11ª Emisión

DIPLOMADO

Desarrollo de Sistemas con Tecnología Java

Módulo 7

Persistencia con Spring Data

Dr. Omar Mendoza González

omarmendoza564@aragon.unam.mx

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC

DGTIC

50 AÑOS Educación Continua 1971 - 2021

Spring MVC

- Permite crear aplicaciones sobre modelo MVC que generen páginas HTML sencillas en las cuales nosotros podamos cargar los contenidos que necesitemos de forma sencilla pudiendo integrarse con otras tecnologías como jQuery , React o Vue a la hora de generar aplicaciones modernas y flexibles.

- Para poder desarrollar una aplicación sobre Spring MVC es suficiente con descargar un proyecto de Spring Boot que contenga dos de las dependencias más clásicas como Starters
  - ***Spring Web***
  - ***Thymeleaf***

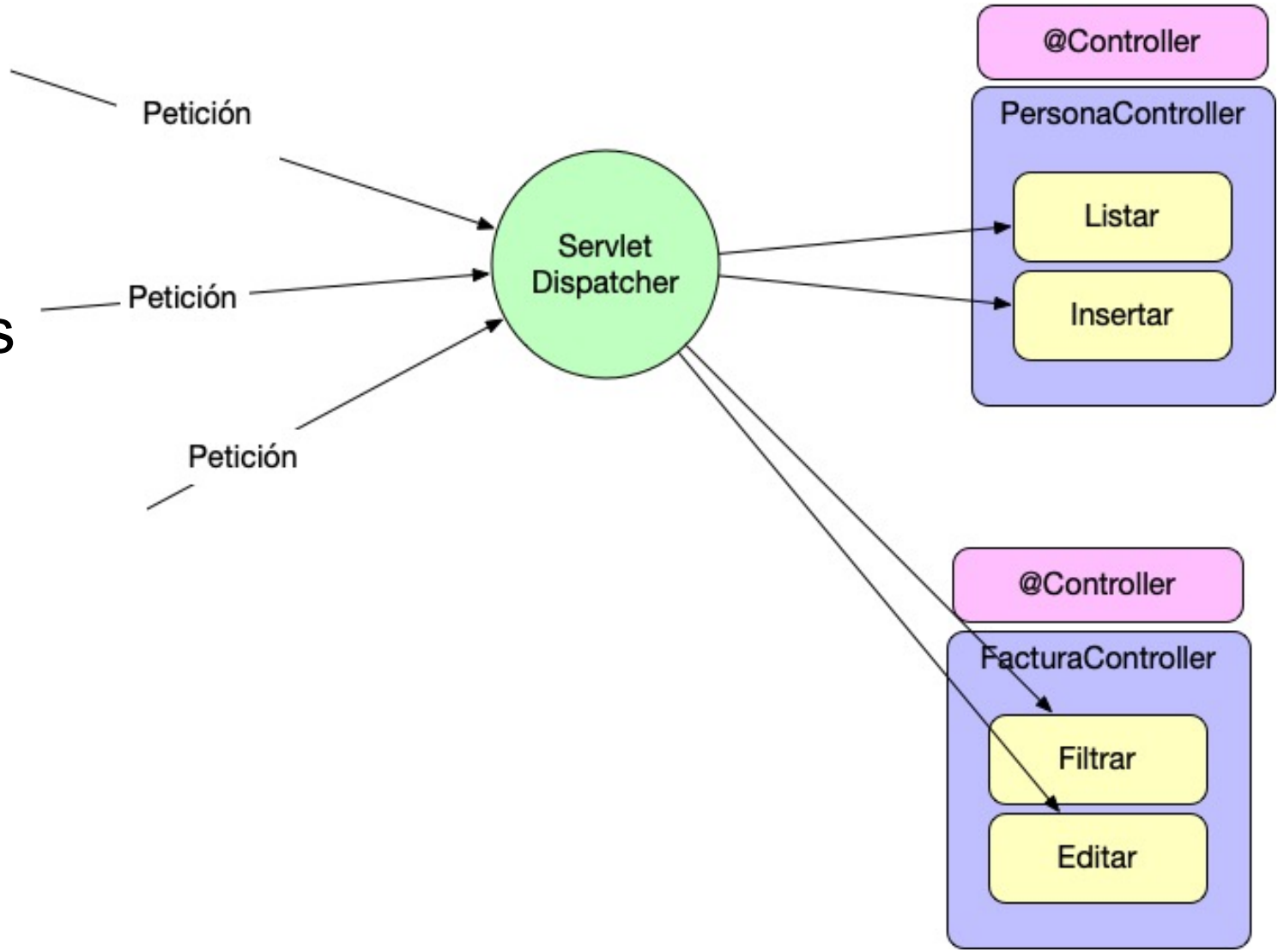
Spring MVC

- Es un framework MVC que trabaja sobre todo con el concepto de Controlador Frontal (***FrontController***) que es el encargado de soportar todas las peticiones Web y redirigirlas a los componentes que sean necesarios .

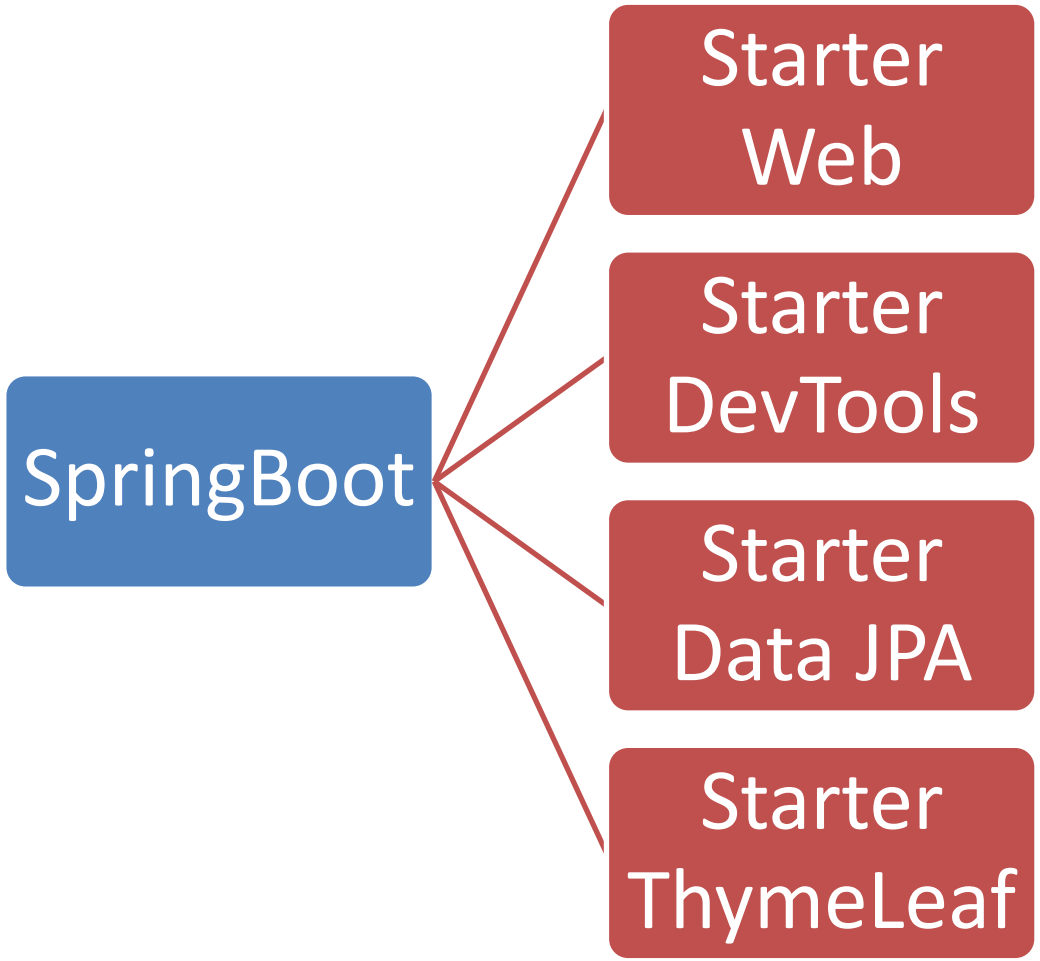
- En este caso el controlador de Spring se denomina ***ServletDispatcher*** y viene configurado por defecto por Spring Boot.

