

» 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 :

- ✓ Concepto de Spring Bean
- ✓ Inyección de dependencias

LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

6.2

Start! 🏁

El Framework Spring MVC

Spring Boot - Initializr

Primer proyecto Spring

Creando un proyecto web Spring Boot con módulo spring-web
Configurando el proyecto
Configuraciones iniciales de Spring Boot

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Conocer las características de Spring Initializr



Aprender a configurar un proyecto Spring



› Creando un proyecto Spring Boot



Creando un proyecto Spring Boot



Al igual que en cualquier biblioteca Java estándar, en Spring Boot se incluyen los correspondientes archivos JAR (Java Archive) o WAR (Web Application Archive) en el Classpath. Java recurre a esta ruta del sistema para buscar los archivos ejecutables. Puedes crear los archivos de almacenamiento de Spring Boot de dos maneras distintas:



- **Instala y utiliza Maven** para crear por tu cuenta todo el marco del proyecto, incluidas las dependencias necesarias.
- **Utiliza el servicio web Spring Initializr** para establecer la configuración de Spring Boot y, después, descárgala como plantilla de proyecto final.





Creando un proyecto Spring Boot



Una característica de **Spring Initializr** es que proporciona una interfaz web muy fácil de usar para crear los archivos JAR, lo que simplifica considerablemente el proceso. Como **Initializr también emplea Maven** para generar los archivos, el resultado es el mismo que si lo haces de forma manual.





Creando un proyecto Spring Boot



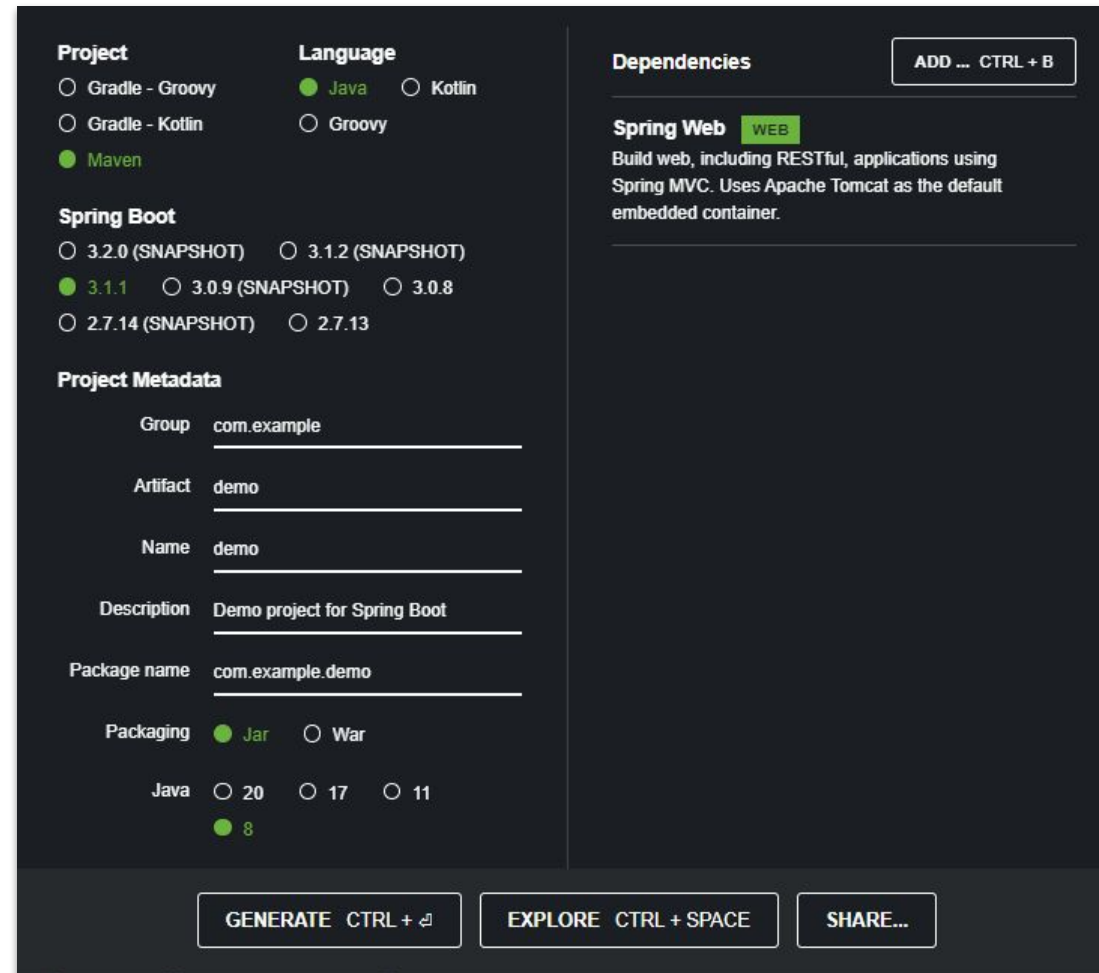
Para iniciar un proyecto Spring Boot:

