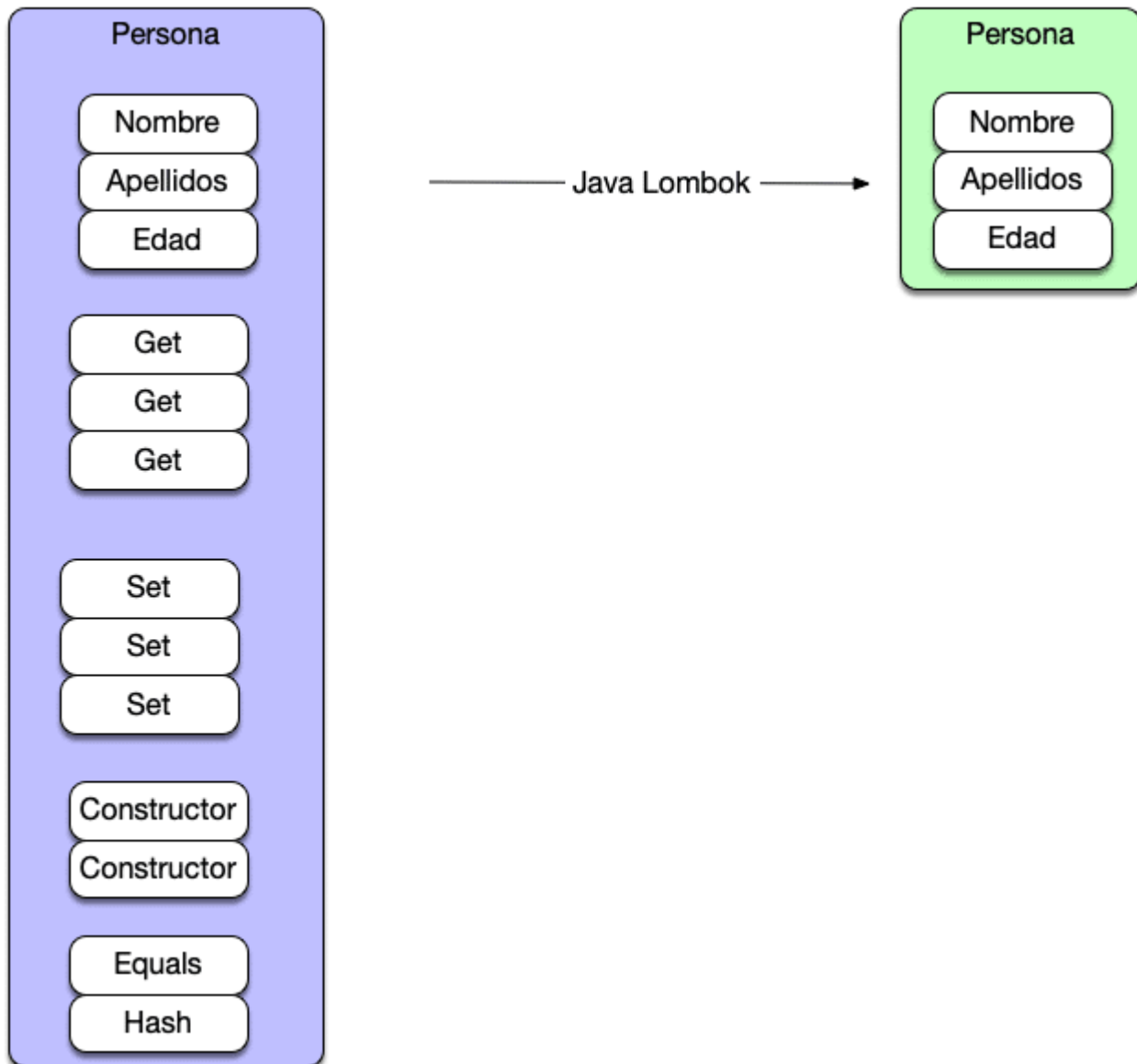
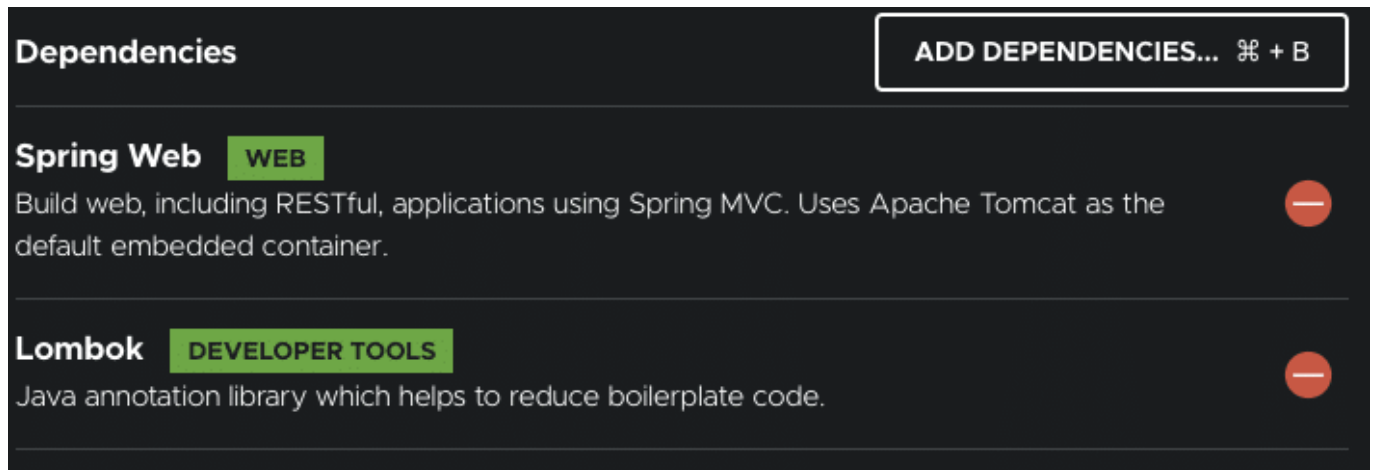


