



Documentación

GRUPO XX

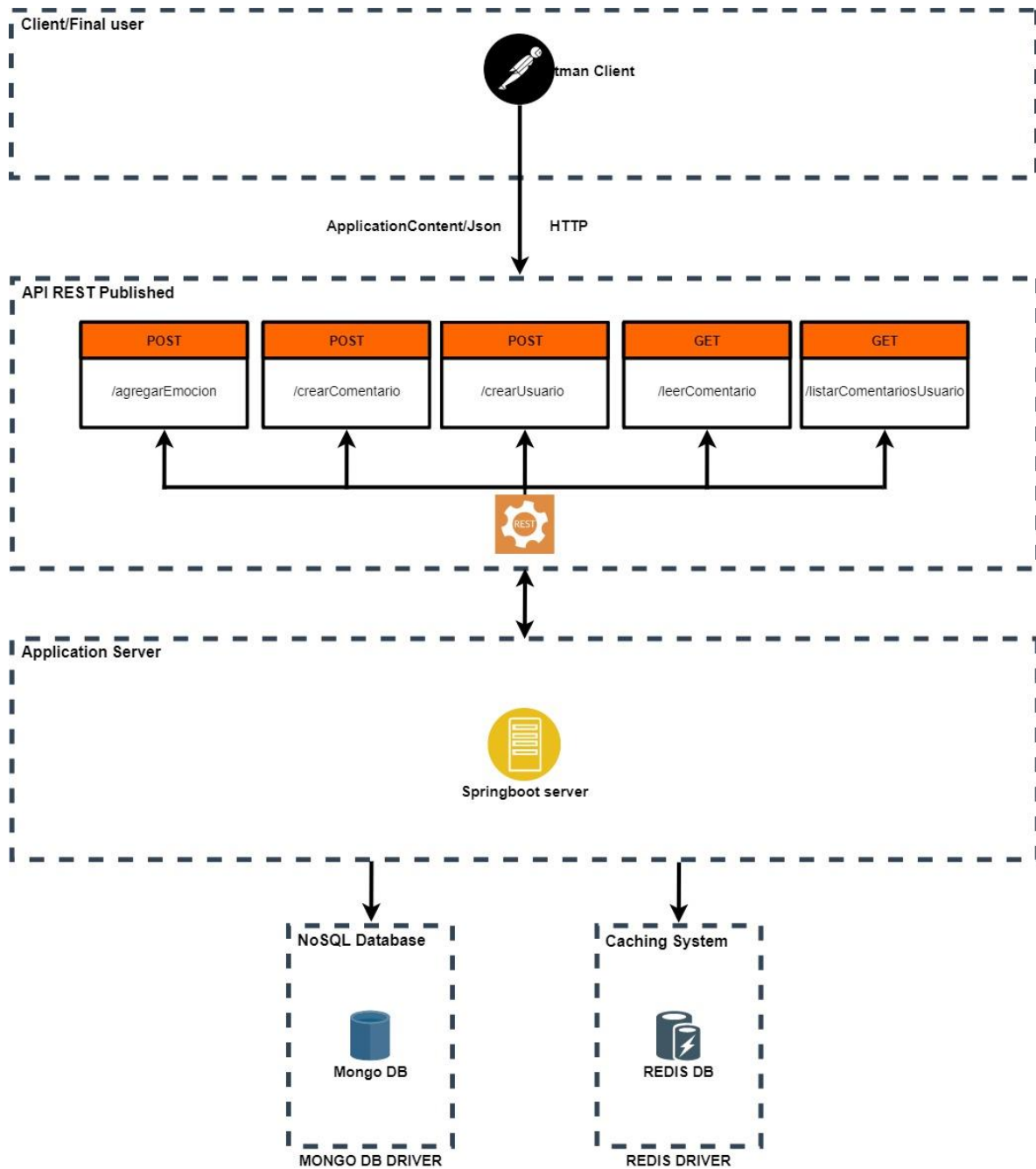
FEDERICO BENTANCOR
GONZALO SANTA MARÍA
FEDERICO SIERRA
FACUNDO LABORDE

Contenido

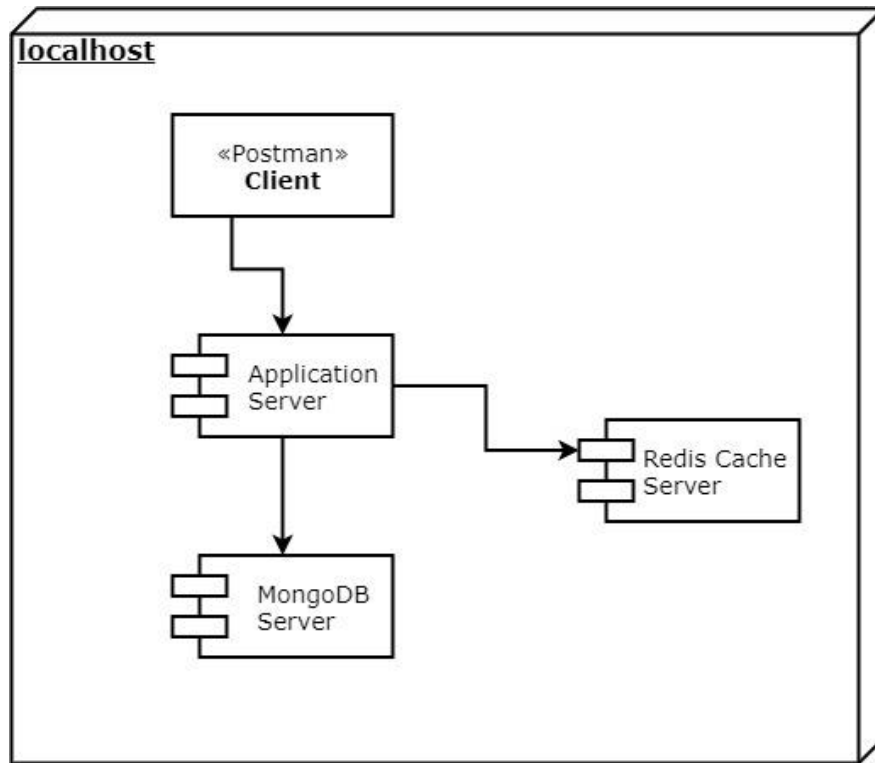
Diagramas.....	2
• Diagrama de arquitectura	2
• Diagrama de despliegue.....	3
Formato de intercambio de datos	4
• Métodos HTTP.....	4
○ POST:	4
○ GET:	4
Instalación y configuración	5
• Instalación	5
○ MongoDB.....	5
○ Redis	6
○ Postman	6
○ Aplicación	7
• Configuración	8
○ Redis.....	8
○ Postman	9
Plataforma, bases de datos, lenguaje y servidor	11
• Plataforma.....	11
• Base de datos	11
• Lenguaje	11
• Servidor de aplicaciones.....	11
Caché.....	11
• Estrategias para la implementación de caché.....	11

Diagramas

- Diagrama de arquitectura



- Diagrama de despliegue



Formato de intercambio de datos

- Métodos HTTP

- POST:

- [/agregarEmocion/{comId}/{email}/{emocion}](#)
 - comId (string)
 - email (string)
 - emocion (boolean)

- [/crearcomentario/{texto}/{email}](#)
 - texto (string)
 - email (string)

- [/crearusuario/{email}](#)
 - email (string)

- GET:

- [/leercomentario/{comId}](#)
 - comId (String)

Response:

```
{
  "cantMeGusta": int,
  "cantNoMeGusta": int,
  "comId": "string",
  "texto": "string",
  "usrId": "string"
}
```