- 1- Ingresar al sitio web Spring Initializr en start.spring.io .
- 2- Configurar el proyecto, incluyendo las dependencias (en este caso comenzaremos con la dependencia Spring Web).



› Configurando un proyecto Spring Boot

Configurando un proyecto Spring Boot



The screenshot shows the Spring Boot project configuration interface. It is divided into three main sections: Project, Language, and Dependencies.

Project

- ☐ Gradle - Groovy
- ☐ Gradle - Kotlin
- ☒ Maven

Language

- ☒ Java
- ☐ Kotlin
- ☐ Groovy

Spring Boot

- ☐ 3.2.0 (SNAPSHOT)
- ☐ 3.1.2 (SNAPSHOT)
- ☒ 3.1.1
- ☐ 3.0.9 (SNAPSHOT)
- ☐ 3.0.8
- ☐ 2.7.14 (SNAPSHOT)
- ☐ 2.7.13

Project Metadata

Group:

Artifact:

Name:

Description:

Package name:

Packaging: ☒ Jar ☐ War

Java: ☐ 20 ☐ 17 ☐ 11

☒ 8

Dependencies

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Buttons at the bottom: **GENERATE** (CTRL + G), **EXPLORE** (CTRL + SPACE), **SHARE...**



Configurando un proyecto Spring Boot



Configuraciones:

Project: Permite elegir la herramienta de construcción de la aplicación. En Java las dos herramientas más usadas son Maven y Gradle. Recomendamos Maven al ser la más expandida.



Language: Lenguaje de programación que se va a utilizar en la aplicación. Los tres tipos están soportados por la máquina virtual JVM. Java es la opción más extendida y tiene mejor soporte de los editores de programación.

Spring Boot: Versión del Spring Boot a usar. Siempre que se pueda se optará por la última estable, compuesta únicamente por números (sin palabras entre paréntesis).



Configurando un proyecto Spring Boot



Project Metadata:

- **Group:** Utilizado para clasificar el proyecto en los repositorios de binarios. Normalmente se suele usar una referencia similar a la de los packages de las clases. Por ejemplo, com.example para disponer todas las aplicaciones web en el mismo directorio.
- **Artifact:** Para indicar el nombre del proyecto y del binario resultante. La combinación de groupId y artifactId (más la versión) son identificadores únicos.
- **Name:** Asignar un nombre al proyecto.
- **Description:** Determina una descripción al proyecto a crear.
- **Packaging name:** Se refiere al nombre principal del paquete de nuestra aplicación.



Configurando un proyecto Spring Boot



Project Metadata:

- **Packaging:** Indica qué tipo de binario se debe construir. Si la aplicación se ejecutará por sí sola se seleccionará JAR, éste contiene todas las dependencias dentro de él y se podrá ejecutar con `java -jar binario-jar`. Si por el contrario, la aplicación se ejecutará en un servidor J2EE existente o en un Tomcat ya desplegado se deberá escoger WAR.
- **Java:** Se selecciona la versión de Java a usar. En este caso, se recomienda usar la versión **estable** de Java más antigua para garantizar la compatibilidad con otras librerías o proyectos que se quieran incluir, así será más probable encontrar documentación existente que siga siendo válida. Reduce el riesgo de toparse con funcionalidades no muy maduras.