El uso de Java Lombok es bastante habitual en los proyectos de Java . Java es un lenguaje clásico y como lenguaje que tiene 20 años tiene que asumir que cosas que se realizaron en el pasado ... pues pueden ser mejorables . Este es el caso de el manejo de Getters , Setters , Constructores , Equals y hashCode. La mayor parte de estos métodos los generamos de forma automática con Eclipse . Pero tenemos el problema de que hay que generarlos para cada clase algo que consume bastante tiempo . La libreria de Java Lombok nos permite simplificar estas cosas y reducirlas prácticamente a la nada.



Vamos a ver un ejemplo con Spring Boot . Para ello elegimos las dependencias en [Spring](#)

Initializer.



Como podemos ver Spring Boot soporta directamente Lombok como librería a configurar . Una vez realizado este paso descargamos el proyecto y lo abrimos . Es momento de construir la clase Persona esta clase habitualmente se encuentra rellena por métodos get/set , constructores , Equals/HashCode etc. Con Java Lombok como dependencia podemos eliminar todo este código y centrarnos en lo que realmente importa.

```
package com.arquitecturajava.lombok;
```

```
import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.Setter;
import lombok.ToString;
```

```
@Getter
@Setter
@EqualsAndHashCode
@ToString
@AllArgsConstructor
public class Persona {
```

```
        private String nombre;  
        private String apellidos;  
        private int edad;  
    }  
}
```

Una vez realizada esta operación podemos generarnos un servicio REST para las Personas y publicar una sencilla Lista.

```
package com.arquitecturajava.lombok;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class ServicioREST {  
  
    @RequestMapping("/personas")  
    List<Persona> buscarTodas() {  
        List<Persona> lista= new ArrayList<Persona>();  
        lista.add(new Persona ("pedro","perez",20));  
        lista.add(new Persona ("juan","sanchez",30));  
        return lista;  
    }  
}
```

Realizadas estas operaciones podemos lanzar la aplicación de **Spring Boot** y ver el resultado en el navegador.

```
[{"nombre":"pedro","apellidos":"perez","edad":20},{\"nombre\":\"juan\",\"apellidos\":\"sanchez\",\"edad\":30}]
```

Acabamos de usar Java Lombok para configurar de una forma mucho más sencilla nuestros Java Beans.

Java Lombok y @Data

Otra posible configuración es en vez de usar cada una de las anotaciones de forma independiente es usar directamente la anotación @Data que añade todas las anteriores de forma directa.

```
package com.arquitecturajava.lombok;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
@Data
```

```
@AllArgsConstructor
```

```
public class Persona {
```

```
    private String nombre;
```

```
    private String apellidos;
```

```
    private int edad;
```

```
}
```

Eclipse y Java Lombok

Uno de los problemas que todo el mundo tiene con Lombok cuando empieza es su integración con Eclipse . Para que la integración sea natural debemos descargarnos el Jar de Lombok y ejecutarlo en nuestro equipo el localizará el Eclipse que corresponda y lo actualizará para que las clases Javas al estar vacías no den errores.

Eclipse, Spring Tool Suite, (Red Hat) JBoss Developer Studio, MyEclipse

The **Eclipse** editor is compatible with lombok. Eclipse offshoots are also compatible with lombok, specifically:

- **MyEclipse**
- **Red Hat JBoss Developer Studio**
- **JBoss Developer Studio (JBDS)**
- **Spring Tools Suite (STS)**

Double-click `lombok.jar` (downloadable from this site, or from your maven repository; it's the same jar). This starts the eclipse installer which will find eclipse (and eclipse variants as listed above), and offers to install lombok into these eclipse installations. The same tool can also uninstall lombok:



Otros Artículos relacionados

- ¿Que es Spring Framework?
- Java Encapsulamiento y reutilización
- Java Herencia vs Interfaces