Spring MVC

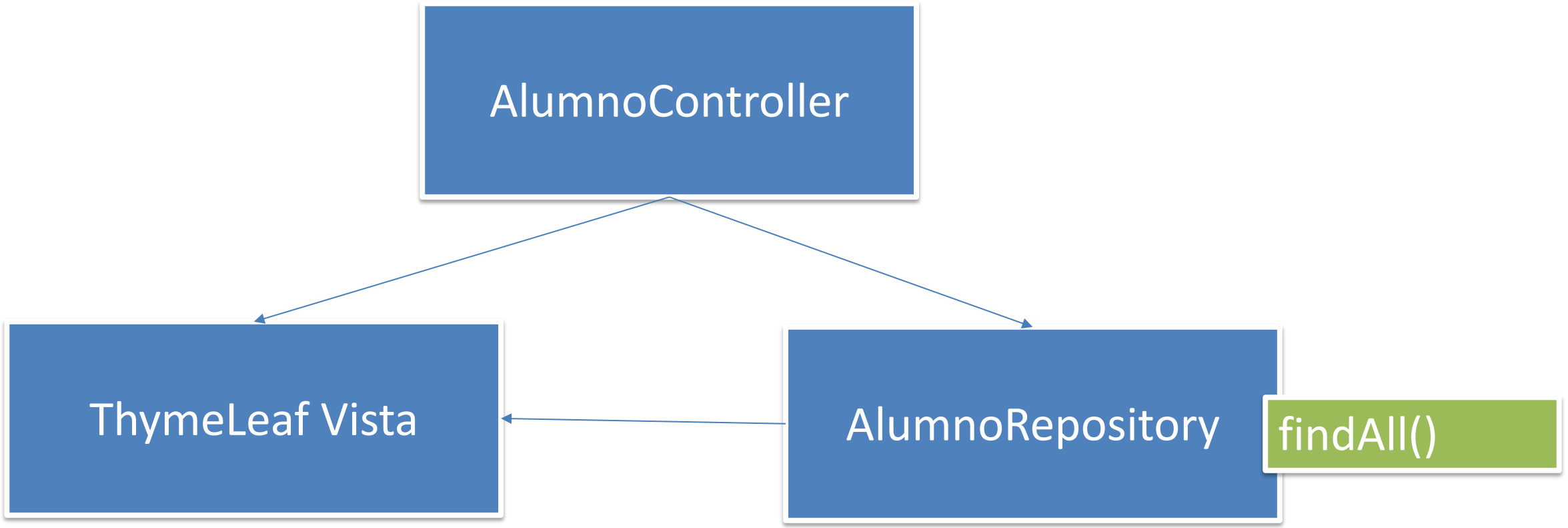
- Existe un único ***Servlet*** que recibe todas las peticiones (GET,POST etc) y según la información que recibe delega en el método adecuado de cada uno de los controladores.