Configurando un proyecto Spring Boot

Del lado derecho de la pantalla podremos encontrar el botón de buscador de dependencias **Add Dependencies**: Buscador de dependencias con los starters de Spring boot disponibles. Las dependencias más habituales son:

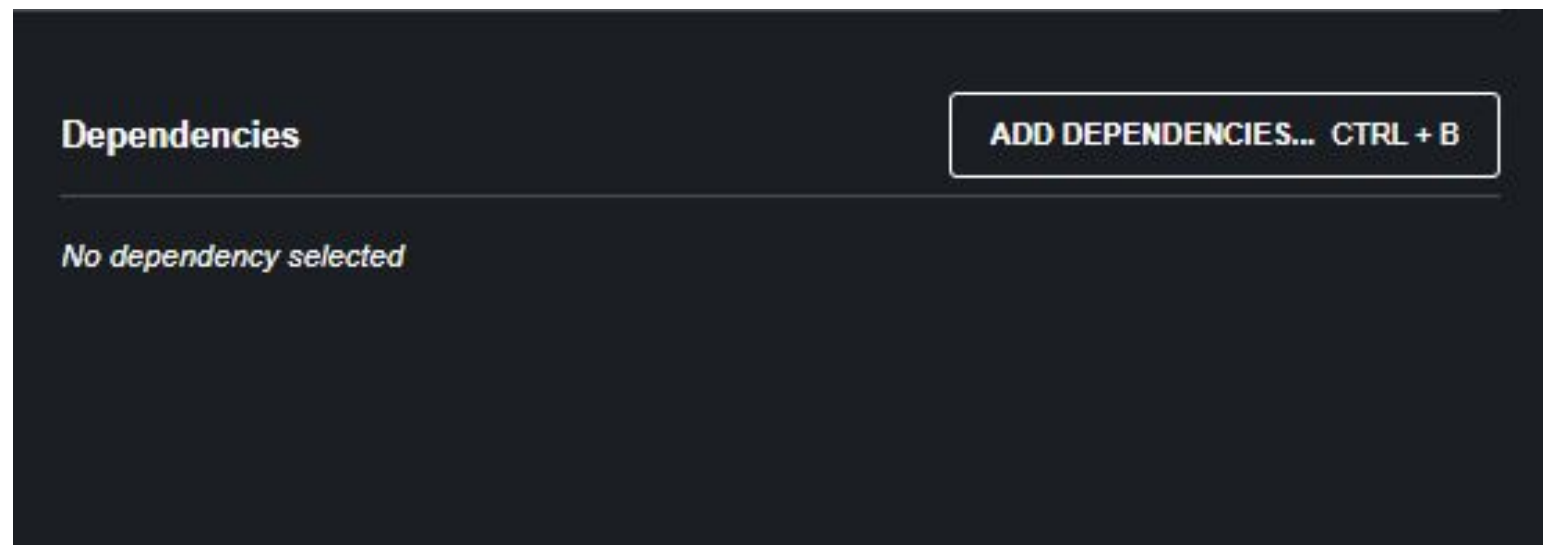
- **Spring Web**: Orientado al desarrollo de aplicación web o microservicios.
- **Thymeleaf**: Incorpora el motor de plantillas para HTML dinámico (sucesor de JSP - Java Server Page).
- **Spring Data JPA**: Capa estándar de acceso a base de datos SQL denominada Java Persistence Api.
- **Spring Security**: Incorporar controles de acceso en base a usuarios y roles sobre URLs de la aplicación. Asimismo, habilita el control de ejecución de métodos de servicio en base a roles según los estándares J2EE.

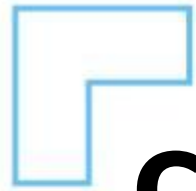


Configurando un proyecto Spring Boot



- **Lombok:** Facilita la programación como la creación de @Getters y @Setters automáticamente para las clases que forman parte del conjunto de mensajes.
- **Mysql/Postgresql:** Incluye el JAR que contiene el driver JDBC necesario para configurar la capa de JPA según la base de datos a usar.





Configurando un proyecto Spring Boot



Después de seleccionar los parámetros necesarios se requiere generar el proyecto, para lo cual se tiene que hacer click en el botón "Generate Ctrl+" y automáticamente se descargará el archivo zip con el nombre del Artifact que contendrá la carpeta con la estructura de la aplicación lista para importar desde el IDE de tu preferencia.

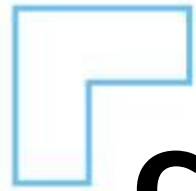


GENERATE CTRL + ↵

EXPLORE CTRL + SPACE

SHARE...





Configurando un proyecto Spring Boot



Después de la descarga del archivo, se debe descomprimir el archivo ZIP con la finalidad de importar o abrir el proyecto creado.



Una vez extraído el archivo, ya podemos abrirlo en Eclipse IDE, seleccionando la pestaña File, luego Open Projects from File System y a continuación eligiendo la carpeta del directorio que contiene el proyecto descomprimido.



