# El Framework Spring MVC

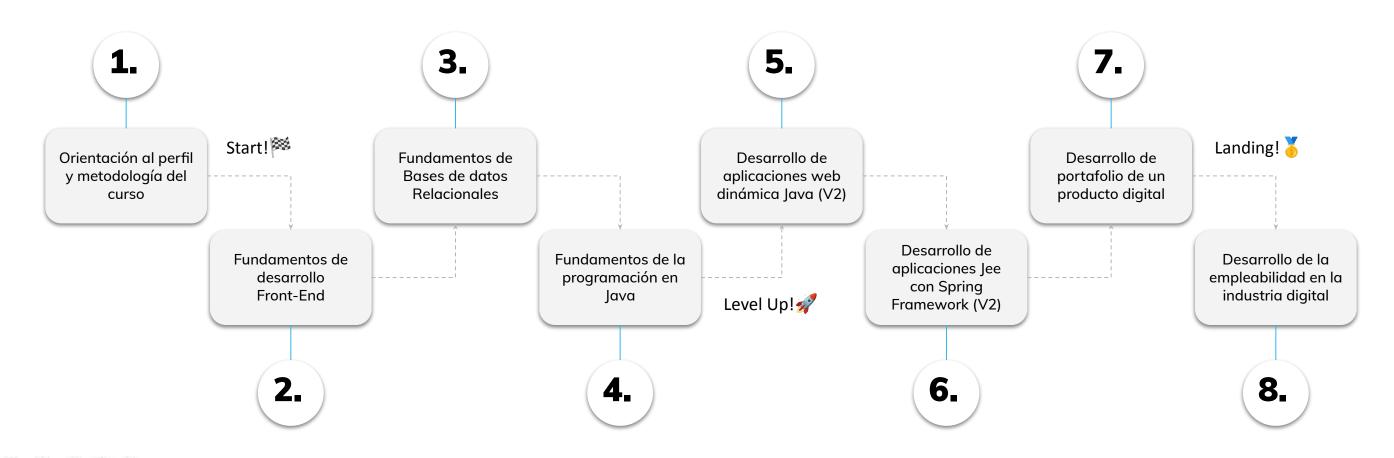
Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0





#### HOJA DE RUTA

¿Cuáles skill conforman el programa?









#### REPASO CLASE ANTERIOR



En la clase anterior trabajamos 📚:



- Fase de empaquetamiento
- Instalación y ejecución



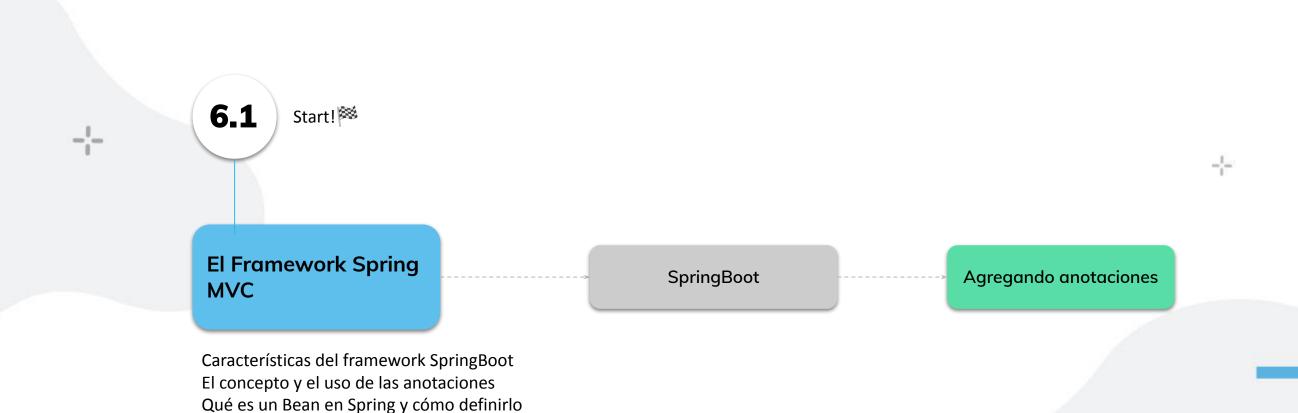




#### **LEARNING PATHWAY**

Inyección de dependencias y la anotación Autowired

¿Sobre qué temas trabajaremos?







#### OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



- Conocer las características de Spring Boot
- Aprender conceptos de anotaciones y Bean
- Comprender el concepto de inyección de dependencias









Spring Boot es un popular y poderoso framework de desarrollo de aplicaciones Java que simplifica considerablemente la creación de aplicaciones basadas en Spring Framework. Ofrece una amplia gama de características y beneficios que lo convierten en una elección atractiva para desarrolladores y equipos de desarrollo de todo el mundo.







- 1. **Simplifica la configuración**: Spring Boot ofrece una configuración por defecto qu es bastante sensata, lo que significa que puedes comenzar a desarrollar sin tener que preocuparte por detalles de configuración complicados. Además, proporciona anotaciones y propiedades que permiten personalizar la configuración según tus necesidades.
- 2. **Incorpora un servidor de aplicaciones embebido**: Spring Boot incluye servidores de aplicaciones embebidos, como Tomcat, Jetty o Undertow. Esto significa que no es necesario desplegar tu aplicación en un servidor externo; puedes ejecutarla directamente como una aplicación independiente.







- **3. Gestión de dependencias**: Spring Boot incorpora una gestión de dependencias sólida a través de Apache Maven o Gradle. Esta característica permite especificar las dependencias necesarias en el archivo pom.xml o build.gradle, y Spring Boot se encarga de descargar automáticamente las bibliotecas correctas y gestionar las versiones de dependencias.
- **4. Creación de aplicaciones autónomas**: Lo que significa que una aplicación puede funcionar de manera independiente sin requerir configuraciones externas complicadas. Esto facilita la implementación y la ejecución de aplicaciones en diferentes entornos, como servidores locales, contenedores o plataformas en la nube.







- **5. Autoconfiguración inteligente**: Significa que ajusta automáticamente la configuración de la aplicación en función de las bibliotecas y dependencias que detecta en el proyecto. Esto reduce la necesidad de configuraciones manuales y garantiza que la aplicación esté configurada de manera óptima.
- **6. Desarrollo basado en anotaciones**: Las anotaciones como @Controller, @Service, @Repository, @RequestMapping, entre otras, simplifican el desarrollo de aplicaciones web y permiten una estructura clara y modular.







- **7. Integración de bases de datos:** Proporcionando herramientas como Spring Data JPA y Spring Data MongoDB, que facilitan la implementación de la capa de persistencia. Además, ofrece soporte para bases de datos en memoria para pruebas y desarrollo.
- **8. Soporte para pruebas unitarias:** Se integra perfectamente con frameworks de pruebas como JUnit y TestNG. Esto facilita la escritura y ejecución de pruebas unitarias y de integración, lo que garantiza la calidad del código y la funcionalidad de la aplicación.







- **9. Actuadores y monitoreo:** Spring Boot proporciona una serie de "actuadores" que permiten supervisar y administrar la aplicación en tiempo real. Puedes obtener información sobre métricas, salud de la aplicación, estado de las conexiones de base de datos y más.
- **10. Compatibilidad con Spring Ecosystem:** Lo que te permite aprovechar las ventajas de las características y bibliotecas de Spring existentes. Esto incluye Spring Data, Spring Security, Spring Cloud y muchos otros proyectos de Spring.







- **11. Desarrollo ágil y rápido prototipado:** La combinación de autoconfiguración, servidores de aplicaciones embebidos y una estructura de proyectos clara permite a los desarrolladores centrarse en la lógica de negocio en lugar de preocuparse por la configuración y la infraestructura.
- **12. Soporte para aplicaciones basadas en microservicios:** Spring Boot es una elección popular para desarrollar aplicaciones basadas en microservicios debido a su capacidad para crear aplicaciones autónomas y su integración con Spring Cloud, que proporciona herramientas para la gestión de microservicios.





