

Clase 10

SpringBoot + WEB + REST

SpringBoot

- Librería de código abierto
- Basado en spring core (extiende)
- Simplifica el desarrollo configurando el proyecto inicial
- A través de módulos se configuran otras librerías (starter pack)
- Servidor web embebido (incluido en el paquete)
- Herramientas de configuración y monitoreo
- Configuración por defecto y simplificada
- Configuración personalizada simple

Spring Initializr



Project

☐ Gradle - Groovy

☐ Gradle - Kotlin

☒ Maven

Language

☒ Java

☐ Kotlin

☐ Groovy

Spring Boot

☐ 3.1.1 (SNAPSHOT)

☒ 3.1.0

☐ 3.0.8 (SNAPSHOT)

☐ 3.0.7

☐ 2.7.13 (SNAPSHOT)

☐ 2.7.12

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java

☐ 20

☒ 17

☐ 11

☐ 8

Dependencies

ADD DEPENDENCIES... ⌘ + B

Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Web

WEB

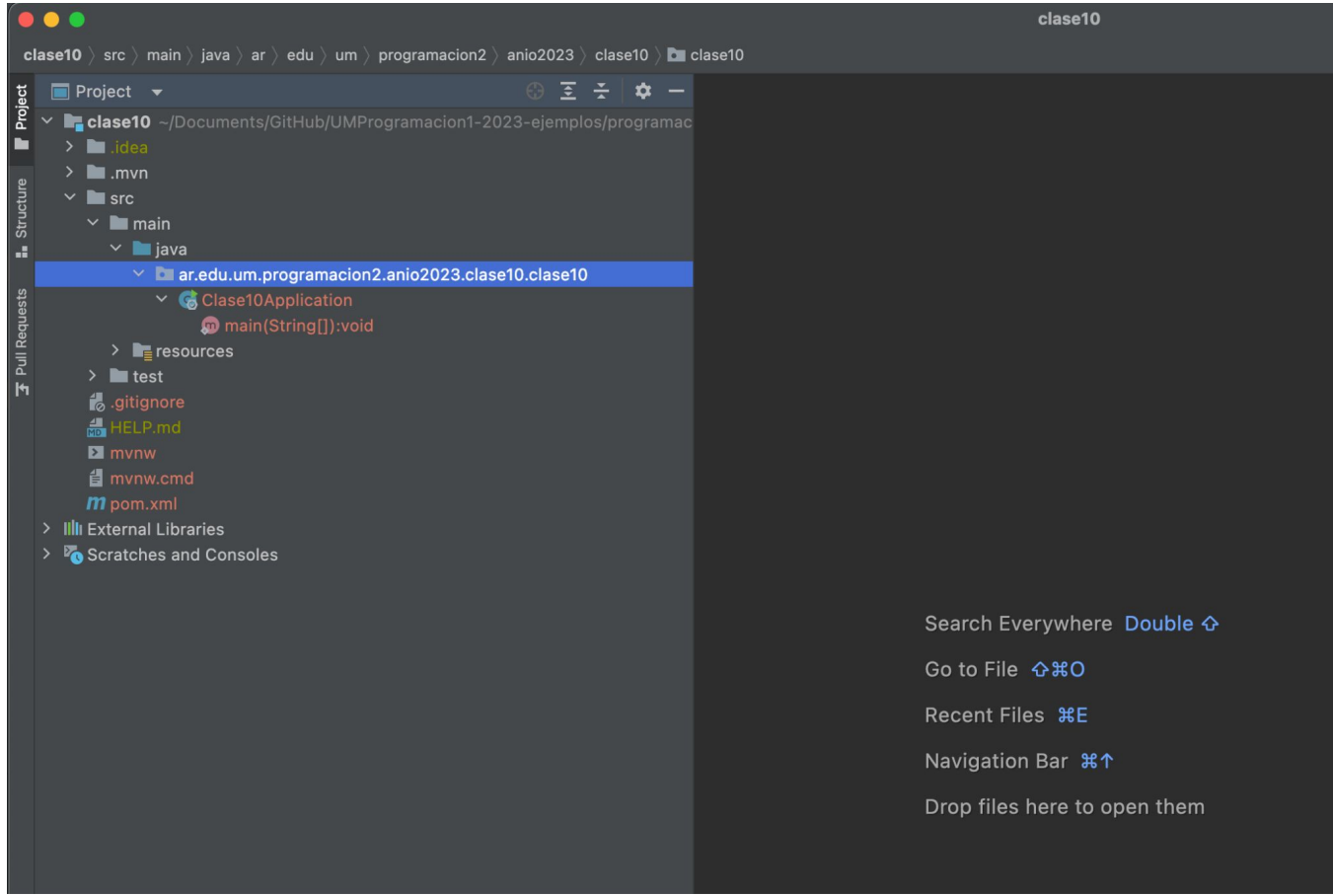
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring HATEOAS

