

Curso de introducción a Lombok y primeras anotaciones

Desarrollo en Java

¡HOLA!

Soy David Sebastián

Soy Senior developer y arquitecto

Me dedico al diseño y desarrollo de soluciones arquitectónicas

 [linkedin.com/in/david-sebastián-manjón-695b12108/](https://www.linkedin.com/in/david-sebastián-manjón-695b12108/)

ÍNDICE

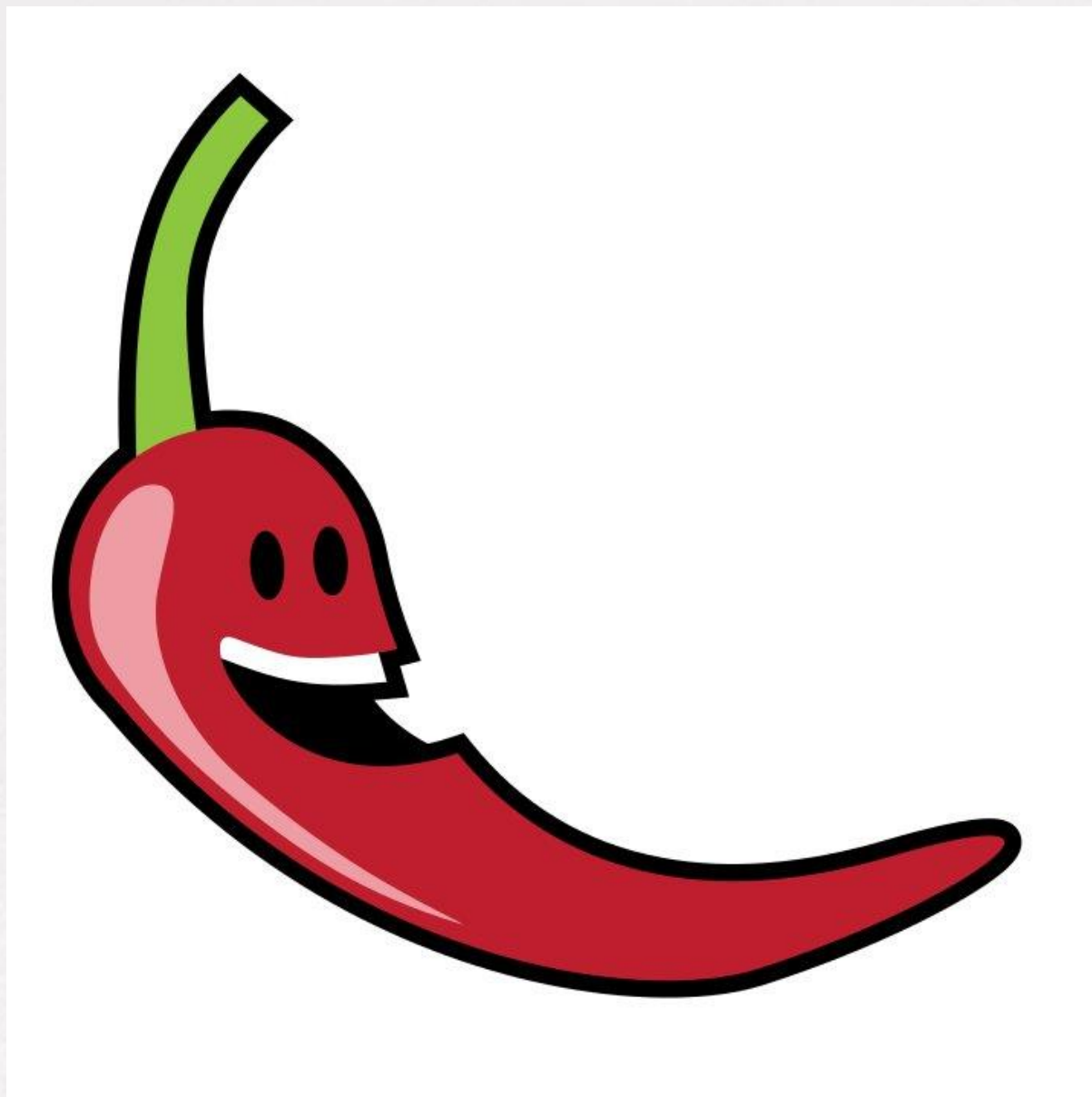
- **Introducción**
- **Alternativas a Lombok**
- **Instalación**
- **Anotaciones**
- **Ejercicio práctico**
- **Cierre**

REQUISITOS **PREVIOS**

- **Desarrollo en Java**



SOBRE LOS CURSOS DE **LOMBOK**



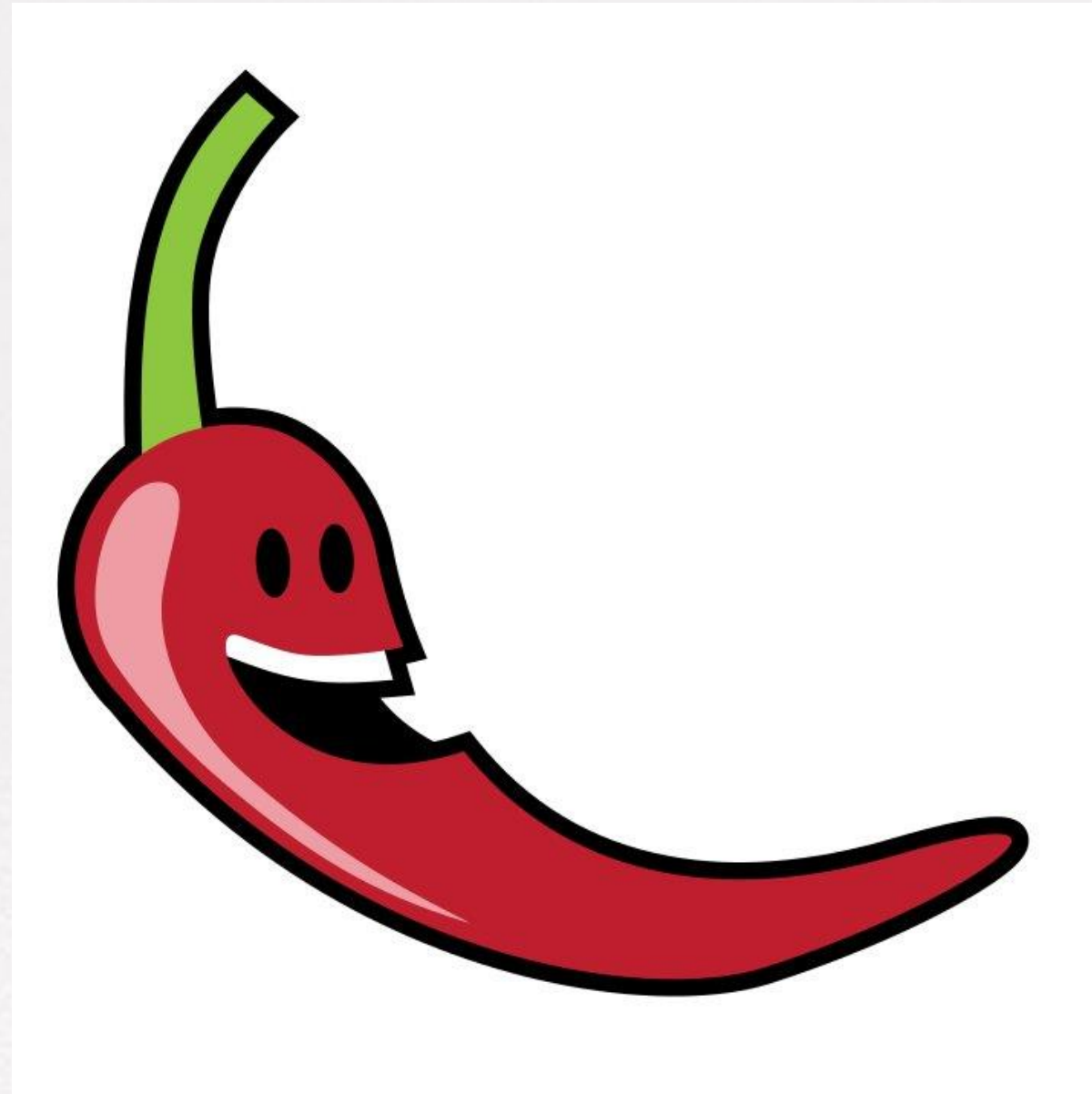
Introducción a Lombok y primeras anotaciones

