

Creación de un proyecto con Spring MVC

Creación de un proyecto con Spring MVC	1
¿Qué aprenderás?	2
Introducción	2
Requisitos y herramientas	2
Conocimientos Previos	3
Paso a Paso	3



¡Comencemos!

¿Qué aprenderás?

- Crear un proyecto Spring MVC desde cero utilizando spring Initializr y maven.
- Crear un proyecto utilizando el framework Spring Boot.
- Configurar en Eclipse IDE el arranque de un proyecto Spring Boot.

Introducción

En base a la teoría que se ha aprendido anteriormente, en este apartado se darán los pasos iniciales para comenzar con un proyecto spring boot desde cero. Este proyecto se llamará holamundospringmvc, y nos servirá para todos los ejemplos prácticos de este documento.

¡Vamos con todo!



Requisitos y herramientas

- Instalación y configuración de Java JDK 1.8.
- Instalación y configuración de Maven 3.6.x.
- Instalación de Eclipse IDE for Enterprise Java Developers versión 4.12.0.
- Spring Tools 4 - for Spring Boot (STS 4) - Instalado en Eclipse IDE.
- Spring Tool 3 Add-On for Spring Tools 4.3.9.10-CI - Instalado en Eclipse IDE.
- Spring Tool 3 (Standalone Edition) 3.9.9.RELEASE - Instalado en Eclipse IDE.
- Maven (Java EE) Integration for Eclipse WTP (Luna/Mars) 1.2.0 - Instalado en Eclipse IDE.

Conocimientos Previos

- Programación en lenguaje Java.
- Conocimientos básicos de Maven.
- Conocimientos básicos de Eclipse IDE.
- Conocer qué es JSTL.
- Conocer MVC.
- Conocer qué es el Front Controller y el Dispatcher (despachador) y su funcionamiento básico.

Paso a Paso

1. Crear el proyecto HolaMundoSpringMVC mediante [start.spring](https://start.spring.io).
 - a. Ir a la web oficial de [Spring Boot](https://start.spring.io), para crear el proyecto, e ingresar los datos del proyecto como se indica a continuación:

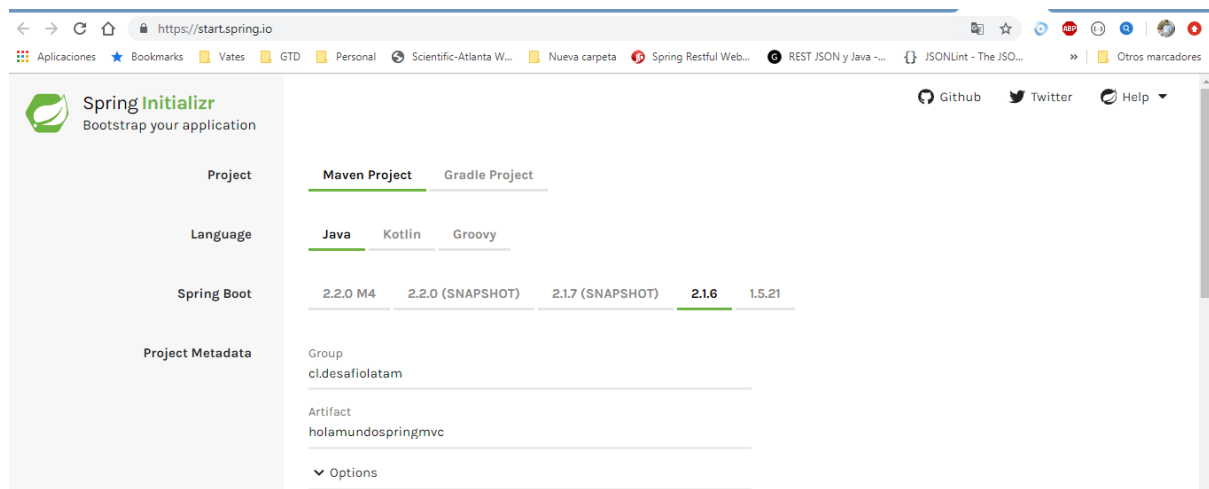


Imagen 1. Crear proyecto HolaMundoSpringMVC.
Fuente: Desafío Latam.

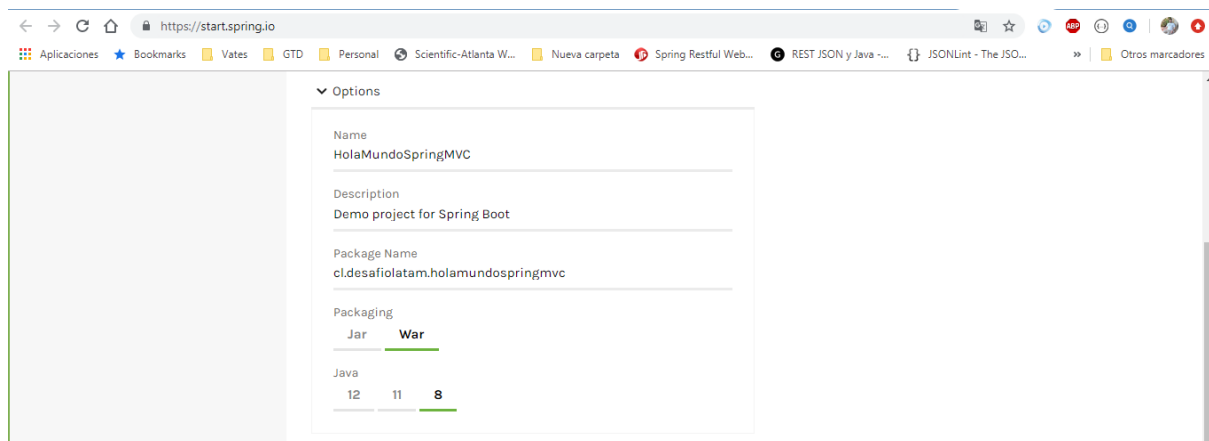


Imagen 2. Seleccionando War y Java 8.
Fuente: Desafío Latam.

b. Agregar las dependencias.