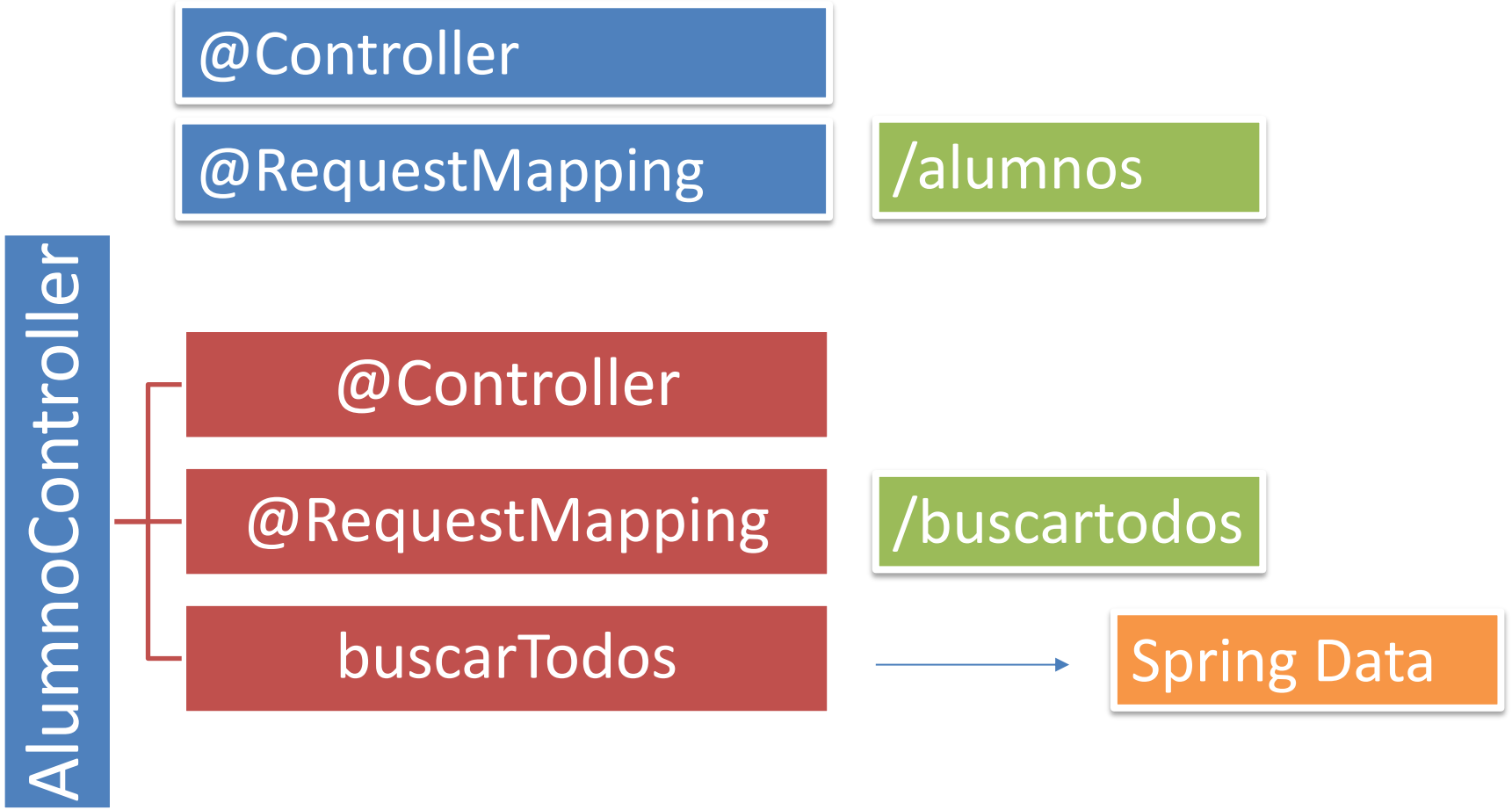
Spring Data y Spring MVC



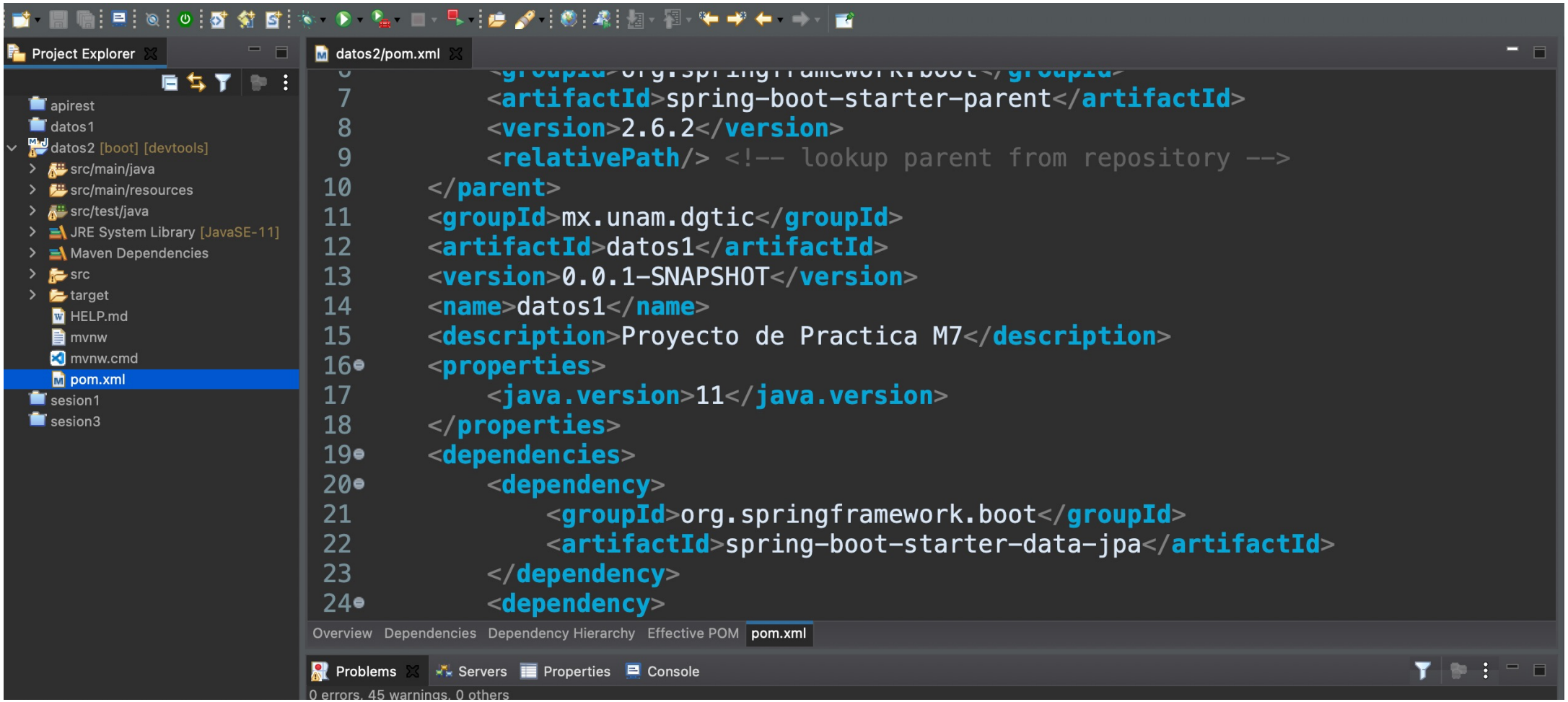
Spring Data y Spring MVC



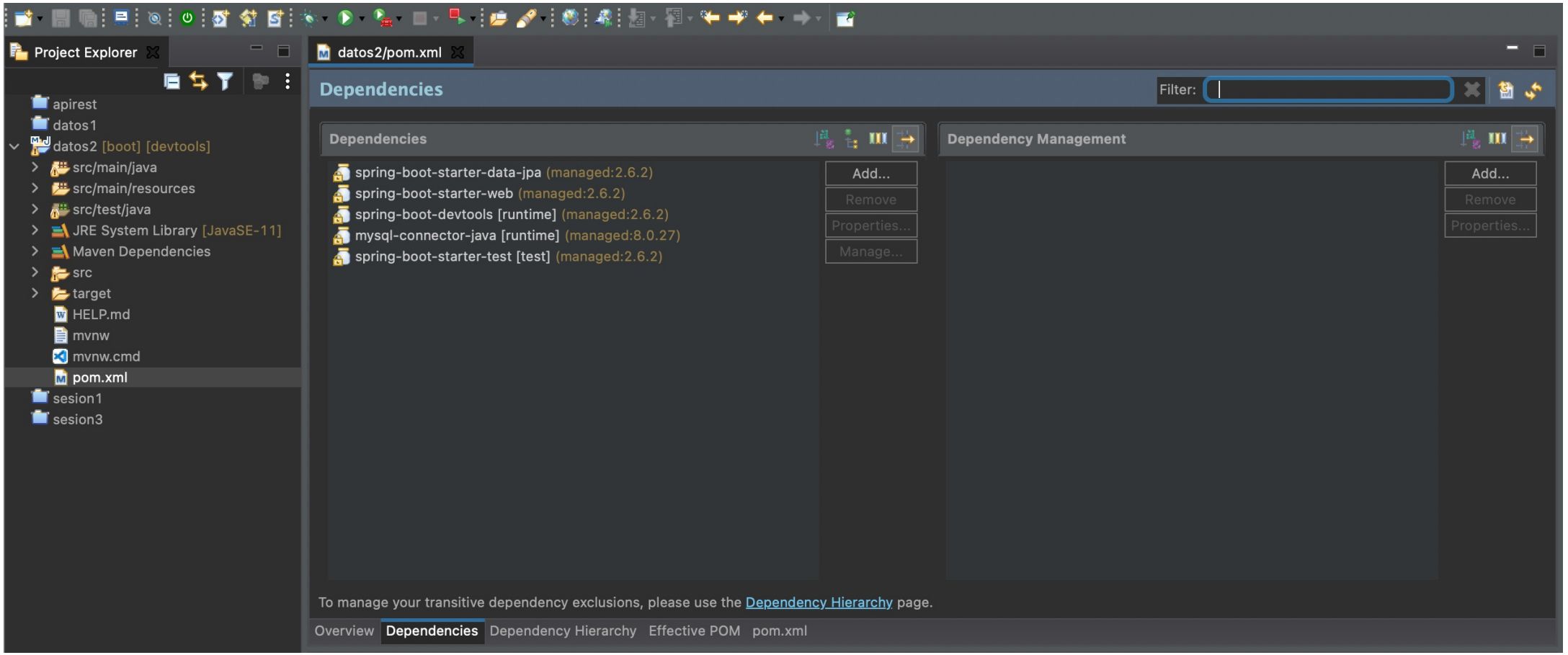
Spring MVC


@Controller
@RequestMapping
/alumnos
AlumnoController
@Controller
@RequestMapping
/buscartodos
buscarTodos
Spring Data

pom.xml


Project Explorer
apirest
datos1
datos2 [boot] [devtools]
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
sesion1
sesion3
datos2/pom.xml
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.6.2</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>mx.unam.dgtic</groupId>
<artifactId>datos1</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>datos1</name>
<description>Proyecto de Practica M7</description>
<properties>
<java.version>11</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml
Problems
Servers
Properties
Console
0 errors, 45 warnings, 0 others

pom.xml


Project Explorer
apirest
datos1
datos2 [boot] [devtools]
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
sesion1
sesion3
datos2/pom.xml
Dependencies
Filter:
Dependencies
spring-boot-starter-data-jpa (managed:2.6.2)
spring-boot-starter-web (managed:2.6.2)
spring-boot-devtools [runtime] (managed:2.6.2)
mysql-connector-java [runtime] (managed:8.0.27)
spring-boot-starter-test [test] (managed:2.6.2)
Add...
Remove
Properties...
Manage...
Dependency Management
Add...
Remove
Properties...
To manage your transitive dependency exclusions, please use the Dependency Hierarchy page.
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml

pom.xml

https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-thymeleaf/2.6.2

Spring Boot Starter Thymeleaf » 2.6.2

Starter for building MVC web applications using Thymeleaf views

License	Apache 2.0
Organization	Pivotal Software, Inc.
HomePage	https://spring.io/projects/spring-boot
Date	(Dec 22, 2021)
Files	pom (2 KB) jar (4 KB) View All
Repositories	Central
Used By	532 artifacts

