
spring

6. Proyectos Web Spring Boot

6.1 Tecnologías

- **HTML**

Define la estructura básica de las páginas web, facilitando la presentación de información.

- **CSS**

Mejora la apariencia visual de la interfaz de la app asegurando una experiencia de usuario agradable.

- **JavaScript**

Implementa funciones interactivas en la interfaz de usuario, mejorando la experiencia del cliente.

- **ThymeLeaf**

Thymeleaf es una biblioteca Java que implementa un motor de plantillas de XML/XHTML/HTML5 (también extensible a otros formatos) que puede ser utilizado tanto en modo web como en otros entornos no web.

Se acopla muy bien para trabajar en la capa vista del MVC de aplicaciones web, pero puede procesar cualquier archivo XML, incluso en entornos desconectados.

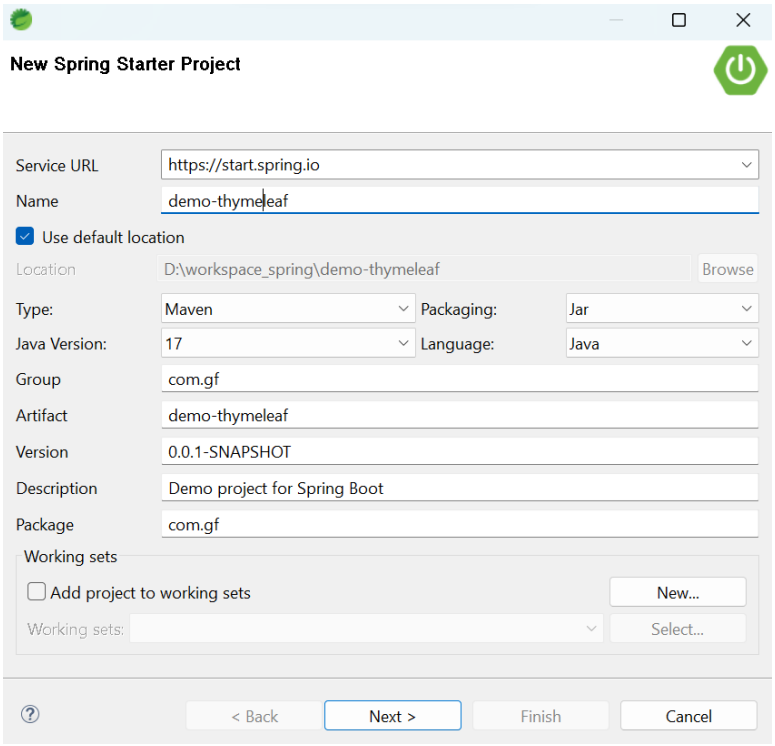
El objetivo principal de Thymeleaf es permitir la creación de plantillas de una manera elegante y un código bien formateado.

