Contenido

[**Conceptos básicos** 2](#_Toc145324789)

[**Sección 79: Spring Framework 5: Creando webapps con Spring Boot** 2](#_Toc145324790)

[594. Importante: antes de comenzar con esta sección de Spring 2](#_Toc145324791)

[595. Introducción 2](#_Toc145324792)

[596. Instalación del Spring Tools Suite IDE 2](#_Toc145324793)

[597. Creando un proyecto Web con Spring Boot 2](#_Toc145324794)

[598. Estructura de una aplicación Spring Boot 3](#_Toc145324795)

[599. Creando el controlador y la vista 3](#_Toc145324796)

[600. Pasar datos a la vista 3](#_Toc145324797)

[601. Anotación @RequestMapping sobre el Controlador 3](#_Toc145324798)

[602. Introducción MVC 6](#_Toc145324799)

[603. El objeto Model 6](#_Toc145324800)

[604. Directiva if de Thymeleaf 6](#_Toc145324801)

[605. Directiva each (foreach) de Thymeleaf 6](#_Toc145324802)

[606. Anotación @ModelAttribute 6](#_Toc145324803)

[607. Anotacion @RequestParam 6](#_Toc145324804)

[608. Obtener varios parámetros de la URL con @RequestParam 6](#_Toc145324805)

[609. Anotación @PathVariable 6](#_Toc145324806)

[610. Obtener varios parámetros de la ruta con @PathVariable 6](#_Toc145324807)

[611. Agregando recursos estáticos y estilos CSS personalizados 6](#_Toc145324808)

[612. Inyectar valores usando la anotación @Value 6](#_Toc145324809)

[613. Agregando otros archivos properties 6](#_Toc145324810)

[personalizados para los textos 6](#_Toc145324811)

[614. Retornando redirect y forward como respuesta en métodos del controlador 6](#_Toc145324812)

[615. Despliegue y ejecución desde terminal (deploy) 6](#_Toc145324813)

# **Conceptos básicos**

1. **Spring**: Se utiliza para simplificar el desarrollo de aplicaciones empresariales en Java al proporcionar una estructura y bibliotecas.
2. **Spring Boot**: Se utiliza para crear aplicaciones Java de manera rápida y sencilla al eliminar gran parte de la configuración manual y ofrecer un inicio rápido.
3. **Framework**: Se utiliza para facilitar y estandarizar el desarrollo de software al proporcionar herramientas, reglas y convenciones predefinidas para resolver problemas comunes en un dominio específico.