WEB

Eases the creation of RESTful APIs that follow the HATEOAS principle when working with Spring / Spring MVC.

Proyecto



pom.xml

```
<parent>
  <groupId>org.springframework.boot </groupId>
  <artifactId>spring-boot-starter-parent </artifactId>
  <version>3.1.0</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>ar.edu.um.programacion2.anio2023.clase10 </groupId>
<artifactId>clase10</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>clase10</name>
<description>Proyecto web</description>
<properties>
  <java.version>17</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot </groupId>
    <artifactId>spring-boot-starter-hateoas </artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot </groupId>
    <artifactId>spring-boot-starter-web </artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot </groupId>
    <artifactId>spring-boot-starter-test </artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Proyecto

```
> Maven: ch.qos.logback:logback-classic:1.4.7
> Maven: ch.qos.logback:logback-core:1.4.7
> Maven: com.fasterxml.jackson.core:jackson-annotations:2.15.0
> Maven: com.fasterxml.jackson.core:jackson-core:2.15.0
> Maven: com.fasterxml.jackson.core:jackson-databind:2.15.0
> Maven: com.fasterxml.jackson.datatype:jackson-datatype-jdk8:2.15.0
> Maven: com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.15.0
> Maven: com.fasterxml.jackson.module:jackson-module-parameter-naming:2.15.0
> Maven: com.jayway.jsonpath:json-path:2.8.0
> Maven: com.vaadin.external.google:android-json:0.0.20131108.vaadin1
> Maven: io.micrometer:micrometer-commons:1.11.0
> Maven: io.micrometer:micrometer-observation:1.11.0
> Maven: jakarta.activation:jakarta.activation-api:2.1.2
> Maven: jakarta.annotation:jakarta.annotation-api:2.1.1
> Maven: jakarta.xml.bind:jakarta.xml.bind-api:4.0.0
> Maven: net.bytebuddy:byte-buddy:1.14.4
> Maven: net.bytebuddy:byte-buddy-agent:1.14.4
> Maven: net.minidev:accessors-smart:2.4.9
> Maven: net.minidev:json-smart:2.4.10
> Maven: org.apache.logging.log4j:log4j-api:2.20.0
> Maven: org.apache.logging.log4j:log4j-to-slf4j:2.20.0
> Maven: org.apache.tomcat.embed:tomcat-embed-core:10.1.8
> Maven: org.apache.tomcat.embed:tomcat-embed-el:10.1.8
> Maven: org.apache.tomcat.embed:tomcat-embed-websocket:10.1.8
> Maven: org.apiguardian:apiguardian-api:1.1.2
> Maven: org.assertj:assertj-core:3.24.2
> Maven: org.hamcrest:hamcrest:2.2
> Maven: org.junit.jupiter:junit-jupiter:5.9.3
> Maven: org.junit.jupiter:junit-jupiter-api:5.9.3
> Maven: org.junit.jupiter:junit-jupiter-engine:5.9.3
> Maven: org.junit.jupiter:junit-jupiter-params:5.9.3
> Maven: org.junit.platform:junit-platform-commons:1.9.3
> Maven: org.junit.platform:junit-platform-engine:1.9.3
> Maven: org.mockito:mockito-core:5.3.1
> Maven: org.mockito:mockito-junit-jupiter:5.3.1
```