#### Características principales:

- **13. Comunidad activa y amplia documentación:** Esto facilita la obtención de ayuda, la solución de problemas y el aprendizaje de nuevas características y mejores prácticas.
- **14. Implementación en la nube:** Se integra con facilidad en plataformas en la nube como AWS, Google Cloud, Microsoft Azure y Heroku. Esto permite la implementación de aplicaciones en entornos en la nube de manera sencilla y eficiente.
- **15. Seguridad:** Spring Boot ofrece funcionalidades de seguridad, como autenticación y autorización, que se pueden configurar de manera sencilla para proteger las aplicaciones web. También se integra con Spring Security, un proyecto de Spring ampliamente utilizado para la gestión de la seguridad.



---



### Anotaciones



### Anotaciones

Las anotaciones proporcionan una forma de agregar metadatos enriquecidos a los elementos del código. Esto permite a los desarrolladores adjuntar información adicional a clases, métodos o campos, como autor, fecha de creación, descripción, restricciones de seguridad y más. Estos metadatos pueden ser utilizados por herramientas de desarrollo, generación de documentación y otras tareas que requieren información contextual sobre el código.







### Anotaciones

#### Personalización del comportamiento:

Las anotaciones se utilizan para personalizar el comportamiento de clases, métodos y campos. Esto se logra mediante la configuración de anotaciones específicas que indican cómo una clase o método debe comportarse en tiempo de ejecución. Por ejemplo, las anotaciones como **@Override** o **@Deprecated** influyen en cómo el compilador y otras herramientas tratan el código.

#### Validación y restricciones:

Las anotaciones se emplean para aplicar restricciones y validaciones en los datos de entrada y los objetos. Por ejemplo, las anotaciones de validación como **@NotNull**, **@Min**, **@Max** y **@Pattern** permiten definir restricciones en los campos de una entidad, lo que garantiza que los datos cumplan con ciertos criterios antes de ser procesados.







#### Extensibilidad y creación de anotaciones personalizadas:

Java permite a los desarrolladores crear sus propias anotaciones personalizadas. Esto brinda una flexibilidad adicional para definir metadatos y comportamientos específicos de la aplicación. La capacidad de crear anotaciones personalizadas ha llevado a la creación de numerosas bibliotecas y marcos que hacen un uso extensivo de anotaciones adaptadas a casos de uso específicos.









#### Compatibilidad con anotaciones de Java SE y EE:

Java SE y EE proporcionan un conjunto de anotaciones predefinidas que son ampliamente utilizadas en el desarrollo de aplicaciones empresariales y aplicaciones de escritorio. Estas anotaciones incluyen **@Override**, **@Deprecated**, **@Entity**, **@Resource**, **@WebService** y muchas más. La compatibilidad con estas anotaciones está integrada en el lenguaje Java y en su infraestructura de desarrollo.









#### Compatibilidad con anotaciones de Java SE y EE:

Java SE y EE proporcionan un conjunto de anotaciones predefinidas que son ampliamente utilizadas en el desarrollo de aplicaciones empresariales y aplicaciones de escritorio. Estas anotaciones incluyen **@Override**, **@Deprecated**, **@Entity**, **@Resource**, **@WebService** y muchas más. La compatibilidad con estas anotaciones está integrada en el lenguaje Java y en su infraestructura de desarrollo.







#### LIVE CODING

Ejemplo en vivo

#### Overview de un proyecto Spring Boot:

En esta primera parada, veremos la estructura de un proyecto Spring Boot básico con las anotaciones iniciales que ayudan a su funcionamiento.

Tiempo: 10 minutos





### ¿Alguna consulta?



#### RESUMEN

¿Qué logramos en esta clase?



- Conocer el framework Spring Boot y sus características
- ✓ Aprender sobre anotaciones







#### **#WorkingTime**

Continuemos ejercitando



¡Antes de cerrar la clase! Te invitamos a: 👇 👇 👇

- 1. Repasar nuevamente la grabación de esta clase
- 2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Lectura Modulo 6, Lección 2: páginas 1 6
- 3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.







# Muchas Gracias!

Nos vemos en la próxima clase 🤎



*M* alkemy

>:

Momento:

### Time-out!

**⊘**5 min.