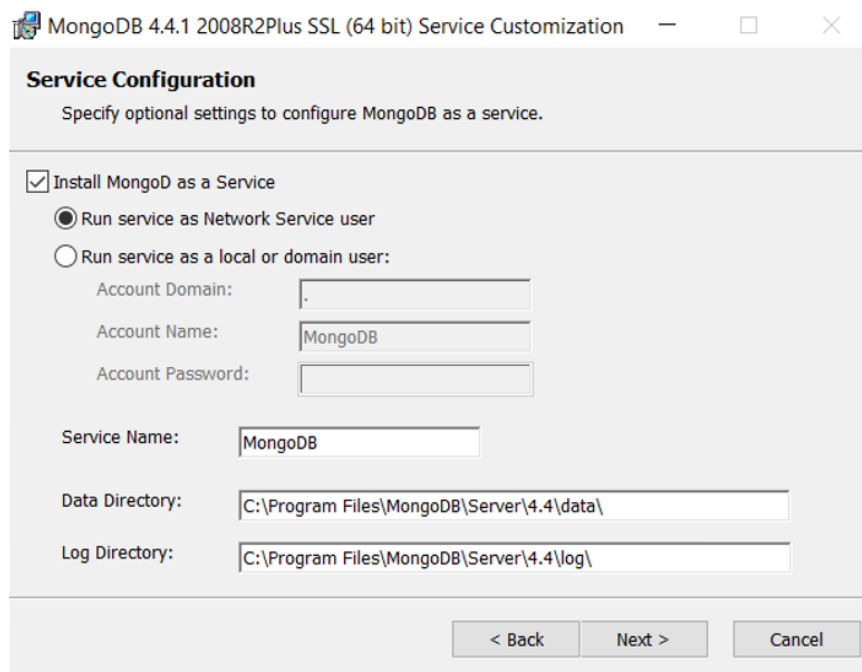
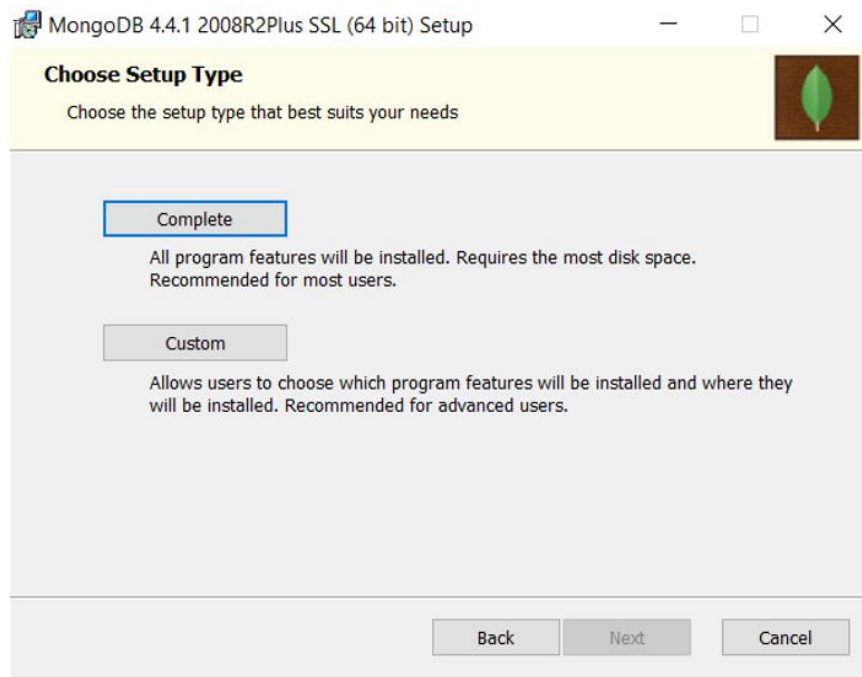
- [/listarcomentariosusuario/{email}](#)
 - email (String)

Response:

```
{
  "cantMeGusta": int,
  "cantNoMeGusta": int,
  "comId": "string",
  "texto": "string",
  "usrId": "string"
}
```

Instalación y configuración

- Instalación
 - MongoDB
 - <https://www.mongodb.com/try/download/community>



- Redis

- Requisitos:
 - Ubuntu 20.04 Terminal en Windows 10
- Instalación:
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
 - `sudo apt-get install redis-server`
 - `redis-cli -v`
- Levantar servicio:
 - `sudo service redis-server restart`
- Abrir terminal:
 - `$ redis-cli`

- Postman

- <https://www.postman.com/downloads/>
- Instalación no necesaria con la versión portable.

