

Uso de Spring Initializr

Pregunta: Identifica y explica cuales son los archivos que creó Spring Initializr en este proyecto.

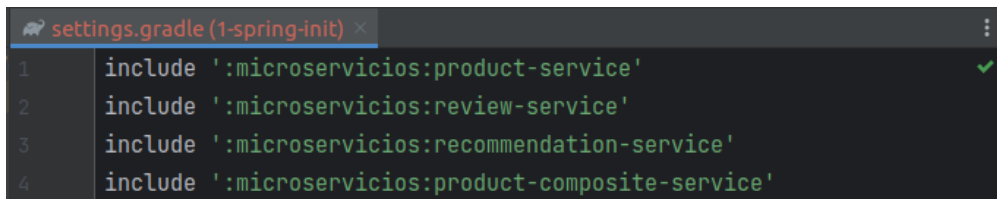
Spring Initializr creó los siguientes archivos:

- ProductServiceApplication.java: Clase de aplicación, inicia la aplicación desde el método main.
- application.properties: Se usa para definir propiedades relacionadas a la aplicación.
- ProductServiceApplicationTests.java: Define las pruebas a realizar de la aplicación.
- .gitignore: Indica los archivos que no se incluirán en el sistema de control de versiones Git.
- gradle-wrapper.jar: Contiene código para descargar la distribución de Gradle especificada en el archivo gradle-wrapper.properties.
- gradle-wrapper.properties: Almacena información acerca de la distribución de Gradle
- settings.gradle: Indica el nombre del proyecto, en este caso product-service
- gradlew: Es un script que invoca una versión de Gradle declarada, descargándola de antemano si es necesario.
- build.gradle: Describe como construir el proyecto.

Configuración de compilaciones de proyectos múltiples en Gradle

Pregunta: comprueba todos estos resultados y muestra los resultados obtenidos.

Creación de archivo settings.gradle:



```
1 include ':microservicios:product-service'
2 include ':microservicios:review-service'
3 include ':microservicios:recommendation-service'
4 include ':microservicios:product-composite-service'
```

Copia de archivos

ejecutables de Gradle:

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/1-spring-init$ cp -r microservicios/product-service/gradle .
cp microservicios/product-service/gradlew .
cp microservicios/product-service/.gitignore .
```

Eliminación de archivos ejecutables de Gradle generados en cada proyecto:

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/1-spring-init$ find microservicios -depth -name "gradle" -exec rm -rfv "{}" \;
find microservicios -depth -name "gradlew*" -exec rm -fv "{}" \;
removed 'microservicios/product-service/gradle/wrapper/gradle-wrapper.jar'
removed 'microservicios/product-service/gradle/wrapper/gradle-wrapper.properties'
removed directory 'microservicios/product-service/gradle/wrapper'
removed directory 'microservicios/product-service/gradle'
removed 'microservicios/review-service/gradle/wrapper/gradle-wrapper.jar'
removed 'microservicios/review-service/gradle/wrapper/gradle-wrapper.properties'
removed directory 'microservicios/review-service/gradle/wrapper'
removed directory 'microservicios/review-service/gradle'
removed 'microservicios/recommendation-service/gradle/wrapper/gradle-wrapper.jar'
removed 'microservicios/recommendation-service/gradle/wrapper/gradle-wrapper.properties'
removed directory 'microservicios/recommendation-service/gradle/wrapper'
removed directory 'microservicios/recommendation-service/gradle'
removed 'microservicios/product-composite-service/gradle/wrapper/gradle-wrapper.jar'
removed 'microservicios/product-composite-service/gradle/wrapper/gradle-wrapper.properties'
removed directory 'microservicios/product-composite-service/gradle/wrapper'
removed directory 'microservicios/product-composite-service/gradle'
removed 'microservicios/product-service/gradlew'
removed 'microservicios/review-service/gradlew'
removed 'microservicios/recommendation-service/gradlew'
removed 'microservicios/product-composite-service/gradlew'
```

Construcción del proyecto con un solo comando:

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/1-spring-init$ ./gradlew build

> Task :microservicios:product-composite-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

> Task :microservicios:product-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

> Task :microservicios:recommendation-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

> Task :microservicios:review-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

BUILD SUCCESSFUL in 30s
28 actionable tasks: 28 executed
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/1-spring-init$
```

Agregar API RESTful

Pregunta: Explica el funcionamiento de la declaración de la interfaz Java.