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

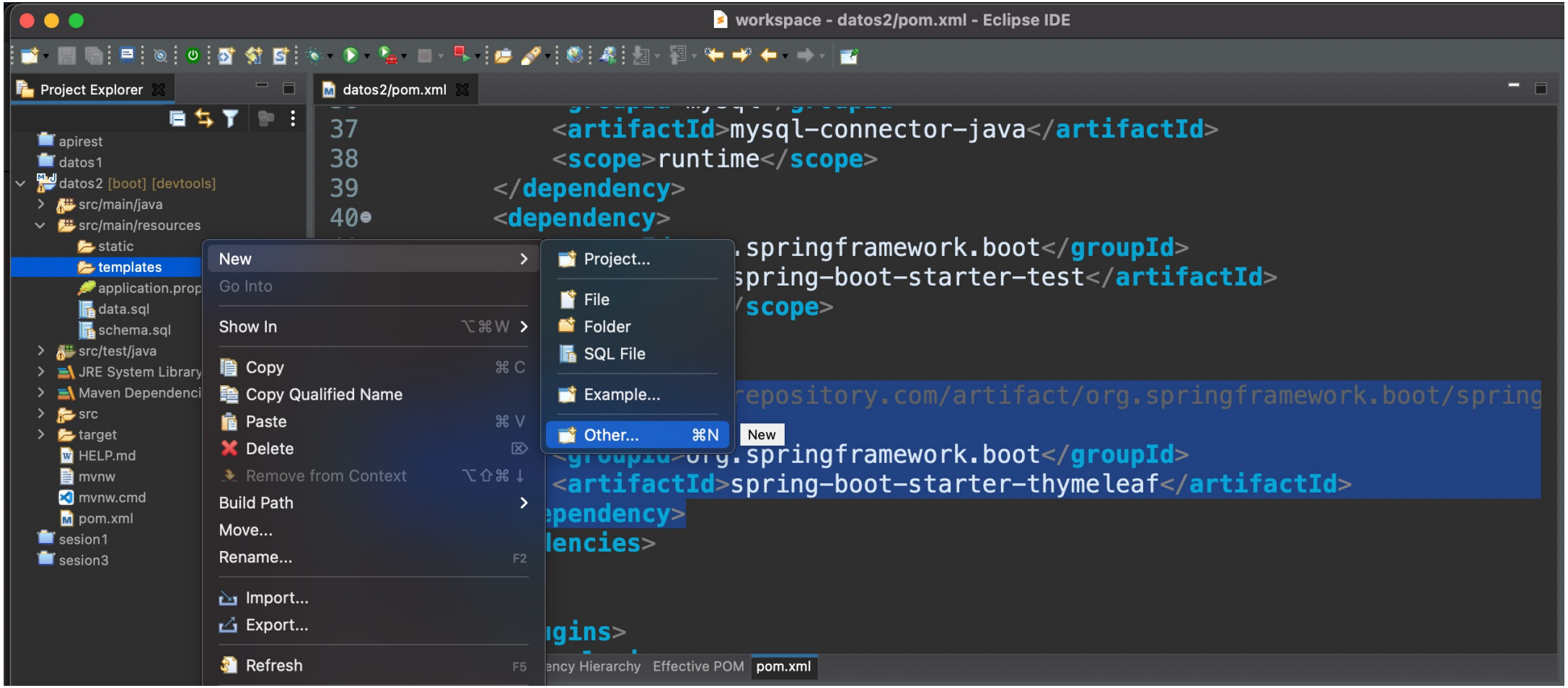
```
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-thymeleaf -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
    <version>2.6.2</version>
</dependency>
```