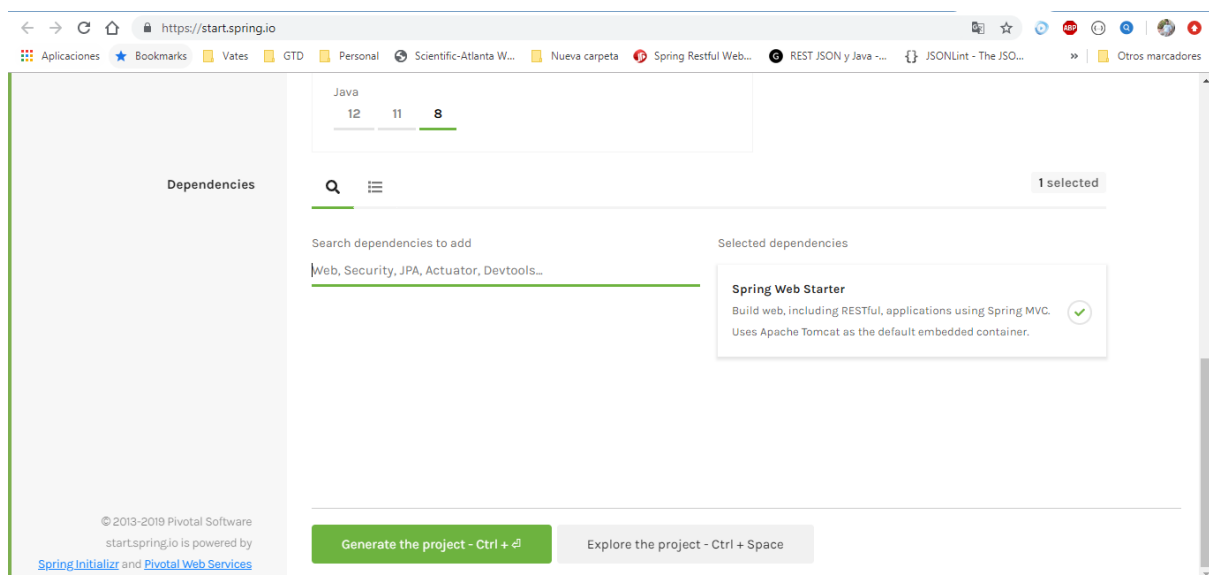


Imagen 3. Agregar dependencias.
Fuente: Desafío Latam.

En Search dependencies to add, buscar la palabra web.

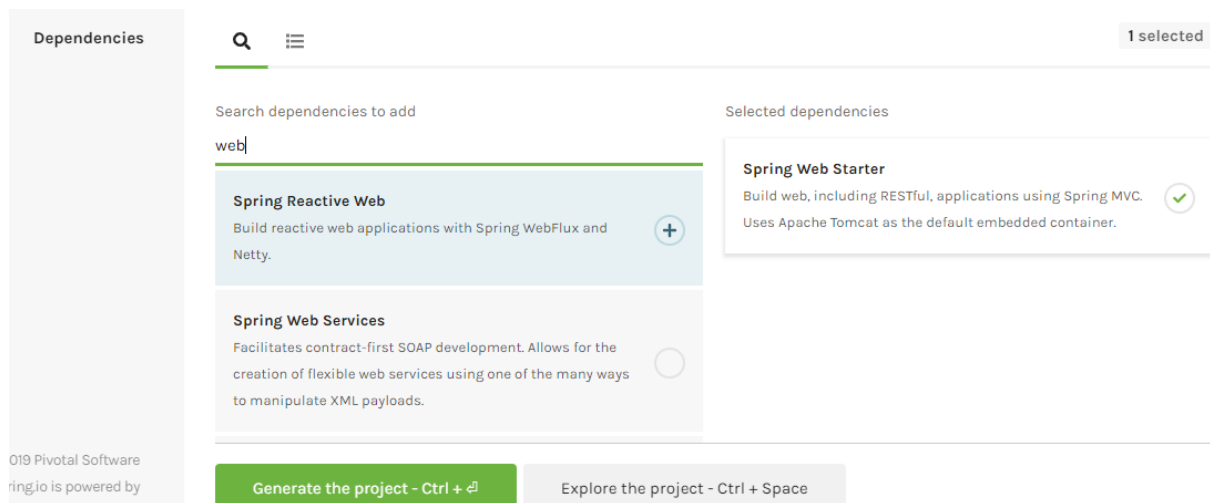


Imagen 4. Buscar dependencia Web.
Fuente: Desafío Latam.

- c. Generar el proyecto. Presionar el botón Generate the project - Ctrl +. El proyecto se descargará automáticamente en un archivo .zip.

