

# SPRING FRAMEWORK

## ¿Qué es Lombok ?

**Lombok es una biblioteca de Java que ofrece una serie de anotaciones y herramientas que simplifican y automatizan tareas comunes en el desarrollo de software.** Su objetivo principal es reducir la verbosidad del código Java eliminando la necesidad de escribir getters, setters, constructores, métodos equals, hashCode, y toString, entre otros.

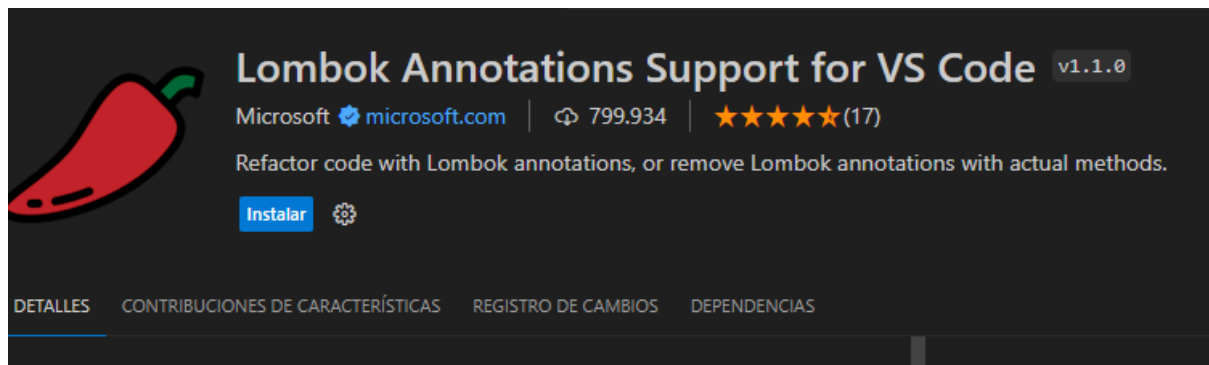
Al utilizar Lombok, puedes agregar anotaciones a tus clases para que Lombok genere automáticamente el código repetitivo durante el proceso de compilación. Esto hace que el código sea más limpio, legible y fácil de mantener, al tiempo que reduce el riesgo de errores y la cantidad de código redundante que necesitas escribir manualmente.

El principal uso de esta herramienta es en la maquetación de nuestras entidades, pero se puede utilizar Lombok en otros tipos de clases como componentes de Spring (@Component, @Service, @Controller, etc.), clases de utilidad, clases de DTO (Data Transfer Object), y en general en cualquier clase donde desees aprovechar las anotaciones de Lombok para generar automáticamente métodos y constructores.

Sin embargo, es importante recordar que aunque Lombok puede hacer que tu código sea más conciso y fácil de mantener, debes usarlo con moderación y consideración. En algunos casos, la claridad del código puede verse comprometida si abusas de las anotaciones de Lombok, especialmente en clases complejas o donde se requiera una lógica de negocio específica.

## **1. Descarga de la extensión en VSCode:**

1. Busca en la pestaña extensiones, la correspondiente a Lombok.



2. Instala la herramienta en tu editor de código.
3. Ya puedes hacer uso de los beneficios de la herramienta.

## Descargar la herramienta para otro editor de código:

Si estás utilizando otro editor de código, puedes descargar el instalador correspondiente desde <https://projectlombok.org/>. Una vez descargado, ejecútalo según el entorno en el que trabajes y sigue el asistente que se te propone. Lo primero que hará el instalador será escanear los IDEs que tengas instalados en tu sistema. Si no encuentra ninguno, deberás buscarlo manualmente, seleccionando "Specify location..." y luego pulsar en "Install/Update" para instalar Lombok.

## 2. Declaración de la dependencia en archivo pom.xml:

Debemos declarar el uso de la dependencia en el archivo pom.

