

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

# Actividad 3: automatización de pruebas de aceptación, funcionales, de sistema y de carga

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

## Desarrollo:

Para el desarrollo de esta actividad lo que se hizo primero fue levantar el back-end ejecutando los siguientes comandos:

```
git clone https://github.com/spring-petclinic/spring-petclinic-rest.git
cd spring-petclinic-rest
./mvnw spring-boot:run
```

Después de ejecutar estos comandos se comprobó que los servicios están activos mediante la visualización con swagger en <http://localhost:9966/petclinic/swagger-ui/index.html>

```

C:\Users\Usuario\Desktop>spring-petclinic-rest
[INFO] --- spring-boot-maven-plugin:3.0.2:run (default-cli) @ spring-petclinic-r
[INFO] Attaching agents: []
[INFO] Spring Boot 3.0.2
[INFO] Version - HV000001: Hibernate Validator 6.0.0.Final
[INFO] PetClinicApplication - Starting PetClinicApplication using Java 17.0.7 wit
[INFO] PID 7624 (C:\Users\Usuario\Desktop>spring-petclinic-rest)
[INFO] PetClinicApplication - Running with Spring Boot v3.0.2, Spring web v0.0.
[INFO] PetClinicApplication - The following profiles are active: "hsqldb", "spr
[INFO] Spring-data-jpa
[INFO] RepositoryConfigurationDelegate - Bootstrapping Spring Data JPA repositori
[INFO] RepositoryConfigurationDelegate - Finished Spring Data repository scanning
[INFO] TomcatWebServer - Tomcat initialized with port(s): 9966 (http)
[INFO] Http11NioProtocol - Initializing ProtocolHandler ["http-nio-9966"]
[INFO] StandardEngine - Starting Servlet engine: [Apache Tomcat/10.1.5]
[INFO] [petclinic] - Initializing Spring embedded WebApplicationContext
[INFO] ServletWebServerApplicationContext - Root WebApplicationContext: initializ
[INFO] HikariDataSource - HikariPool-1 - Starting...
[INFO] HikariDataSource - HikariPool-1 - Driver does not support get/set network timeout
[INFO] HikariPool - HikariPool-1 - Added connection org.hsqldb.jdbc.JDBCConnectio
[INFO] HikariDataSource - HikariPool-1 - Start completed.
[INFO] Log4j2 - HH000004: Processing PersistenceUnitInfo [name: default]
[INFO] Version - HH000412: Hibernate ORM core version 6.1.6.Final
[WARN] deprecation - HH000002: Encountered deprecated setting [java.persistence
[INFO] JtaPlatformInitiator - HH000049: Using JtaPlatform implementation: [org.h
[INFO] LocalContainerEntityManagerFactoryBean - Initialized JPA EntityManagerFact
[INFO] UserDetailsServiceConfiguration -
Using generated security password: 74bfa059-ed93-4ac3-956b-1b7366b13991

```

Figura 1

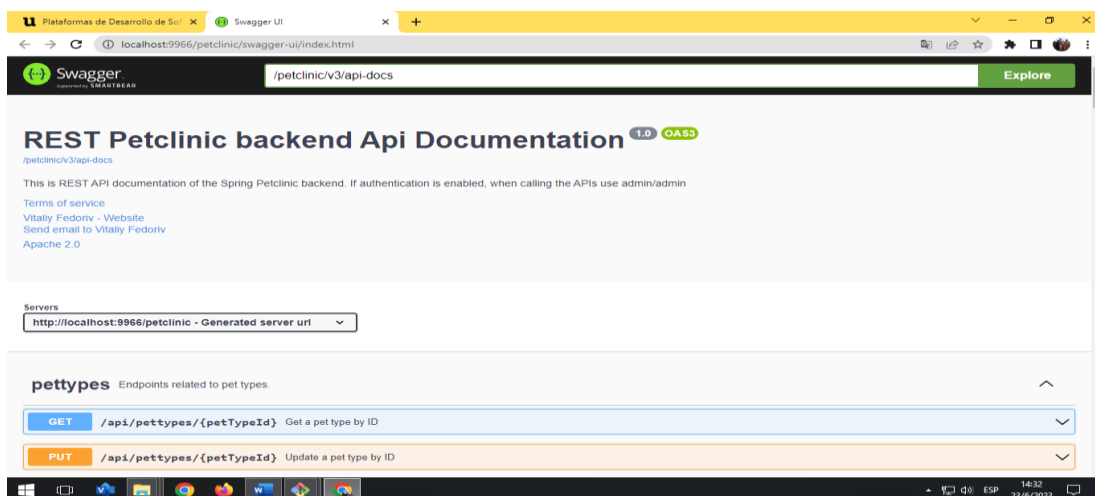


Figura 2

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Después se levantó el *front-end* ejecutando los siguientes comandos:

`npm uninstall -g angular-cli @angular/cli`

`npm cache clean`

`npm install -g @angular/cli@latest`

`git clone https://github.com/spring-petclinic/spring-petclinic-angular.git`

`cd spring-petclinic-angular`

`npm install --force`

`ng serve`

Se comprobó que se pueda acceder al interfaz gráfico desde <http://localhost:4200/petclinic/welcome>

```

MINGW64/c/Users/Usuario/Desktop/spring-petclinic-angular
$ cd spring-petclinic-angular
$ ng serve
workspace extension with invalid name (defaultProject) found.
This is a simple server for use in testing or debugging Angular applications locally.
It hasn't been reviewed for security issues.

DON'T USE IT FOR PRODUCTION!

- Generating browser application bundles (phase: setup)...
TypeScript compiler options "target" and "useDefineForClassFields" are set to "ES2022" and "false" respectively by the Angular CLI. To control ECMAScript version and features use the Browserslist
configuration. For more information, see https://angular.io/guide/build#configuring-browser-compatibility
NOTE: You can set the "target" to "ES2022" in the project's tsconfig to remove this warning.
- Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size | Estimated Transfer Size
main.js             | main  | 1.18 MB | 239.23 kB
styles.css, styles.js | styles | 366.28 kB | 76.33 kB
scripts.js          | scripts | 165.47 kB | 47.82 kB
polyfills.js        | polyfills | 161.56 kB | 41.25 kB
runtime.js          | runtime | 1.77 kB | 904 bytes
Initial Total | 2.06 MB | 409.53 kB

Build at: 2023-06-23T19:25:49.606Z - Hash: 1d7af9b287515bb8 - Time: 54262ms
Warning: C:\Users\Usuario\Desktop\spring-petclinic-angular\node_modules\@angular\material-moment-adap...
Warning: 2 rules skipped due to selector errors:
.form-floating > ~ label -> Did not expect successive traversals.
.form-floating > ~ label -> Did not expect successive traversals.

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

/ Compiled successfully.

```

Figura 3

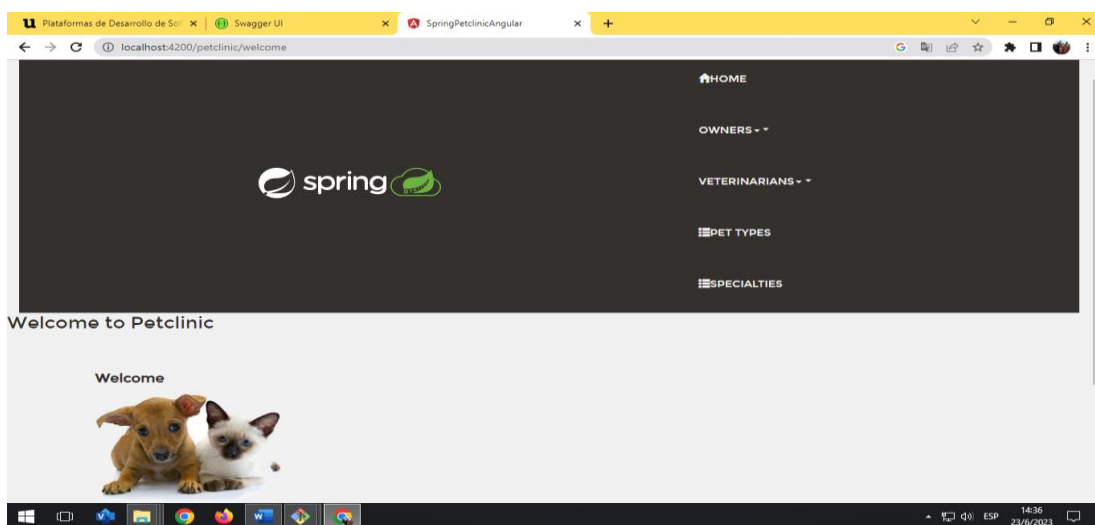


Figura 4

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

## Pruebas en el Back End

Hice las pruebas del back end con dos entidades (Propietarios y Especialidades)

Estos son los endpoints que se identificaron de propietarios para realizar las pruebas en la herramienta Postman

GET /api/owners/{ownerId} URL:

http://localhost:9966/petclinic/api/owners/{ownerId} Obtiene un propietario de mascota por su ID.

PUT /api/owners/{ownerId} URL:

http://localhost:9966/petclinic/api/owners/{ownerId} Actualiza los detalles de un propietario de mascota por su ID.