Las anotaciones Lombok orientadas a objetos

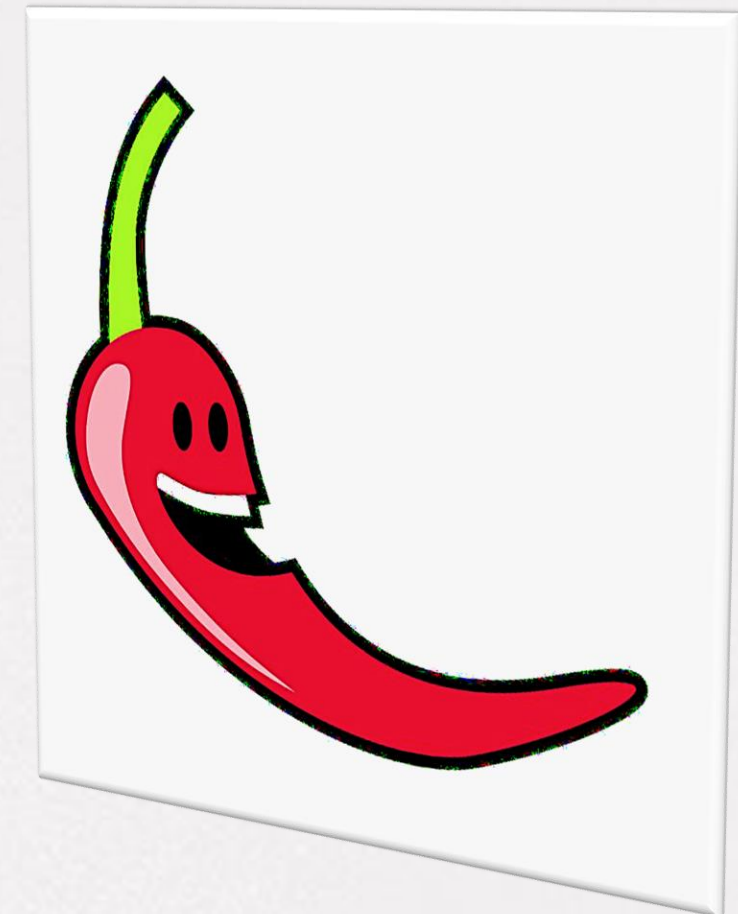
Las anotaciones Lombok para constructores, equals y hash code

Construye objetos variables y constantes, excepciones, sincronización y logs

INTRODUCCIÓN A LOMBOK



- **Introducción a Lombok**
- Alternativas a Lombok
- Instalación
- Anotaciones
- Ejercicio práctico
- Cierre



QUÉ ES **LOMBOK**

1. Project Lombok
2. Biblioteca de Java
3. Conecta al editor/compilador
4. Facilita el desarrollo

QUÉ ES **LOMBOK**

5. Evita escribir código:
 - a. Getters/Setters
 - b. Equals
 - c. Constructor(es)
 - d. Variables/constantes

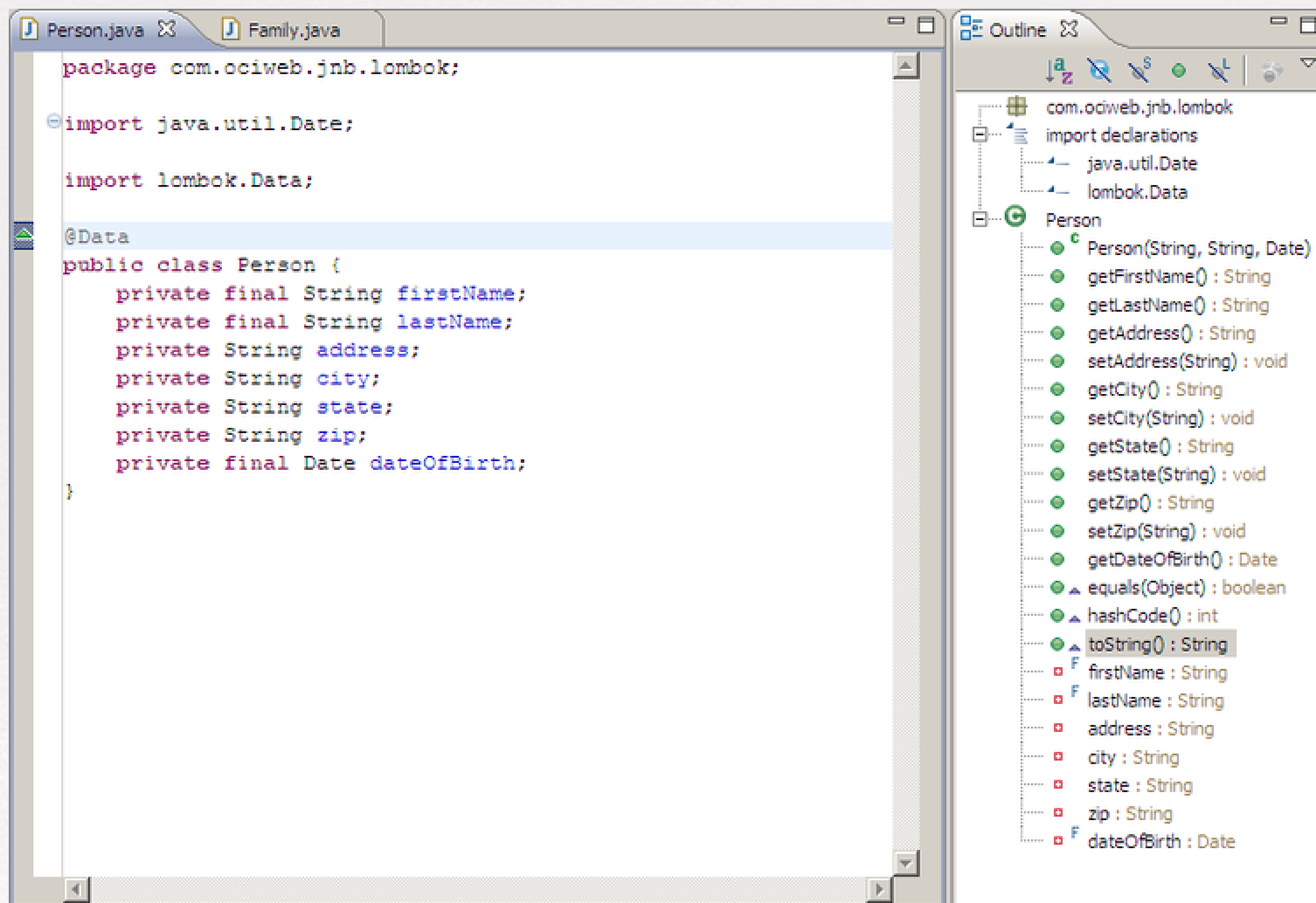
QUÉ ES **LOMBOK**

6. Elimina código repetitivo
7. Genera automáticamente
8. Integra con el propio IDE
 - a. Inyecta código
 - b. Al instante lo tiene el desarrollador

QUÉ ES **LOMBOK**

- 9. Ejemplo: “@Data”
 - a. Constructor
 - b. Getters/setters
 - c. equals
 - d. hashCode
 - e. toString

QUÉ ES LOMBOK



INTRODUCCION A LA **INSTALACIÓN**

1. Tendremos una sección, pero emplearemos más Maven
2. Existen más herramientas
 - a. Maven
 - b. Gradle
 - c. Ivy
 - d. SBT
 - e. Java

INTRODUCCION A LA **INSTALACIÓN**

a. Maven

```
<dependencies>  
  <dependency>  
    <groupId>org.projectlombok</groupId>  
    <artifactId>lombok</artifactId>  
    <version>1.18.12</version>  
    <scope>provided</scope>  
  </dependency>  
</dependencies>
```

INTRODUCCION A LA **INSTALACIÓN**

a. Maven

`<version>1.18.12</version>`

<https://mvnrepository.com/artifact/org.projectlombok/lombok>

INTRODUCCION A LA **INSTALACIÓN**

b. Gradle \geq 2.12

```
compileOnly "org.projectlombok:lombok:1.18.12 "
```


INTRODUCCION A LA **INSTALACIÓN**

b. Gradle < 2.12

```
provided "org.projectlombok:lombok:1.18.12 "
```

INTRODUCCION A LA **INSTALACIÓN**

c. Ivy

```
<dependency  
  org="org.projectlombok"  
  name="lombok"  
  rev="1.18.12"  
  conf="build" />
```

INTRODUCCION A LA **INSTALACIÓN**

d. SBT

```
libraryDependencies += "org.projectlombok" % "lombok" % "1.18.12"
```

INTRODUCCION A LA **INSTALACIÓN**

e. Java

```
java -jar lombok.jar
```

<https://projectlombok.org/downloads/lombok.jar>

ALGUNAS DE LAS ANOTACIONES

- ✓ @NonNull
- ✓ @Cleanup
- ✓ @Getter/@Setter
- ✓ @ToString
- ✓ @EqualsAndHashCode
- ✓ @NoArgsConstructor
- ✓ @RequiredArgsConstructor
- ✓ @AllArgsConstructor

ALGUNAS DE LAS ANOTACIONES

- ✓ @Data
- ✓ @Value
- ✓ @Builder
- ✓ @SneakyThrows
- ✓ @Synchronized
- ✓ @With
- ✓ @Log

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

1. Incapacidad de detectar los constructores de una superclase.

Esto significa que si una superclase no tiene un constructor por defecto, ninguna subclase puede usar la anotación `@Data` sin escribir explícitamente un constructor para hacer uso del constructor de superclase disponible.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

2. Las anotaciones fueron pensadas para información meta.

Las anotaciones no deben usarse si al eliminarlas el código deja de compilar. Esto es justo lo que sucede con las anotaciones de Lombok. De estas anotaciones surgen nuevos métodos que están destinados a ser utilizados en el propio código de la aplicación. Además, para el correcto funcionamiento requiere soporte por el IDE.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

3. Código privativo con brujería.

El proyecto Lombok hace uso de un API no públicas para lograr la "brujería". Esto significa que existe el riesgo de que Project Lombok se rompa con las versiones posteriores de IDE o JDK.

