

DESARROLLO DEL LABORATORIO 02

EVIDENCIAS

1- Conectar con el proyector “productos-api”

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2023-09-07T14:19:06 -0500
Data center	default	Uptime	00:21
		Lease expiration enabled	false
		Renews threshold	5
		Renews (last min)	2

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
PRODUCTOS-API	n/a (2)	(2)	UP (2) - productos-api:ebcd407a-7079-4bc7-ae64-88660202c070 , productos-api:b10d5643-c8fd-410b-b9ca-11e7411154bf

General Info

Name	Value
total-avail-memory	72mb
num-of-cpus	12
current-memory-usage	34mb (47%)
server-up-time	00:21
registered-replicas	
unavailable-replicas	
available-replicas	

Instance Info

2-

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2023-09-07T15:53:09 -0500
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	8
		Renews (last min)	0

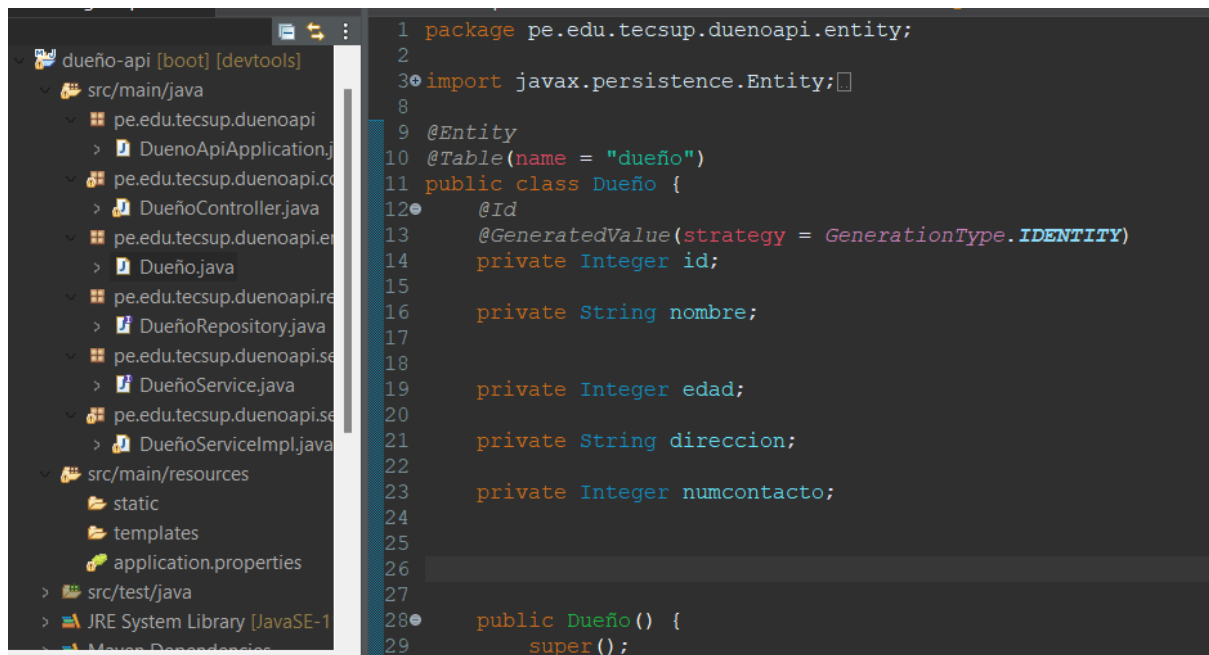
DS Replicas

Instances currently registered with Eureka

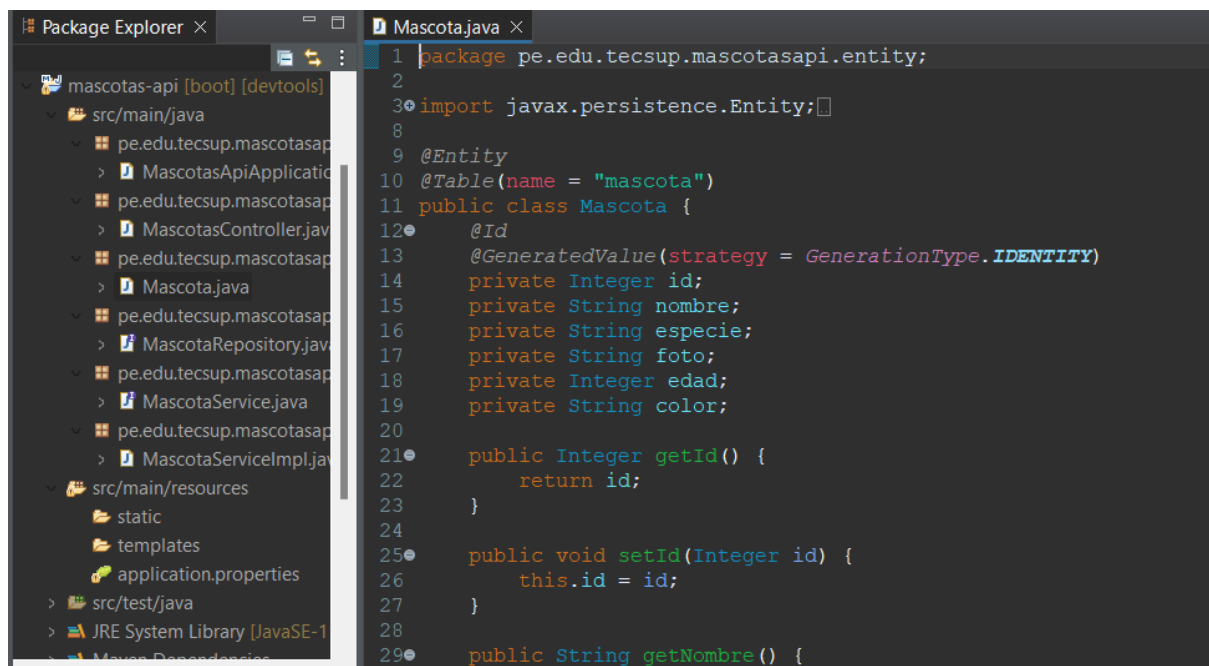
Application	AMIs	Availability Zones	Status
MASCOTAS-API	n/a (2)	(2)	UP (2) - mascotas-api:03be7e66-6f9c-438d-9d85-dd80e64d4834 , mascotas-api:9a0939d2-e4d1-4ade-aa9c-c3423f1d228f
PRODUCTOS-API	n/a (2)	(2)	UP (2) - productos-api:ebcd407a-7079-4bc7-ae64-88660202c070 , productos-api:b10d5643-c8fd-410b-b9ca-11e7411154bf