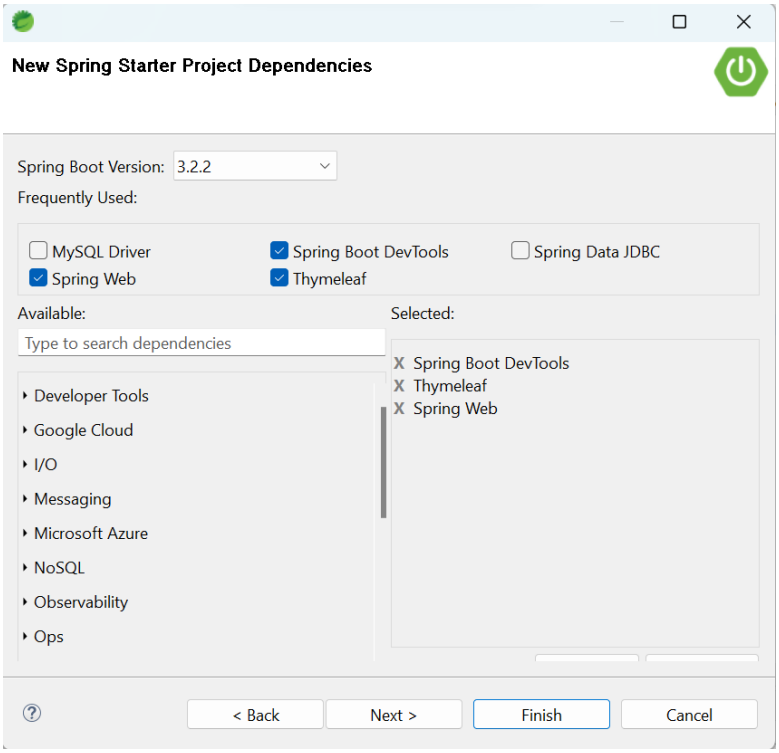
- **Bootstrap**

Bootstrap es un framework que permite a los desarrolladores web construir páginas web responsives de una forma más rápida y sencilla. En este sentido, proporciona un conjunto de componentes y plantillas CSS, HTML y JavaScript que cualquiera puede utilizar o modificar de manera gratuita.

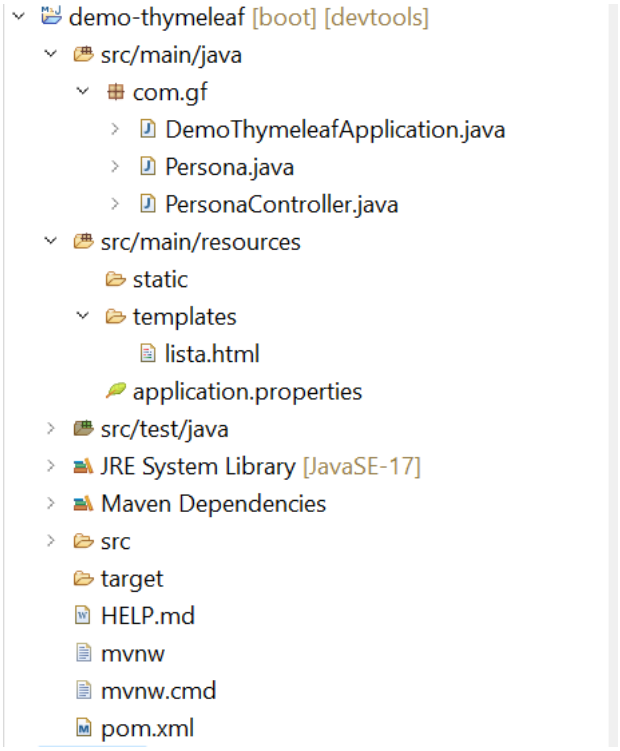
✓ Ejemplo Spring Web + Thymeleaf

✓ Creación de un nuevo started





✓ La estructura del Proyecto quedará de la siguiente manera:



✓ Implementación de la entidad Persona

```java
public class Persona {

    private String nombre;
    private String apellidos;
    private int edad;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public Persona(String nombre, String apellidos, int edad) {
        super();
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.edad = edad;
    }

    public Persona(String nombre) {
        this.nombre = nombre;
    }

}
```

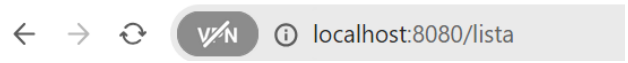
✓ Implementar la clase PersonaController.

```java
@Controller
public class PersonaController {

    @GetMapping("/lista")
    public String lista(Model modelo) {

        Persona p1 = new Persona("noel", "prieto", 20);
        Persona p2 = new Persona("sergio", "otero", 30);
        Persona p3 = new Persona("lucía", "garcia", 40);

        List<Persona> personas = Arrays.asList(p1, p2, p3);
        modelo.addAttribute("personas", personas);
        return "lista";

    }
}
```