CONTROVERSIAS Y LIMITACIONES DE LOMBOK

4. @SneakyThrows permite lanzar excepciones fantasma.

Discordia entre los que apoyan que las excepciones se deben mostrar o no.

RESUMEN (1/3)

1. Es una herramienta de ayuda al desarrollador
2. Elimina código repetitivo
3. Permite crear clases de limpias, concisas y fáciles de mantener.

RESUMEN (2/3)

1. En IntelliJ no es muy estable
2. Existe el riesgo de rotura con las actualizaciones de IDE y JDK
3. Anotaciones ¿para información meta?

RESUMEN (3/3)

La pregunta es

¿El Proyecto Lombok puede proporcionar más valor que costo?

REFERENCIAS

1. Web

- <https://projectlombok.org>

2. Configuración

- <https://projectlombok.org/setup/overview>

3. Instalación

- <https://projectlombok.org/mavenrepo/index.html>

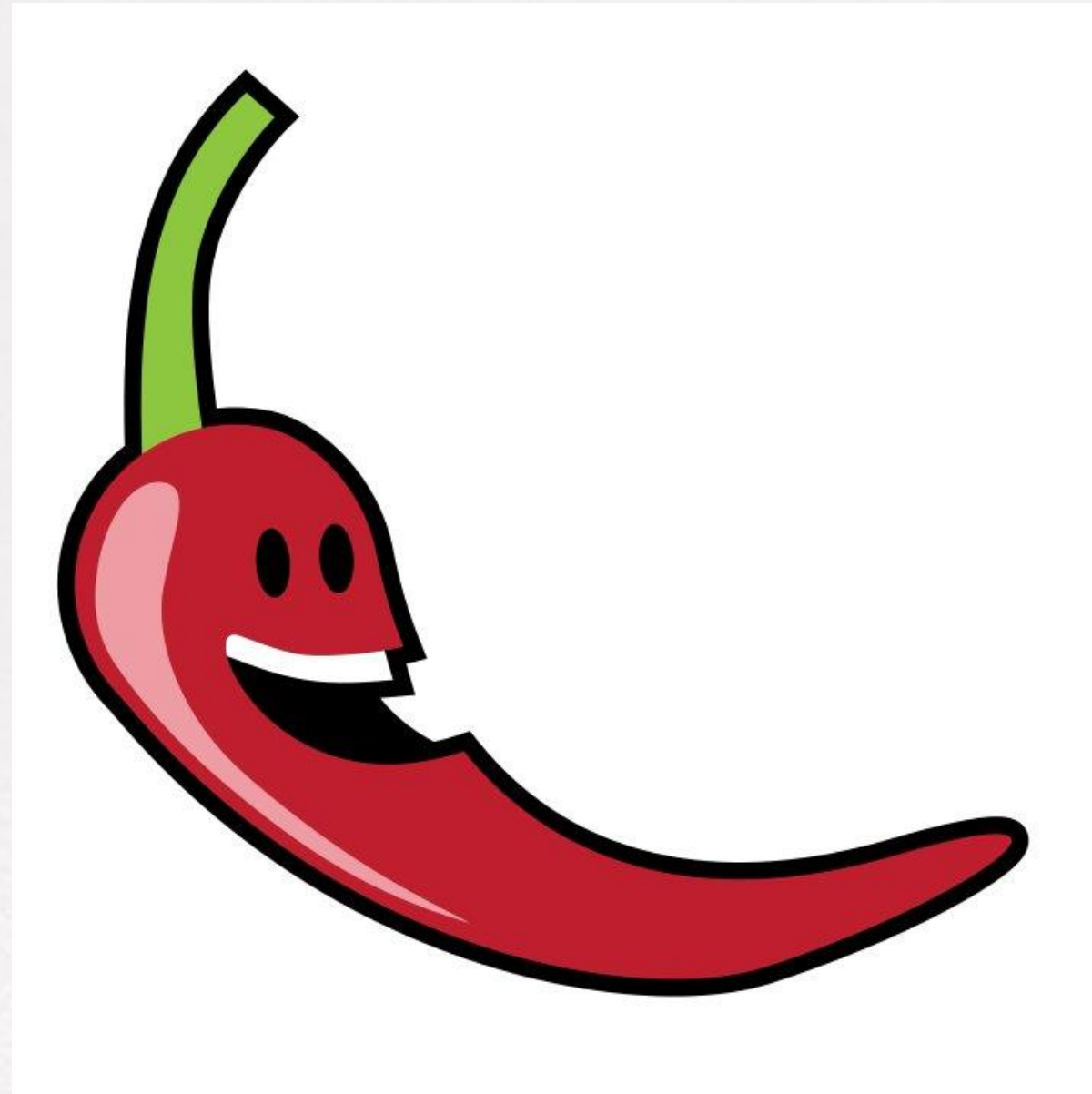
4. API

- <https://projectlombok.org/api/index.html>

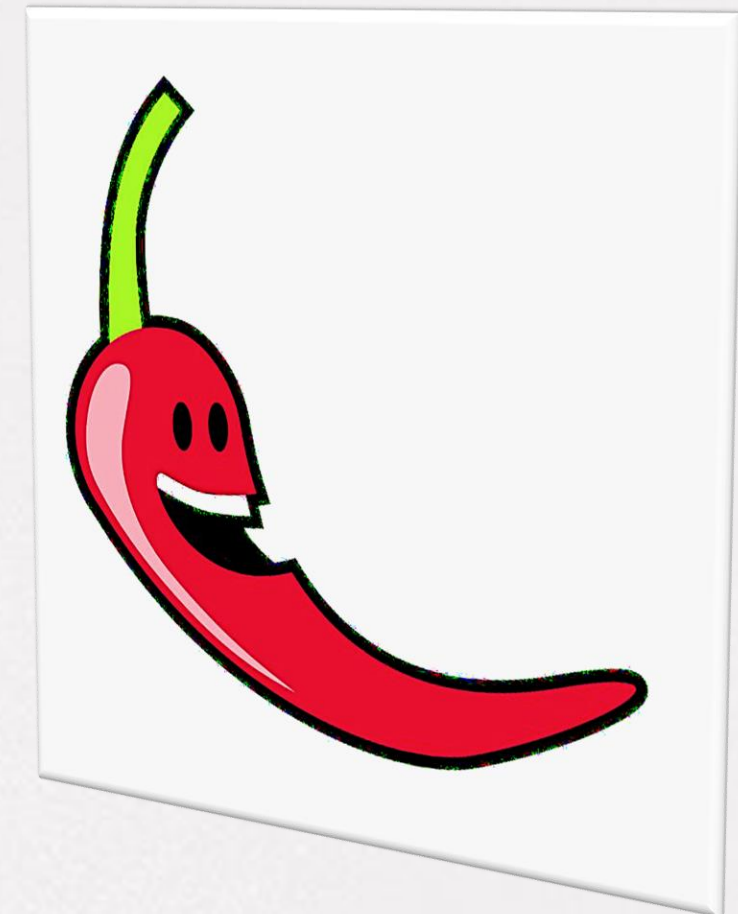
5. Google groups

- <https://groups.google.com/g/project-lombok>

ALTERNATIVAS A LOMBOK



- Introducción a Lombok
- **Alternativas a Lombok**
- Instalación
- Anotaciones
- Ejercicio práctico
- Cierre



ALTERNATIVAS: **SPRING**

1. Spring: <https://spring.io>

Spring no es sólo un framework, sino una familia.

Facilita el desarrollo incrementando las capacidades de Java



ALTERNATIVAS: **SPRING**

Microservicios

Permite muy fácilmente crear microservicios que pueden evolucionar de forma independiente.

Reactivo

La arquitectura asincrónica y sin bloqueo.

