

# Implementación

## Instrucciones

Para la construcción del proyecto se ha utilizado *Maven* de tal manera que para la ejecución del código fuente será necesario que esté instalado. En la siguiente imagen se muestra los detalles de la versión instalada en el computador donde se desarrolló el prototipo:



```
fabian@fabian-laptop:~/Desktop/gs-rest-service/initial$ mvn -v
Apache Maven 3.5.2
Maven home: /usr/share/maven
Java version: 1.8.0_181, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-oracle/jre
Default locale: es_ES, platform encoding: UTF-8
OS name: "linux", version: "4.15.0-43-generic", arch: "amd64", family: "unix"
```

Una vez se esté en la ruta que se logra visualizar en la imagen anterior, entonces será posible comenzar a ejecutar el software.

Inicialmente se deben descargar todas las dependencias que se han utilizado en el proyecto por lo cual será necesario correr el comando:

```
mvn install
```

Una vez las dependencias terminen de descargar entonces podremos construir y correr el proyecto. El *framework* que se ha utilizado es *Spring* por lo cual el comando para ejecutar el proyecto debe ser:

```
mvn spring-boot:run
```

Para realizar las respectivas pruebas se ha utilizado la herramienta *curl* con la cual se envían las peticiones *HTTP* para poder verificar las funcionalidades, por lo cual será necesario que esté en el computador para la verificación.

**Nota Importante:** en la implementación del prototipo solicitado se podrá notar que difiere del Diseño planteado para las funcionalidades: “registrar una opinión” y “eliminar una opinión”; esto se debe a inconvenientes en el desarrollo que por cuestiones de tiempo no los he solucionado, por lo cual he preferido entregar un MVP con las funcionalidades solicitadas así no coincidan ambas propuestas.

### - registrar una opinión

Para esta funcionalidad se ha nombrado el recurso: *registrarOpinion* y seguido de este se adicionan los respectivos atributos que componen una *opinión* según los requerimientos especificados:

- **id:** correspondiente al identificador del comentario que para las pruebas se asignará manualmente.

- **fecha:** fecha en la que es creado el comentario, para este prototipo se tiene solamente información del año, mes y día.
- **usuarioid:** correspondiente al identificador del usuario que ha generado la *opinión*.
- **tiendaid:** correspondiente al identificados de la tienda a la cual se le está generando la *opinión*.
- **comprald:** este campo hace referencia al serial de la orden, que puede llamarse también *número de orden*.
- **puntuacion:** valor asignado a la satisfacción del servicio.
- **comentario:** información adicional que el usuario desee adicionar.

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://localhost:8080/registrarOpinion?id=7\&fecha=2011-11-10\&usuarioid=111\&tiendaid=
456\&comprald=333\&puntuacion=4\&comentario=rica%20comida
```

---

#### - **eliminar una opinión**

En esta funcionalidad se ha asignado el recurso: *eliminarOpinion* y como parámetro (*opinionId*) es necesario que se especifique el identificador de la opinión que se desea eliminar. El error que se especifica en esta funcionalidad es sólo en caso de que con el ID indicado no se encuentre ninguna coincidencia.

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://localhost:8080/eliminarOpinion?opinionId=1
```

---

#### - **obtener la opinión para una compra**

Para la funcionalidad en mención se ha asignado el recurso: *opinion* en donde espera un parámetro *comprald* con el que se hará la respectiva búsqueda para asociar la opinión.

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://localhost:8080/opinion?comprald=789
```

---

#### - **obtener las opiniones para una tienda en un rango de fechas**

El recurso: *opiniones* es el que se utiliza para esta funcionalidad y adicionalmente se esperan los siguientes parámetros:

- **tiendaid:** hace referencia al identificador de la tienda de la cual se desea conocer las opiniones.
- **fechalni:** correspondiente a la fecha de inicio del lapso en el que se desean consultar las opiniones. Debe estar escrita en formato *aaaa-mm-dd*.

- **fechaFin:** correspondiente a la fecha final del lapso en el que se desean consultar las opiniones. Debe estar escrita en formato *aaaa-mm-dd*.

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET  
http://localhost:8080/opiniones?tiendaId=456\&fechaIni=2011-01-01\&fechaFin=2019-12-22
```

Según lo especificado en el enunciado no se debía construir ningún tipo de persistencia por lo cual todos los datos son volátiles y adicional se crean unos datos de prueba dentro del mismo software.