```
public interface ProductService {

    @GetMapping(
        value = "/product/{productId}",
        produces = "application/json")
    Product getProduct(@PathVariable int productId);
}
```

Esta interfaz declara un método abstracto `getProduct`, el cual requiere como argumento un `productId` y retorna un objeto `Product`. Este método es llamado cuando se ingresa a la dirección `"/product/{productId}"` y retorna un archivo JSON.

Implementando la API

Pregunta: Realiza los mismos procedimientos para los otros servicios de la actividad.

Para recommendation service:

1.- Se agrega los proyectos api y util como dependencias del archivo build.gradle

```
recommendation-service/build.gradle
1  plugins {
2      id 'java'
3      id 'org.springframework.boot' version '3.1.1'
4      id 'io.spring.dependency-management' version '1.1.0'
5  }
6
7  group = 'com.kapumota.microservicios.core.recommendation'
8  version = '0.0.1-SNAPSHOT'
9
10 java {
11     sourceCompatibility = '17'
12 }
13
14 repositories {
15     mavenCentral()
16 }
17
18 dependencies {
19     implementation project(':api')
20     implementation project(':util')
```

2.- Se agrega la anotación @ComponentScan a la clase de aplicación main:

```
RecommendationServiceApplication.java x
1 package com.kapumota.microservicios.core.recommendation;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.context.annotation.ComponentScan;
6
7 @SpringBootApplication
8 @ComponentScan("com.kapumota")
9 public class RecommendationServiceApplication {
10     public static void main(String[] args) {
11
12         SpringApplication.run(RecommendationServiceApplication.class, args);
13     }
14 }
```

3.- Se crea la clase de implementación del servicio, RecommendationServiceImpl.java

```
@RestController
public class RecommendationServiceImpl implements RecommendationService {

    3 usages
    private final ServiceUtil serviceUtil;

    no usages
    @Autowired
    public RecommendationServiceImpl(ServiceUtil serviceUtil) {
        this.serviceUtil = serviceUtil;
    }

    no usages
    @Override
    public List<Recommendation> getRecommendations(int productId) {
        List<Recommendation> recommendations = new ArrayList<>();
        Recommendation recommendation1 = new Recommendation(productId, recommendationId: 1,
            author: "Carlos", rate: 4, content: "bueno", serviceUtil.getServiceAddress());
        Recommendation recommendation2 = new Recommendation(productId, recommendationId: 2,
            author: "Juan", rate: 3, content: "regular", serviceUtil.getServiceAddress());
        recommendations.add(recommendation1);
        recommendations.add(recommendation2);
        return recommendations;
    }
}
```

4.- Se crea el archivo `application.yml` dentro del directorio `resources`, asignando el puerto 7003 al servicio de recomendación:

```
1 server.port: 7003
2 server.error.include-message: always
3
4 logging:
5   level:
6     root: INFO
7     com.kapumota.microservicios: DEBUG
```

5.- Se crea e inicia el microservicio:

```
walterrrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ ./gradlew build
> Task :microservicios:recommendation-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