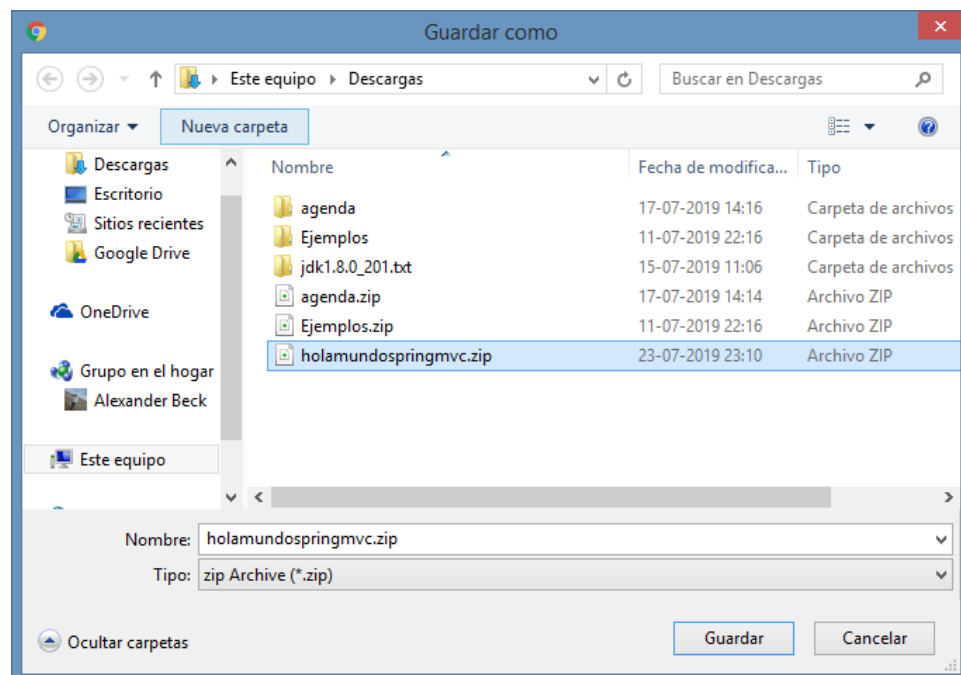


Imagen 5. Descargar el proyecto .zip y guardar.
Fuente: Desafío Latam.

- Presionar el botón guardar.

- d. Descomprimir el archivo en nuestro workspace configurado en Eclipse IDE.

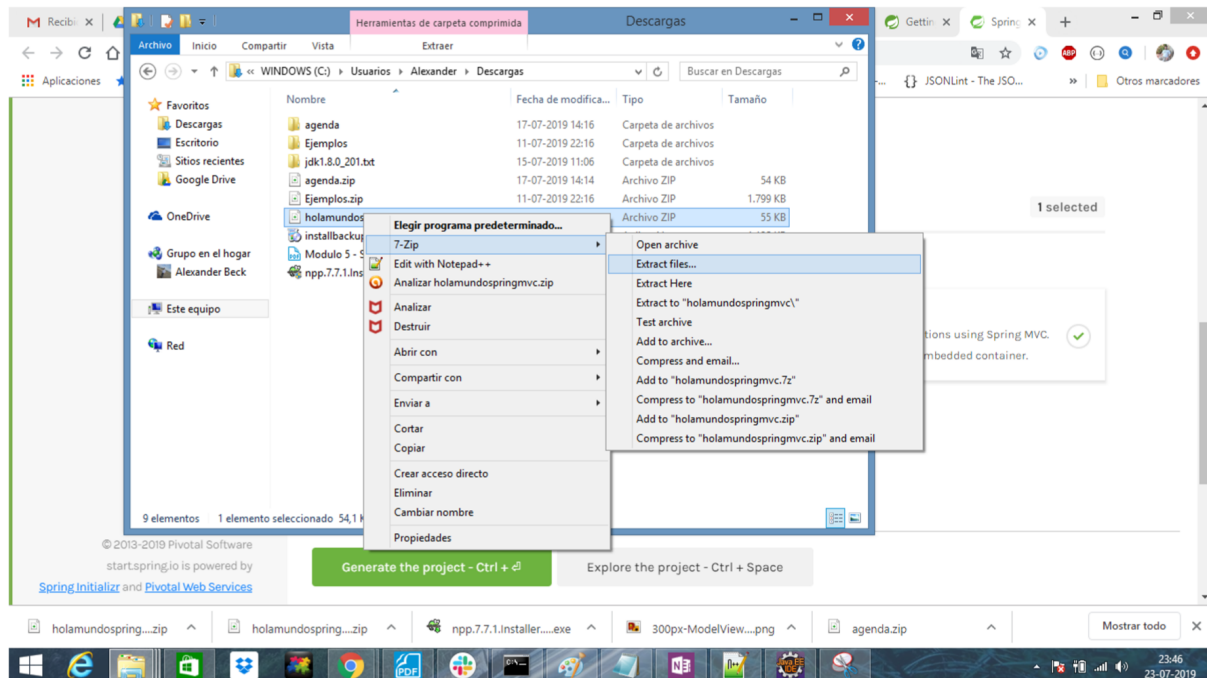


Imagen 6. Extraer el .zip.

Fuente: Desafío Latam.

- Click derecho sobre el archivo .zip de nuestro proyecto y presionar Extract files.
- Con el botón, buscar la carpeta o directorio de nuestro espacio de trabajo.

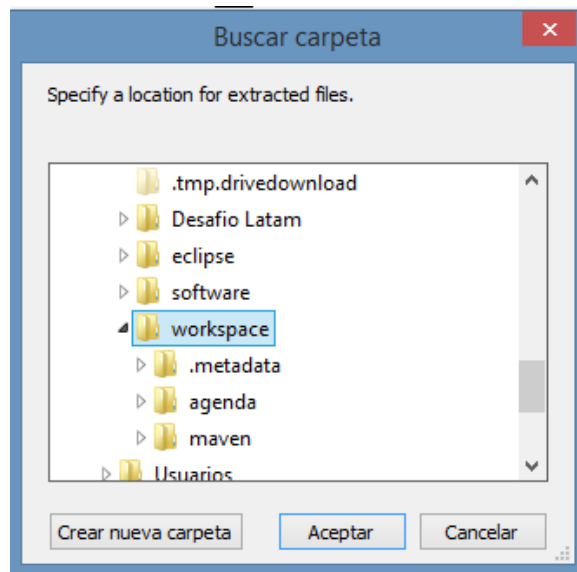


Imagen 7. Buscar carpeta.
Fuente: Desafío Latam.

- Presionar botón Aceptar y luego el botón OK de la aplicación Zip incluida en Window 8.