# **Sección 79: Spring Framework 5: Creando webapps con Spring Boot**

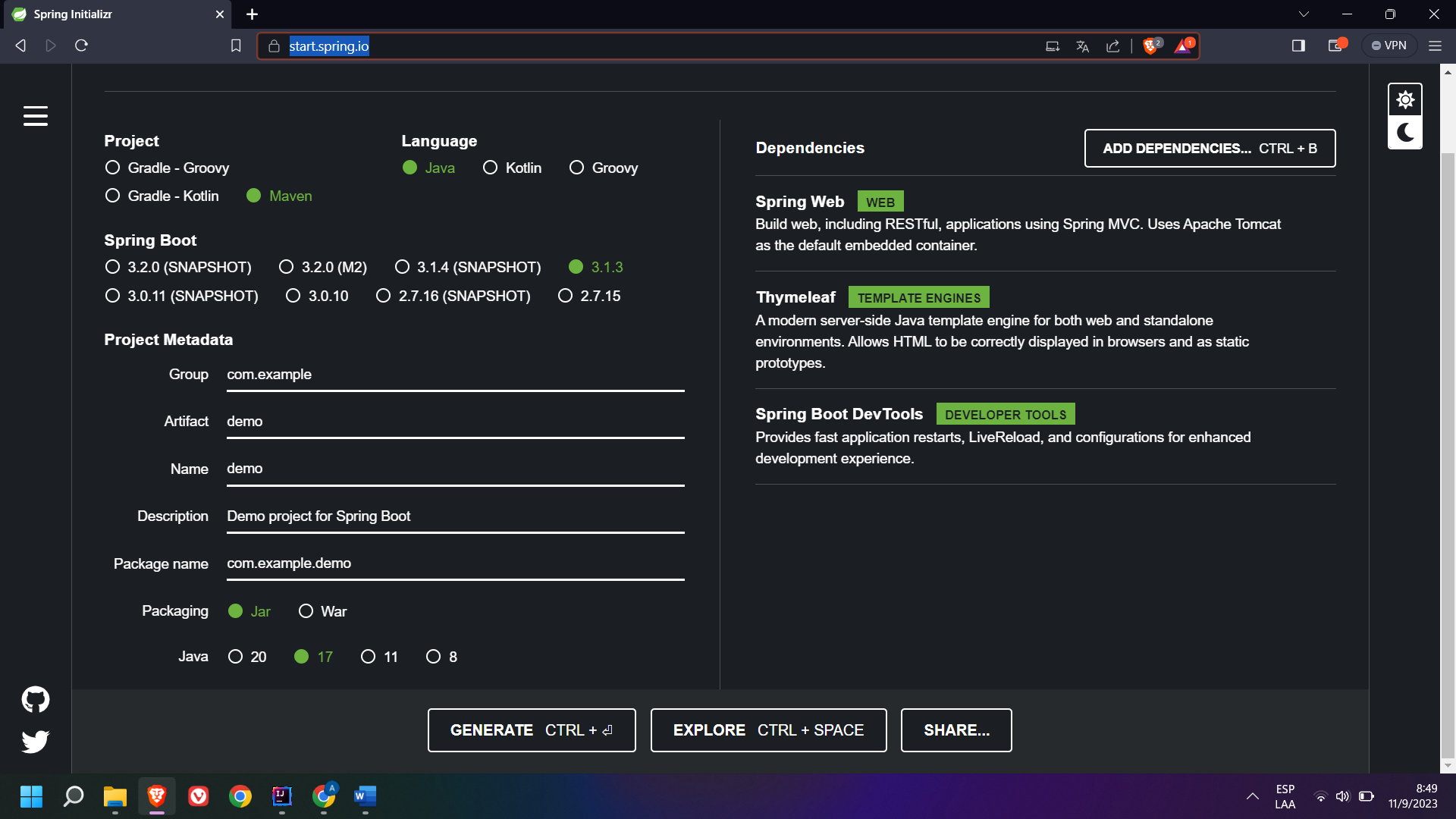
# 594. Importante: antes de comenzar con esta sección de Spring

# 595. Introducción

# 596. Instalación del Spring Tools Suite IDE

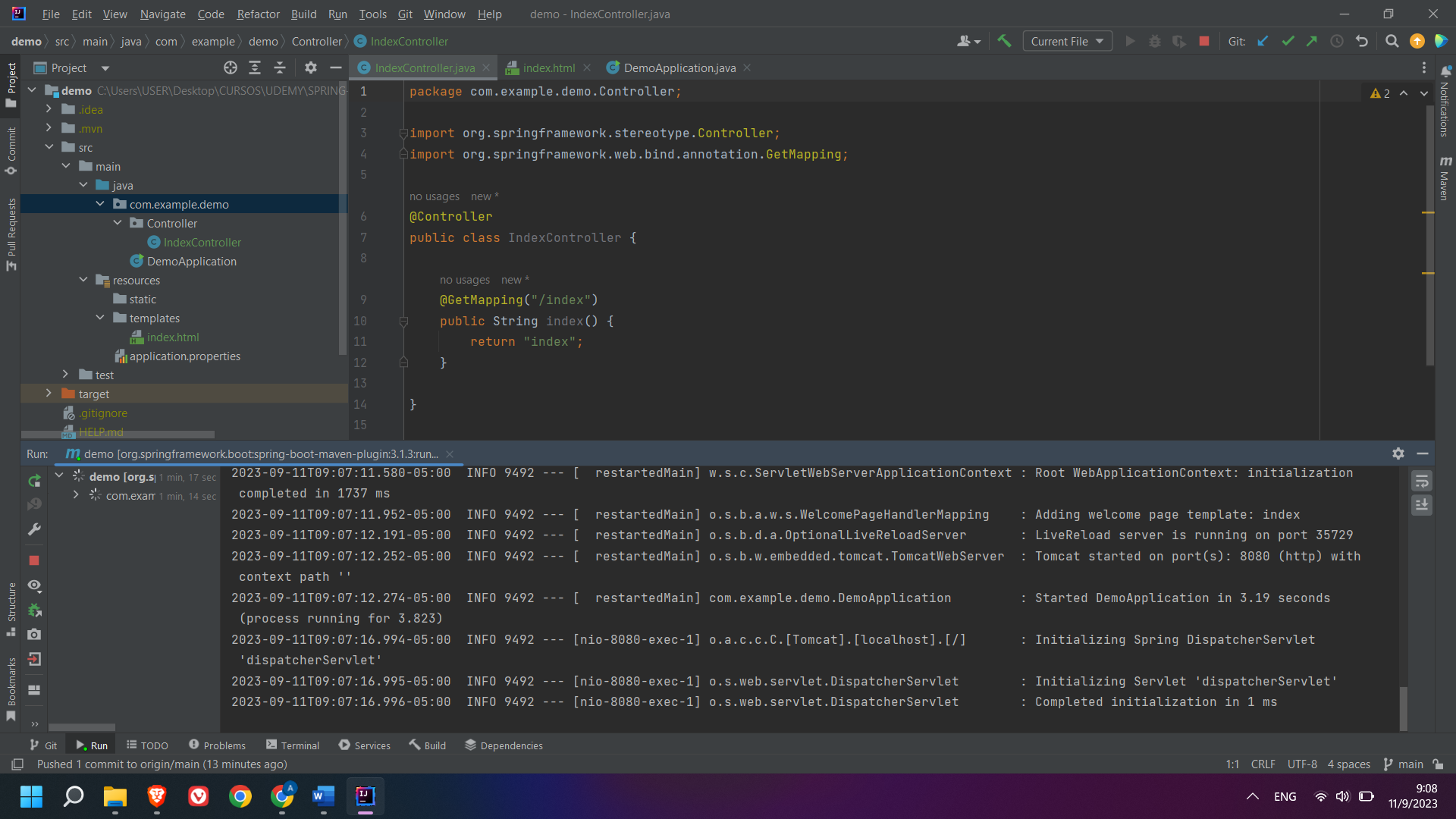
# 597. Creando un proyecto Web con Spring Boot