- Aplicación

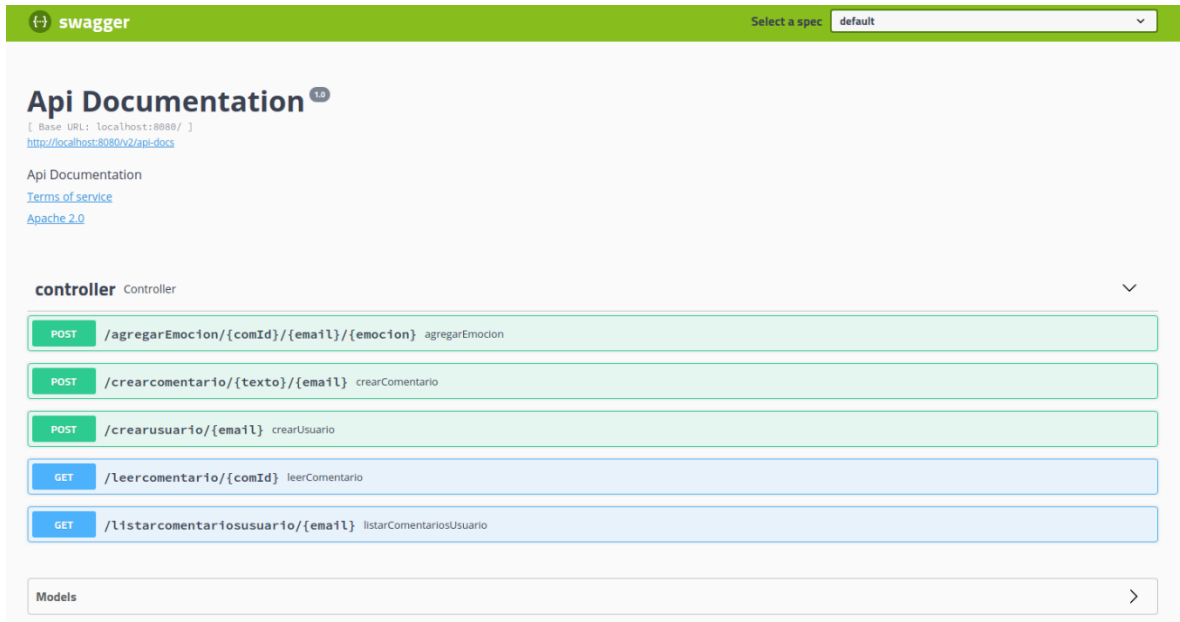
- Requisitos:
 - Maven
 - Spring
- Instalación:
 - `mvn spring-boot::run`

```
Federico@federico: ~/Documentos/tecnologo/noSQL/REPO/obligatorioNoSQL
Federico@federico:~/Documentos/tecnologo/noSQL/REPO/obligatorioNoSQL$ mvn spring-boot::run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:obligatorioNosql >-----
[INFO] Building obligatorioNosql 0.0.1-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] >>> spring-boot-maven-plugin:2.3.4.RELEASE:run (default-cli) > test-compile @ obligatorioNosql >>>
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ obligatorioNosql ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ obligatorioNosql ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ obligatorioNosql ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/federico/Documentos/tecnologo/noSQL/REPO/obligatorioNoSQL/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ obligatorioNosql ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< spring-boot-maven-plugin:2.3.4.RELEASE:run (default-cli) < test-compile @ obligatorioNosql <<<
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.4.RELEASE:run (default-cli) @ obligatorioNosql ---
[INFO] Attaching agents: []

Spring Boot
(v2.3.4.RELEASE)

2020-11-10 21:58:16.760 INFO 40714 --- [main] c.e.o.ObligatorioNosqlApplication : Starting ObligatorioNosqlApplication on federico with PID 40714 (/home/federico/Documentos/tecnologo/noSQL/REPO/obligatorioNoSQL/target/classes started by federico in /home/federico/Documentos/tecnologo/noSQL/REPO/obligatorioNoSQL)
2020-11-10 21:58:16.763 INFO 40714 --- [main] c.e.o.ObligatorioNosqlApplication : No active profile set, falling back to default
2020-11-10 21:58:17.200 INFO 40714 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2020-11-10 21:58:17.211 INFO 40714 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 8m
2020-11-10 21:58:17.483 INFO 40714 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
```