Include comment with link to declaration

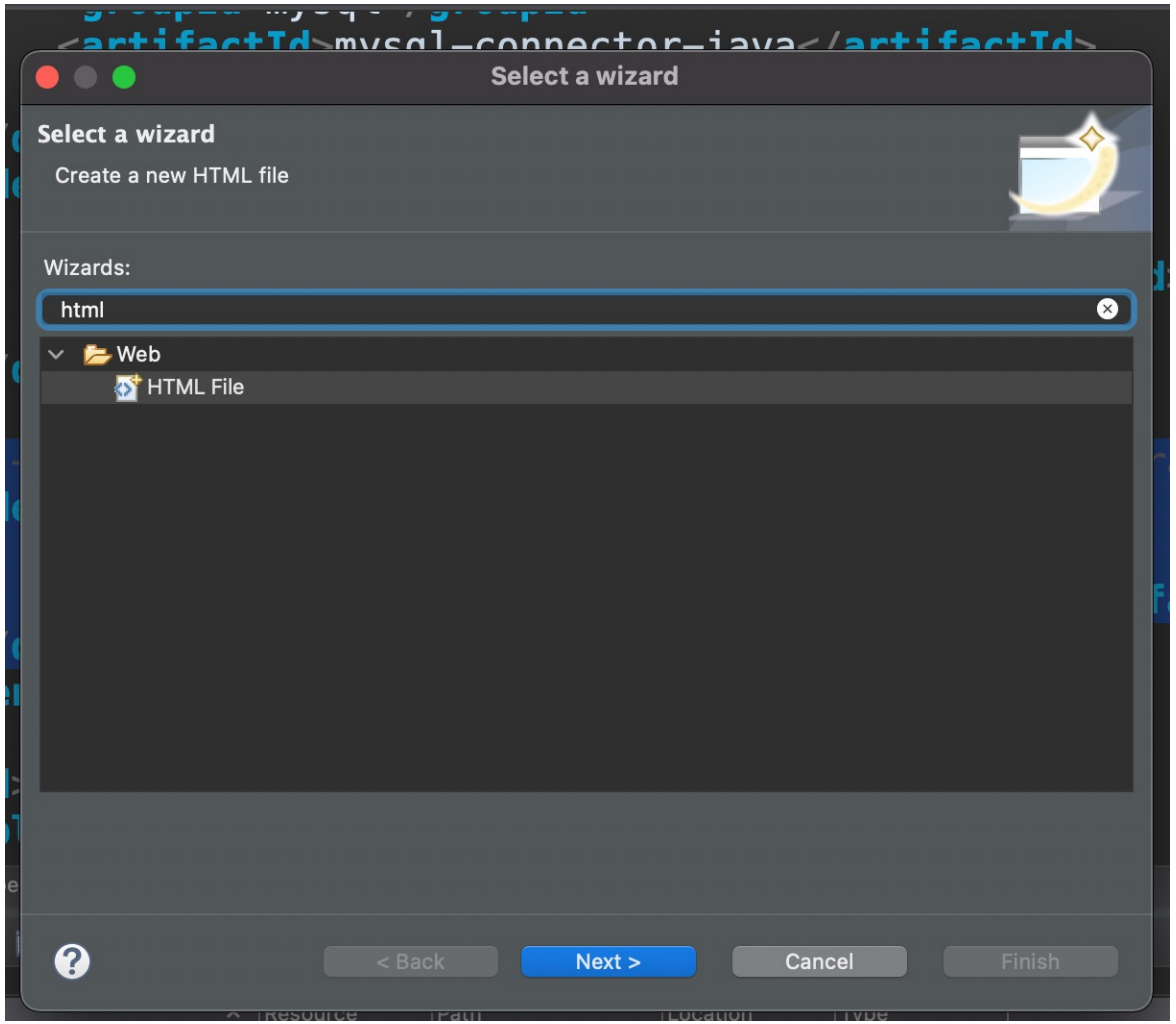
pom.xml

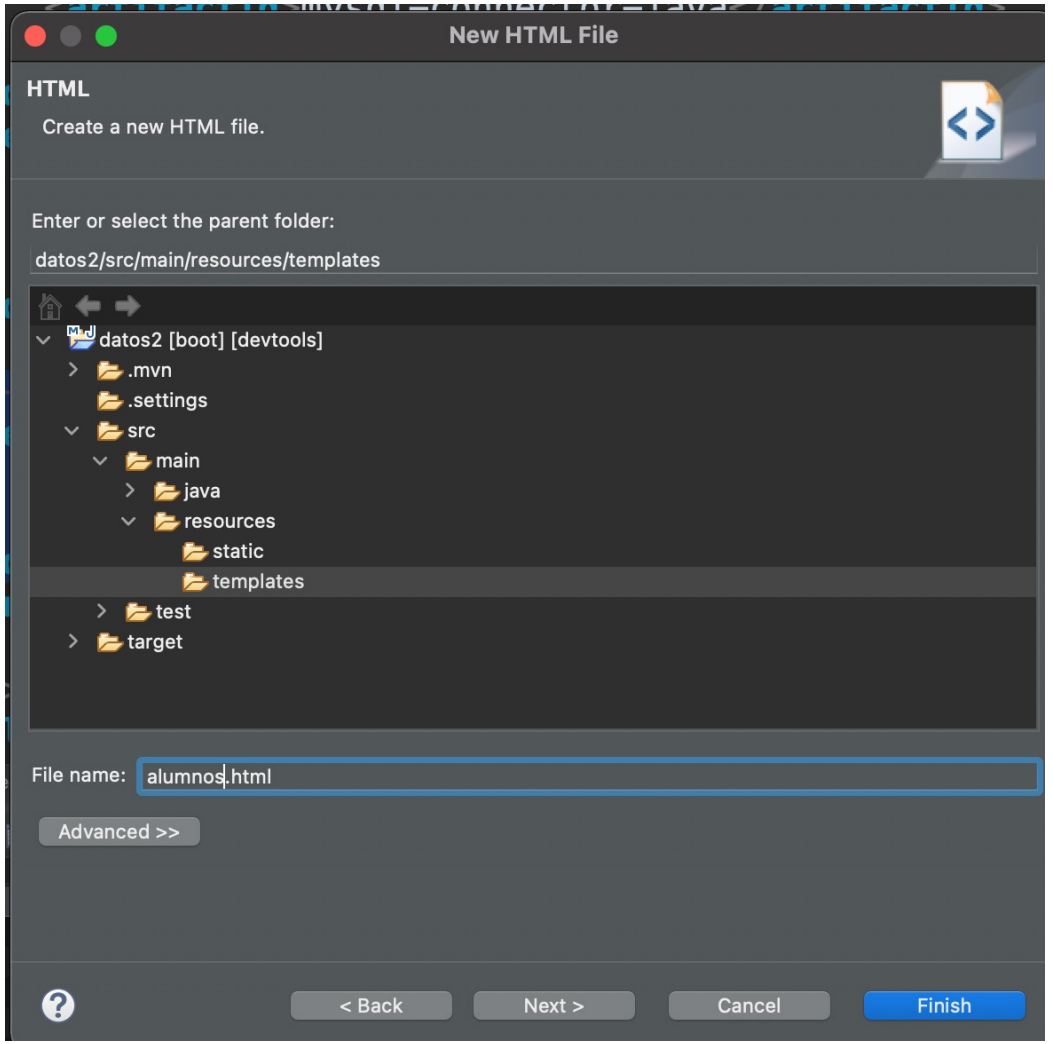
```
<!--
https://mvnrepository.com/artifact/org.springframework.boot/spring-
boot-starter-thymeleaf -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

Agregar archivo html

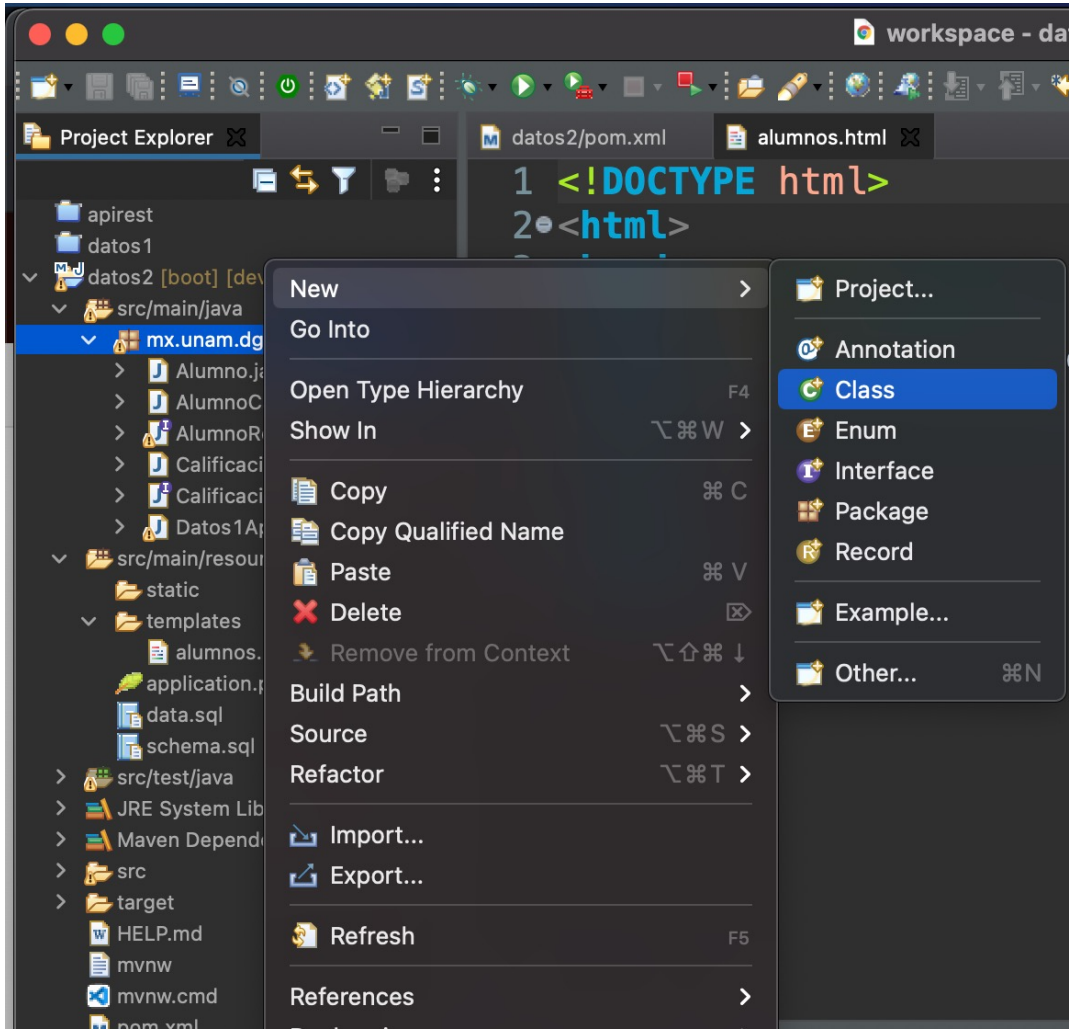


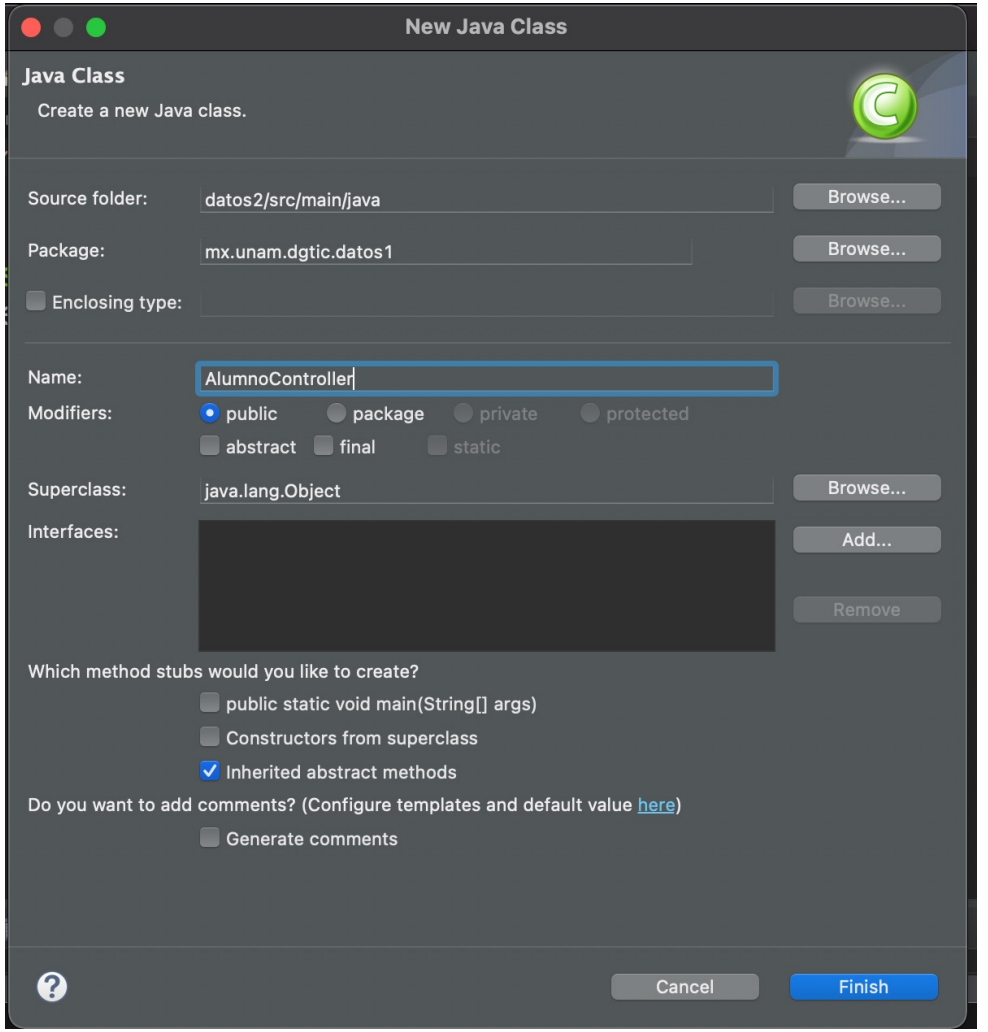
Agregar archivo html





Agregar Clase AlumnoController





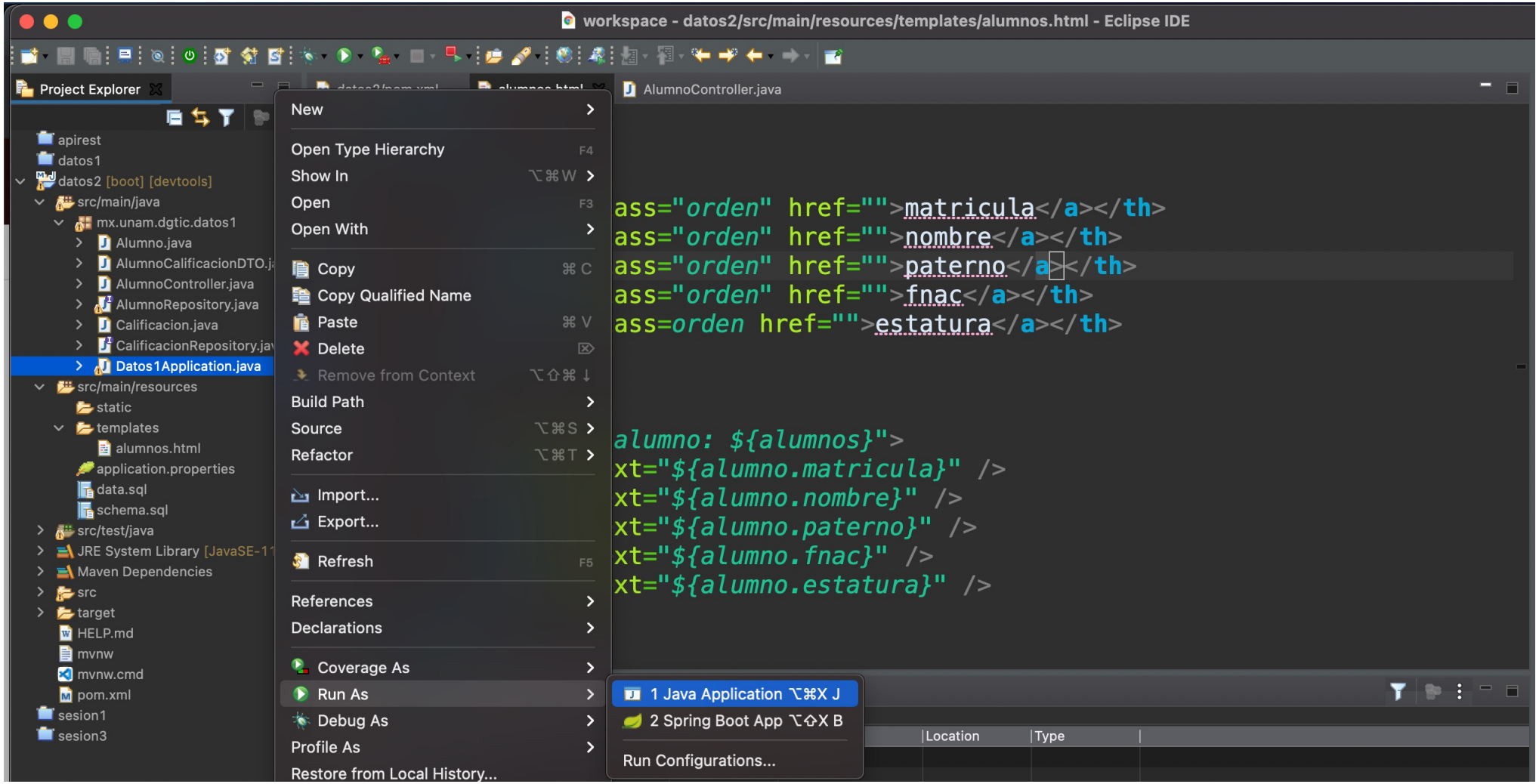
Agregar Clase AlumnoController