<https://start.spring.io/>



# 598. Estructura de una aplicación Spring Boot

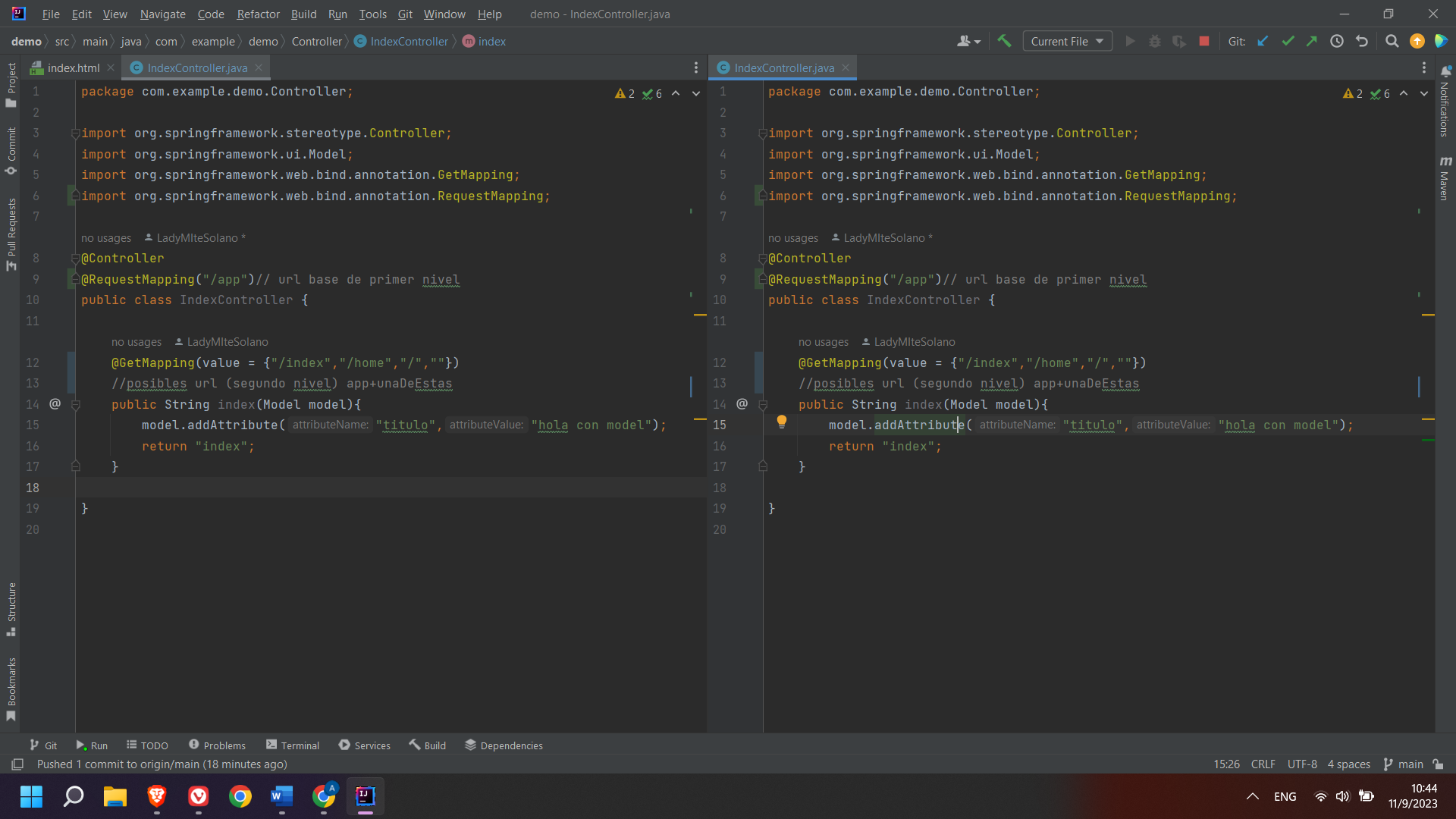
# 599. Creando el controlador y la vista