➤ Ejecutando el proyecto



Ejecutando el proyecto



Importación del proyecto en tu IDE

Descomprime el archivo ZIP descargado y abre el proyecto en tu Entorno de Desarrollo Integrado (IDE) favorito. La mayoría de los IDE populares, como IntelliJ IDEA, Eclipse o Visual Studio Code, son compatibles con proyectos de Spring Boot.





Ejecutando el proyecto

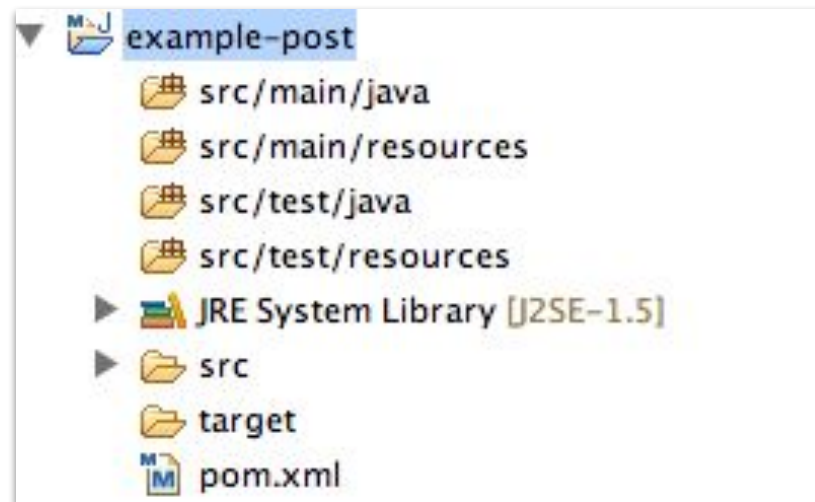
Estructura del proyecto

Un proyecto Spring Boot creado con Spring Initializr tendrá una estructura de directorios predefinida:

- **src/main/java**: Aquí se encuentra la ubicación principal para tus clases de Java.
- **src/main/resources**: Aquí se almacenan los recursos, como archivos de configuración.
- **src/main/webapp** (opcional): Si necesitas recursos web estáticos, como HTML, CSS o JavaScript, puedes ubicarlos aquí.
- **src/test**: Este directorio contiene las pruebas unitarias y de integración para tu aplicación.
- **pom.xml** (si usaste Maven): Estos archivos contienen la configuración del proyecto y las dependencias.



Ejecutando el proyecto



Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE CODING

Ejemplo en vivo

Creando un proyecto con Spring Initializr:

Vamos a poner en práctica lo aprendido! En este ejemplo vamos a crear un proyecto Spring Boot y abrirlo en nuestro IDE.

*1- En primer lugar deberás acceder a la página
<https://start.spring.io/>*

  **Tiempo: 15 minutos**



LIVE CODING

Ejemplo en vivo

A continuación, genera las siguientes configuraciones:

1. *Proyecto: Maven*
2. *Lenguaje: Java*
3. *Versión de Spring Boot: 3.1.2*
4. *Grupo: com.alke*
5. *Artefacto: wallet*
5. *Nombre: AlkeWallet*
6. *Descripción: Billetera Virtual*
7. *Paquete: com.alke.wallet*
8. *Empaquetamiento: WAR*
9. *Versión de Java: 8*



LIVE CODING

Ejemplo en vivo

A continuación, genera las siguientes configuraciones:

Luego, del lado derecho de la pantalla, deberás incluir las siguientes dependencias:

- *Spring Web*
- *Spring Data JPA*
- *Spring Data JDBC*
- *MySQL Driver*
- *Thymeleaf*

Finalmente descargar y descomprimir el archivo para poder abrir el proyecto en Eclipse.



Ejercicio N° 1

Primer proyecto Spring



Primer proyecto Spring

Contexto: 🙌

Ahora es tu turno de crear tu primer proyecto. Crea un proyecto Spring Boot en Spring Initializr con las siguientes particularidades:

Paso a paso: ⚙️

- 1- Proyecto Maven
- 2- Nombre de paquete raíz: com.ejercicio.spring
- 3- Version de Java: 8
- 4- Dependencias: Spring Web
- 5- Descargar .zip
- 6- Abrir en Eclipse.



Tiempo 🕒: 15 minutos



○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ Conocer Spring Boot Initializr y sus características
- ✓ Aprender la estructura de un proyecto Spring



#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 - 10*
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 🙌



Momento: ✚

Time-out!

🕒 5 min.