```
> Maven: org.objenesis:objenesis:3.3
> Maven: org.opentest4j:opentest4j:1.2.0
> Maven: org.ow2.asm:asm:9.3
> Maven: org.projectlombok:lombok:1.18.26
> Maven: org.skyscreamer:jsonassert:1.5.1
> Maven: org.slf4j:jul-to-slf4j:2.0.7
> Maven: org.slf4j:slf4j-api:2.0.7
> Maven: org.springframework.boot:spring-boot: Project library
> Maven: org.springframework.boot:spring-boot-autoconfigure:3.1.0
> Maven: org.springframework.boot:spring-boot-starter:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-hateoas:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-json:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-logging:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-test:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-tomcat:3.1.0
> Maven: org.springframework.boot:spring-boot-starter-web:3.1.0
> Maven: org.springframework.boot:spring-boot-test:3.1.0
> Maven: org.springframework.boot:spring-boot-test-autoconfigure:3.1.0
> Maven: org.springframework.hateoas:spring-hateoas:2.1.0
> Maven: org.springframework.plugin:spring-plugin-core:3.0.0
> Maven: org.springframework:spring-aop:6.0.9
> Maven: org.springframework:spring-beans:6.0.9
> Maven: org.springframework:spring-context:6.0.9
> Maven: org.springframework:spring-core:6.0.9
> Maven: org.springframework:spring-expression:6.0.9
> Maven: org.springframework:spring-jcl:6.0.9
> Maven: org.springframework:spring-test:6.0.9
> Maven: org.springframework:spring-web:6.0.9
> Maven: org.springframework:spring-webmvc:6.0.9
> Maven: org.xmlunit:xmlunit-core:2.9.1
> Maven: org.yaml:snakeyaml:1.33
```

SpringBoot - Arranque

Clase por donde arranca la aplicación

```
@SpringBootApplication
public class Clase10Application {

    public static void main(String[] args) {
        SpringApplication.run(Clase10Application.class, args);
    }
}
```

Controlador

- Registra en la aplicación spring una URL para acceder desde un cliente web
- Listo para registrar puntos de entrada
- Manipula entrada y salida de datos
- Singleton similar a `@Service` y `@Configuration`
- Multihilo

```
@Controller
@RequestMapping("/prueba/")
public class PruebaController {

    @GetMapping("funcion1")
    @ResponseBody
    public String funcion1() {
        return "llamado funcion 1";
    }
}
```


Controlador REST

```
@RestController
@RequestMapping ("/persona/")
public class PersonaController {

    @Autowired
    PersonaRepositorio personas;

    @GetMapping("/{codigo}")
    public Persona get(@PathVariable int codigo) {
        Persona p = this.personas.get(codigo);
        return p;
    }

    @GetMapping
    public List<Persona> getAll() {
        List<Persona> personas = this.personas.getAll();
        return personas;
    }

    @PostMapping
    public Persona post(@RequestBody Persona persona) {
        this.personas.add(persona);
        return persona;
    }
}
```

Servicio - Interfaz

```
public interface PersonaRepositorio {  
    public Persona get(int codigo);  
    public void add(Persona persona);  
    public void remove(int codigo);  
    public void remove(Persona p);  
    public List<Persona> getAll();  
}
```

Servicio

```
@Service
public class PersonaMemoriaService implements PersonaRepositorio {
    protected Map<Integer,Persona> personas;

    public PersonaMemoriaService () { this.personas = new HashMap<>(); }

    @Override
    public Persona get(int codigo) {
        Persona p = this.personas.get(codigo);
        return p;
    }

    @Override
    public void add(Persona p) {
        this.personas.put(p.getCodigo(), p);
    }

    @Override
    public List<Persona> getAll () {
        List<Persona> personas;
        personas =
this.personas.values().stream().collect(Collectors.toCollection(ArrayList::new));
        return personas;
    }
}
```

Configuración

