

**PRIMER
PROYECTO
THYMELEAF**



NUESTRO IDE

- ▶ **Spring Tool Suite**
(basado en eclipse)

Otras posibilidades

- ▶ IntelliJ (IDEA)
- ▶ Netbeans



SPRING

Conceptos de repaso

Spring Web MVC, Spring Boot, Spring Data
JPA

SPRING **WEB MVC**

- ▶ Framework para implementar aplicaciones web bajo el patrón Modelo-Vista-Controlador
- ▶ Inversión de control e Inyección de dependencias.
- ▶ En la vista, utilizaremos Thymeleaf

SPRING **BOOT**

- ▶ Convención sobre configuración
- ▶ Nos permite configurar ~~rápida~~ **rápidamente** instantáneamente una aplicación Spring (por ejemplo Web + ORM + Base de datos)
- ▶ Configuración *customizable*.

SPRING DATA JPA

- ▶ Una capa de abstracción sobre JPA (Java Persistence API).
- ▶ JPA+Hibernate: nos permite manejar entidades y no tablas, generar DDL, hacer consultas, ...
- ▶ Spring Data JPA añade algunas características: CRUDs automáticos, generación sencilla de consultas, ...

Spring Data JPA
JPA
Hibernate
JDBC
Base de datos

OTROS COMPONENTES

Usaremos algún componente más, aparte de Spring

- ▶ H2 Database
 - ▷ Integrada en el proyecto. 1 fichero
 - ▷ Dependencia Maven
- ▶ Webjars: Bootstrap y jQuery
 - ▷ Recursos estáticos a través de dependencias Maven.
 - ▷ Versión *agnóstica*.

CONFIGURACIÓN

Pasos a seguir

PRIMEROS PASOS

- ▶ *File > New > Spring Starter Project*
- ▶ Rellenar datos iniciales
- ▶ Seleccionar
 - ▷ Web (MVC)
 - ▷ JPA (Data JPA, Hibernate)
 - ▷ H2 (base de datos)
 - ▷ **Thymeleaf**

PRIMEROS PASOS

- ▶ Añadir dependencias de webjars (<https://www.webjars.org/>)
 - ▷ Bootstrap (3.3.X) ([ver](#))
 - ▷ jQuery (3.3.X) ([ver](#))
 - ▷ Versión agnóstica (*locator*) ([ver](#))

En próximas lecciones explicamos cómo *customizar* la configuración.

PLANTILLA BASE

PLANTILLA **BASE**

- ▶ Partimos de un diseño web *base*.
- ▶ Utilizamos una plantilla básica de Bootstrap 3

(ver)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
    *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

IMPLEMENTACIÓN **DE LA PLANTILLA**

IMPLEMENTACIÓN DE LA PLANTILLA

- ▶ *New > File > Others ... > HTML* dentro de la carpeta *src/main/resources/templates*.
- ▶ Copiamos el código fuente.
- ▶ Adaptamos las rutas de la librerías para usar las *webjars*.

```
<link href="css/bootstrap.min.css" rel="stylesheet">  
<link href="/webjars/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>  
<script src="/webjars/jquery/jquery.min.js"></script>
```

```
<script src="js/bootstrap.min.js"></script>  
<script src="/webjars/bootstrap/js/bootstrap.min.js"></script>
```

DE HTML A THYMELEAF

- ▶ Para indicar que este documento HTML es una plantilla de Thymeleaf, añadimos un espacio de nombres con prefijo **th**.

```
<!DOCTYPE html>
```

```
<html lang="es" xmlns:th="http://www.thymeleaf.org">
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    ....
```


IMPLEMENTACIÓN DEL CONTROLADOR

CONTROLADOR BÁSICO

@Controller

```
public class MainController {
```

Ruta (o rutas) que podemos introducir en el navegador




```
    @GetMapping({"/", "/welcome"})
```

```
    public String welcome(@RequestParam(name="nombre",  
required=false,
```

```
    defaultValue="Mundo") String nombre, Model model) {
```

Información que estamos enviando a la plantilla



```
        model.addAttribute("nombre", nombre);
```

```
        return "index";
```

```
    }
```

Ruta de la plantilla

Prefijo por defecto: /src/main/resources/templates

Sufijo por defecto: .html

Ruta completa: /src/main/resources/templates/index.html

```
}
```

IMPLEMENTACIÓN DE LA PLANTILLA (2)

PLANTILLA **BASE**

- Actualizamos la plantilla recogiendo el valor del controlador.

