



Certified Tech Developer

The Ultimate Degree

Back end I

Consigna para las mesas de trabajo

Crear un proyecto Spring Boot

Objetivo

Tal como te pedimos en la clase asincrónica, ahora vamos a crear otro proyecto con Spring Boot.

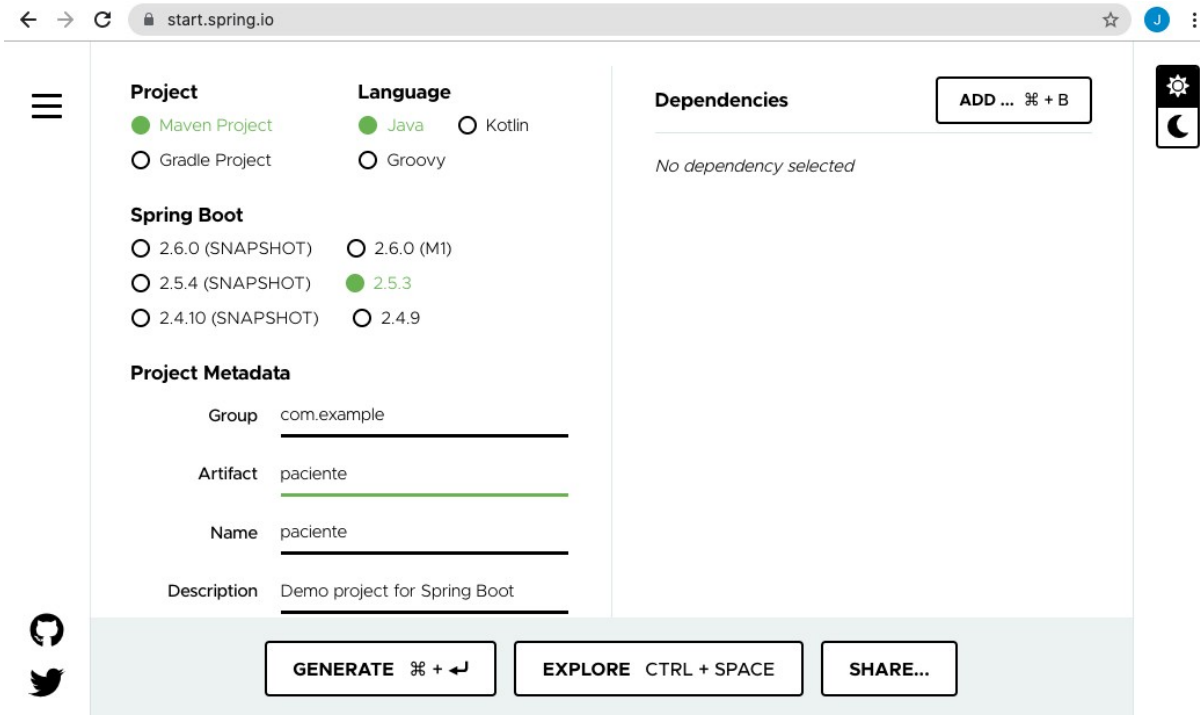
A tener en cuenta:

- Ejercicio individual
- Complejidad: baja 🐣

Instrucciones

1. Primero, debemos ingresar al siguiente link: <https://start.spring.io>.
2. Allí visualizarás la siguiente pantalla. En el apartado "Name" y "Artifact" escribí "paciente" (atajo: si lo escribís primero en "Artifact", se completa por defecto el "Name").