Cloud

Permite desarrollar servicios independiente de la plataforma.



ALTERNATIVAS: **SPRING**

Aplicaciones web

Frameworks para aplicaciones web con facilitar para acceder cualquier data store.

Serverless

La máxima flexibilidad. Permite escalar y desescalar bajo demanda.

Orientado a eventos

Permite reaccionar bajo la lógica de negocio, permitiendo transmisión en tiempo real.



ALTERNATIVAS: **SPRING**

Batch

Tareas automatizadas y programadas.



ALTERNATIVAS: **GOOGLE AUTO**

2. Google auto: <https://github.com/google/auto>

Es un conjunto de herramientas orientadas a generar código fuente para Java.

Los subproyectos de Auto son una colección de generadores de código que automatizan las labores de generación de código repetitivo.

ALTERNATIVAS: **GOOGLE AUTO**

- **AutoFactory**: factoría compatible con JSR-330
 - <https://github.com/google/auto/tree/master/factory>
- **AutoService**: provee archivos de configuración para ServiceLoader
 - <https://github.com/google/auto/tree/master/service>
- **AutoValue**: generación de código para Java 7+.
 - <https://github.com/google/auto/tree/master/value>
- **Common**: utilidades para escribir procesadores de anotaciones.
 - <https://github.com/google/auto/tree/master/common>

ALTERNATIVAS: INMUTABLES

3. Inmutables: <https://immutables.github.io>

Es un procesador de anotaciones Java para generar objetos de valor simples, seguros y consistentes.

ALTERNATIVAS: SCALA

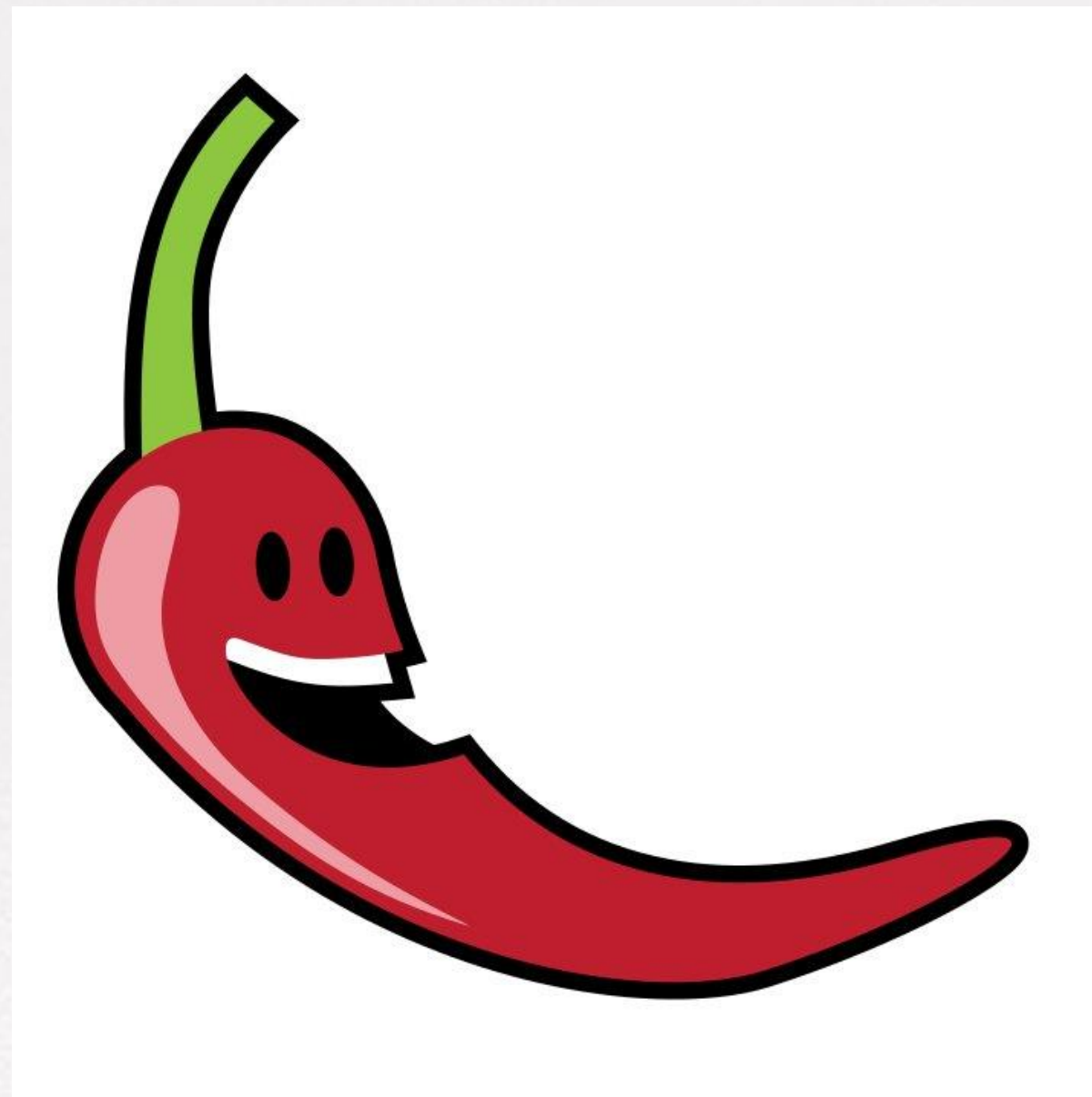
4. Scala: <https://www.scala-lang.org>

Combina programación funcional y orientada a objetos en un lenguaje conciso y de alto nivel.

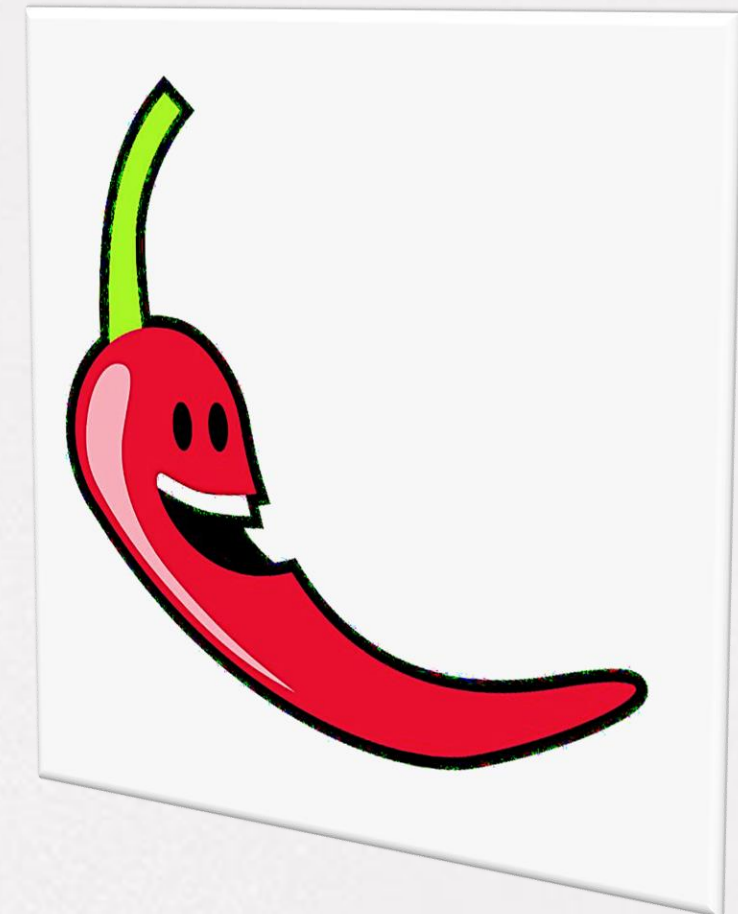
Orientado a evitar errores en aplicaciones complejas.

Los tiempos de ejecución de JVM permiten construir sistemas de alto rendimiento.