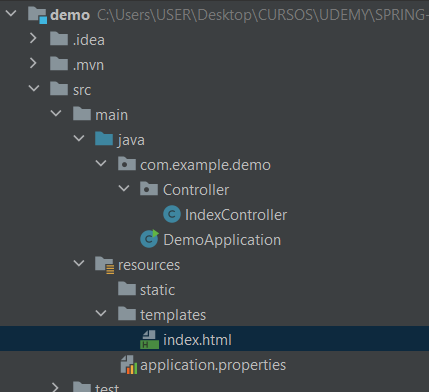


# 600. Pasar datos a la vista

La misma del 601 más request mappgin

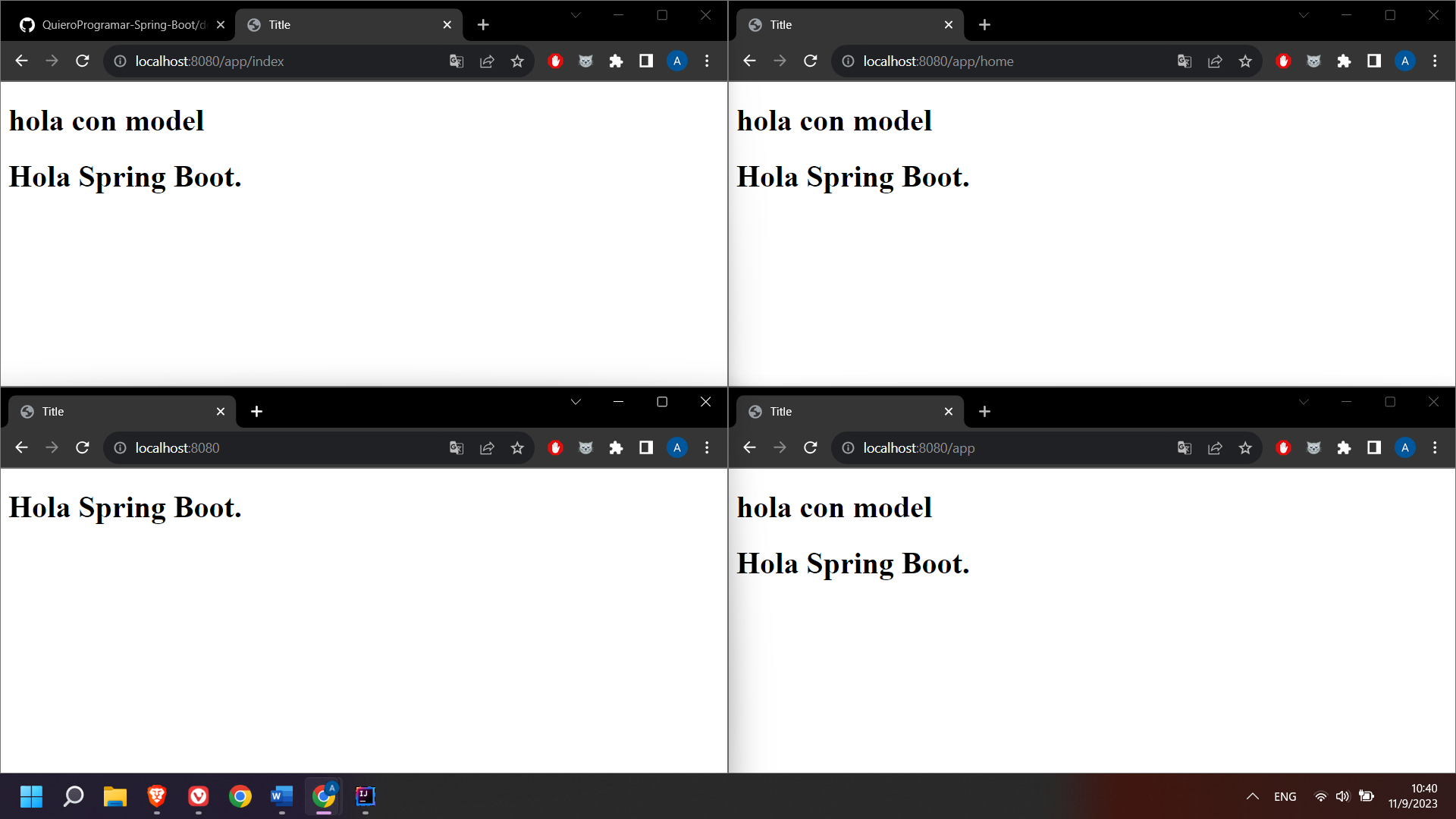
# 601. Anotación @RequestMapping sobre el Controlador