```java
@Controller
@RequestMapping("/alumnos")
public class AlumnoController {

    @Autowired
    AlumnoRepository repositorioAlumno;

    @RequestMapping(value="buscartodos")
    public String buscarTodos(Model modelo) {
        // obteniendo datos desde spring data
        Iterable<Alumno> alumnos = null;
        alumnos = repositorioAlumno.findAll();

        modelo.addAttribute("alumnos", alumnos );
        return "alumnos";
    }
}
```

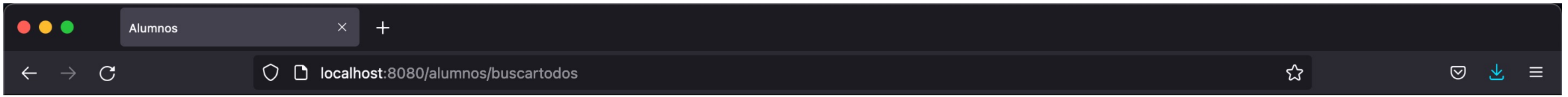
alumnos.html

```
<table>
    <tr>

        <th><a class="orden" href="">matricula</a></th>
        <th><a class="orden" href="">nombre</a></th>
        <th><a class="orden" href="">paterno</a></th>
        <th><a class="orden" href="">fnac</a></th>
        <th><a class=orden href="">estatura</a></th>
    </tr>