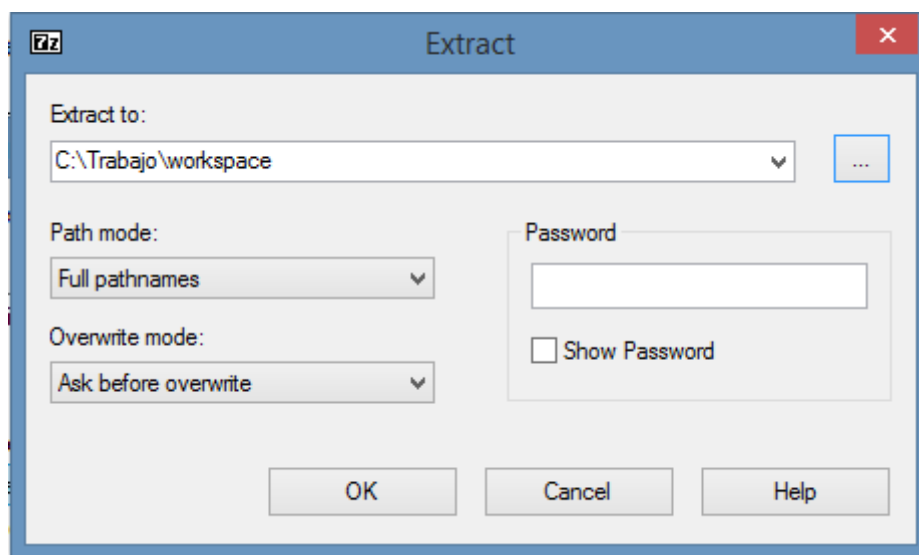


Imagen 8. Extraer archivos.
Fuente: Desafío Latam.

2. Importar el proyecto en Eclipse IDE.
 - a. Abrir Eclipse IDE, e ir al menú File > Import.
 - b. Seleccionar en la lista Existing Maven Project, y luego presionar el botón Next >

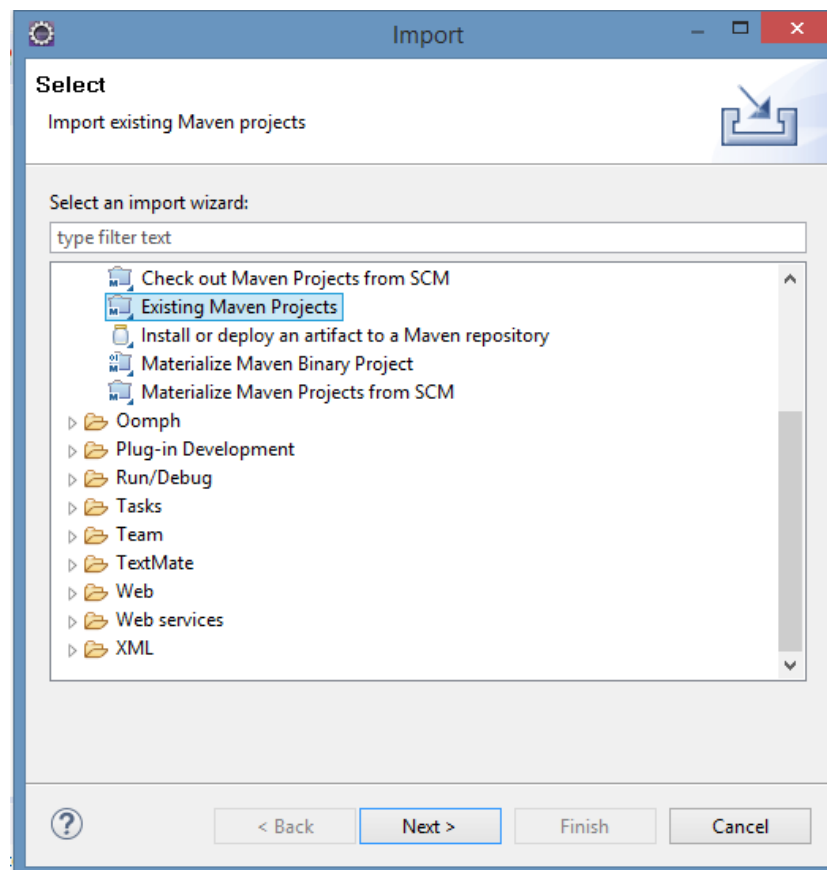


Imagen 9. Proyecto maven existente.
Fuente: Desafío Latam.

Con el botón Browser, ir al directorio en donde guardamos el proyecto descomprimido. Para este caso, el directorio correspondiente al workspace configurado en Eclipse IDE.

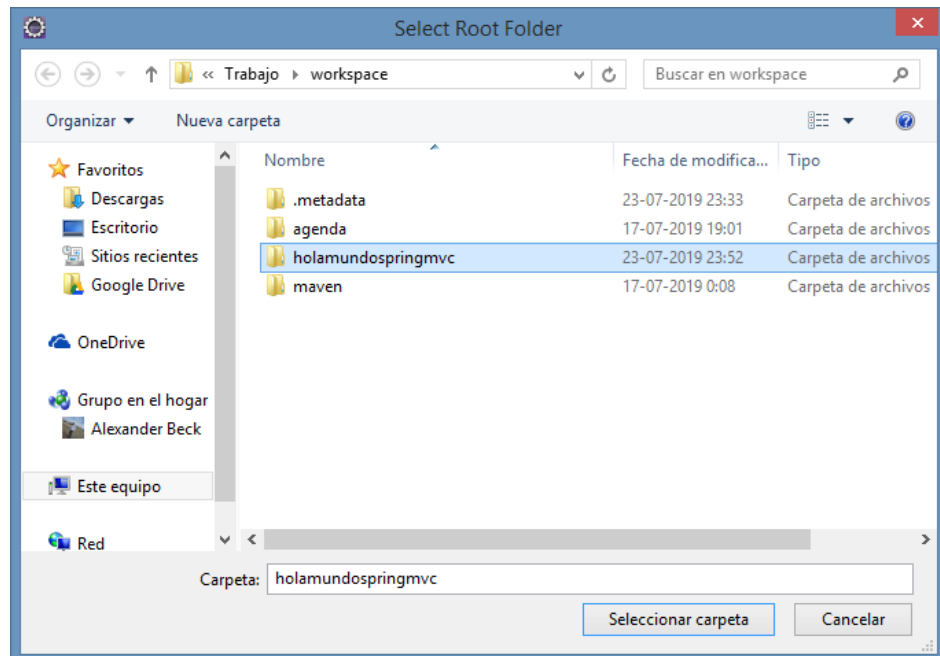


Imagen 10. Buscar el proyecto descomprimido.