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <scope>provided</scope>
</dependency>
```

\*No es necesario declarar una versión específica. De este modo, hará búsqueda de la última versión disponible al compilar.

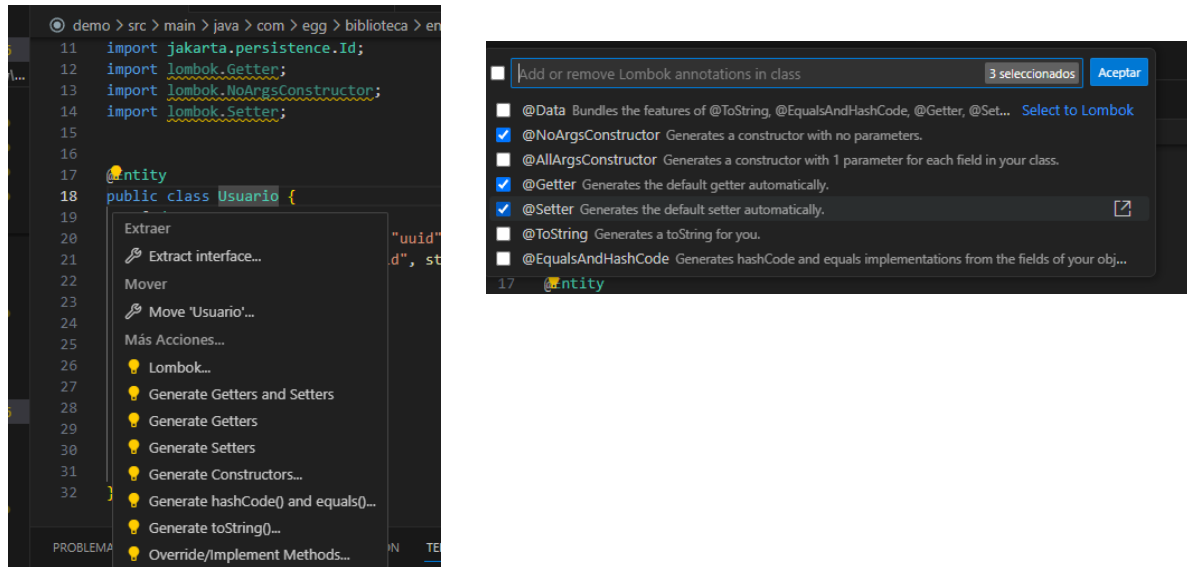
## 3. Ejemplo de Uso: Clase Entidad

Una de las formas de implementar su uso de manera rápida, es posicionarse en una línea de código, el asistente automáticamente pondrá un acceso para

las acciones disponibles.

```
@Entity
public class Usuario {
    @Id
    @GeneratedValue (generator
    @GenericGenerator (name
```

Podremos acceder a las utilizadas con más frecuencia o buscar alguna específica.



Una vez elegidos los requerimientos, lombok se ocupará de incorporar las anotaciones y realizar las importaciones necesarias.

```
9 import jakarta.persistence.Enumerated;
10 import jakarta.persistence.GeneratedValue;
11 import jakarta.persistence.Id;
12 import lombok.Getter;
13 import lombok.NoArgsConstructor;
14 import lombok.Setter;
15
16
17 @Setter
18 @Getter
19 @NoArgsConstructor
20 @Entity
21 public class Usuario {
22     @Id
23     @GeneratedValue (generator = "uuid")
24     @GenericGenerator (name = "uuid", strategy = "uuid2")
25     private String id;
26
27     private String nombre;
28     private String email;
29     private String password;
30
31     @Enumerated(EnumType.STRING)
```

¡Listo! Ya puedes continuar con la maquetación de tu proyecto. Lombok realizará todo lo necesario durante la compilación del código.

## 4. Algunas anotaciones utilizadas con frecuencia:

<b>@Getter y @Setter</b>	Estas anotaciones son usadas para generar los métodos getter y setter de manera automática.
<b>@ToString</b>	Esta anotación la utilizaremos para generar el método toString(), la aplicamos a nivel de clase y cuenta con las siguientes configuraciones: Por defecto la cadena generada corresponde al nombre de la clase seguido de una lista de los campos y su valor actual separados por coma.
<b>@NoArgsConstructor</b> <b>@AllArgsConstructor</b>	y Usaremos estas anotaciones para generar el constructor de la clase, la primera crea un constructor sin argumentos y el segundo un constructor que permite inicializar cada uno de los campos de la clase.
<b>@Data</b>	Esta anotación es una combinación de todas las anotaciones que vimos anteriormente en este tutorial con excepción de @AllArgsConstructor, es muy útil cuando trabajamos con herramientas como el ORM Hibernate.

Existen muchas otras anotaciones. En la web <https://projectlombok.org/> veras varios ejemplos de cómo utilizarlas.