    <tr th:each="alumno: ${alumnos}">
        <td th:text="${alumno.matricula}" />
        <td th:text="${alumno.nombre}" />
        <td th:text="${alumno.paterno}" />
        <td th:text="${alumno.fnac}" />
        <td th:text="${alumno.estatura}" />
    </tr>
</table>
```

Lanzar la aplicacion

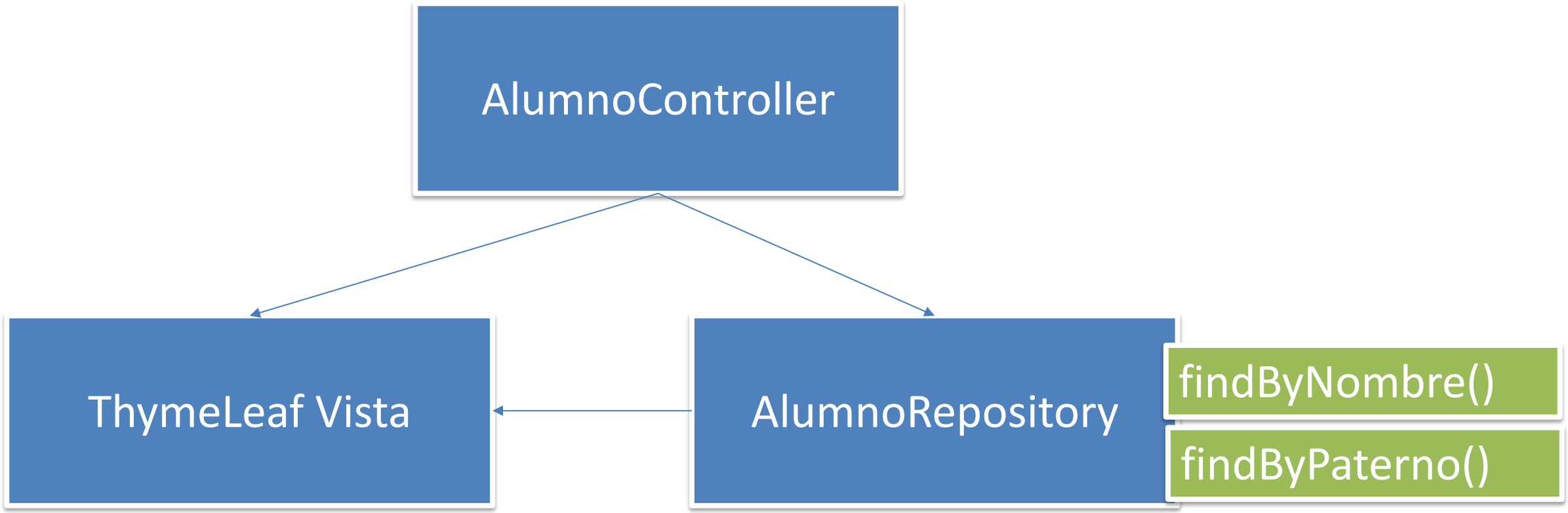


Ver resultado

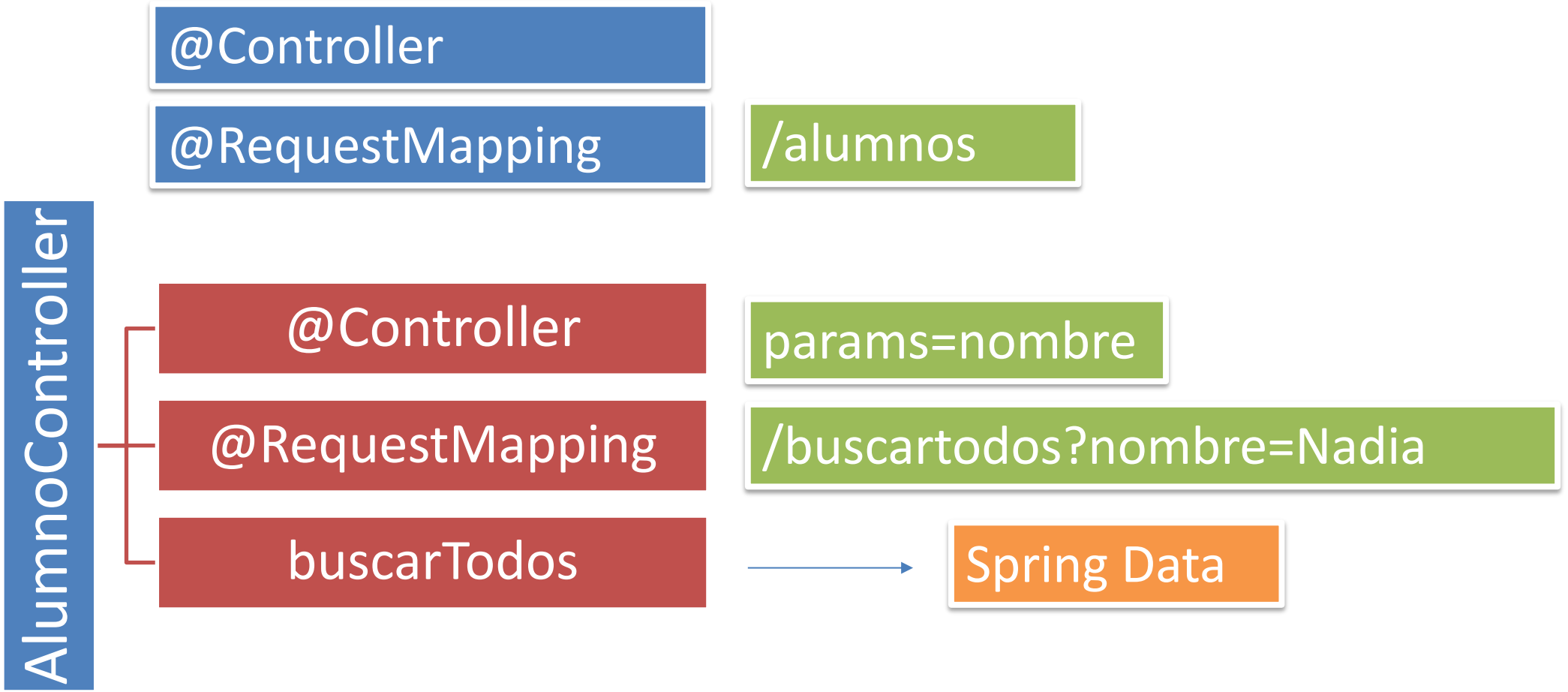


matricula	nombre	paterno	fnac	estatura
1A	Juan	Lopez	2001-01-01 00:00:00.0	1.78
2A	Nadia	Perez	2001-01-10 00:00:00.0	1.56
3B	Perla	Calles	2001-01-20 00:00:00.0	1.6
4A	Carlos	Madero	2001-01-01 00:00:00.0	1.68
5A	Javier	Amaro	2001-02-10 00:00:00.0	1.75
6C	Jesus	Garcia	2001-03-20 00:00:00.0	1.65
7B	Gema		2001-03-20 00:00:00.0	1.53

Filtrado


AlumnoController
ThymeLeaf Vista
AlumnoRepository
findByNombre()
findByPaterno()

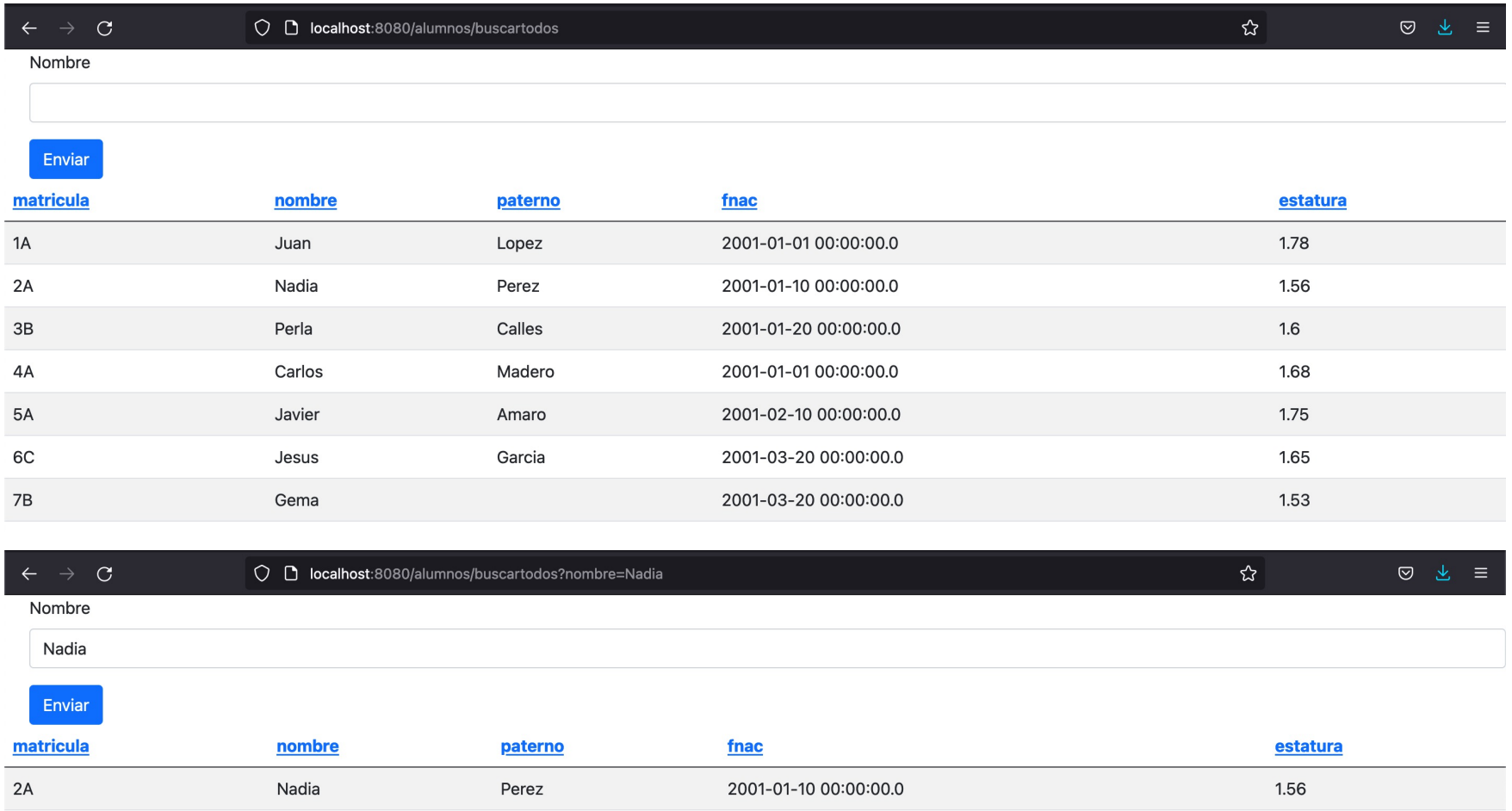
Filtrado


@Controller
@RequestMapping
/alumnos
AlumnoController
@Controller
params=nombre
@RequestMapping
/buscartodos?nombre=Nadia
buscarTodos
Spring Data

Filtrado findByNombre