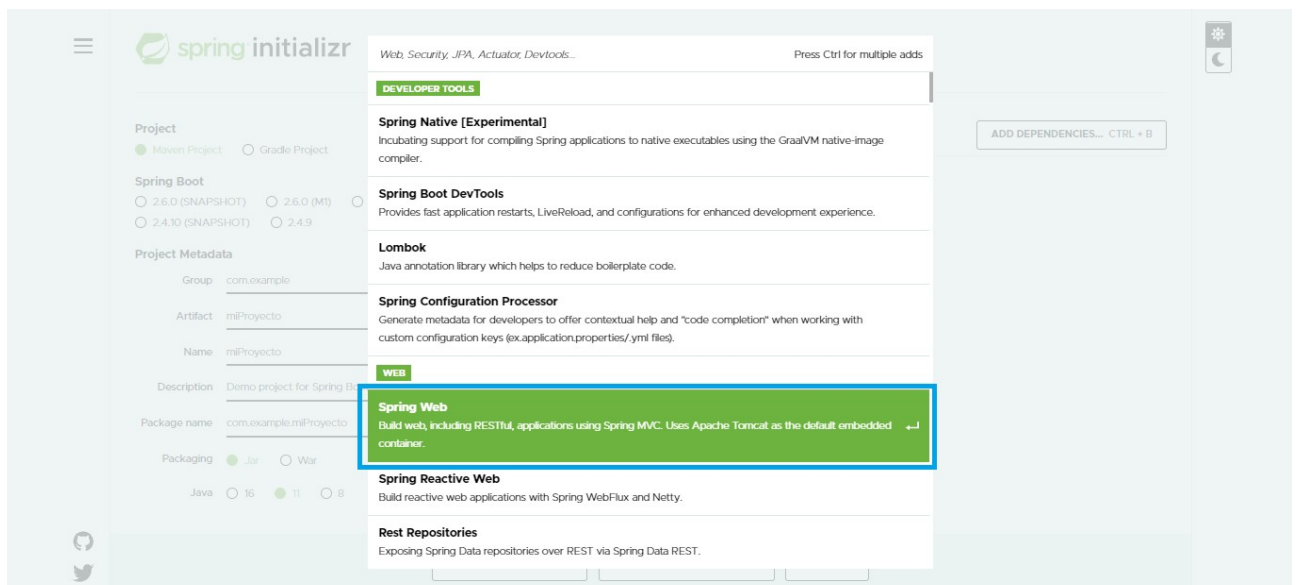
Los otros campos debés dejarlos tal cual aparecen.



The screenshot shows the start.spring.io web application generator. The interface is divided into several sections:

- Project:** Includes radio buttons for "Maven Project" (selected) and "Gradle Project".
- Language:** Includes radio buttons for "Java" (selected) and "Kotlin", and "Groovy".
- Spring Boot:** Includes radio buttons for versions: "2.6.0 (SNAPSHOT)", "2.6.0 (M1)", "2.5.4 (SNAPSHOT)", "2.5.3" (selected), and "2.4.10 (SNAPSHOT)", "2.4.9".
- Project Metadata:** Includes input fields for "Group" (com.example), "Artifact" (paciente), "Name" (paciente), and "Description" (Demo project for Spring Boot).
- Dependencies:** A section with the text "No dependency selected" and an "ADD ... ⌘ + B" button.
- Buttons:** At the bottom, there are three buttons: "GENERATE ⌘ + ↵", "EXPLORE CTRL + SPACE", and "SHARE...".