INSTALACIÓN DE LOMBOK

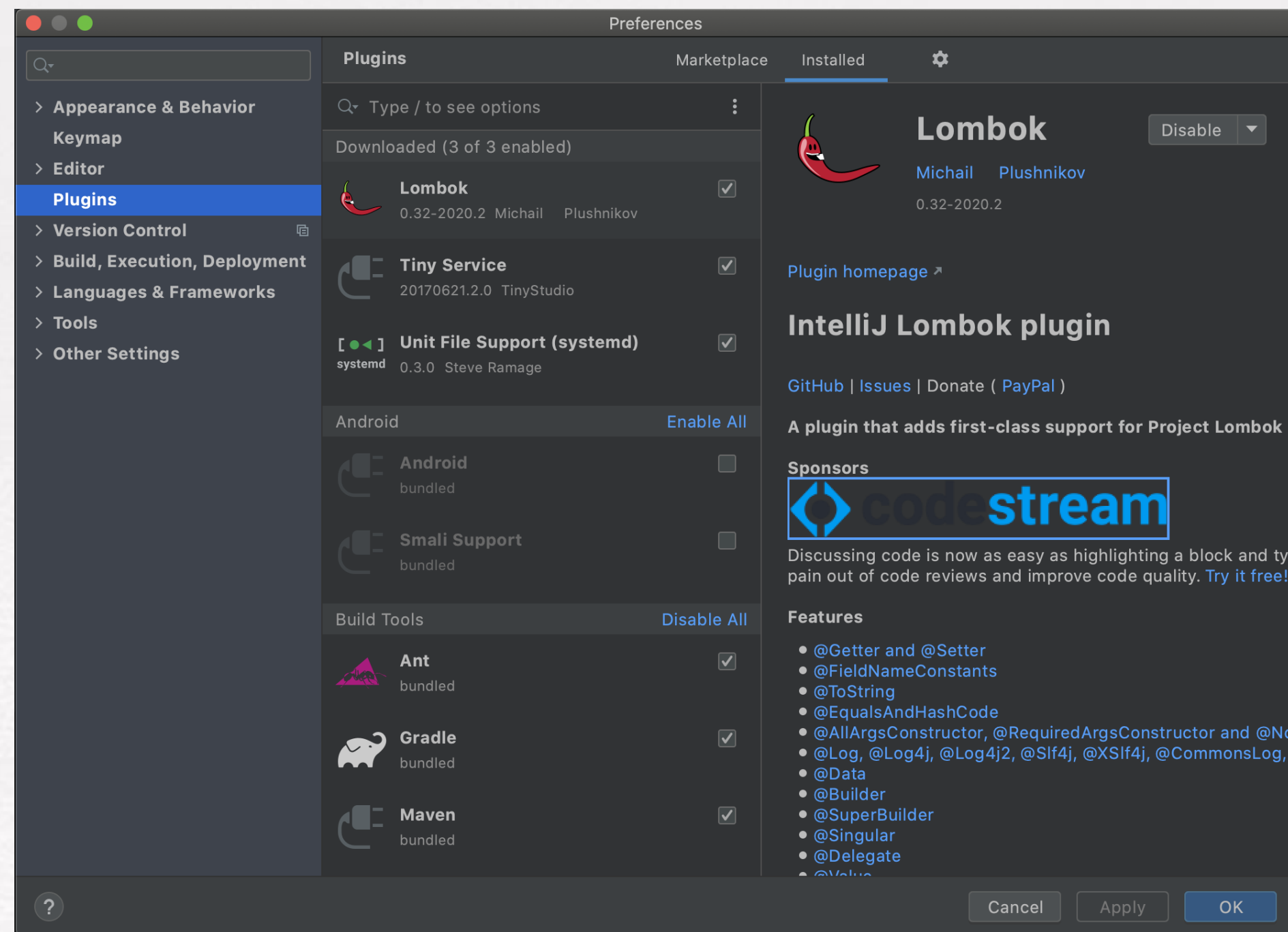


- Introducción a Lombok
- Alternativas a Lombok
- **Instalación**
- Anotaciones
- Ejercicio práctico
- Cierre



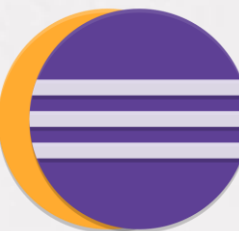
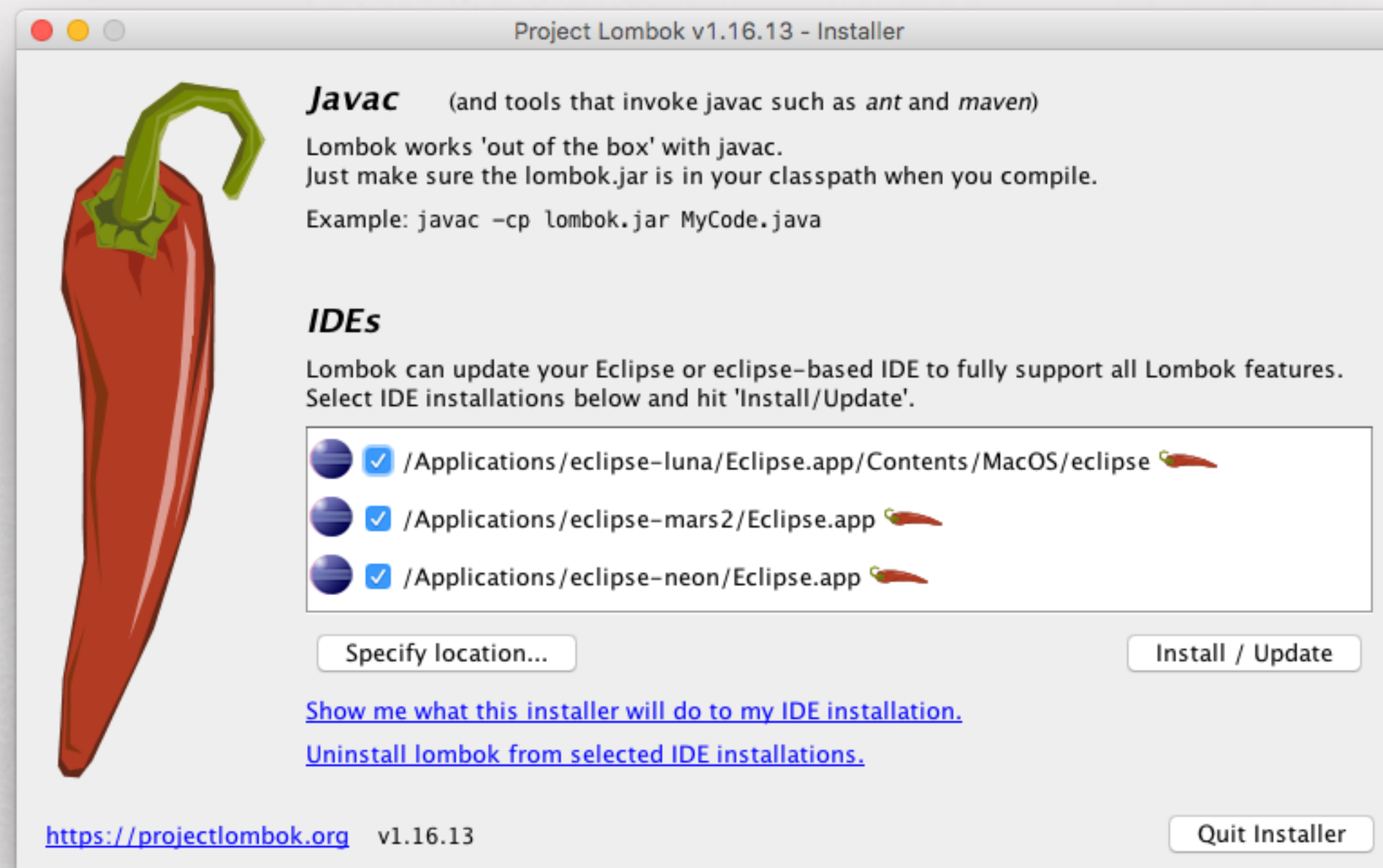
INSTALACIÓN DE LOMBOK