```
@Data
@Configuration
@ConfigurationProperties (prefix = "fernando", ignoreUnknownFields = true)
public class PruebaConfiguration {
    protected String url;
    protected String puerto;
    protected Persona pers;
}
```

Configuración (Annotation Processor)

pom.xml

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <optional>true</optional>
</dependency>
```

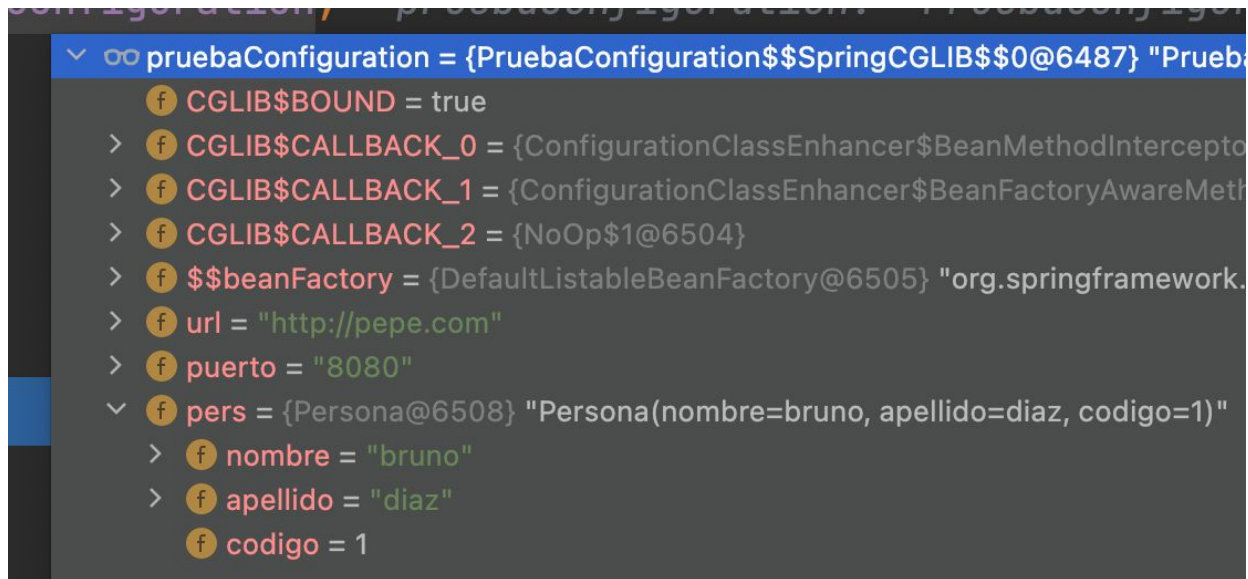
application.yml

```
server:
  port: 8080
fernando:
  url: http://pepe.com
  puerto: 8080
pers:
  nombre: bruno
  apellido: diaz
  codigo: 1
```

Configuración

```
@Data
@Configuration
@ConfigurationProperties(prefix = "fernando", ignoreUnknownFields = true)
public class PruebaConfiguration {
    protected String url;
    protected String puerto;
    protected Persona pers;
}
```

```
fernando:
url: http://pepe.com
puerto: 8080
pers:
    nombre: bruno
    apellido: diaz
    codigo: 1
```



```
pruebaConfiguration = {PruebaConfiguration$$SpringCGLIB$$0@6487} "Prueb
  f CGLIB$BOUND = true
  > f CGLIB$CALLBACK_0 = {ConfigurationClassEnhancer$BeanMethodIntercepto
  > f CGLIB$CALLBACK_1 = {ConfigurationClassEnhancer$BeanFactoryAwareMeth
  > f CGLIB$CALLBACK_2 = {NoOp$1@6504}
  > f $$beanFactory = {DefaultListableBeanFactory@6505} "org.springframework.
  > f url = "http://pepe.com"
  > f puerto = "8080"
  v f pers = {Persona@6508} "Persona(nombre=bruno, apellido=diaz, codigo=1)"
    > f nombre = "bruno"
    > f apellido = "diaz"
    f codigo = 1
```