

DATUM



Spring Framework

Trata de un marco de trabajo, en este caso empresarial y de código abierto.

Ventajas de usar Spring:

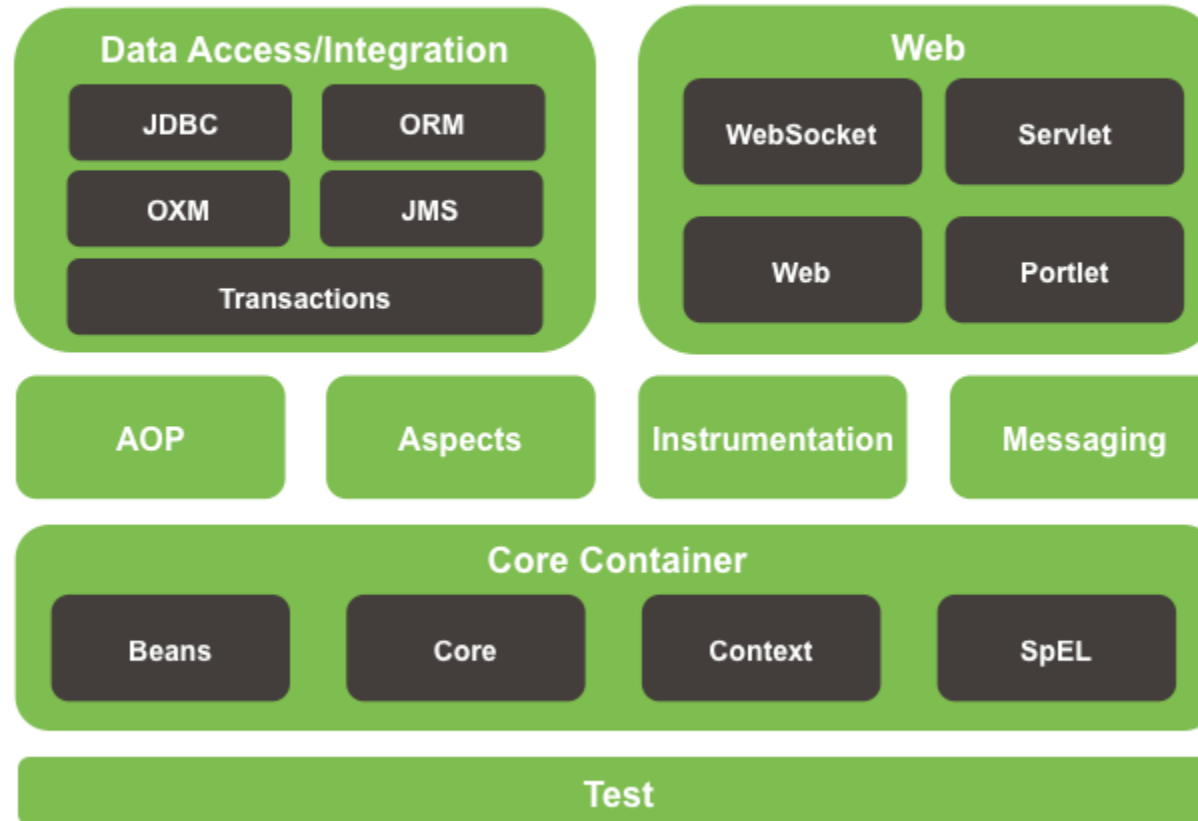
- Desarrollo de MicroServicios.
- Reactivo.
- Cloud



Spring Framework - Estructura



Spring Framework Runtime



Spring Boot

Es una herramienta de Spring Framework que busca facilitarnos la creación de proyectos con Spring Framework eliminando la configuración de archivos XML.

