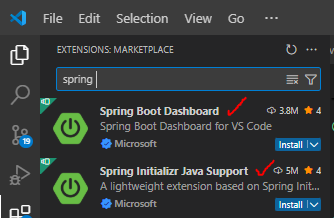
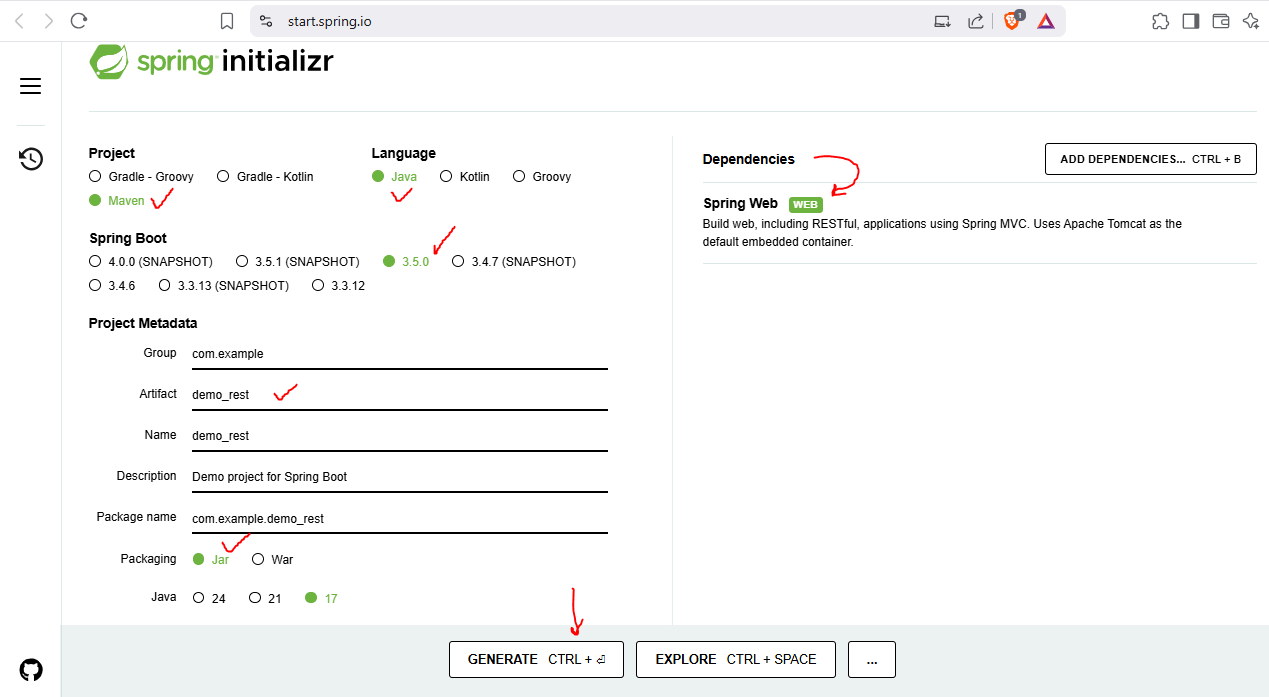
Prueba de Sistema a Sistema

En VS Code instalar las siguientes extensiones:



Ir al sitio <https://start.spring.io/> y generar un proyecto con las características:



Eso generará un archivo .zip Descomprimirlo en la carpeta deseada

Nuevamente en VS Code, abrir un nuevo proyecto, y seleccionar:

* + - * **Project**: Maven
      * **Language**: Java
      * **Dependencies**: **Spring Web** (para REST).

**Importar el proyecto** Spring en VS Code (Menú File > Open Folder)

**Crear el controlador (Servicio REST)**

Crear la clase SaludoController.java en src/main/java/com/ejemplo/demo:

package com.ejemplo.demo;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class SaludoController {

@PostMapping("/saludar-y-sumar")

public Respuesta saludarYSumar(@RequestBody Solicitud solicitud) {

String mensaje = "¡Hola! " + solicitud.getNombre();

int suma = solicitud.getInt1() + solicitud.getInt2();

return new Respuesta(mensaje, "La suma es: " + suma);

}

}

### ****Definir las clases de Solicitud y Respuesta****

Crear dos clases para estructurar los datos de entrada y salida:

Solicitud.java (Datos de entrada):

package com.ejemplo.demo;

public class Solicitud {

private String nombre;

private int int1;

private int int2;

*// Getters y Setters (necesarios para Spring)*

public String getNombre() { return nombre; }

public void setNombre(String nombre) { this.nombre = nombre; }

public int getInt1() { return int1; }

public void setInt1(int int1) { this.int1 = int1; }

public int getInt2() { return int2; }

public void setInt2(int int2) { this.int2 = int2; }

}

Respuesta.java (Datos de salida):

package com.ejemplo.demo;

public class Respuesta {

private String mensaje;

private String suma;

*// Constructor*

public Respuesta(String mensaje, String suma) {

this.mensaje = mensaje;

this.suma = suma;

}

*// Getters (necesarios para la respuesta JSON)*

public String getMensaje() { return mensaje; }

public String getSuma() { return suma; }

}

**Instalar la dependencia en pom.xml**

**Editar y agregar**:

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

</dependencies>

**Ejecutar y probar el servicio**

**Iniciar la aplicación**:

* + Buscar la clase principal (DemoApplication.java) y ejecutarla.
  + Se verá un mensaje como: Tomcat started on port 8080.

**Probar con Postman** o **curl**:

* + **Método**: POST
  + **URL**: http://localhost:8080/saludar-y-sumar
  + **Body (JSON)**:

{

"nombre": "Juan",

"int1": 10,

"int2": 5

}

* + **Respuesta esperada**:

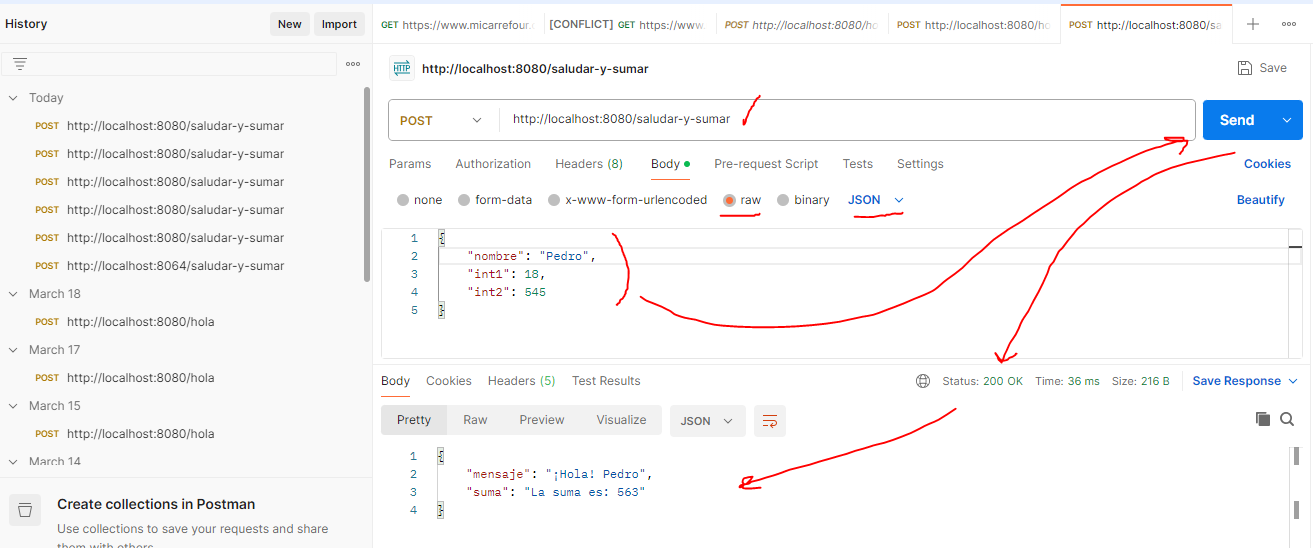
{

"mensaje": "¡Hola! Juan",

"suma": "La suma es: 15"

}

En Postman:



### ****Detalles****:

1. @RestController: Indica que la clase manejará peticiones REST.
2. @PostMapping: Mapea la ruta /saludar-y-sumar al método.
3. @RequestBody: Convierte automáticamente el JSON recibido en un objeto Solicitud.
4. **Spring Boot**: Se encarga del servidor embebido (Tomcat) y la serialización JSON.