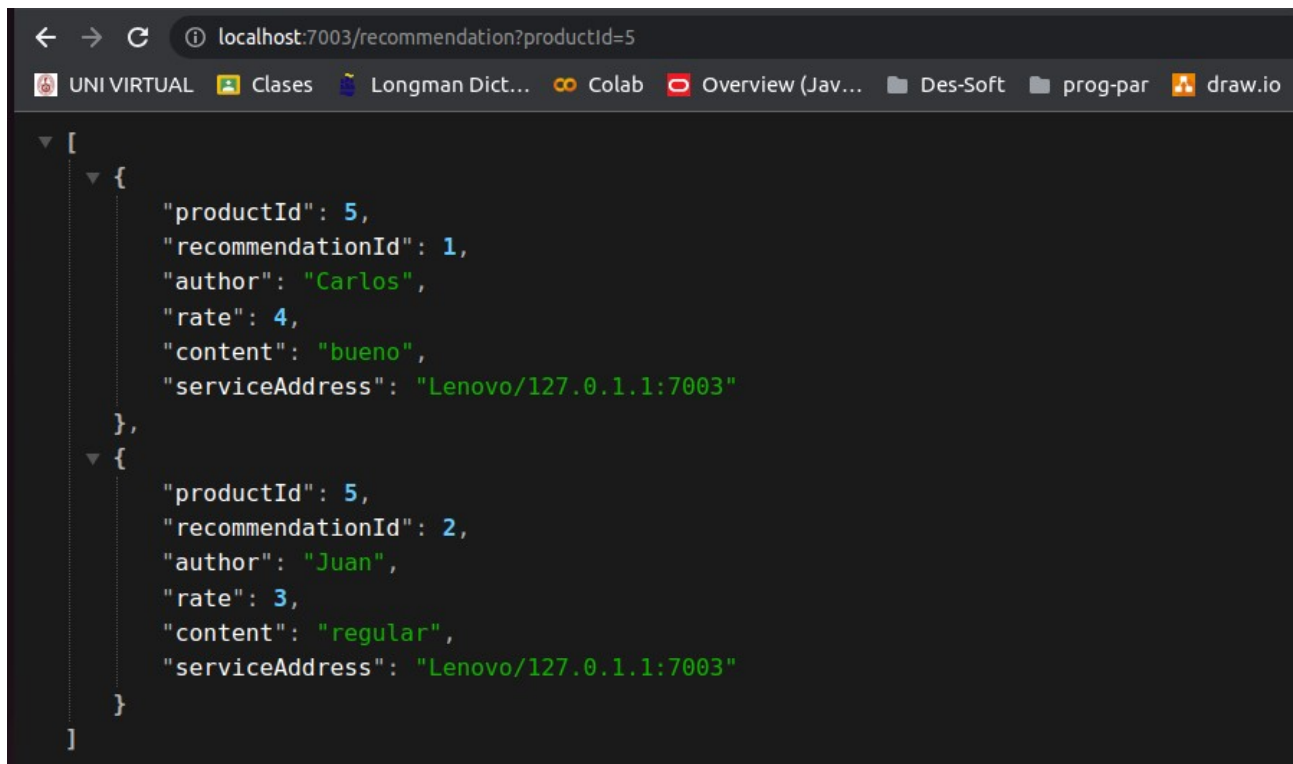
BUILD SUCCESSFUL in 11s
32 actionable tasks: 5 executed, 27 up-to-date
walterrrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ java -jar microservicios/recommendation-service/build/libs/*.jar &
[2] 14547
```

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$
```

```
. _--_      _    -- -  
/\ / ___'_ _ __ _(_)_ _ _ _ \\\ \  
( )\___| '_ |'_|| '|_ \|_ _ | \\ \\  
\V ____)|_|)|_|_|_|_|(|_|) )) )  
' |_|| ._|_| |_|_|_|_\__, | // / /  
=====|_|=====|____/_/_/_/  
  
:: Spring Boot ::                (v3.1.1)
```

```
2023-07-10T13:07:22.360+05:00 INFO 14547 --- [           main] k.m.c.r.RecommendationServiceApplica  
tion : Starting RecommendationServiceApplication v0.0.1-SNAPSHOT using Java 17.0.7 with PID 14547 (/home/walterrg/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos/microservicios/recommenda  
tion-service/build/libs/recommendation-service-0.0.1-SNAPSHOT.jar started by walterrg in /home/walterrg/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos)  
2023-07-10T13:07:22.369+05:00 DEBUG 14547 --- [           main] k.m.c.r.RecommendationServiceApplica  
tion : Running with Spring Boot v3.1.1, Spring v6.0.10  
2023-07-10T13:07:22.373+05:00 INFO 14547 --- [           main] k.m.c.r.RecommendationServiceApplica  
tion : No active profile set, falling back to 1 default profile: "default"  
2023-07-10T13:07:24.921+05:00 INFO 14547 --- [           main] o.s.b.a.e.web.EndpointLinksResolver  
       : Exposing 1 endpoint(s) beneath base path '/actuator'  
2023-07-10T13:07:25.660+05:00 INFO 14547 --- [           main] o.s.b.w.embedded.netty.NettyWebSer  
ver     : Netty started on port 7003  
2023-07-10T13:07:25.690+05:00 INFO 14547 --- [           main] k.m.c.r.RecommendationServiceApplica  
tion : Started RecommendationServiceApplication in 4.412 seconds (process running for 5.268)
```

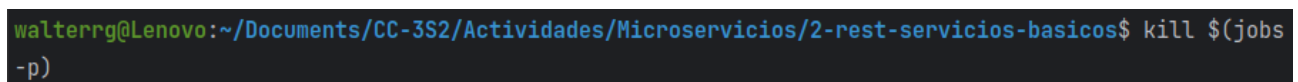
6.- Se realiza una llamada de prueba al servicio de recomendación:



The screenshot shows a web browser window with the address bar displaying `localhost:7003/recommendation?productId=5`. The browser's tab bar includes several tabs: 'UNI VIRTUAL', 'Clases', 'Longman Dict...', 'Colab', 'Overview (Jav...', 'Des-Soft', 'prog-par', and 'draw.io'. The main content area displays a JSON array of two recommendation objects. The first object has a product ID of 5, recommendation ID of 1, author 'Carlos', a rate of 4, content 'bueno', and service address 'Lenovo/127.0.1.1:7003'. The second object has a product ID of 5, recommendation ID of 2, author 'Juan', a rate of 3, content 'regular', and the same service address.

```
[
  {
    "productId": 5,
    "recommendationId": 1,
    "author": "Carlos",
    "rate": 4,
    "content": "bueno",
    "serviceAddress": "Lenovo/127.0.1.1:7003"
  },
  {
    "productId": 5,
    "recommendationId": 2,
    "author": "Juan",
    "rate": 3,
    "content": "regular",
    "serviceAddress": "Lenovo/127.0.1.1:7003"
  }
]
```

7.- Se detiene el servicio:

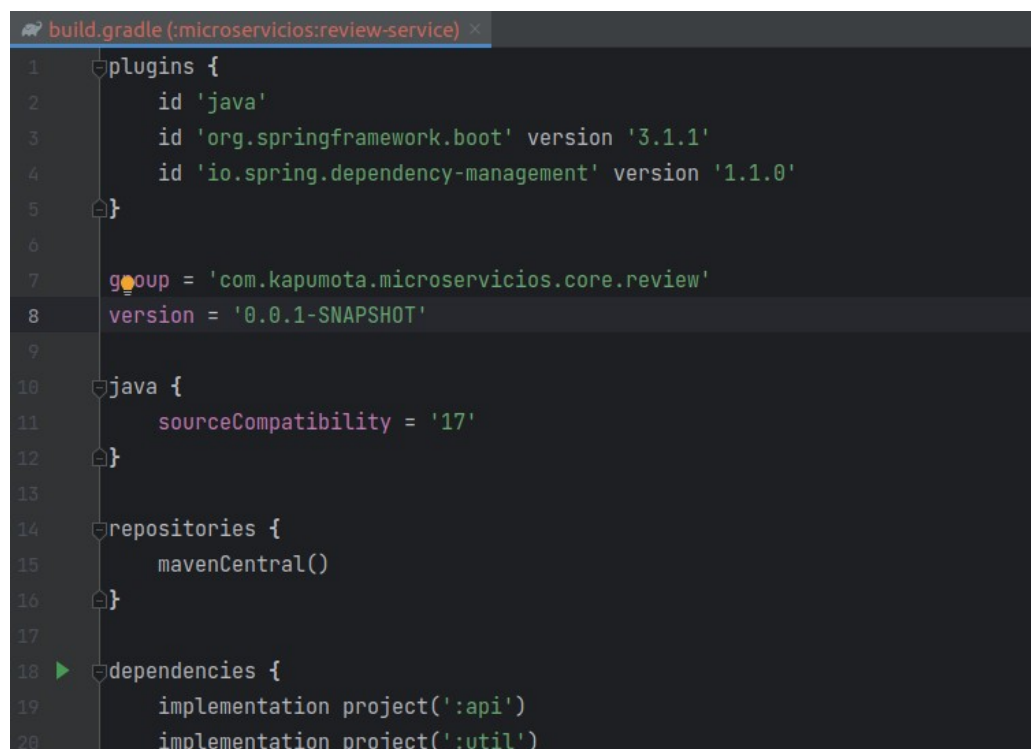


The screenshot shows a terminal window with the prompt `walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$`. The command `kill $(jobs -p)` has been entered to stop the running service.

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ kill $(jobs -p)
```

Se realiza los mismos pasos ara review service:

1.- Se agrega los proyectos api y util como dependencias del archivo build.gradle



The screenshot shows an IDE window titled `build.gradle (:microservicios:review-service)`. The file contains Gradle configuration for a Java project. It includes plugins for Java, Spring Boot, and Spring Dependency Management. It also defines a group, version, source compatibility, repositories, and dependencies. The dependencies section includes `implementation project(':api')` and `implementation project(':util')`.