Fuente: Desafío Latam.

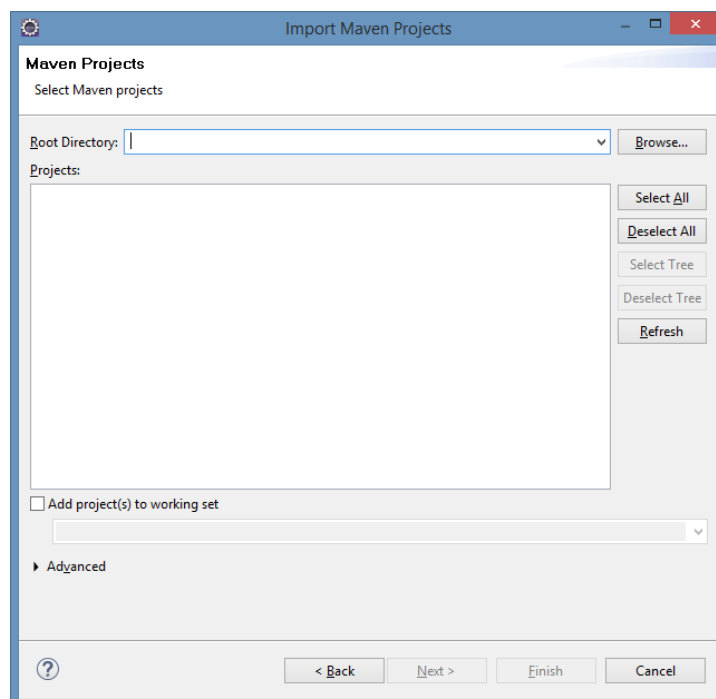


Imagen 11. Seleccionar la carpeta del proyecto.

Fuente: Desafío Latam.

Seleccionar la carpeta del proyecto, y luego presionar el botón Seleccionar Carpeta.

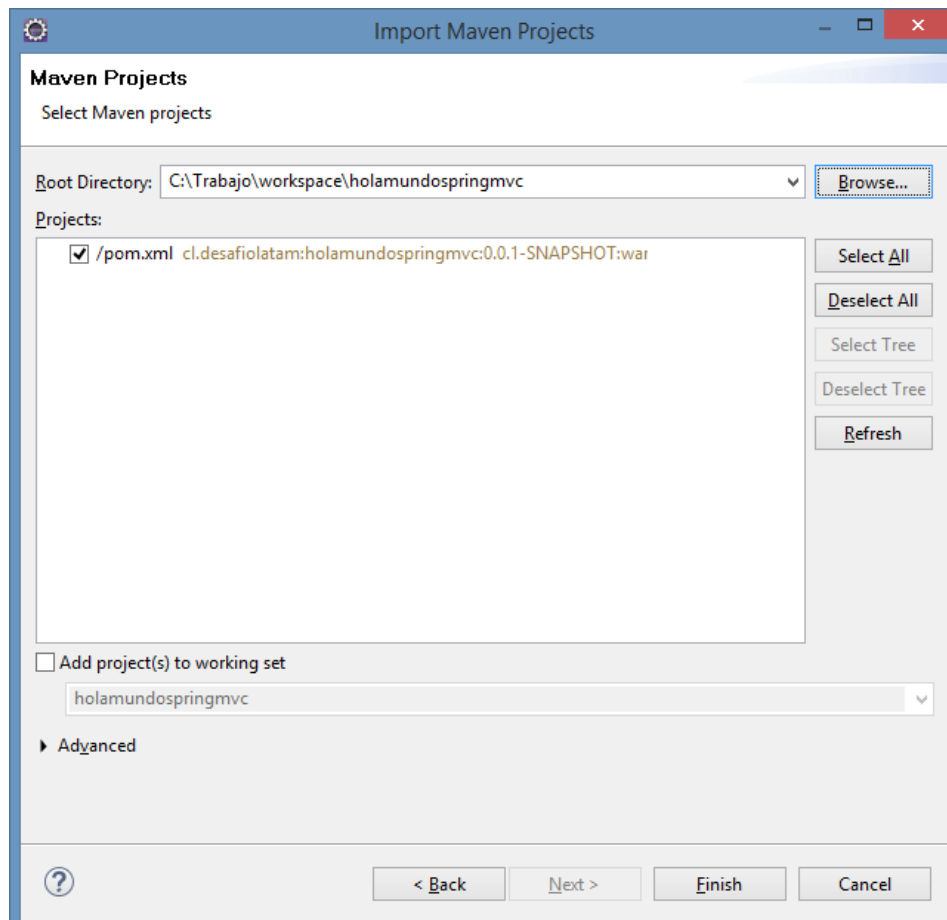


Imagen 12. Importar el proyecto.
Fuente: Desafío Latam.

- c. Presionar el botón Finish. El proyecto debiese quedar configurado y creado en Eclipse IDE, con la siguiente estructura.