✓ Implementamos HTML con etiquetas Thymeleaf

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<body>
<table>
<tr th:each="persona: ${personas}">
    <td th:text="${persona.nombre}" />
    <td th:text="${persona.apellidos}" />
    <td th:text="${persona.edad}" />
</tr>
</table>
</body>
</html>
```

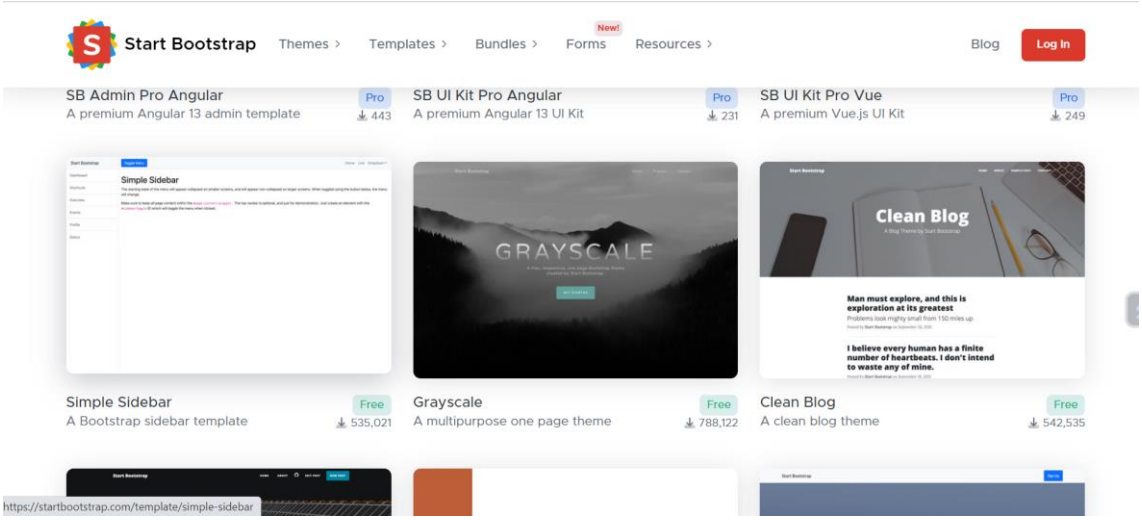
✓ Ejecutamos la aplicación de Spring Boot con la siguiente URL

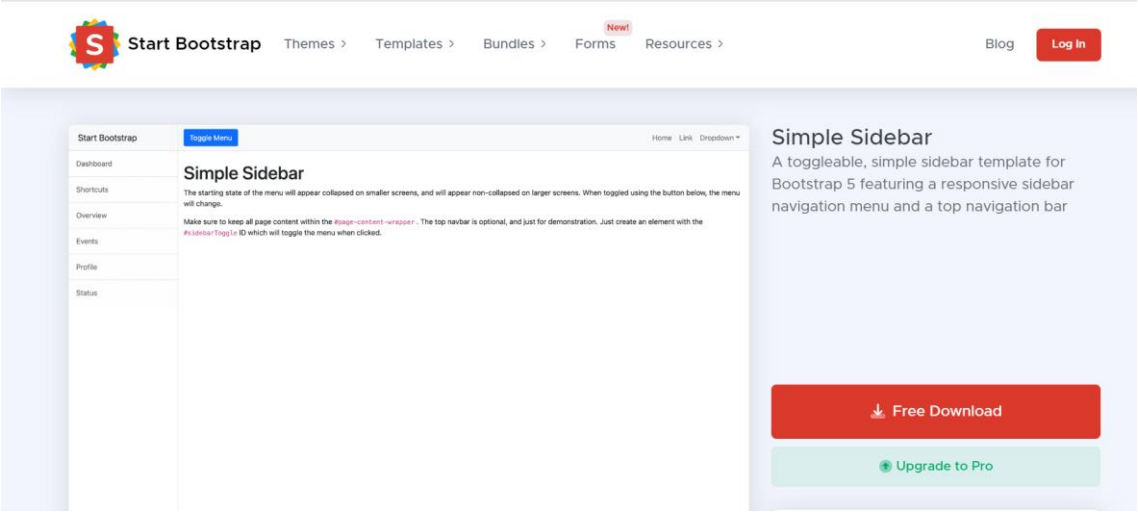


Listado personas

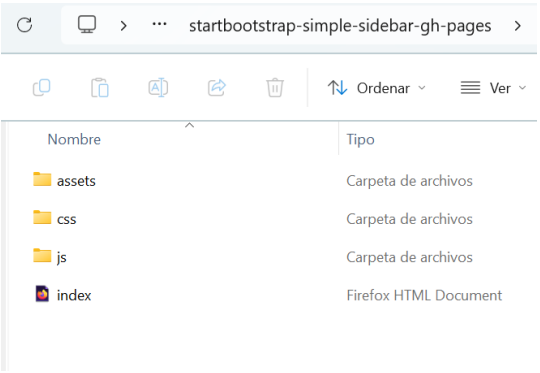
noel prieto 20