Características de Spring Tool:

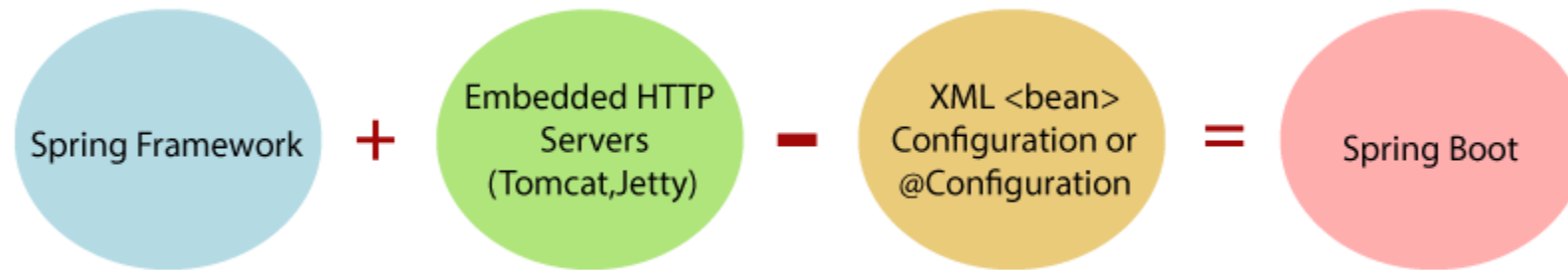
- Permite crear aplicaciones independientes
- Posee los embebidos necesarios para su implementación
- Es auto configurable
- Cuenta con funciones de producción
- No necesita una configuración XML



Spring Framework - Spring Boot

¿Existen diferencias marcadas entre Spring Boot y Spring Framework?

La respuesta sencilla, es que sí, existen. No obstante, tienes que tener en cuenta que Spring Boot, es una herramienta que facilita el desarrollo de Spring Framework.

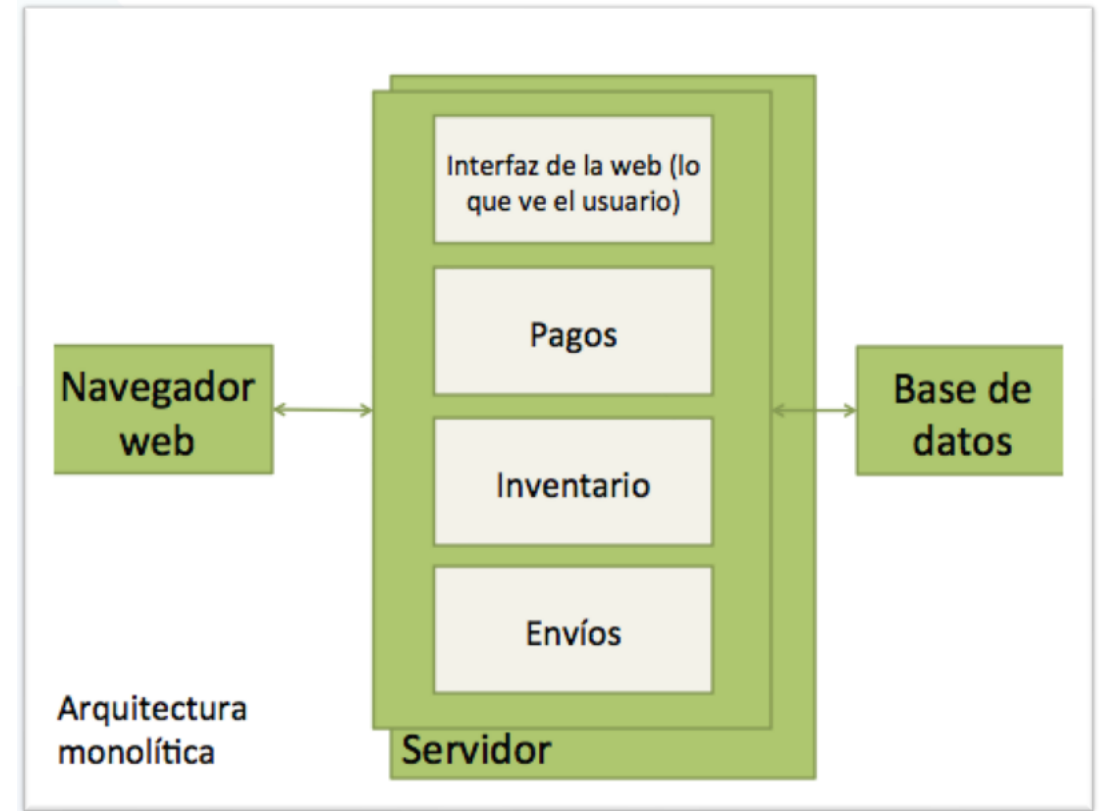


Instalación de Spring Tool Suit IDE



Servicios Monolíticos

Aplicación de software en la que todas sus capas (interfaz de usuario, lógica de negocio y acceso a datos) están combinadas en un mismo programa y sobre una misma plataforma.