package com.example.demo.Controller;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
  
@Controller  
@RequestMapping("/app")// url base de primer nivel  
public class IndexController {  
  
 @GetMapping(value = {"/index","/home","/",""})  
 //posibles url (segundo nivel) app+unaDeEstas  
 public String index(Model model){  
 model.addAttribute("titulo","hola con model");  
 return "index";  
 }  
  
}

<!DOCTYPE html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
<h1 th:text="${titulo}">Hola Spring Boot.</h1>  
<h1>Hola Spring Boot.</h1>  
</body>  
</html>



Usamos app como url principal, también vimos para mapear varias url en el mismo método (sirve para crud)

# 602. Introducción MVC

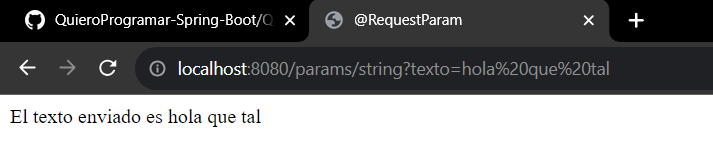
# 603. El objeto Model

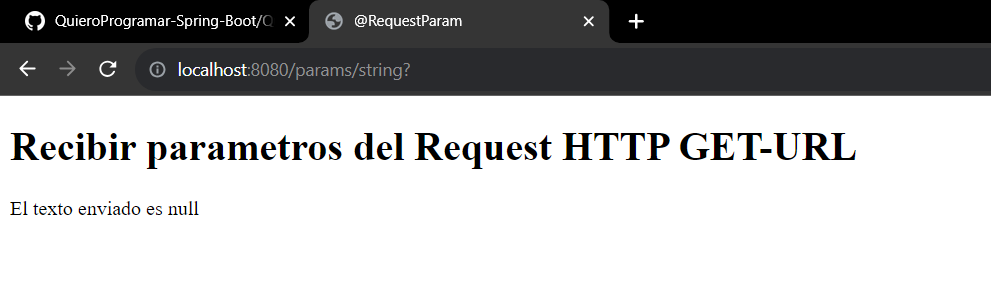
# 604. Directiva if de Thymeleaf

# 605. Directiva each (foreach) de Thymeleaf

# 606. Anotación @ModelAttribute

# 607. Anotacion @RequestParam



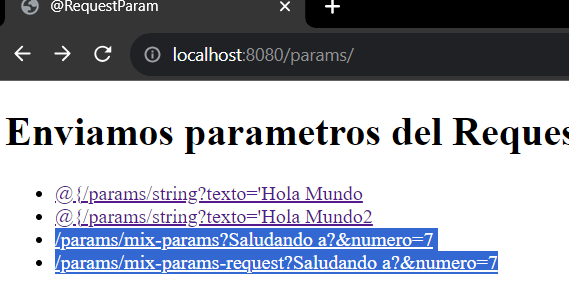


////



# 608. Obtener varios parámetros de la URL con @RequestParam



# 609. Anotación @PathVariable

# 610. Obtener varios parámetros de la ruta con @PathVariable

# 611. Agregando recursos estáticos y estilos CSS personalizados

# 612. Inyectar valores usando la anotación @Value

# 613. Agregando otros archivos properties

# personalizados para los textos

# 614. Retornando redirect y forward como respuesta en métodos del controlador

# 615. Despliegue y ejecución desde terminal (deploy)