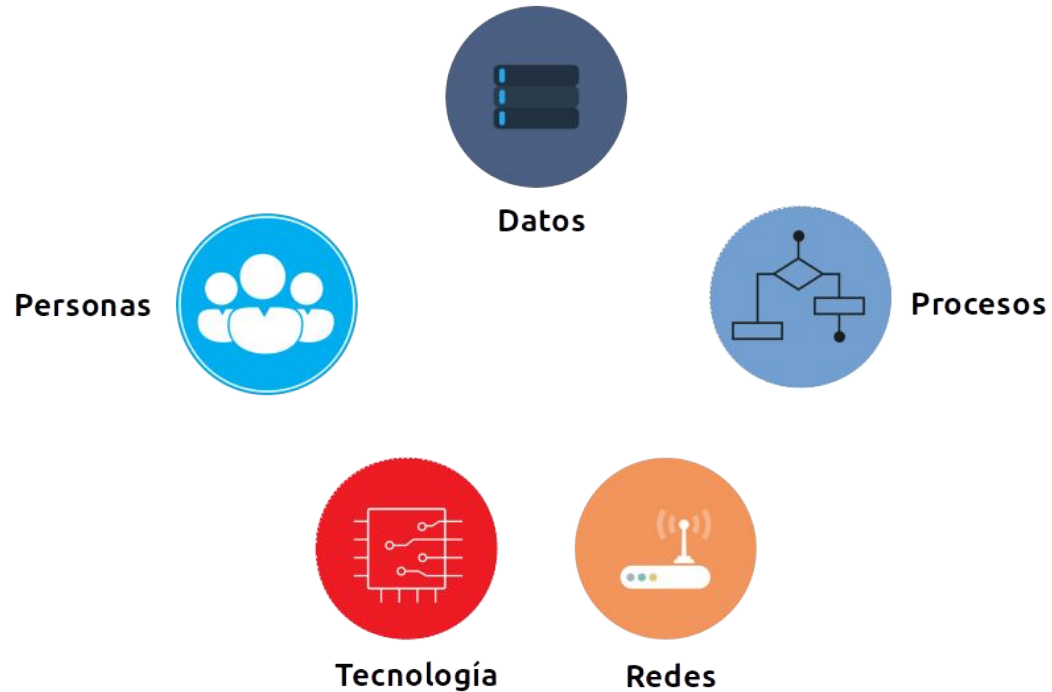
1.

APLICACIÓN EMPRESARIAL

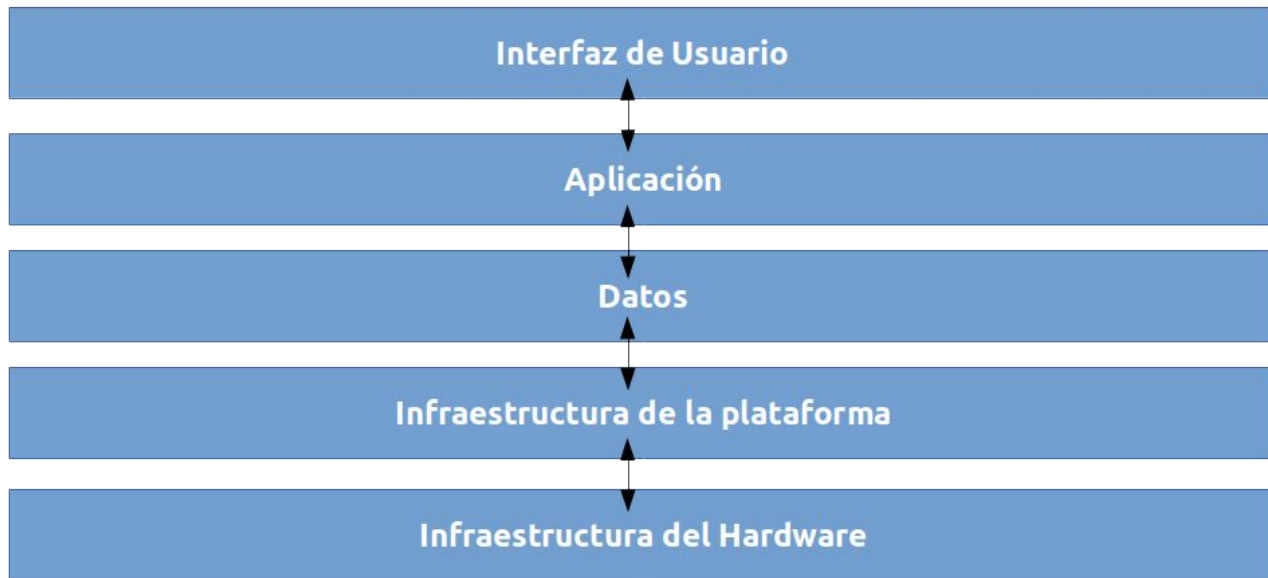
¿QUÉ ES UNA APLICACIÓN EMPRESARIAL?