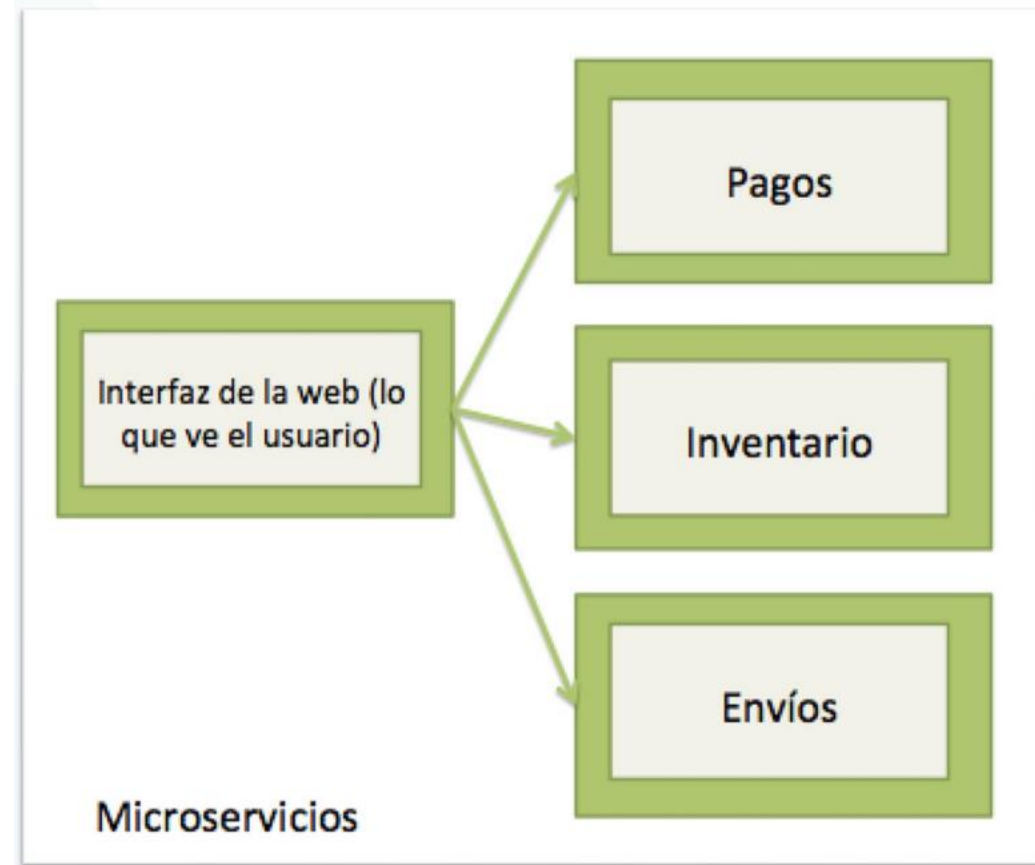
Problema de los Servicios Monolíticos

- Si se produce un fallo del sistema se caen todos los servicios
- Es difícil de escalar
- Imposibilidad de innovación tecnológica
- Despliegues o actualizaciones conflictivas
- Complejo de gestionar equipos de desarrollo
- Alto número de desarrolladores
- Nivel de conocimiento de todo el Sistema
- Difícil desarrollar funcionalidades en paralelo



MicroServicio

Pequeñas aplicaciones con una funcionalidad muy concreta y un alto nivel de especialización que trabajan en conjunto.



