

Taller: Creación de una página web usando un motor de renderizado diferente a EJS

Objetivo: Que los estudiantes exploren y utilicen un motor de renderizado distinto al usado en clase (EJS con Node.js), ampliando su conocimiento sobre tecnologías web y su capacidad de adaptación.

1. Elige un motor de renderizado diferente a EJS.

Java + Spring Boot + Thymeleaf

2. Investiga cómo se configura y se usa ese motor.

- **Configuración:**

Thymeleaf se configura automáticamente al crear un proyecto **Spring Boot** con la dependencia del motor de plantillas y opcionalmente el soporte para web.

En Maven, se agrega en el archivo `pom.xml` las dependencias:

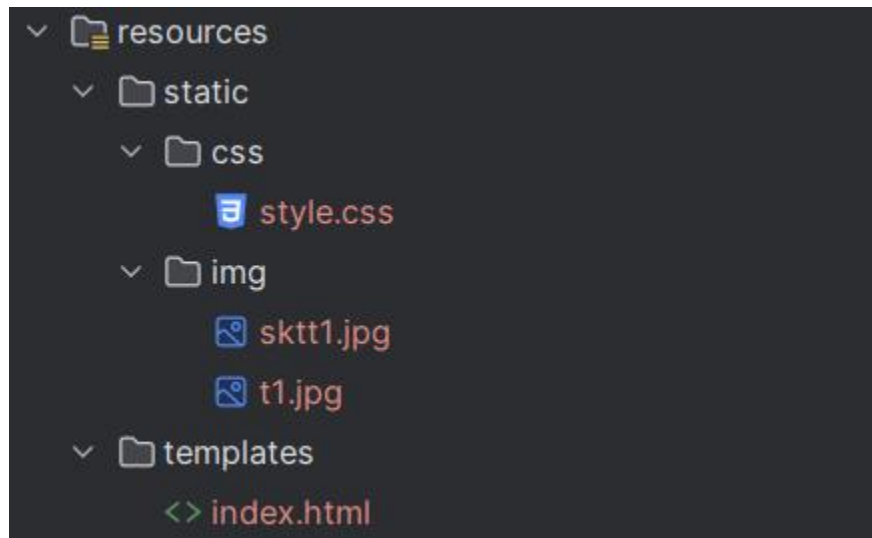
```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Spring Boot detecta la dependencia y configura Thymeleaf de forma predeterminada. Las vistas se ubican en:

`src/main/resources/templates/`

y los archivos estáticos (imágenes, CSS, JS) en:

`src/main/resources/static/`



- **Uso:**

En los controladores (@Controller), se retorna el nombre del archivo HTML sin extensión.
Ejemplo:

```
@Controller
public class HomeController {

    @GetMapping("/{id}", @RequestMapping("/index"))
    public String index(Model model) {
        model.addAttribute("title", "SKT T1 y T1 - Campeonatos Mundiales de League of Legends");
        model.addAttribute("yearsSKT", new int[]{2013, 2015, 2016});
        model.addAttribute("yearsT1", new int[]{2023, 2024, 2025});
        return "index";
    }
}
```

En la plantilla HTML, se utilizan los atributos de Thymeleaf, como th:text, th:each, th:href, etc.:

```
<title th:text="${title}">SKT T1 - Worlds</title>
```

3. Crea una página web sencilla (por ejemplo, una biografía personal, portafolio o página de producto).

4. Presenta tu trabajo explicando:

- Por qué elegiste ese motor.

Lo elegí debido a que ya tenía experiencia utilizando **Spring MVC** y **Thymeleaf** anteriormente, por lo tanto, su implementación me resultó sencilla.

- Qué diferencias encontraste respecto a EJS:

Diferencias principales

- **EJS** se usa con **Node.js** y JavaScript, mientras que **Thymeleaf** se integra con **Java y Spring Boot**.
- **EJS** genera HTML en el servidor con sintaxis JavaScript (`<%= %>`), y **Thymeleaf** usa expresiones propias de Java (`th:text`, `th:each`, etc.).
- **Thymeleaf** puede renderizar tanto en el servidor como directamente en el navegador (modo “natural template”).

- Qué ventajas o desventajas observaste.

Ventajas de Thymeleaf

- Se integra fácilmente con Spring Boot.
- Las plantillas pueden verse correctamente en el navegador sin ejecutar el servidor.
- Buen soporte para enlaces, iteraciones y validación de formularios en Spring MVC.

Desventajas frente a EJS

- Más “verbal” o pesada la sintaxis.
- Menos flexible a comparación de entorno que utilizan JavaScript o Python.
- Requiere conocimiento del ecosistema Java/Spring