- ▶ *Gran* aplicación comercial
- ▶ Compleja, escalable, distribuida, crítica
- ▶ Despliegue en redes corporativas o internet
- ▶ Centrada en los datos.
- ▶ Intuitiva, de uso fácil.
- ▶ Requisitos de seguridad y mantenibilidad.

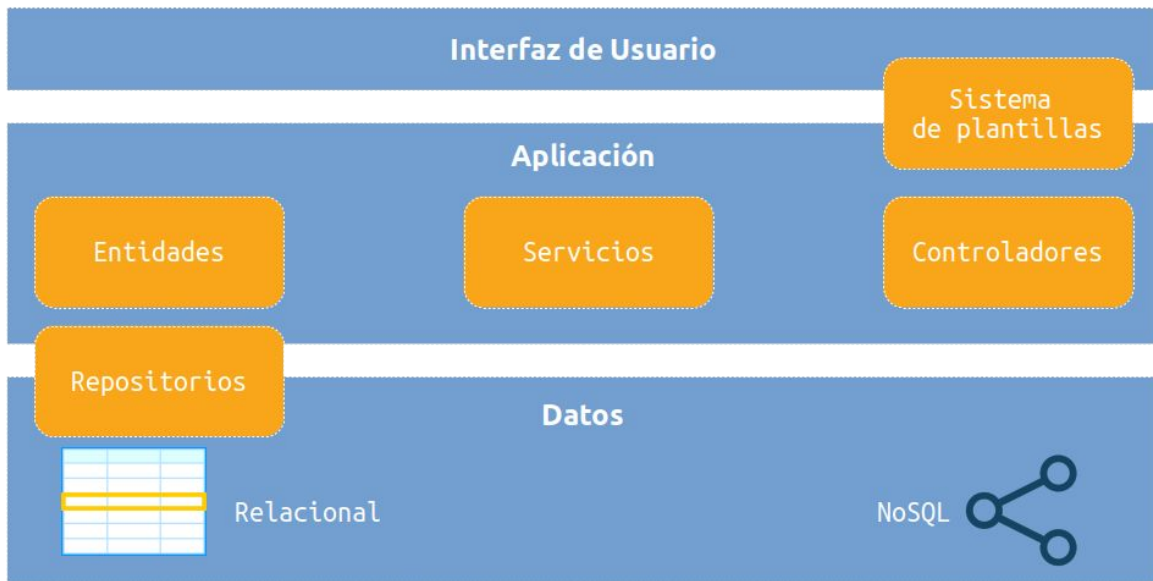
ESTRUCTURA DE UNA APLICACIÓN EMPRESARIAL



ESTRUCTURA DE UNA APLICACIÓN EMPRESARIAL



SPRING EN UNA APLICACIÓN EMPRESARIAL





2.

PATRONES DE DISEÑO

PATRÓN DE DISEÑO SOFTWARE

- ▶ Tengo un problema al diseñar mi aplicación.
- ▶ ¿Seguro que no le ha pasado a nadie más?
- ▶ En un alto porcentaje de situaciones, alguien ya tuvo ese problema y lo resolvió.

Un patrón de diseño es una solución a un problema, que ya ha sido utilizada, y cuya efectividad ha sido probada; además, es reutilizable en problemas con circunstancias similares.

TIPOS DE PATRONES DE DISEÑO

Propuestos por *Gang of Four*.

		PROPÓSITO		
		Creacional	Estructural	Comportamiento
ÁMBITO	Clase	Factory Method	Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Bridge Composite Decorator Facade Flyweight Proxy	Chain of responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

TIPOS DE PATRONES DE DISEÑO

Propuestos por otros autores:

- ▶ Inversión de control
- ▶ Inyección de dependencias
- ▶ *View Helper*
- ▶ Modelo-Vista-Controlador
- ▶ *Data Access Object*
- ▶ *Front-Controller*
- ▶ ...

¿QUÉ PATRONES DE DISEÑO USAREMOS CON SPRING?

- ▶ **Inyección de dependencias**
- ▶ *Singleton*: restringe la creación de objetos de un tipo a uno solo
- ▶ *Prototype*: permite la creación de nuevos objetos duplicándolos.
- ▶ *Proxy*: proporciona un sustituto o representante de un objeto, para controlar el acceso a este

Ámbitos de
un bean

Spring AOP
(Programación
orientada a
aspectos)

¿QUÉ PATRONES DE DISEÑO USAREMOS CON SPRING?

- ▶ Modelo-Vista-Controlador: separa la lógica y los datos de la interfaz y el control de peticiones.
- ▶ Front Controller: un solo controlador maneja todas las peticiones.
- ▶ *Factory*: centralización de la construcción de objetos.

*Spring Web
MVC*

*Spring IoC
Container*