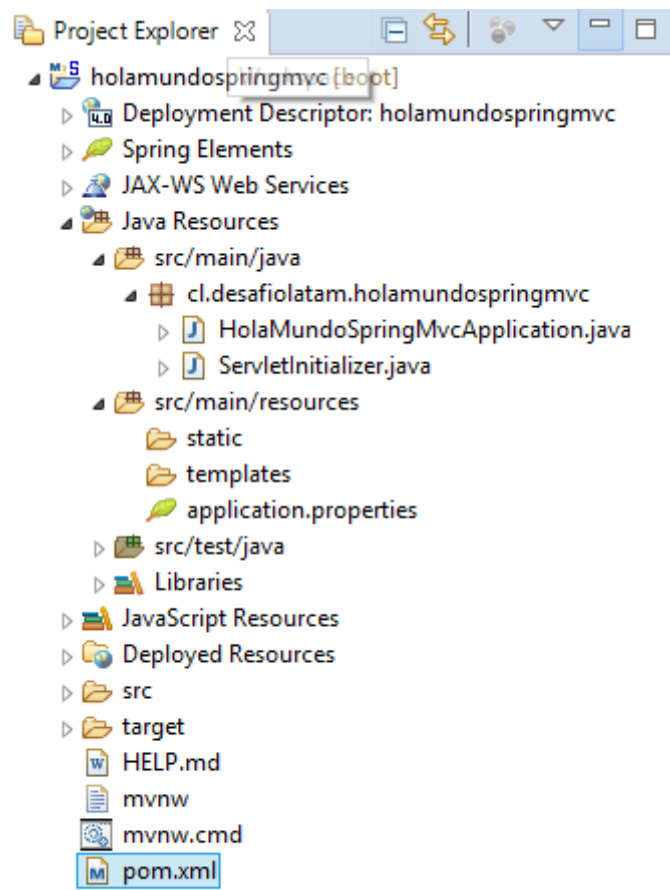


Imagen 13. Estructura del proyecto.
Fuente: Desafío Latam.

3. Comprobar que el proyecto, Spring Boot arranque correctamente.
 - a. Click derecho en el proyecto, e ir al menú Debug As > Debug Configurations.

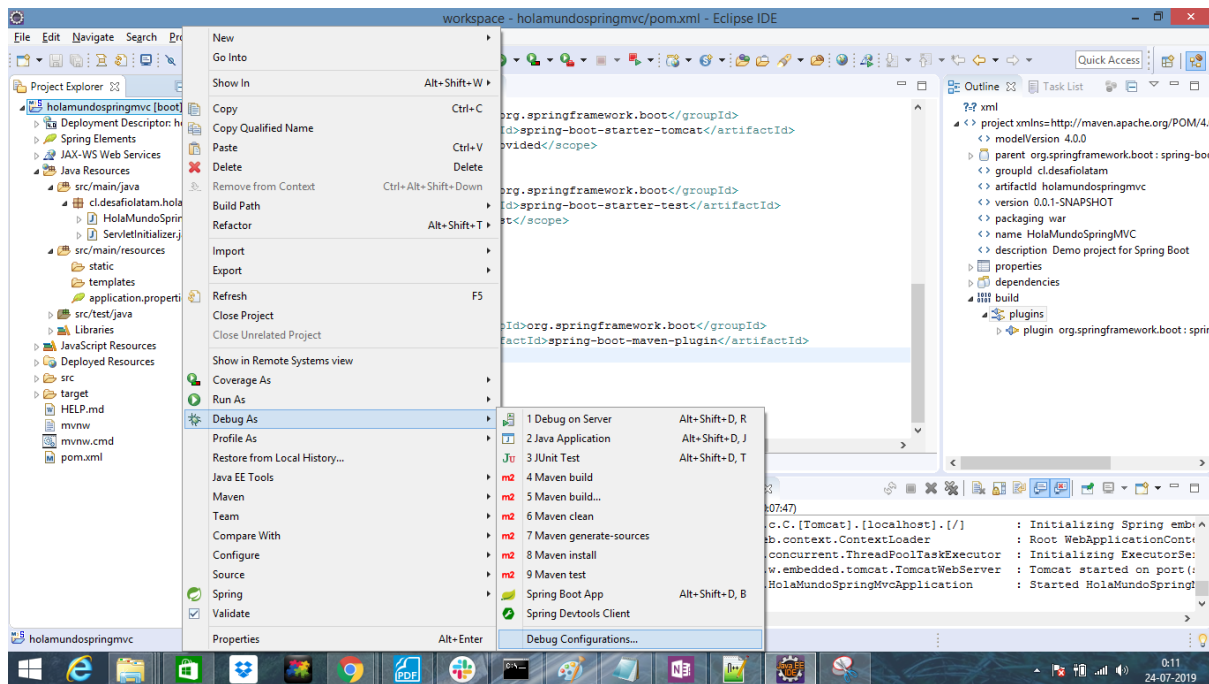


Imagen 14. Abrir configuración de arranque del proyecto.

Fuente: Desafío Latam.

- b. Configurar el arranque del proyecto, como se indica a continuación.