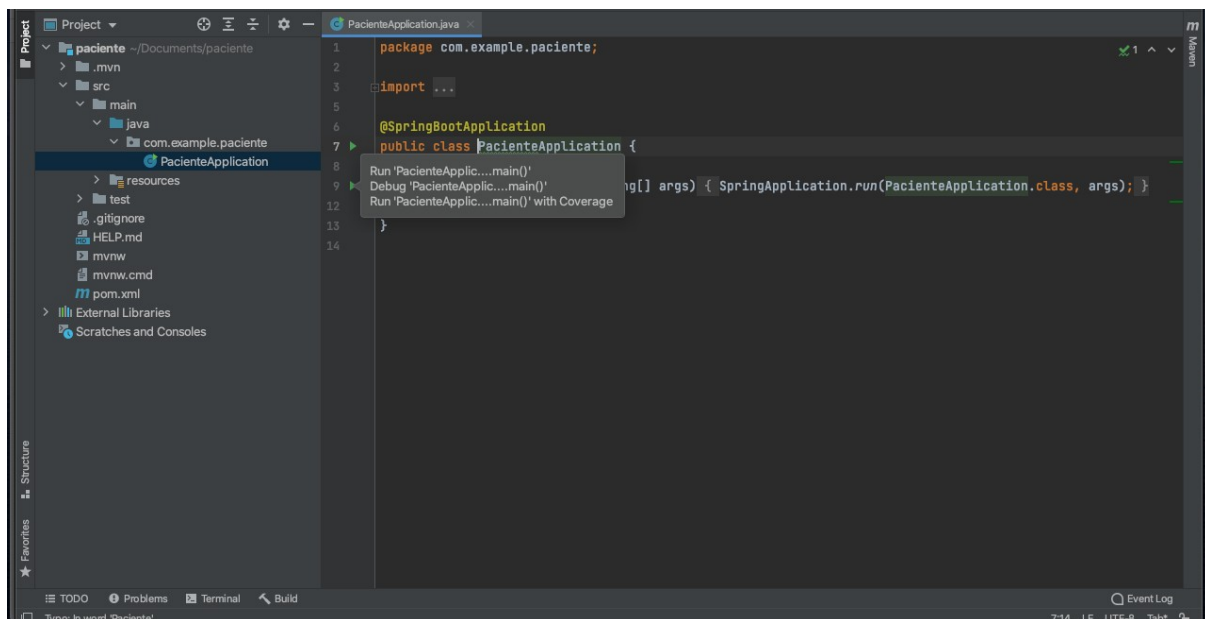
- Una vez completado lo mencionado anteriormente, presioná el botón "ADD DEPENDENCIES".
- Allí se desplegará la siguiente pantalla, donde tenés que elegir la opción "Spring Web".



The screenshot shows the spring initializr web application generator. The interface is divided into several sections:

- Project:** Includes radio buttons for "Maven Project" (selected) and "Gradle Project".
- Spring Boot:** Includes radio buttons for versions: "2.6.0 (SNAPSHOT)", "2.6.0 (M1)", "2.4.10 (SNAPSHOT)", and "2.4.9".
- Project Metadata:** Includes input fields for "Group" (com.example), "Artifact" (miProyecto), "Name" (miProyecto), and "Description" (Demo project for Spring Boot).
- Package name:** com.example.miProyecto
- Packaging:** Includes radio buttons for "Jar" (selected) and "War".
- Language:** Includes radio buttons for "Java" (selected), "16", "11", and "8".
- Dependencies:** A section with the text "Press Ctrl for multiple adds" and an "ADD DEPENDENCIES... CTRL + B" button. Below this, there is a list of dependencies:
 - DEVELOPER TOOLS:**
 - Spring Native [Experimental]** Incubating support for compiling Spring applications to native executables using the GraalVM native-image compiler.
 - Spring Boot DevTools** Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
 - Lombok** Java annotation library which helps to reduce boilerplate code.
 - Spring Configuration Processor** Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).
 - WEB:**
 - Spring Web** Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. (This option is highlighted with a blue box and a cursor.)
 - Spring Reactive Web** Build reactive web applications with Spring WebFlux and Netty.
 - Rest Repositories** Exposing Spring Data repositories over REST via Spring Data REST.

5. Luego, presioná "Generate". Ahora se va a descargar un archivo llamado paciente.zip, el cual deberás descomprimir y abrir IntelliJ IDEA.
6. En IntelliJ IDEA, deberás ir a "Archivo -> Nuevo -> Project from existing source". Luego, tenés que buscar la carpeta "paciente" que descomprimiste y abrirla desde el archivo pom.xml.
7. Buscá la clase PacienteApplication y ejecutá el método main.



8. Ahora, copió y pegá este código para poder crear el Controller:

```
package com.example.paciente;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class PacienteApplication {

    public static void main(String[] args) {
        SpringApplication.run(MiProyectoApplication.class, args);
    }

    @GetMapping
    public String holaPAciente(){
        return "Hola Paciente";
    }
}
```



```
}  
  
}
```

9. Luego, debés ir al navegador y correr el localhost:8080.
10. ¿Obtuviste el “Hola Paciente” en pantalla?

Ahora ya tenés un proyecto Spring Boot creado y listo para usarse.