



Universidad de
Oviedo



SPRING

Enol García González
Universidad de Oviedo
31 de octubre de 2023

Spring es un framework para desarrollo web al igual que JSF. Spring se diferencia de JSF en dos características fundamentales:

- El enfoque de la presentación.
 - JSF es un framework basado en componentes.
 - Su filosofía es crear componentes (a poder ser reutilizables) y definir su comportamiento.
 - En JSF se abstraen lo máximo posible las peticiones entre el servidor y el cliente y se centra en la presentación.
 - Spring se basa en crear controladores para decidir como responder a peticiones http y delega la presentación en un motor de plantillas como Thymeleaf o Mustache.

- ¿Qué elementos se implementan en el framework?
 - JSF se centra en la capa de presentación. Sólo incluye componentes para la gestión de las vistas.
 - Spring es un framework mucho más amplio. Esta compuesto por multitud de módulos que se pueden componer para abarcar diferentes aspectos del proyecto.
 - Spring tiene desarrollados componentes para tratar con casi todo: gestión de seguridad, acceso a base de datos (con diferentes tecnologías), gestión de eventos, etc.

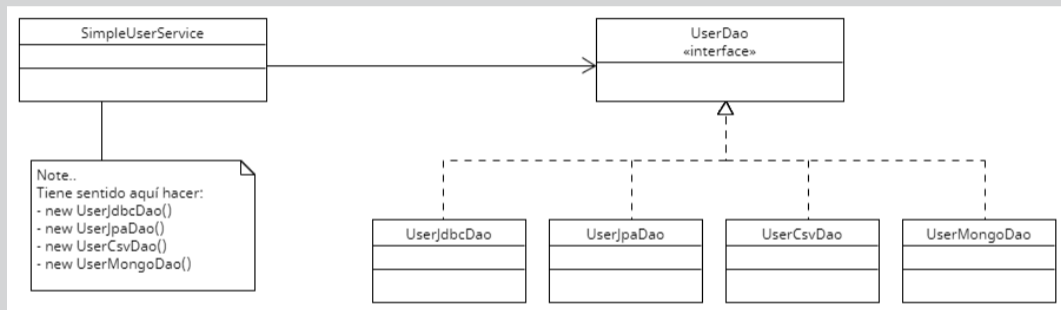
INYECCIÓN DE DEPENDENCIAS

- Una de los motivos por los que Spring ganó muchísima popularidad fue por su capacidad y facilidad para la inyección de dependencias.
- Esto está relacionado con el buen diseño de las aplicaciones. En la clase de patrones de diseño os hablé por encima de los principios SOLID.

PRINCIPIOS DE DISEÑO

- S** Single Responsibility Principle. Cada clase debe tener un sólo motivo para cambiar.
- O** Open/Closed Principle. El diseño general de las clases debe permitir que estén abiertas a introducir nuevas funcionalidades de forma fácil pero cerradas a modificar el comportamiento ya existente.
- L** Liskov Substitution Principle. Si S es un subtipo de T, entonces los objetos de tipo T en un programa pueden ser sustituidos por objetos de tipo S sin afectar a su funcionamiento y sin necesidad de conocer las diferencias entre S y T.
- I** Interface Segregation Principle. Muchas interfaces cliente específicas son mejores que una interfaz de propósito general.
- D** Dependency Inversion Principle. Los módulos de alto nivel no deberían depender de los módulos de bajo nivel.

PROBLEMA DE ACOPLAMIENTO



PROBLEMA DE ACOPLAMIENTO

- Otra persona tiene que crear el objeto concreto por nosotros.
- Hay dos estrategias:
 - Llamo al objeto responsable para que me lo instancia (Patrón Factoría)
 - “Confío” en que en tiempo de ejecución alguien lo creará y se encargará de dármelo (Inyección)

SPRING BOOT

- La segunda cosa por la que Spring ganó muchos seguidores fue por Spring Boot.
- Spring salió en el año 2003 y con el tiempo se dieron cuenta de que había muchas configuraciones que siempre eran igual.
- En 2014 sacan una versión de Spring con una herramienta que inicializa los proyectos dejándolos casi configurados por completo. Esta versión es Spring Boot.
- Con Spring Boot se pretendía reducir el tiempo de inicio del proyecto drásticamente para dar a los desarrolladores todo lo necesario para comenzar cuanto antes a codificar.
- Además de la configuración por defecto del proyecto, Spring Boot simplifica anotaciones e incluye un servidor integrado en el proyecto, haciendo que no sea necesario tener una instalación como nuestro WildFly de prácticas.

Ejemplo en el campus.