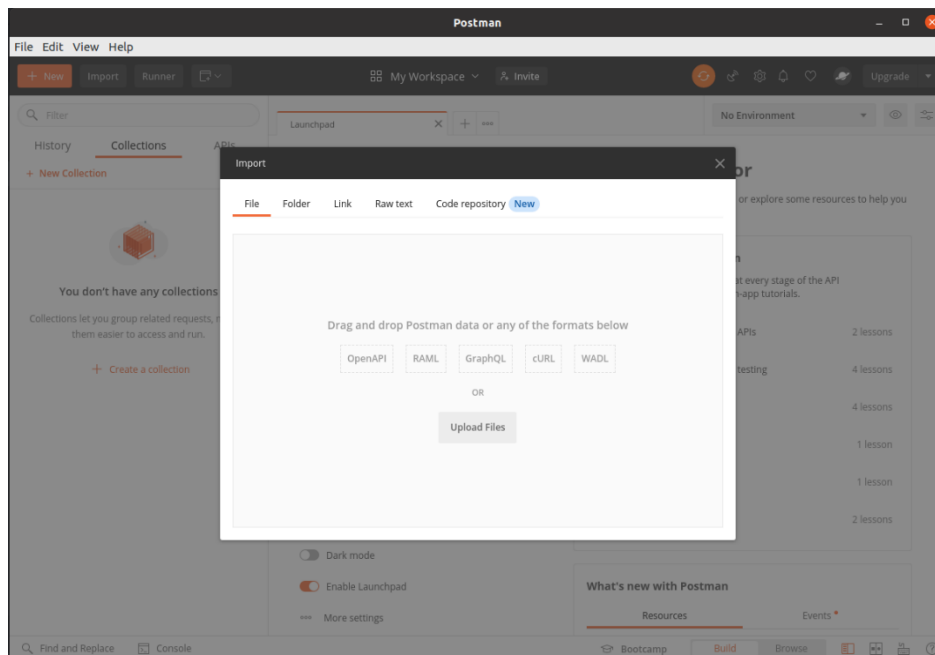
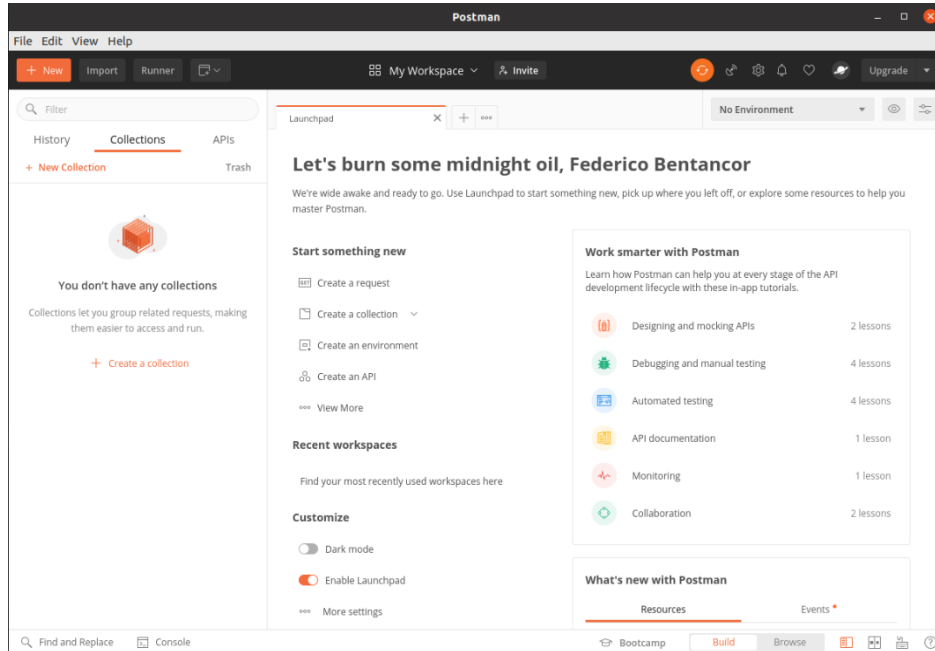

- Se accede a la aplicación por medio del enlace:
<http://localhost:8080/swagger-ui.html>