General Info

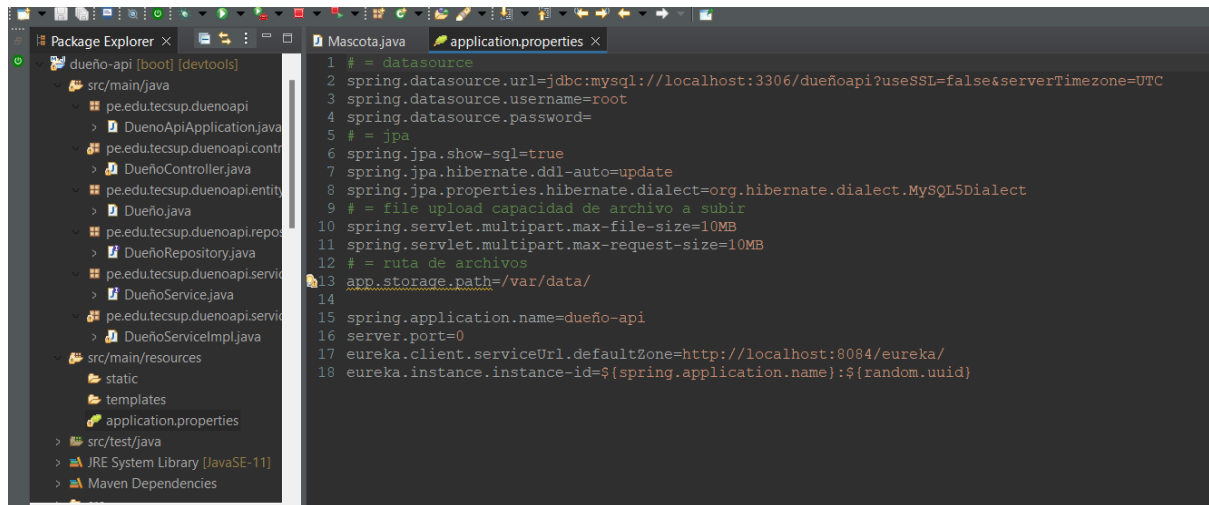
Creación de las clase Dueño



- Creación de la clase mascota

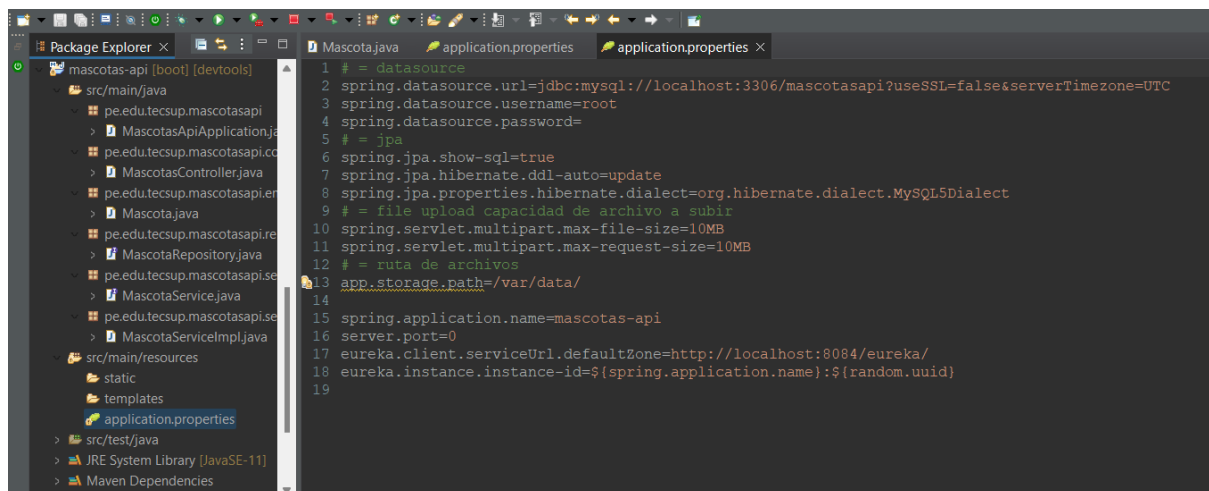


3- archivo properties de entidad “dueño”



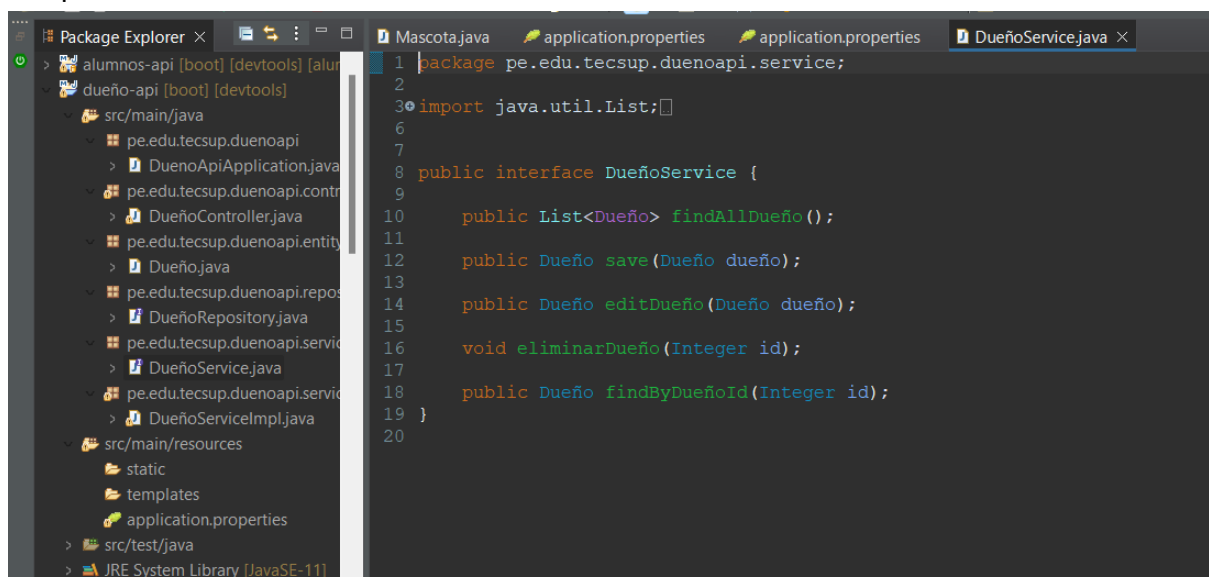
```
1 # = datasource
2 spring.datasource.url=jdbc:mysql://localhost:3306/dueñoapi?useSSL=false&serverTimezone=UTC
3 spring.datasource.username=root
4 spring.datasource.password=
5 # = jpa
6 spring.jpa.show-sql=true
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
9 # = file upload capacidad de archivo a subir
10 spring.servlet.multipart.max-file-size=10MB
11 spring.servlet.multipart.max-request-size=10MB
12 # = ruta de archivos
13 app.storage.path=/var/data/
14
15 spring.application.name=dueño-api
16 server.port=0
17 eureka.client.serviceUrl.defaultZone=http://localhost:8084/eureka/
18 eureka.instance.instance-id=${spring.application.name}:${random.uuid}
```