```
1  plugins {
2      id 'java'
3      id 'org.springframework.boot' version '3.1.1'
4      id 'io.spring.dependency-management' version '1.1.0'
5  }
6
7  group = 'com.kapumota.microservicios.core.review'
8  version = '0.0.1-SNAPSHOT'
9
10 java {
11     sourceCompatibility = '17'
12 }
13
14 repositories {
15     mavenCentral()
16 }
17
18 dependencies {
19     implementation project(':api')
20     implementation project(':util')
```


2.- Se agrega la anotación @ComponentScan a la clase de aplicación main:

```
ReviewServiceApplication.java x
1 package com.kapumota.microservicios.core.review;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5 import org.springframework.context.annotation.ComponentScan;
6
7 @SpringBootApplication
8 @ComponentScan("com.kapumota")
9 public class ReviewServiceApplication {
10     public static void main(String[] args) {
11
12         SpringApplication.run(ReviewServiceApplication.class, args);
13     }
14 }
```

3.- Se crea la clase de implementación del servicio, ReviewServiceImpl.java

```
11 @RestController
12 public class ReviewServiceImpl implements ReviewService {
13
14     3 usages
15     private final ServiceUtil serviceUtil;
16
17     no usages
18     public ReviewServiceImpl(ServiceUtil serviceUtil) {
19         this.serviceUtil = serviceUtil;
20     }
21
22     no usages
23     @Override
24     public List<Review> getReviews(int productId) {
25         List<Review> reviews = new ArrayList<>();
26         Review review1 = new Review(productId, reviewId: 1, author: "Lucas",
27             subject: "Revisión de " + productId, content: "Buen producto",
28             serviceUtil.getServiceAddress());
29         Review review2 = new Review(productId, reviewId: 2, author: "Richard",
30             subject: "Revisión de " + productId, content: "Duradero",
31             serviceUtil.getServiceAddress());
32         reviews.add(review1);
33         reviews.add(review2);
34         return reviews;
35     }
36 }
```

4.- Se crea el archivo `application.yml` dentro del directorio `resources`, asignando el puerto 7002 al servicio de revisión:

```

1  server.port: 7002
2  server.error.include-message: always
3
4  logging:
5  level:
6    root: INFO
7    com.kapumota.microservicios: DEBUG

```

5.- Se crea e inicia el microservicio:

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ ./gradlew build
ild

> Task :microservicios:review-service:test
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap
p classpath has been appended

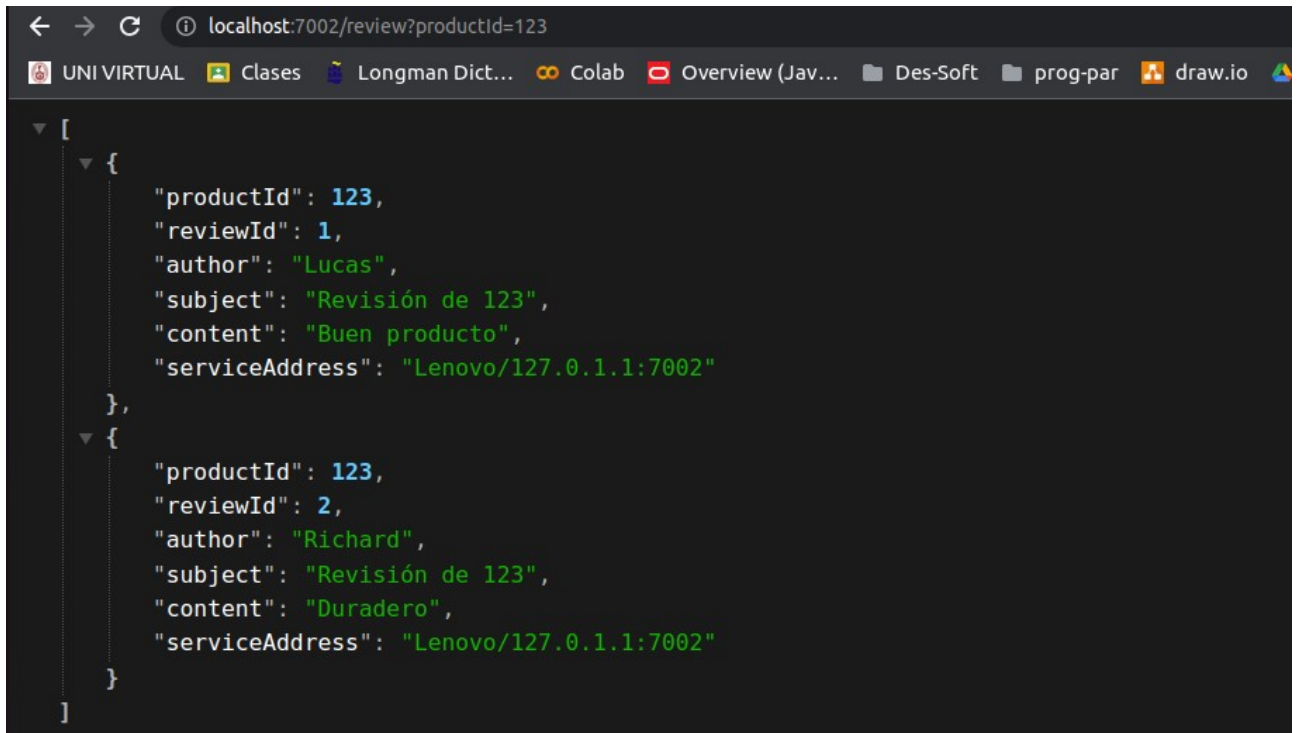
BUILD SUCCESSFUL in 13s
32 actionable tasks: 6 executed, 26 up-to-date
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ java -jar mi
croservicios/review-service/build/libs/*.jar &
[1] 16031
```

```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$
```

```
.   _--_      _    _--_ _  
/\ / ___'_  _ _ _(_)_ _ _ _ \ \ \ \  
( ) \___ | '_ | '_ | | '_ V _' | \ \ \  
\ \ / ___) | | | | | | | | ( _ | | ) ) )  
'  |____| .__|_| |_| | |_\__, | / / / /  
=====|_|=====|___/_/_/_/_/  
  
:: Spring Boot ::                (v3.1.1)
```

```
2023-07-10T14:15:12.231-05:00 INFO 16031 --- [          main] c.k.m.c.review.ReviewServiceApplicat  
ion : Starting ReviewServiceApplication v0.0.1-SNAPSHOT using Java 17.0.7 with PID 16031 (/home/wal  
terrg/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos/microservicios/review-ser  
vice/build/libs/review-service-0.0.1-SNAPSHOT.jar started by walterrg in /home/walterrg/Documents/CC  
-3S2/Actividades/Microservicios/2-rest-servicios-basicos)  
2023-07-10T14:15:12.236-05:00 DEBUG 16031 --- [          main] c.k.m.c.review.ReviewServiceApplicat  
ion : Running with Spring Boot v3.1.1, Spring v6.0.10  
2023-07-10T14:15:12.238-05:00 INFO 16031 --- [          main] c.k.m.c.review.ReviewServiceApplicat  
ion : No active profile set, falling back to 1 default profile: "default"  
2023-07-10T14:15:15.530-05:00 INFO 16031 --- [          main] o.s.b.a.e.web.EndpointLinksResolver  
 : Exposing 1 endpoint(s) beneath base path '/actuator'  
2023-07-10T14:15:16.365-05:00 INFO 16031 --- [          main] o.s.b.web.embedded.netty.NettyWebSer  
ver : Netty started on port 7002  
2023-07-10T14:15:16.407-05:00 INFO 16031 --- [          main] c.k.m.c.review.ReviewServiceApplicat  
ion : Started ReviewServiceApplication in 5.169 seconds (process running for 6.13)
```

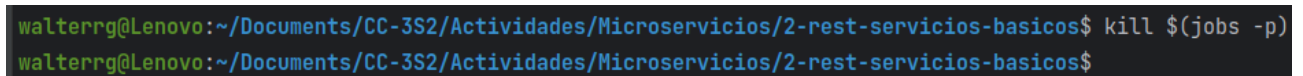
6.- Se realiza una llamada de prueba al servicio de revisión:



```
localhost:7002/review?productId=123
UNI VIRTUAL Clases Longman Dict... Colab Overview (Jav... Des-Soft prog-par draw.io

[
  {
    "productId": 123,
    "reviewId": 1,
    "author": "Lucas",
    "subject": "Revisión de 123",
    "content": "Buen producto",
    "serviceAddress": "Lenovo/127.0.1.1:7002"
  },
  {
    "productId": 123,
    "reviewId": 2,
    "author": "Richard",
    "subject": "Revisión de 123",
    "content": "Duradero",
    "serviceAddress": "Lenovo/127.0.1.1:7002"
  }
]
```

7.- Se detiene el servicio:



```
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$ kill $(jobs -p)
walterrg@Lenovo:~/Documents/CC-3S2/Actividades/Microservicios/2-rest-servicios-basicos$
```