sergio otero 30
lucía garcia 40

✓ Ahora descargamos una plantilla de Bootstrap elegida para utilizar desde nuestro proyecto.





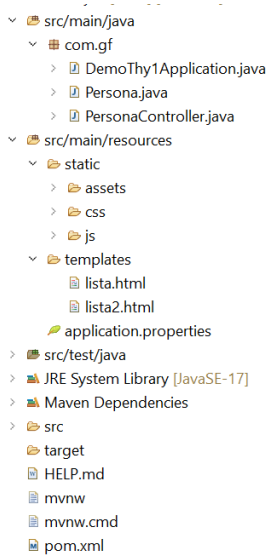
✓ Se descomprime el archivo y se trasladan las carpetas assets, css y jsp a src/main/resources/static del proyecto y el código del archivo index sirve como plantilla para crear lista.html.



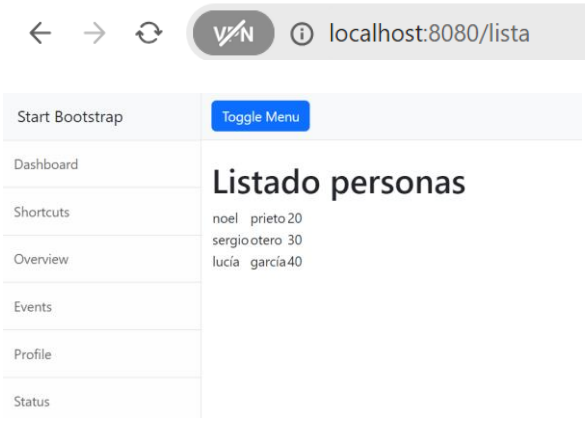
Se incrusta en la plantilla creada lista.html las etiquetas thymeleaf para mostrar el listado de personas

