

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- El marco de trabajo Spring para el desarrollo de aplicaciones Java.
- El concepto de inversión de control.

SPRING

Spring es un *framework* (marco de trabajo) de código abierto para Java que utiliza Maven para gestionar el ciclo de vida de la aplicación. Es una potente herramienta que, considerando sus funcionalidades nucleares, puede ser utilizada para cualquier aplicación Java, sin embargo, también proporciona artefactos útiles para facilitar la construir aplicaciones web basados en Java EE.

Las capacidades de Spring están altamente modularizadas, por lo que para distintas aplicaciones solo necesitas agregar las dependencias necesarias al POM, y no todo el *framework*, para concretar su función objetivo. Una de las capacidades fuertes de Spring es la de facilitar la inyección de dependencias, creando *Beans* basados en *POJOs*.

POJO quiere "Plain Old Java Object" que puede traducirse como "un viejo y simple objeto Java". Hace referencia a una clase de Java minimalista en la que solo se han agregados sus campos, constructores, accesadores y mutadores. Por ejemplo:

```
1 @AllArgsConstructor
2 @NoArgsConstructor
3 @Data
4 public class Alumno{
5     private String rut;
6     private String nombre;
7     private List<Integer> notas;
8     private Float promedio;
9 }
```

A partir de estos POJOs pueden utilizarse anotaciones de Spring (que veremos en los ejercicios de esta sección) para crear un *Bean* o *Component* que luego puede ser inyectado a una clase. Un *Bean* es, básicamente, una clase destinada a almacenar datos para el funcionamiento de nuestra aplicación.

Entonces, con Spring, se facilita la creación de *Beans* (dependencias) por medio de anotaciones simples y la inyección de ellas a otras clases, también por medio de anotaciones, ahorrándonos código.

Spring tiene, además, otras funcionalidades que veremos en las secciones futuras de este módulo.

INVERSIÓN DE CONTROL (IOC)

Spring utiliza el principio de inversión de control para mantener altamente desacopladas las clases de la aplicación. El objetivo principal de la inversión de control es la de remover las dependencias entre los objetos de una aplicación para que el proyecto sea mantenible y desacoplado.

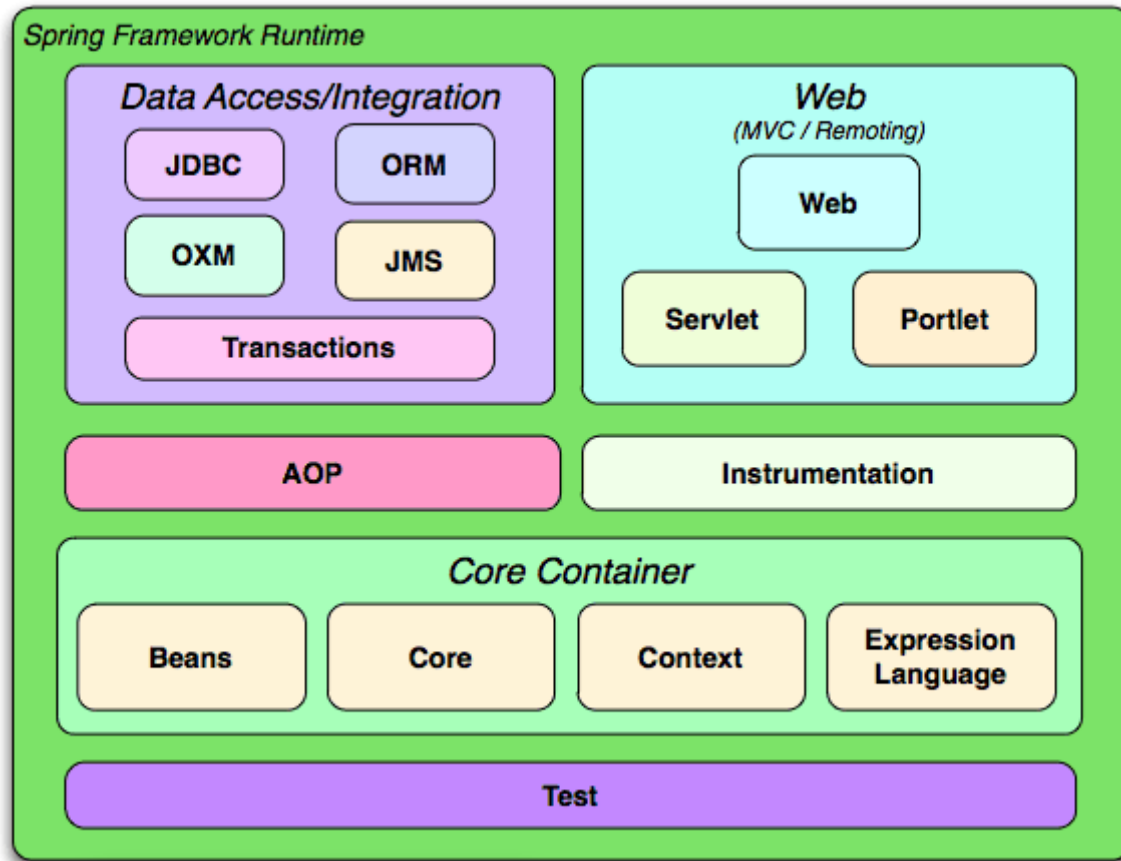
Conceptualmente, la inversión de control o IoC se refiere a la toma de control por parte del *framework* en cuanto a la definición del flujo de actuación, llamando a las distintas clases y sus métodos en el orden necesario, sin que nosotros implementemos un orden específico. Por tanto, nosotros hacemos definición de clases que serán contenedoras (*Beans*) y otras que ejecutan métodos, junto con algunas configuraciones, para que el marco de trabajo ejecute ese código, dependiendo de cuál es su objetivo, sin que sea imperativo que nosotros implementemos el código del flujo de la aplicación en el método *main*.

Para implementar este principio, Spring utiliza su *Spring IoC Container*. Al momento de inicializar la aplicación Spring, este recorrerá las clases con sus distintas anotaciones y configuraciones para registrar los distintos beans declarados y almacenarlos en el *IoC Container*. Veremos cómo lograr esto en los próximos ejercicios de esta sección.

¿POR QUÉ UTILIZAR SPRING?

Spring cuenta con una gran comunidad de desarrolladores que produce gran cantidad de documentación, reduciendo nuestra curva de aprendizaje. Aparte de eso, es un framework basado en el patrón de diseño Modelo Vista Controlador y utiliza numerosos módulos que nos permiten la creación de una aplicación web completa (back-end y front-end) integrando otros frameworks para el uso de bases de datos con diversos motores, el uso de plantillas para generar vistas, seguridad, etc. Todos estos aspectos los veremos en profundidad más adelante en este módulo.

La imagen a continuación muestra un esquema con los módulos que integran el marco de trabajo de Spring ([más información](#)).



En los próximos ejercicios, de esta sección, veremos cómo crear un proyecto Spring, como es su estructura y como utilizar anotaciones para crear *Beans* e inyectarlos.