1. IntelliJ: <https://projectlombok.org/setup/intellij>



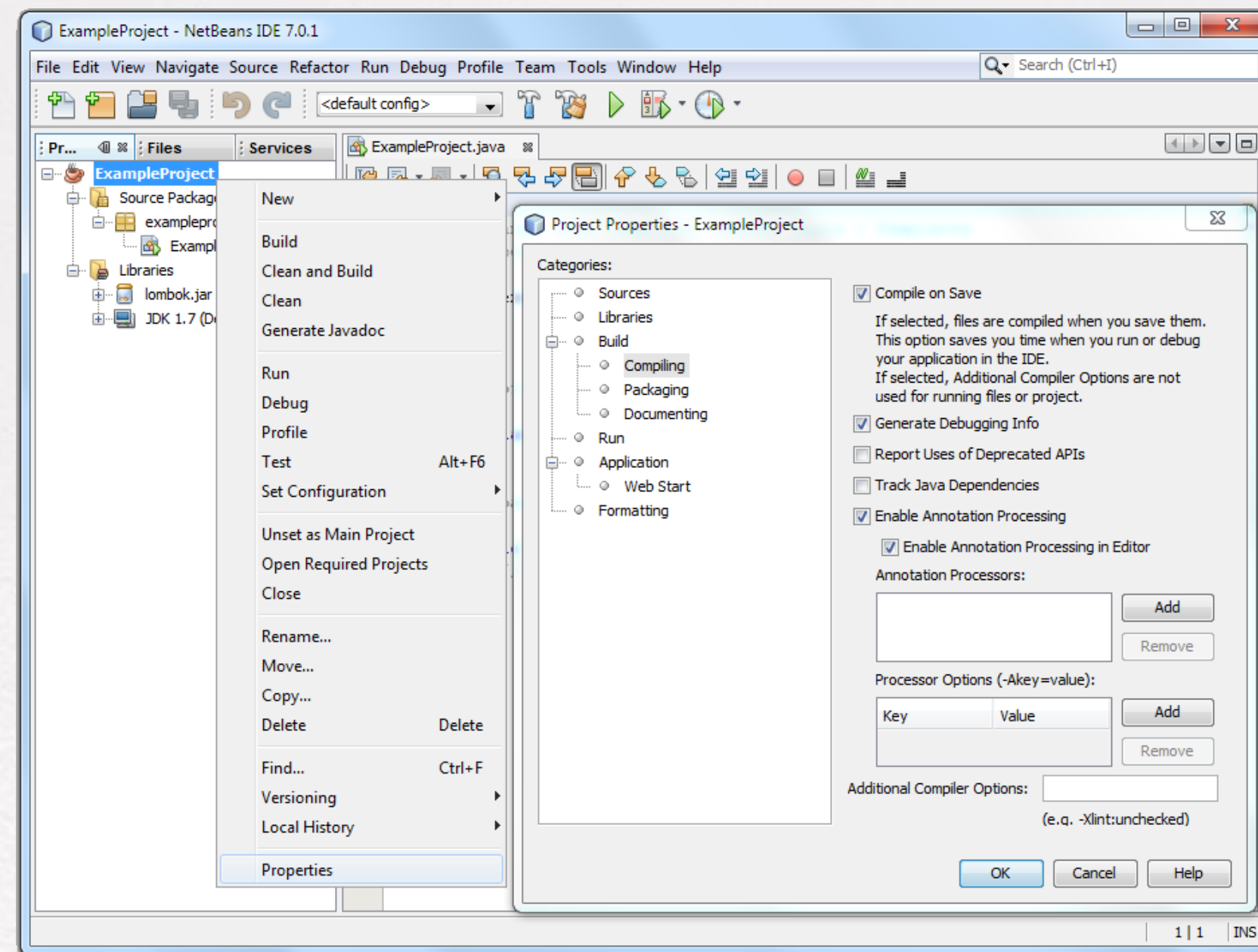
INSTALACIÓN DE LOMBOK

2. Eclipse: <https://projectlombok.org/setup/eclipse>

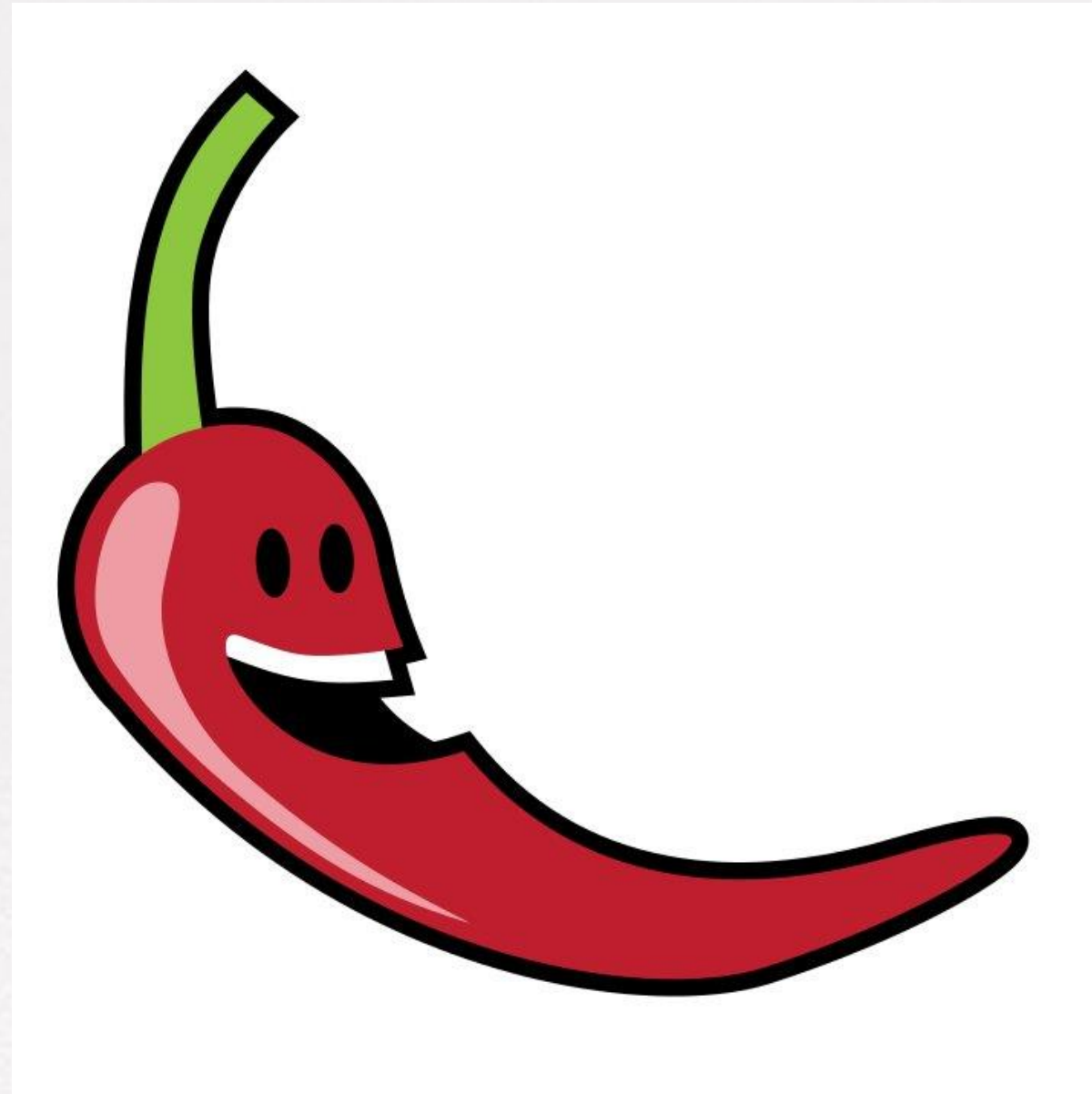


INSTALACIÓN DE LOMBOK

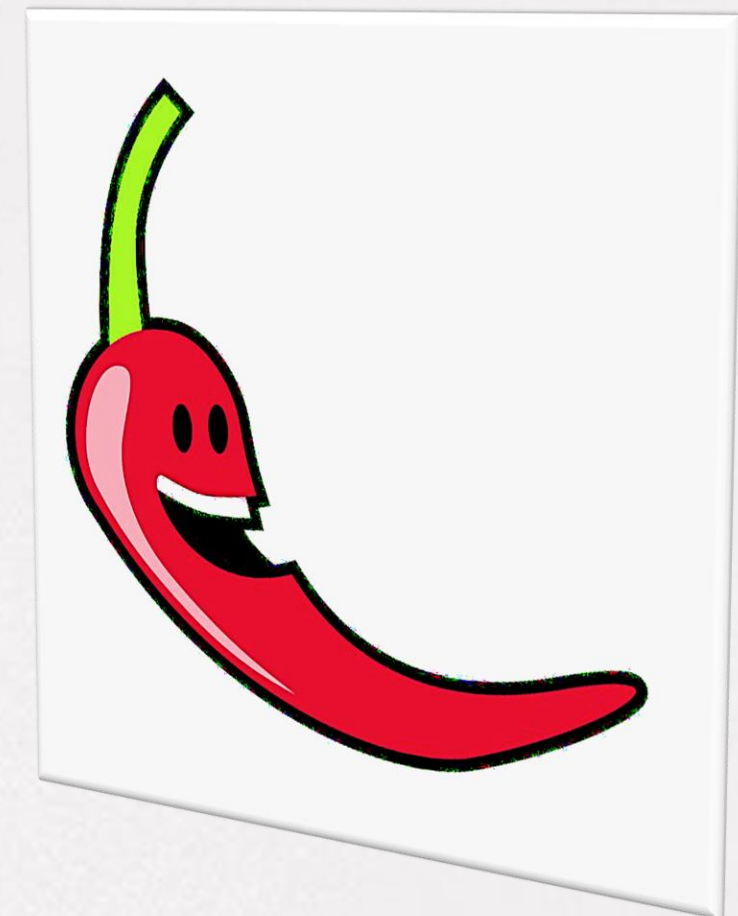
3. NetBeans: <https://projectlombok.org/setup/netbeans>



PRIMERAS ANOTACIONES



- Introducción a Lombok
- Alternativas a Lombok
- Instalación
- **Anotaciones**
- Ejercicio práctico
- Cierre



CONSTANTES **val**

<https://projectlombok.org/features/val>

val se emplea como tipo de declaración de variable local **constante**.