```
@RequestMapping(value="buscartodos", params = "nombre")
public String buscarTodosPorNombre(String nombre, Model modelo) {
    // obteniendo datos desde spring data
    Iterable<Alumno> alumnos = null;
    alumnos = repositorioAlumno.findByNombre(nombre);

    modelo.addAttribute("alumnos", alumnos );
    return "alumnos";
}
```

Filtrado findByNombre



Filtrado findByPaterno

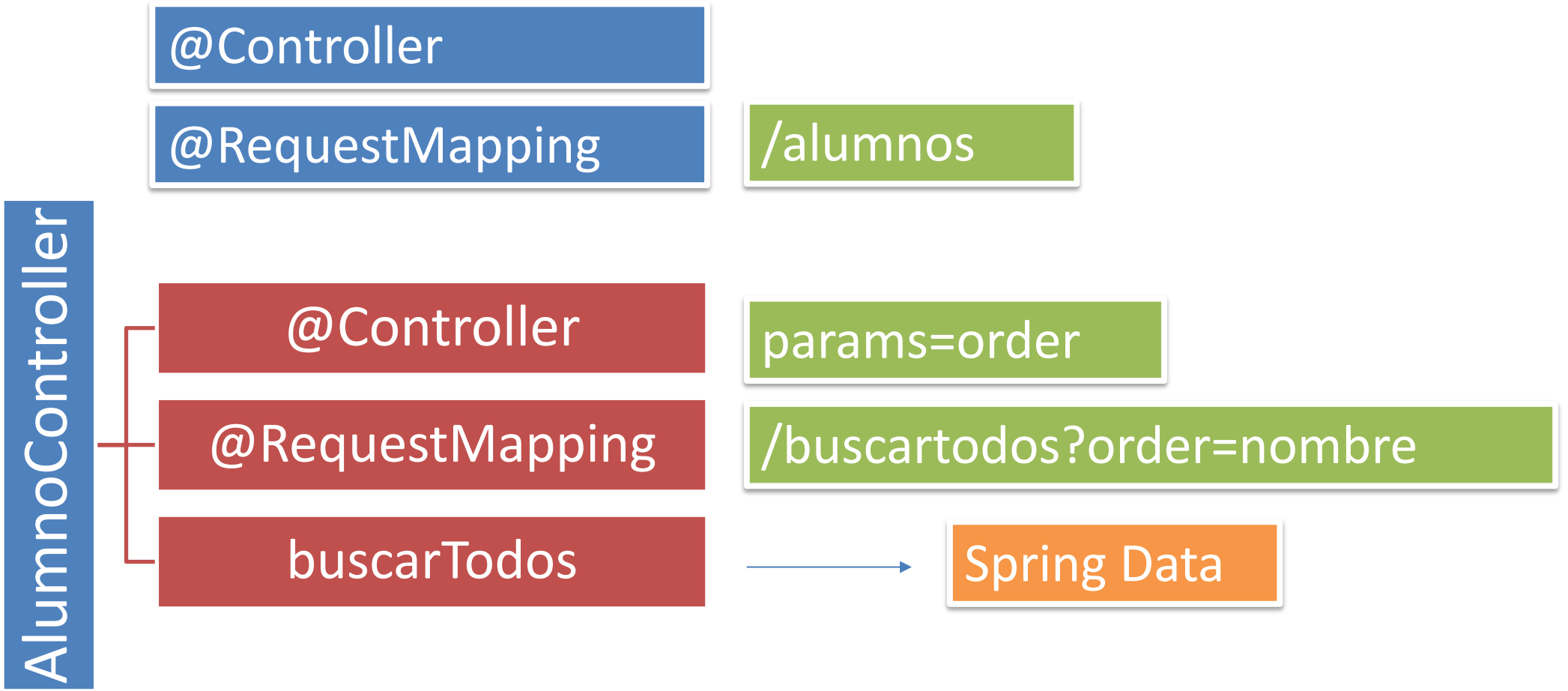
```
@RequestMapping(value="buscartodos", params = "paterno")
public String buscarTodosPorPaterno(String paterno, Model modelo) {
    // obteniendo datos desde spring data
    Iterable<Alumno> alumnos = null;
    alumnos = repositorioAlumno.findByPaterno(paterno);

    modelo.addAttribute("alumnos", alumnos );
    return "alumnos";
}
```

Filtrado findByNombreAndPaterno

```
@RequestMapping(value="buscartodos", params = {"nombre", "paterno"})
public String buscarTodosPorNombreyPaterno(
        String nombre, String paterno, Model modelo) {
    // obteniendo datos desde spring data
    Iterable<Alumno> alumnos = null;
    alumnos = repositorioAlumno.findByNombreAndPaterno(nombre, paterno);

    modelo.addAttribute("alumnos", alumnos );
    return "alumnos";
}
```

Filtrado findByNombreAndPaterno

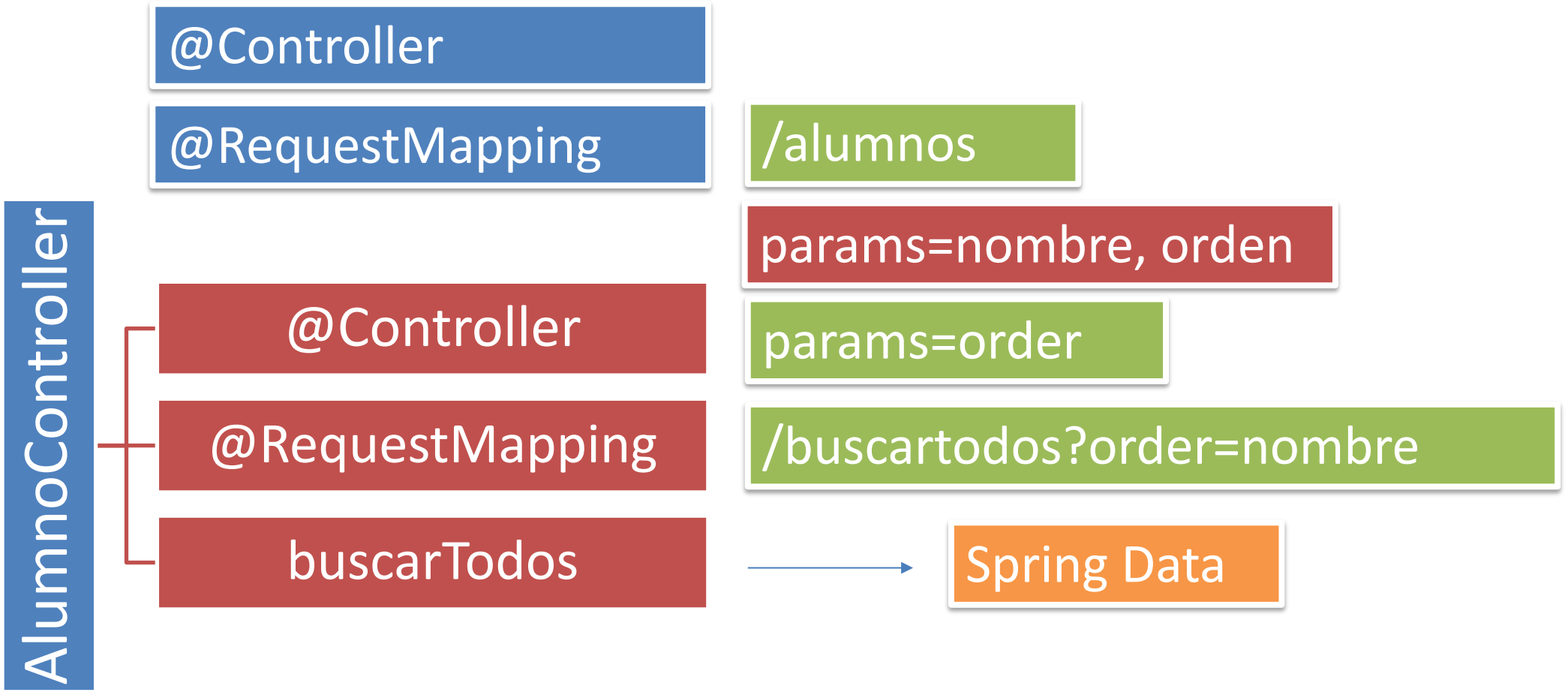
Order


@Controller
@RequestMapping
/alumnos
AlumnoController
@Controller
params=order
@RequestMapping
/buscartodos?order=nombre
buscarTodos
Spring Data

AlumnoController

```java
@RequestMapping(value="buscartodos", params = "orden")
public String buscarTodos(Model modelo,
        @RequestParam(name="orden", defaultValue="matricula") String orden)
    // obteniendo datos desde spring data
    Iterable<Alumno> alumnos = null;
    alumnos = repositorioAlumno.findAll(Sort.by(orden));

    modelo.addAttribute("alumnos", alumnos );
    return "alumnos";
}
```

Order


@Controller
@RequestMapping
/alumnos
params=nombre, orden
AlumnoController
@Controller
params=order
@RequestMapping
/buscartodos?order=nombre
buscarTodos
Spring Data

Contacto

Dr. Omar Mendoza González

omarmendoza564@aragon.unam.mx