```
                        aria-controls="navbarSupportedContent" aria-expanded="false"
                        aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarSupportedContent">
                        <ul class="navbar-nav ms-auto mt-2 mt-lg-0">
                            <li class="nav-item active"><a class="nav-link" href="#!">Home</a></li>
                            <li class="nav-item"><a class="nav-link" href="#!">Link</a></li>
                            <li class="nav-item dropdown"><a
                                class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
                                role="button" data-bs-toggle="dropdown" aria-haspopup="true"
                                aria-expanded="false">Dropdown</a>
                                <div class="dropdown-menu dropdown-menu-end"
                                    aria-labelledby="navbarDropdown">
                                    <a class="dropdown-item" href="#!">Action</a> <a
                                        class="dropdown-item" href="#!">Another action</a>
                                    <div class="dropdown-divider"></div>
                                    <a class="dropdown-item" href="#!">Something else here</a>
                                </div></li>
                        </ul>
                    </div>
                </div>
            </nav>
            <!-- Page content-->
            <div class="container-fluid">
                <h1 class="mt-4">Listado personas</h1>
                <table>
                    <tr th:each="persona: ${personas}">
                        <td th:text="${persona.nombre}" />
                        <td th:text="${persona.apellidos}" />
                        <td th:text="${persona.edad}" />
                    </tr>
                </table>
            </div>
        </div>
    </div>
    <!-- Bootstrap core JS-->
    <script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
    <!-- Core theme JS-->
    <script src="js/scripts.js"></script>
</body>
</html>
```

✓ Ahora la estructura del proyecto queda de la siguiente manera



✓ Se ejecuta la aplicación de Spring Boot y se muestra el siguiente resultado



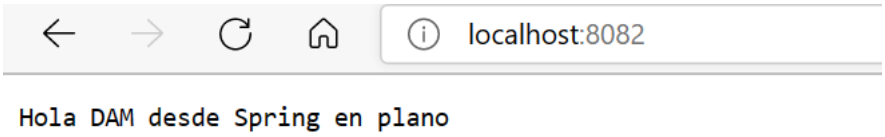
- **API Rest**

Una API REST es una interfaz de comunicación entre sistemas de información que usa el protocolo de transferencia de hipertexto (*hypertext transfer protocol* o HTTP, por su siglas en inglés) para obtener datos o ejecutar operaciones sobre dichos datos en diversos formatos.

✓ *Por ejemplo, creando un servicio RES que devuelve un mensaje en texto plano, HTML pero podría devolver el mismo resultado en JSON o XML directamente.*