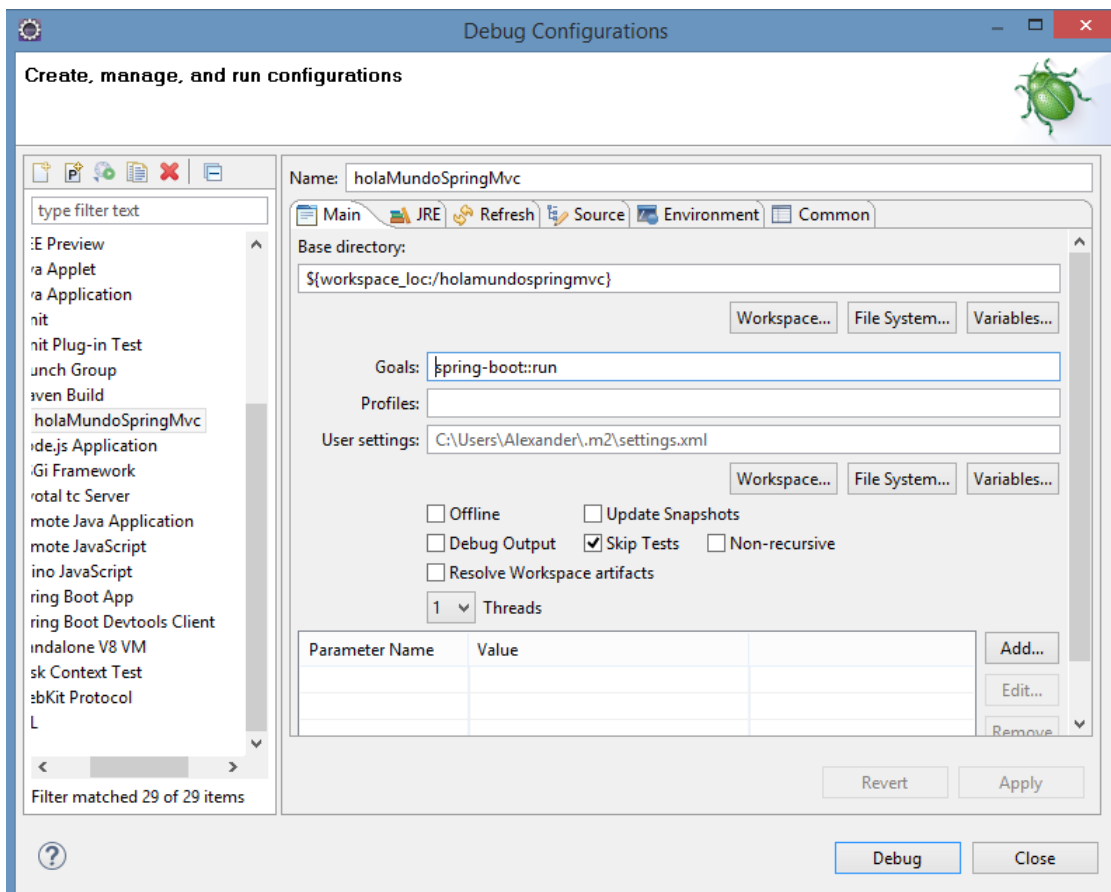


Imagen 15. Configurar el arranque del proyecto.

Fuente: Desafío Latam.

- **goals: spring-boot::run:** Indica a maven que el proyecto se iniciará mediante el plugin de spring boot.
- **Skip Test:** Le indica a maven que no compile los archivos java correspondientes a test unitarios.
- **Offline:** Maven solo funcionará de forma offline, usando los recursos locales.
- **Update Snapshots:** Si está seleccionado, buscará si maven tiene actualizaciones en sus paquetes.
- **Debug Output:** Permite a la aplicación crear mensajes de debug.
- **Resolve Workspace Artifacts:** Si está seleccionada se buscarán las clases de los módulos dependientes en un repositorio de maven en la salida de compilación.

- c. Presionamos el botón Apply y luego Debug. Comenzará el arranque del proyecto bajo el servidor Apache embebido en Spring Boot.

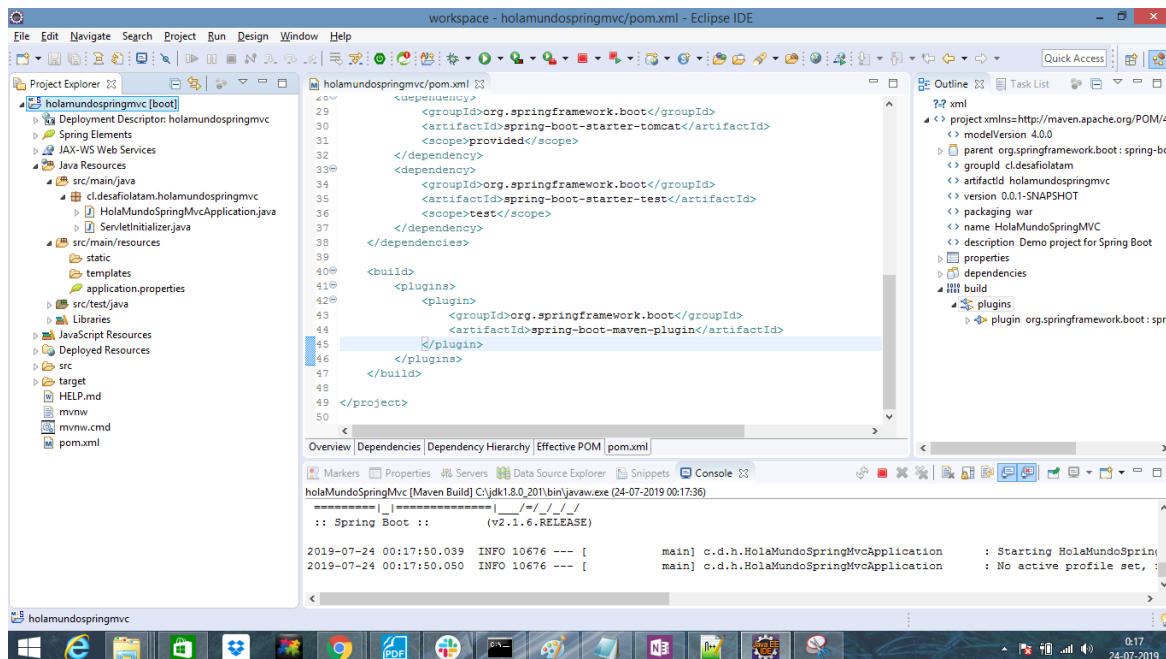


Imagen 16. Iniciando el proyecto.

Fuente: Desafío Latam.

