

Spring vs Spring MVC vs Spring Boot

	Spring	Spring Boot	Spring MVC
¿Qué es?	Marco de trabajo de aplicación web de código abierto basado en Java.	Extensión o módulo basado en el marco Spring.	Marco de trabajo de desarrollo de aplicaciones web basado en Java que sigue el patrón de diseño Modelo-Vista-Controlador (MVC).
¿Para qué sirve?	Proporciona un entorno flexible y completamente configurable mediante herramientas y bibliotecas de código precompilado para crear aplicaciones empresariales en Java y de acoplamiento flexible.	Proporciona la capacidad de crear aplicaciones con Java independientes con un enfoque simplificado y modular que se pueden ejecutar inmediatamente sin necesidad de anotaciones, configuración XML. Además, facilita el uso de marcos basados en Java para crear microservicios y aplicaciones web.	Permite crear aplicaciones sobre modelo MVC para el desarrollo de aplicaciones web. Facilita la creación de controladores, vistas y manejo de solicitudes HTTP de forma organizada y escalable.
¿Características?	<ul style="list-style-type: none"> ■ Inversión de Control (IoC) el cual permite que los objetos no sean responsables de crear o buscar sus dependencias. Las cuales son proporcionadas por un contenedor de IoC, lo que promueve un bajo acoplamiento. ■ Programación Orientada a Aspectos (AOP), la cual permite modularizar aspectos transversales de la aplicación, como la seguridad, la transaccionalidad y el registro de errores, de manera más eficiente y desacoplada. ■ Transacciones Declarativas que permiten definir transacciones mediante anotaciones o XML. ■ Acceso a Datos a través de Spring Data, el cual proporciona un modelo de programación coherente y abstracciones para trabajar con diversas tecnologías de persistencia de datos, como JDBC, JPA, MongoDB, Redis, entre otros. ■ Seguridad a través de Spring Security el cual es un módulo que proporciona características de seguridad para aplicaciones empresariales, como autenticación, 	<ul style="list-style-type: none"> ■ Utiliza anotaciones, convenciones de nombres y dependencias preconfiguradas para configurar automáticamente muchas partes de la aplicación. ■ Proporciona una serie de "starter POMs" (Project Object Models) que simplifican la configuración de dependencias ■ Incluye servidores web embebidos, como Tomcat, Jetty o Undertow eliminando la necesidad de configurar un servidor externo. ■ Spring Boot Actuator proporciona una serie de endpoints HTTP y métricas que permiten monitorear y administrar la aplicación en tiempo de ejecución. ■ Ofrece una interfaz de línea de comandos (CLI) que facilita la creación rápida de prototipos de aplicaciones Spring. 	<ul style="list-style-type: none"> ■ Inversión de control (IoC) y contenedor de inversión de control (IoC container): para gestionar los objetos y sus dependencias. ■ Integración con otros módulos de Spring, como Spring Core, Spring Security, Spring Data, etc. ■ Soporte para múltiples vistas, como JSP (JavaServer Pages), Thymeleaf, FreeMarker, entre otras. ■ Soporte de anotaciones para simplificar la configuración y el desarrollo de controladores y otros componentes. ■ Proporcionar un mecanismo integrado para la validación de datos de entrada del usuario utilizando anotaciones y clases de validación.

	<p>autorización, protección contra ataques CSRF, protección contra XSS, entre otras.</p> <ul style="list-style-type: none"> ■ Pruebas Unitarias y de Integración mediante el uso de IoC y AOP, lo que permite crear tests más fáciles de mantener y ejecutar. 		
¿Tiene servidores integrados?	No. En Spring, tendrás que configurar los servidores explícitamente.	Sí, Spring Boot incluye servidores HTTP integrados como Tomcat y Jetty.	Sí, Spring Boot incluye servidores integrados como Tomcat embebido y compatible con otros servidores integrados como Jetty o Undertow.
¿Necesito saber cómo trabajar con XML?	En Spring, se requiere conocimiento de la configuración XML.	Spring Boot no requiere configuración XML.	Desde Spring 3.0, se introdujo la opción de realizar configuraciones utilizando anotaciones, lo que se conoce como configuración basada en anotaciones o JavaConfig. Esto permite definir los componentes de la aplicación utilizando anotaciones como @Controller, @Service, @Repository, entre otras, eliminando la necesidad de configuración XML en muchos casos.
¿Hay herramientas de la CLI para aplicaciones de desarrollo y pruebas?	El marco Spring por sí solo no proporciona herramientas de la CLI para desarrollar o probar aplicaciones.	Como módulo de Spring, Spring Boot tiene una herramienta de la CLI para desarrollar y probar aplicaciones basadas en Spring.	Sí, Spring Framework y Spring Boot proporcionan varias herramientas de la CLI (Command Line Interface) que son útiles para el desarrollo y pruebas de aplicaciones basadas en Spring MVC.
¿Funciona desde un enfoque con sesgo o sin sesgo?	Sin sesgo*	Con sesgo*	Se puede implementar tanto con un enfoque con sesgo (biased) como sin sesgo (unbiased), dependiendo de cómo se configure y utilice en una aplicación específica.

REFERENCIAS:

Spring

<https://spring.io/projects/spring-framework>

https://es.wikipedia.org/wiki/Spring_Framework

<https://spring.io/>

Spring MVC

<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>

<https://www.arquitecturajava.com/spring-mvc-configuracion/>

Spring Boot

<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-java-spring-boot#:~:text=Spring%20Boot%20es%20un%20marco,desarrollan%20aplicaciones%20web%20y%20microservicios.>

<https://spring.io/projects/spring-boot/#overview>

<https://www.ibm.com/es-es/topics/java-spring-boot#:~:text=Spring%20Boot%20permite%20a%20los,durante%20el%20proceso%20de%20inicializaci%C3%B3n.>