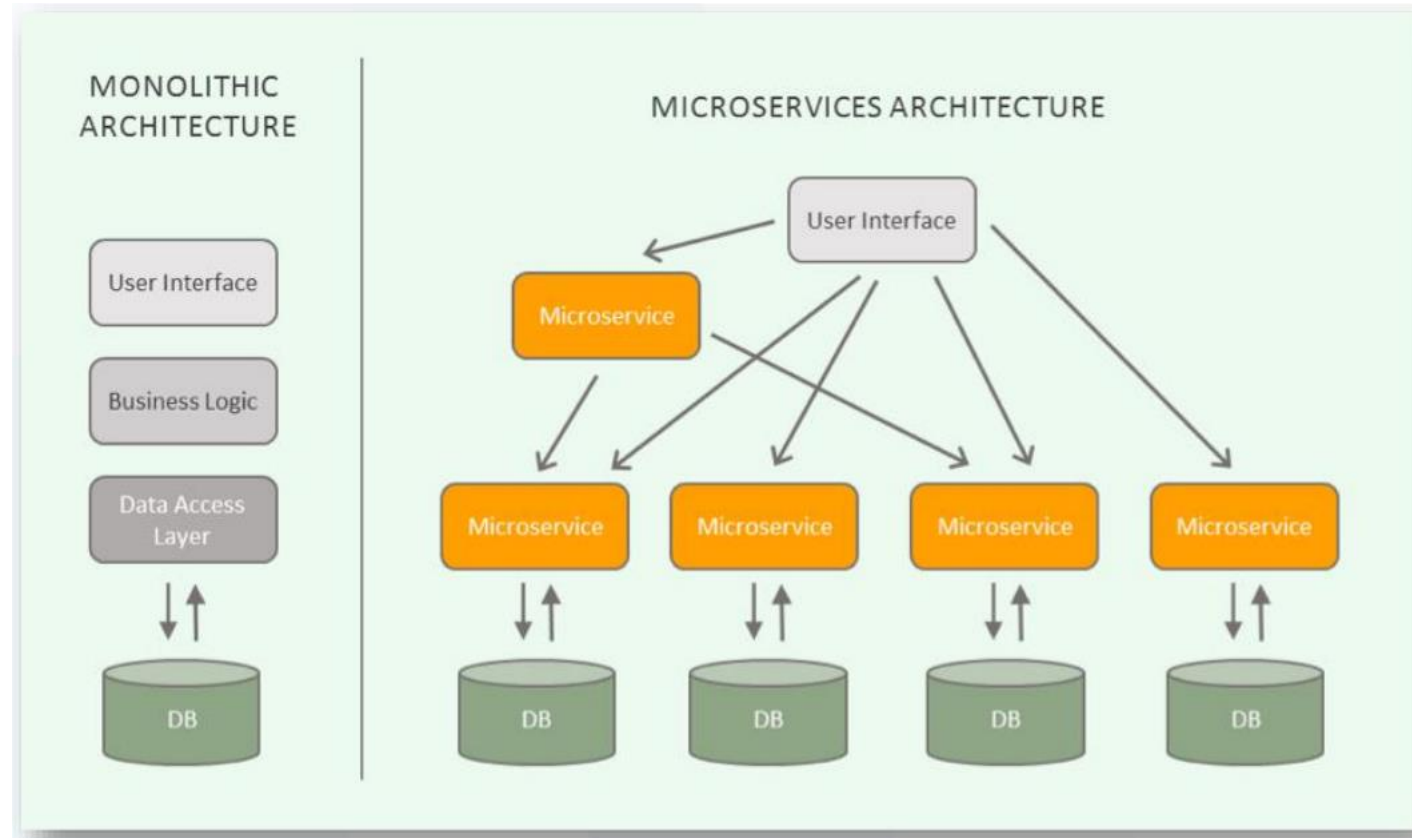
Ventaja de los MicroServicio

- Alta tolerancia a fallos. Si una parte del sistema se cae lo demás sigue funcionando (Circuit breaker)
- Permite escalabilidad del sistema
- Implementación sencilla
- Código más mantenible (menos interdependencias)
- Agilidad de cambios
- Gestión de equipos más simple
- Despliegues y actualizaciones con riesgos controlados
- Permite uso de distintas tecnologías



Arquitectura

MicroServicio vs Servicios Monolíticos



Inyección de Dependencias

La inyección de dependencias es un patrón de Software, que implementa la inversión de control para resolver dependencias; a partir de la cual los objetos definen sus dependencias con otros objetos.

Tipos de Inyecciones:

- Inyección de Dependencias basada en Constructor
- Inyección de Dependencias basada en Setter
- Inyección de Dependencias basada en campos
- Autowiring dependencias





Gracias por la Atención

