



# Spring vs SpringBoot

## ¿Qué es Spring?

Es un framework para el desarrollo de aplicaciones basadas en Java. Proporciona un conjunto de herramientas y funcionalidades para mejorar la productividad.

Algunos de los módulos más comunes de Spring son:

- **Spring Core:** Manejo de la inversión de control (IoC) y la inyección de dependencias, permitiendo una arquitectura más modular y flexible.
- **Spring MVC:** Proporciona un modelo MVC (Modelo - Vista - Controlador) para el desarrollo de aplicaciones web.
- **Spring Test:** Herramientas para el manejo de pruebas unitarias y de integración.
- **Spring ORM:** Integra tecnologías de mapeo Objeto-relacional como Hibernate, JPA, etc.
- **Spring JDBC:** Proporciona herramientas para el manejo y acceso de bases de datos utilizando JDBC de una manera más sencilla.
- **Spring Security:** Ofrece características de seguridad para proteger las aplicaciones web y los servicios REST.

## ¿Qué es Spring Boot?

Spring Boot es una extensión de Spring que simplifica y agiliza el proceso de configuración y desarrollo de aplicaciones basadas en Spring. Algunas características que hacen que Spring Boot sea más sencillo incluyen:

- **Configuración automática:** Cuando se crea un proyecto en Spring Boot, se realiza una configuración automática basada en las dependencias y convenciones, lo que reduce la necesidad de configuraciones manuales.
  - **Servidor embebido:** Spring Boot incorpora un servidor web embebido como Tomcat o Jetty, eliminando la necesidad de desplegar la aplicación en un servidor externo.
  - **Starter dependencies:** Permite agregar fácilmente funcionalidades como base de datos, seguridad mediante la inclusión de dependencias específicas en el archivo de configuración del proyecto.
- 

## Spring MVC

Spring MVC es una herramienta que está dentro del ecosistema de Spring, el cual es utilizado para el desarrollo de aplicaciones web basadas en el patrón modelo vista controlador. Facilita la construcción de aplicaciones web, ya que proporciona un conjunto de clases y mecanismos para implementar componentes del patrón MVC de una manera modular y flexible.

- **DispatcherServlet:** Controlador frontal de Spring MVC que recibe todas las solicitudes HTTP y las envía al controlador adecuado.
  - **Controladores:** Componentes de Spring MVC que gestionan las solicitudes de usuarios.
  - **Beans del modelo:** Representan los datos que se utilizarán en la vista.
  - **Vistas:** Representaciones visuales de los datos que se devuelven al usuario. En Spring MVC, las vistas pueden ser JSP.
- 

Academia Java - David Patiño