```
@RestController
public class Controlador {

    @GetMapping(value="/index", produces = MediaType.TEXT_PLAIN_VALUE)
    public String index() {

        return "HOLA MUNDO SPRING EN PLANO";
    }

}
```

✓ *Ya solo queda realizar una petición web a la URL en donde se ha mapeado el Spring REST Service:*



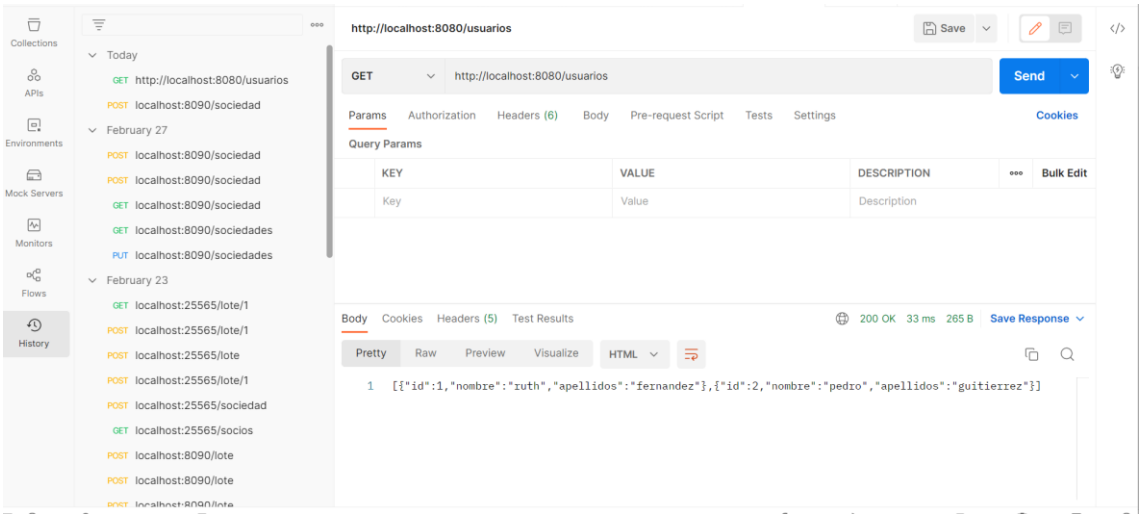
- **Postman**

Postman es una herramienta de colaboración y desarrollo que permite a los desarrolladores interactuar y probar el funcionamiento de servicios web y aplicaciones. proporcionando una interfaz gráfica intuitiva y fácil de usar para enviar solicitudes a servidores web y recibir las respuestas correspondientes.

Este entorno ofrece una GUI que facilita a los desarrolladores el envío de solicitudes HTTP y HTTPS a una API y a gestionar las respuestas recibidas.

Las principales características y funcionalidades de Postman son:

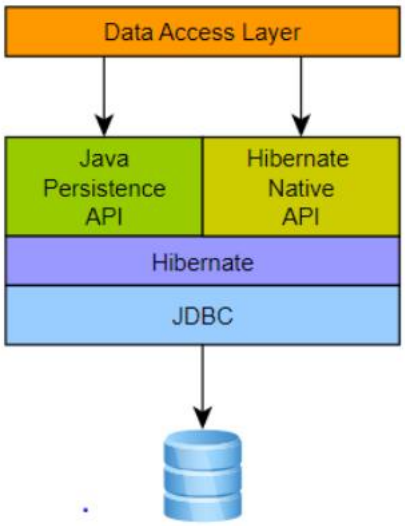
- Envío de solicitudes. Permite enviar solicitudes GET, POST, PUT, DELETE y otros métodos HTTP a una API especificando los parámetros, encabezados y cuerpo de la solicitud.
- Gestión de entornos. Facilita la configuración para diferentes entornos (por ejemplo, desarrollo, prueba, producción) y el cambio sencillo entre ellos (para realizar pruebas y desarrollo en diferentes contextos).
- Colecciones de solicitudes. Agrupa las solicitudes relacionadas en colecciones, lo que facilita la organización y ejecución de pruebas automatizadas.
- Pruebas automatizadas. Es ideal para crear y ejecutar pruebas automatizadas para verificar el comportamiento de una API (detectar errores e incrementar la calidad del software).
- Documentación de API. Genera de forma automatizada, documentación detallada de la API a partir de las solicitudes y respuestas realizadas, lo que facilita su comprensión y uso por parte de otros desarrolladores.



- **JPA**

Java Persistence API (JPA) es un framework que forma parte de Java, y ofrece un conjunto de interfaces y APIs para resolver el problema del almacenamiento de los objetos en una base de datos relacional. JPA no es una implementación en sí misma, sino que brinda interfaces para ser luego implementadas por distintos proveedores. De esta manera, en el código usamos el API de JPA, que luego será implementado por la librería que más nos convenga.

Hibernate entonces es un ORM que implementa JPA, pudiendo trabajar en Hibernate Nativo o a través de JPA, es decir, se puede usar Hibernate para construir una capa de persistencia apoyándose en las definiciones y reglas que la especificación de JPA, aunque esto no sea obligatorio. Sin embargo, esto no quiere decir que Hibernate simplemente implemente el estándar de JPA. **Hibernate es mucho más grande que la especificación de JPA y añade más funcionalidad .**



Una tabla se mapea contra una clase, y cada columna contra un atributo de dicha clase. Usaremos anotaciones de JPA para indicar estas relaciones. De esta forma, la implementación con JPA se encargará de ocultar la complejidad del acceso a datos, exponiendo solamente objetos. **Anotaciones comunes JPA**