El tipo se infiere de la expresión del inicializador.

Ámbito: variables locales y bucles foreach, no en campos globales.

val es un tipo de objeto y existe como una clase real en el paquete lombok.
Se debe importar el paquete lombok.val.

¡¡Esta característica no funciona actualmente en NetBeans!!

VARIABLES **var**

<https://projectlombok.org/features/var>

Mismo comportamiento que “val”

La diferencia radica en que no crea una variable *final*.

Importante:

```
var x = "Hello";  
x = Color.RED; //Does not work;
```

ANOTACION **NonNull**

<https://projectlombok.org/features/NonNull>

@NonNull se usa en el parámetro de un método/constructor para que Lombok genere una declaración de **verificación nula**.

```
if (person == null) {  
    throw new NullPointerException("person is marked @NonNull but is null");  
}
```

Métodos: se insertará en la parte superior.

Constructores: se insertará después de cualquier llamada this()/super()

ANOTACION Cleanup

<https://projectlombok.org/features/Cleanup>

@Cleanup asegura que un recurso determinado se **limpie** automáticamente antes de que se deje de usar.

@Cleanup

```
InputStream in = new FileInputStream ("algunos /  
archivo");
```

ANOTACION Cleanup

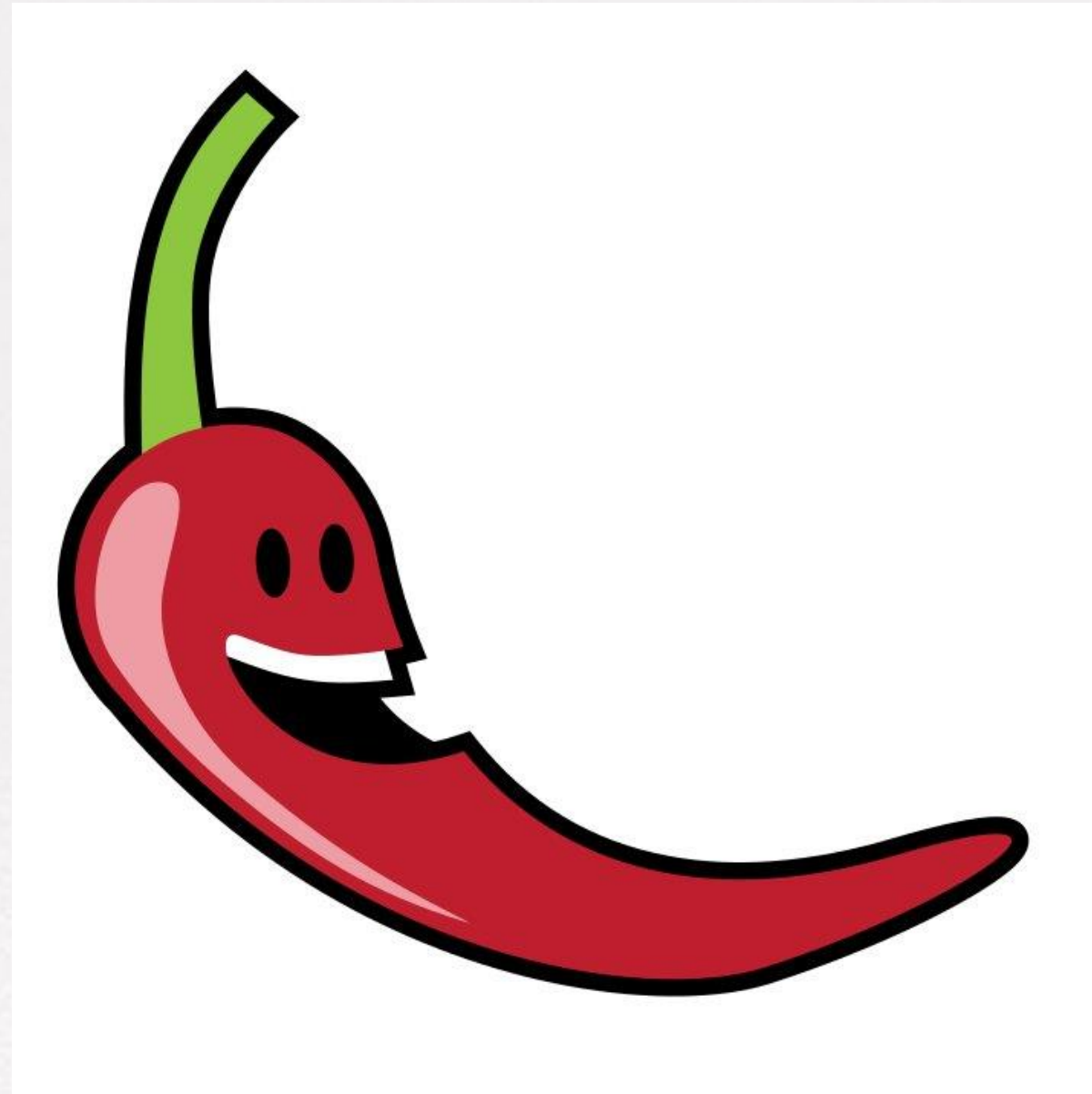
<https://projectlombok.org/features/Cleanup>

Como resultado se asegura llamar

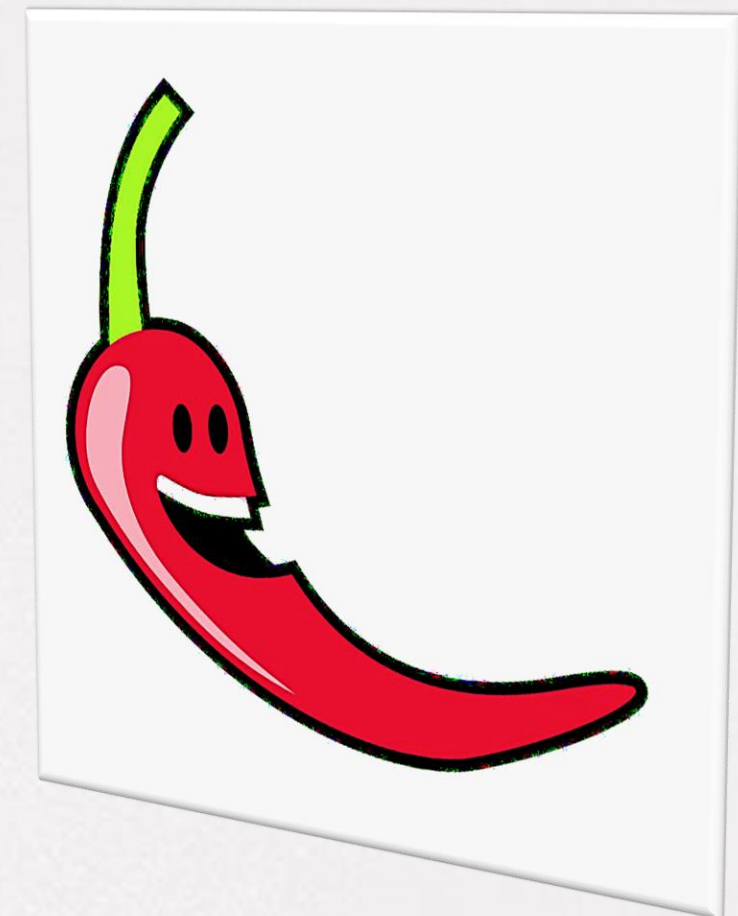
```
in.close()
```

Siempre se llamará dentro de un **try/finally**.