Al final, el log del arranque de la aplicación que se muestra mediante el Eclipse IDE, debe decir lo siguiente:

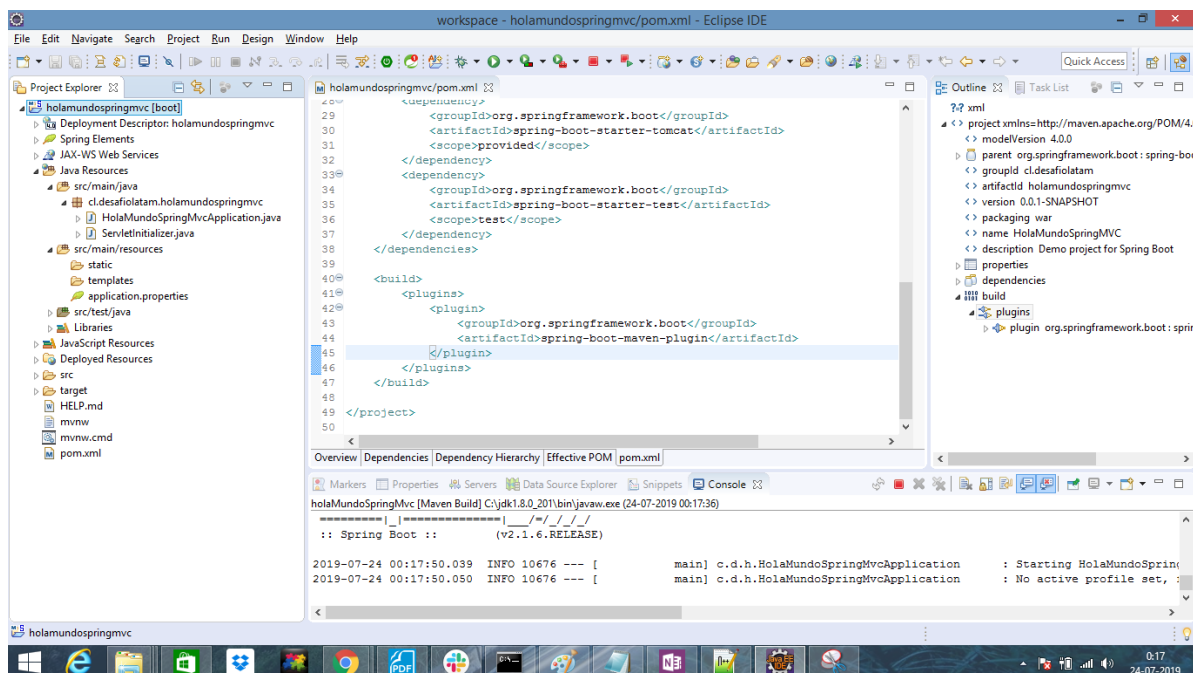


Imagen 17. Log del inicio del proyecto.

Fuente: Desafío Latam.

- d. Spring Boot, incluye todas las librerías necesarias para trabajar con Spring de forma automática, una de estas librerías es la que nos ayudará en la programación con MVC.

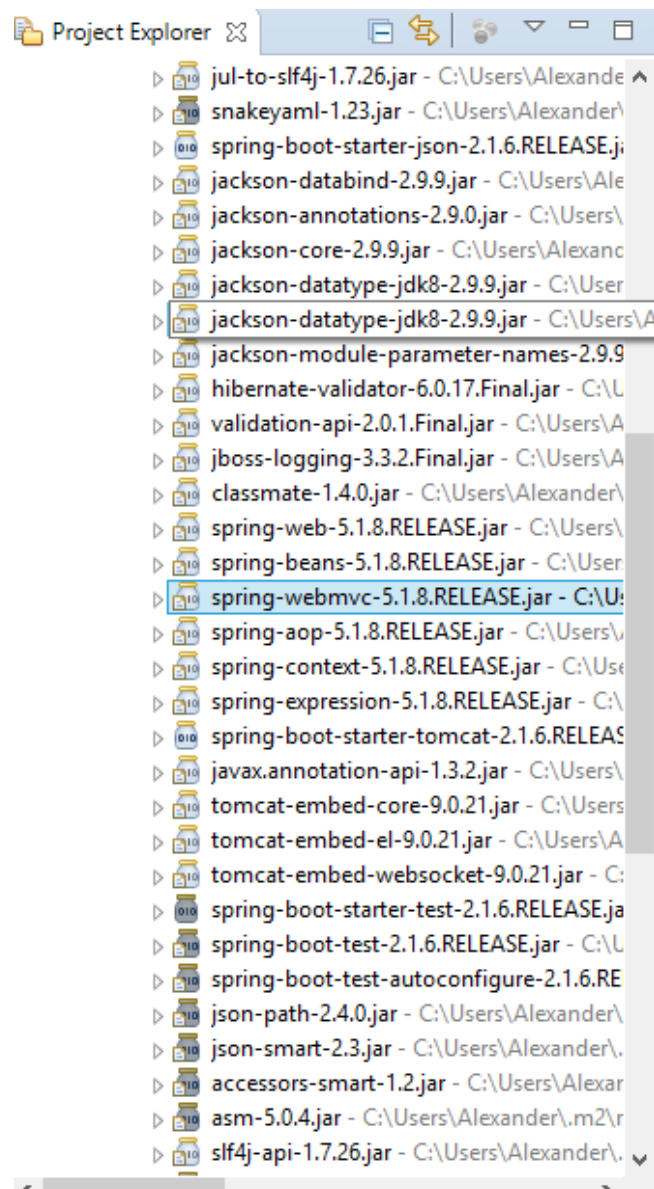


Imagen 18. Librerías incluidas.
Fuente: Desafío Latam.

Más adelante, seguiremos avanzando en este proyecto, para así lograr una pequeña aplicación web, que muestre por pantalla la frase Hola Mundo Spring Mvc!!!. Por lo tanto, nos queda en esta etapa, aprender lo siguiente:

- Configurar el dispatcher de la aplicación.
- Crear la Vista.
- Crear el Controlador.