```
@Entity
@Table
@Basic
@Column
@GeneratedValue
@Id
@Transient
@Temporal
@OneToMany
@ManyToOne
@ManyToMany
@Query (uso de Spring Data JPA)
@Modifying
```

、**@Entity** Antes de que la anotación se utilice en la declaración de declaración de la clase de entidad, indica que la clase Java es una clase de entidad y se asignará a la tabla de base de datos especificada.

、**@Table**

Cuando la clase de entidad y su tabla de base de datos asignada tienen nombres diferentes, debe usar la anotación @Table. Esta anotación se usa en paralelo con la anotación @Entity y se coloca antes de la declaración de declaración de la clase de entidad.

、**@Id**

La anotación @Id se utiliza para declarar que los atributos de una clase de entidad se asignan a la base de datosColumna de clave primaria. El atributo generalmente se coloca antes de la declaración de atributo.

、**@GeneratedValue**

@GeneratedValue se utiliza para marcar la estrategia de generación de la clave principal, que se especifica mediante el atributo de estrategia. Las siguientes estrategias se definen en javax.persistence.GenerationType:

- **AUTO**: La opción por defecto. Deja al proveedor de persistencia elegir cuál de las siguientes tres opciones va a utilizar.
- **SEQUENCE**: Utiliza una secuencia SQL para obtener el siguiente valor de la clave primaria.
- **TABLE**: Necesita una tabla con dos columnas, el nombre de la secuencia y su valor.
- **IDENTITY**: utiliza un generador de identidad como las columnas definidas con auto_increment en MySQL.

、**@Column**

Cuando el atributo de la entidad y la columna de la tabla de la base de datos a la que está asignada tienen nombres diferentes

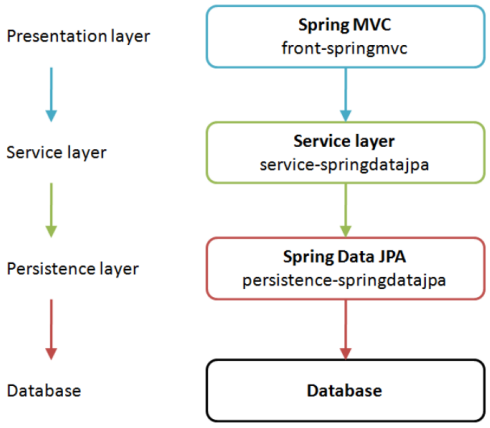
Las clases de entidad son tratadas como tablas relacionales (concepto de JPA), por lo tanto, otras anotaciones para tratar las relaciones entre las clases de entidad son los siguientes:

- **@OneToOne Relation**: Se usa para definir la relación one-to-one (uno a uno)
- **@ManyToMany Relation**: Se usa para definir la relación many-to-many (muchos a muchos)

- **@ManyToOne Relation:** Usas esto para definir la relación many-to-one (muchos a uno).
- **@OneToMany Relation:** Define una relación one-to-many (uno a muchos). mappedBy. Indica que entidad es dueña del uno a muchos de forma única.
- **@JoinColumn:** Para determinar la columna que usaremos como clave para lograr la relación con la entidad principal.

- Otros atributos que se pueden utilizar son:
  - mappedBy, podemos establecer una relación bidireccional ya que a pesar de tener una única FK, podemos relacionar ambas tablas. Al final, el objetivo de las anotaciones es dejar claro dónde está la clave que mapea las relaciones.
  - orphanRemoval= true especifica que la entidad hijo debe ser eliminada automáticamente por el propio ORM si ha dejado de ser referenciada por una entidad padre. p.ej., tenemos una colección de items y eliminamos uno, ese item ha dejado de tener una referencia y será eliminado. No confundir con cascadeType que son operaciones a nivel de base de datos.
  - fetchType=LAZY, Recupera la entidad solo cuando realmente la necesitamos. Importante destacar que la sesión debe estar abierta para poder invocar al Getter correspondiente y recuperar la entidad, ya que hibernate usa el patrón Proxy (object proxying) . En caso contrario (al cerrar la sesión), la entidad pasaría de estado persistent a detach y se lanzaría una excepción LazyInitializationException.

✓ **Spring Data** JPA es uno de los frameworks que se encuentra dentro de la plataforma de Spring. Su objetivo ***es*** *simplificar al desarrollador la persistencia de datos contra distintos repositorios de información****.***

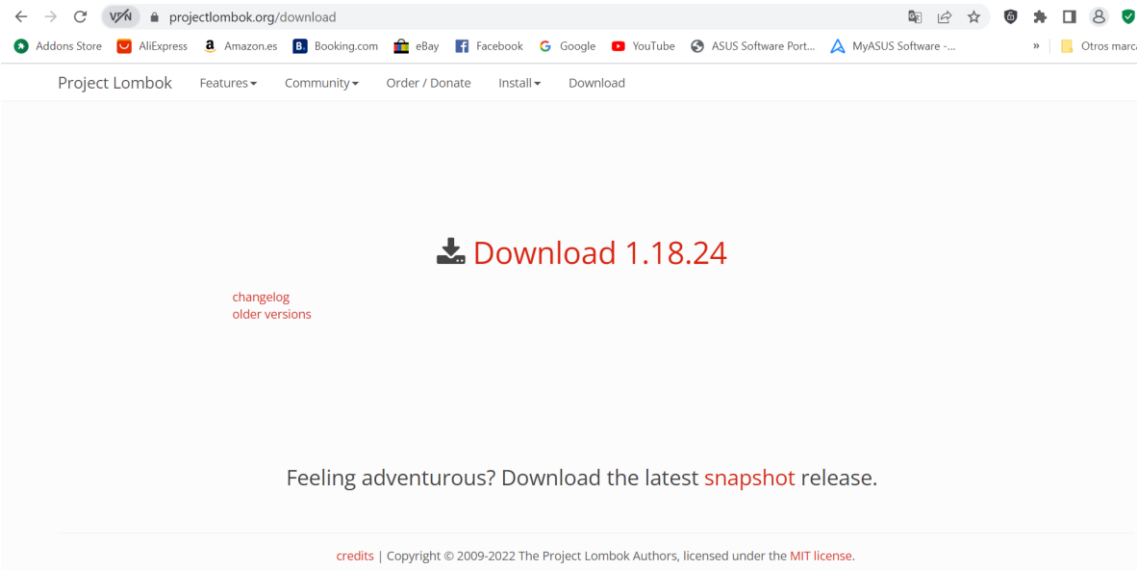


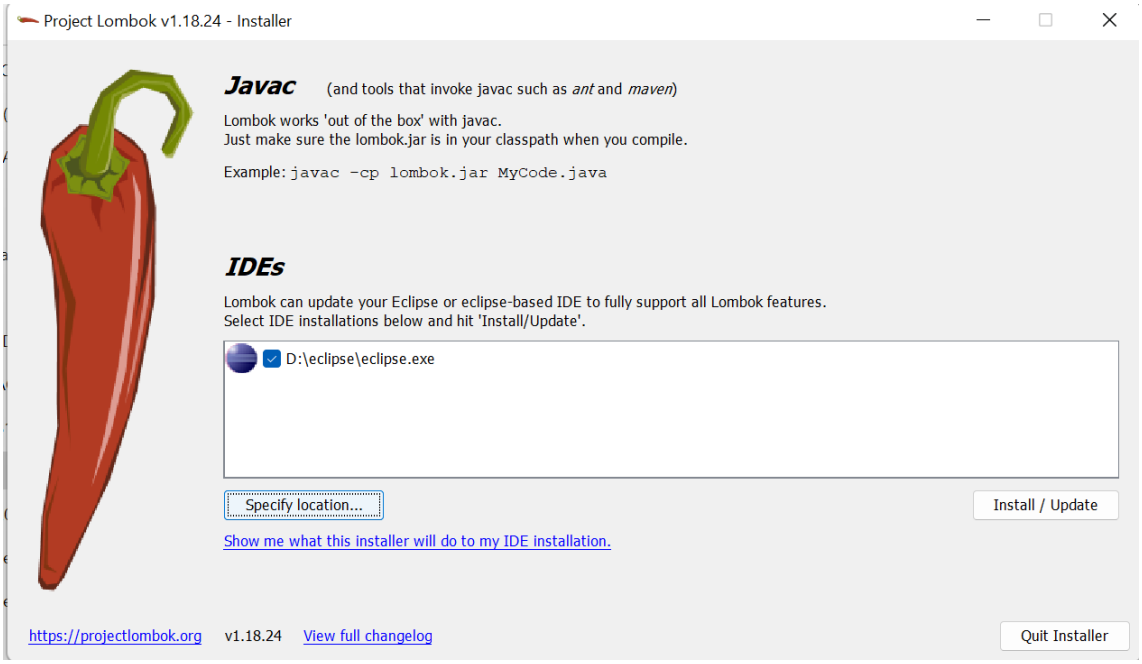
- **Lombok**

Project Lombok es una librería que va a tunear nuestro IDE, que tras instalarla transformará el mismo, ya sea Spring Tool Suite o Eclipse. Además, será una dependencia que tenemos que incluir en nuestro proyecto para que podamos añadir una serie de anotaciones que nos simplifica la generación de getters, setters, equals, hashCode, toString… .

Las anotaciones Lombok aportan agilidad al desarrollo del código, simplificando las clases. Para poder utilizar se debe descargar el plugin correspondiente en Spring Tools desde https://projectlombok.org/p2

Otra opción es descargar de la página oficial el .jar y ejecutarlo.




Project Lombok v1.18.24 - Installer
Javac
(and tools that invoke javac such as ant and maven)
Lombok works 'out of the box' with javac.
Just make sure the lombok.jar is in your classpath when you compile.
Example: javac -cp lombok.jar MyCode.java
IDEs
Lombok can update your Eclipse or eclipse-based IDE to fully support all Lombok features.
Select IDE installations below and hit 'Install/Update'.
D:\eclipse\eclipse.exe
Specify location...
Install / Update
Show me what this installer will do to my IDE installation.
https://projectlombok.org
v1.18.24
View full changelog
Quit Installer