4-archivo properties de entidad “mascota”



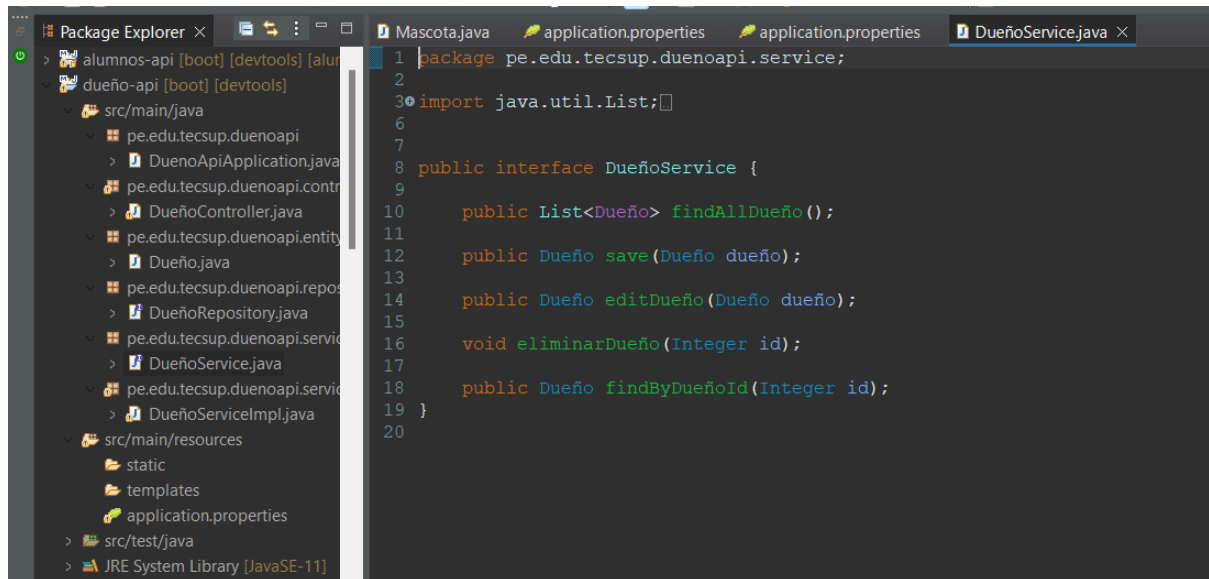
```
1 # = datasource
2 spring.datasource.url=jdbc:mysql://localhost:3306/mascotasapi?useSSL=false&serverTimezone=UTC
3 spring.datasource.username=root
4 spring.datasource.password=
5 # = jpa
6 spring.jpa.show-sql=true
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
9 # = file upload capacidad de archivo a subir
10 spring.servlet.multipart.max-file-size=10MB
11 spring.servlet.multipart.max-request-size=10MB
12 # = ruta de archivos
13 app.storage.path=/var/data/
14
15 spring.application.name=mascotas-api
16 server.port=0
17 eureka.client.serviceUrl.defaultZone=http://localhost:8084/eureka/
18 eureka.instance.instance-id=${spring.application.name}:${random.uuid}
```

5- Operaciones CRUD de la entidad “dueño”



```
1 package pe.edu.tecsup.duenoapi.service;
2
3 import java.util.List;
4
5
6
7
8 public interface DueñoService {
9
10     public List<Dueño> findAllDueño();
11
12     public Dueño save(Dueño dueño);
13
14     public Dueño editDueño(Dueño dueño);
15
16     void eliminarDueño(Integer id);
17
18     public Dueño findByDueñoId(Integer id);
19 }
20
```

6- Operaciones CRUD de la entidad “Mascota”



The screenshot shows an IDE with the Package Explorer on the left and the code editor on the right. The Package Explorer displays the project structure, including the package `pe.edu.tecsup.duenoapi.service` and the file `DueñoService.java`. The code editor shows the following Java code:

```
1 package pe.edu.tecsup.duenoapi.service;
2
3 import java.util.List;
4
5
6
7
8 public interface DueñoService {
9
10     public List<Dueño> findAllDueño();
11
12     public Dueño save(Dueño dueño);
13
14     public Dueño editDueño(Dueño dueño);
15
16     void eliminarDueño(Integer id);
17
18     public Dueño findByIdDueñoId(Integer id);
19 }
20
```