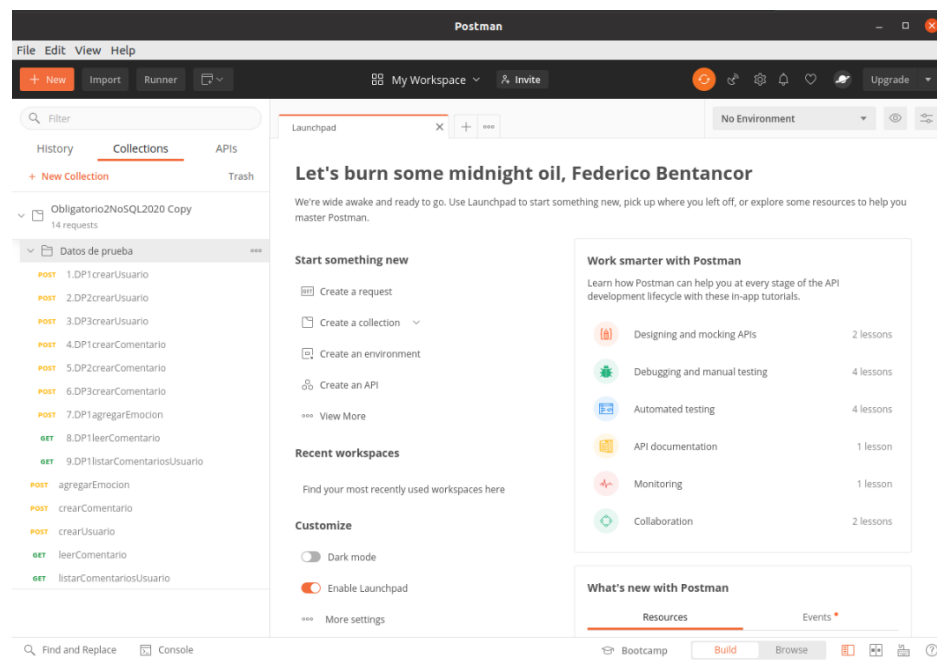
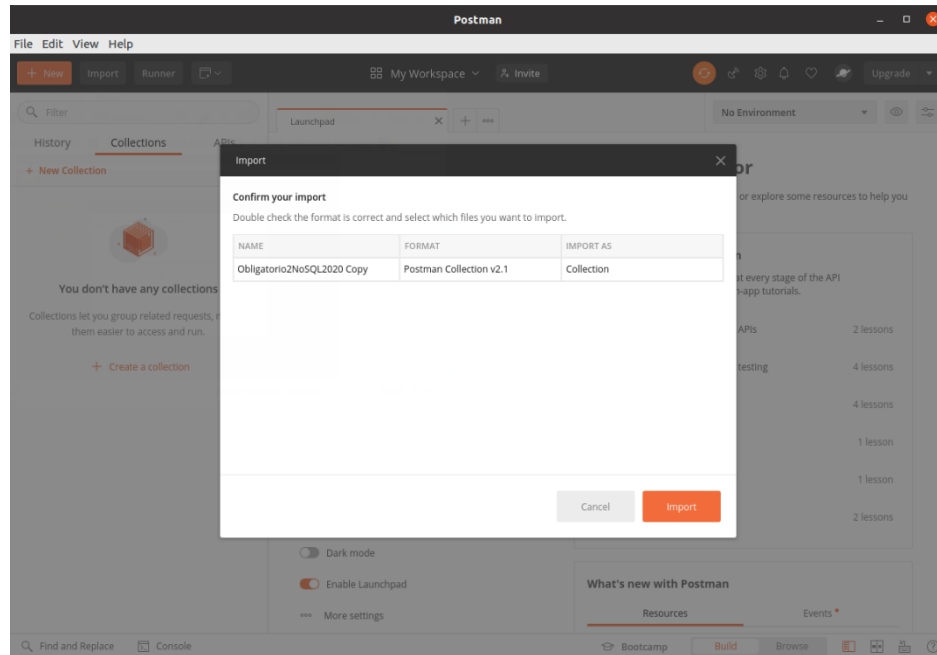


- Configuración
 - Redis
 - En application.properties del proyecto:
 - spring.redis.database=0
 - spring.redis.port=6379
 - spring.redis.timeout=60000

- Postman

- Luego de instalada la aplicación ir al menú Import y luego seleccionar Upload Files y seleccionar el archivo (adjuntado con la documentación):
 - Obligatorio2NoSQL2020.postman_collection.json





Plataforma, bases de datos, lenguaje y servidor

- Plataforma
 - IntelliJ IDEA Ultimate Edition
- Base de datos
 - MongoDB (no relacional)
 - Redis (Cache)
- Lenguaje
 - Java
 - Librerías:
 - Morphia (MongoDB)
 - Jedis (Redis)
- Servidor de aplicaciones
 - Springboot

Caché

Se decidió implementar una base de datos Redis para cumplir así el requerimiento opcional. A continuación, se presenta un punteo de las estrategias definidas para la utilización de este.

- Estrategias para la implementación de caché
 - La estructura almacenada en el caché será:
 - Clave: identificador de comentario.
 - Valor: tipo de datos generado para la devolución de información (DtLeerComentario).
 - El caché almacenará un máximo de diez registros clave-valor.
 - Superado el máximo se eliminará el registro clave-valor más antiguo.
 - En caso de agregar emoción a un comentario existente en los almacenados, este pasará a ser el más reciente del caché.