EJERCICIO PRÁCTICO



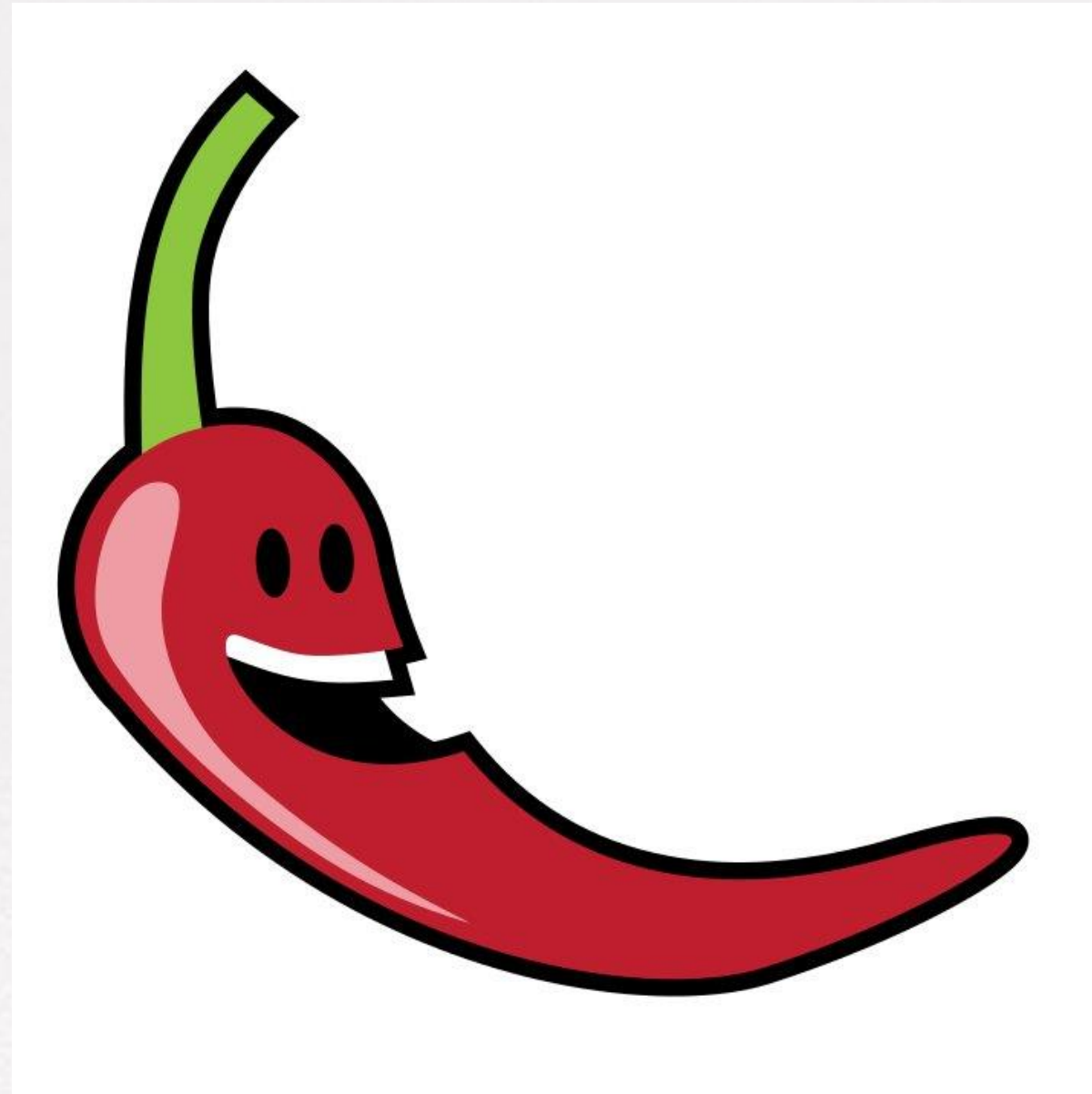
- Introducción a Lombok
- Alternativas a Lombok
- Instalación
- Anotaciones
- **Ejercicio práctico**
- Cierre



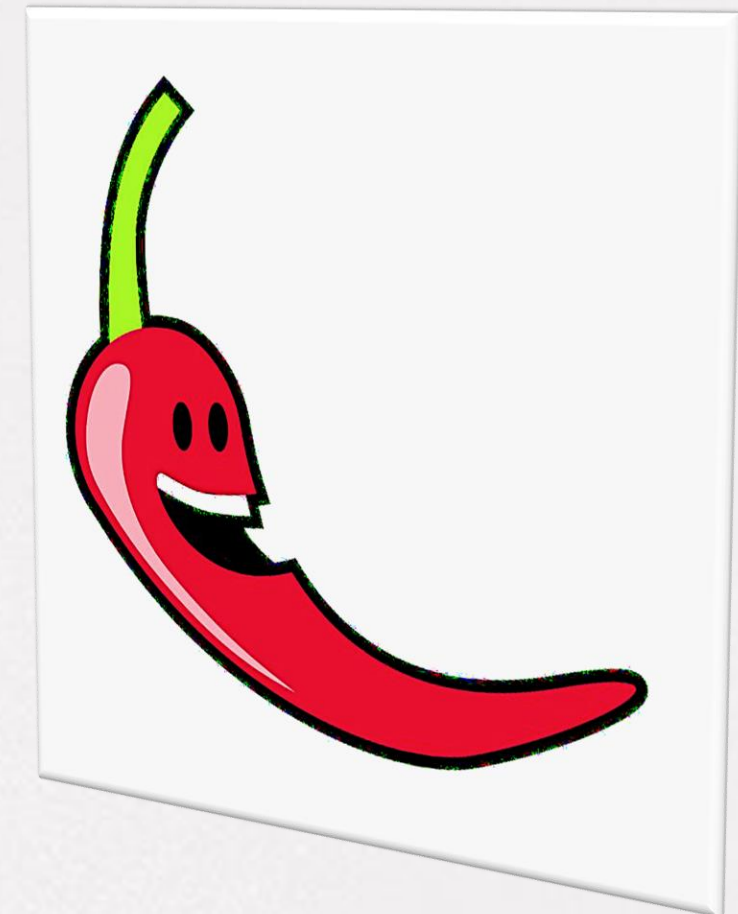
PRÁCTICA #1

- Diseña e implementa una aplicación que realice una búsqueda binaria y
 - <https://www.geeksforgeeks.org/binary-search/>
- Los resultados deben guardarse en un archivo
 - Poniendo el elemento que se ha encontrado
 - En qué posición está sobre qué lista
 - La hora
- Valida que no se puede guardar nada nulo

CIERRE DEL CURSO



- Introducción a Lombok
- Alternativas a Lombok
- Instalación
- Anotaciones
- Ejercicio práctico
- Cierre

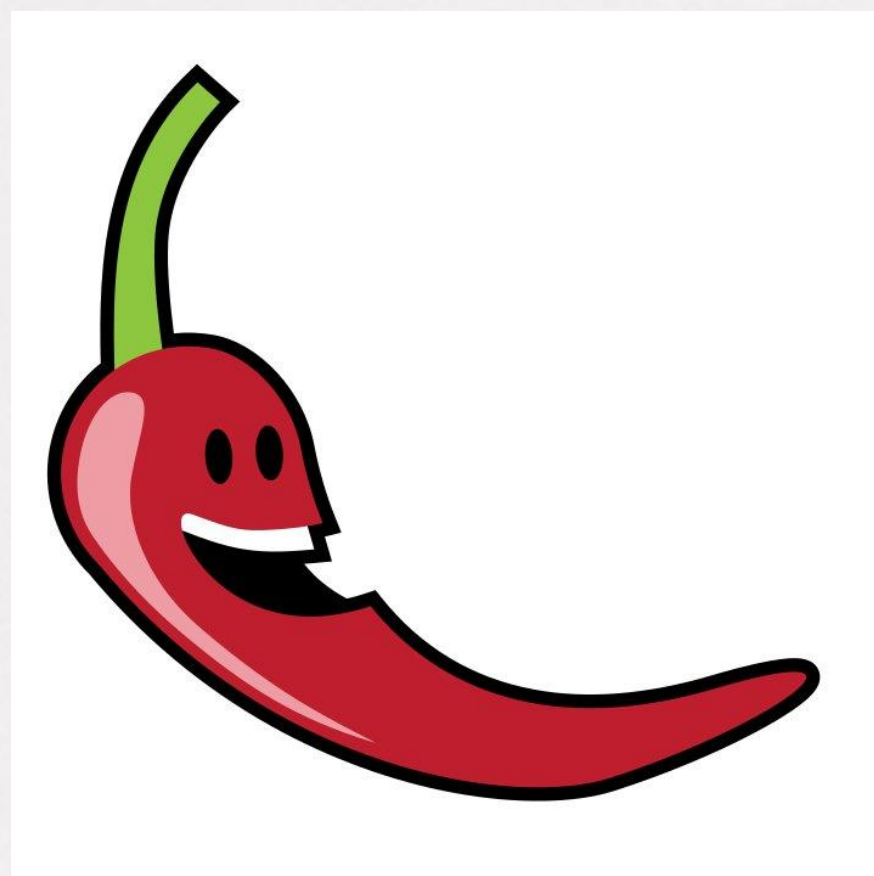


CIERRE DEL CURSO

- Vimos que es Lombok
- Algunas alternativas
- Nos iniciamos con las variables y constantes Lombok
- Vimos dos anotaciones
- Pusimos todo en práctica con un ejercicio

PRÓXIMOS CURSOS

- Las anotaciones Lombok orientadas a objetos
- Las anotaciones Lombok para constructores, equals y hash code
- Construye objetos variables y constantes, excepciones, sincronización y logs



¿PREGUNTAS?

¡HASTA PRONTO!