DELETE /api/owners/{ownerId} URL:

http://localhost:9966/petclinic/api/owners/{ownerId} Elimina un propietario por su ID.

GET /api/owners URL: http://localhost:9966/petclinic/api/owners Enumera todos los propietarios de mascotas.

POST /api/owners URL: http://localhost:9966/petclinic/api/owners Agrega un nuevo propietario de mascota.

Para agregar un nuevo propietario se usó el método Post y devuelve un 201, esto quiere decir que la operación es correcta

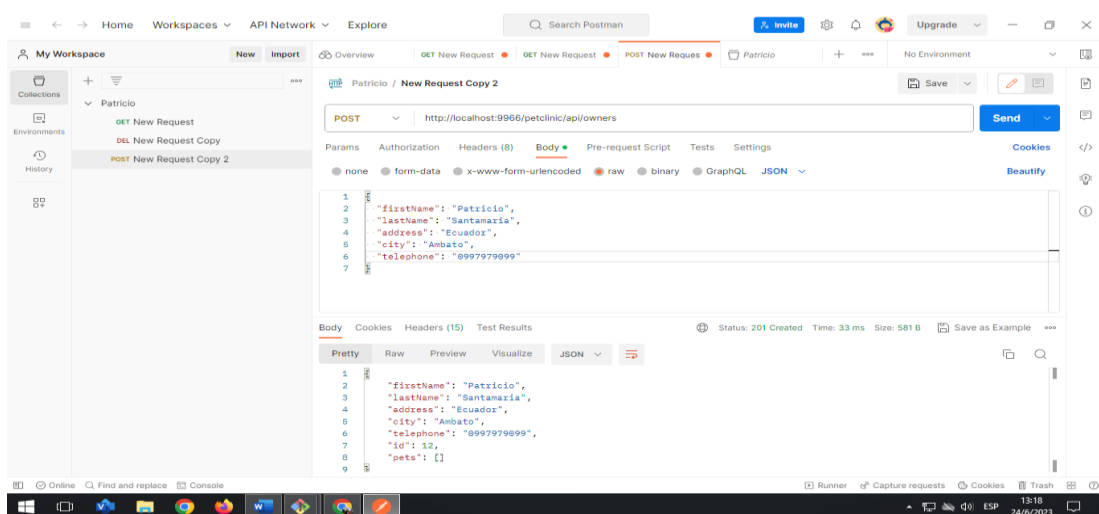


Figura 5

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Para realizar búsquedas de propietarios use el método Get, cuando realice la búsqueda obtuve un 200 de respuesta

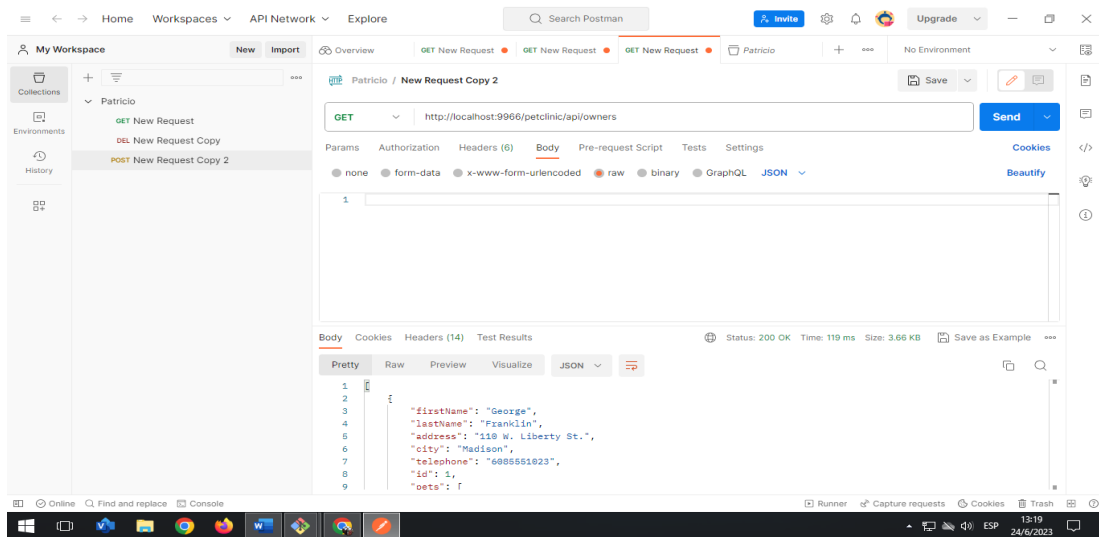


Figura 6

Para realizar búsquedas de propietarios por su id use el método Get, cuando realice la búsqueda obtuve un 200 de respuesta

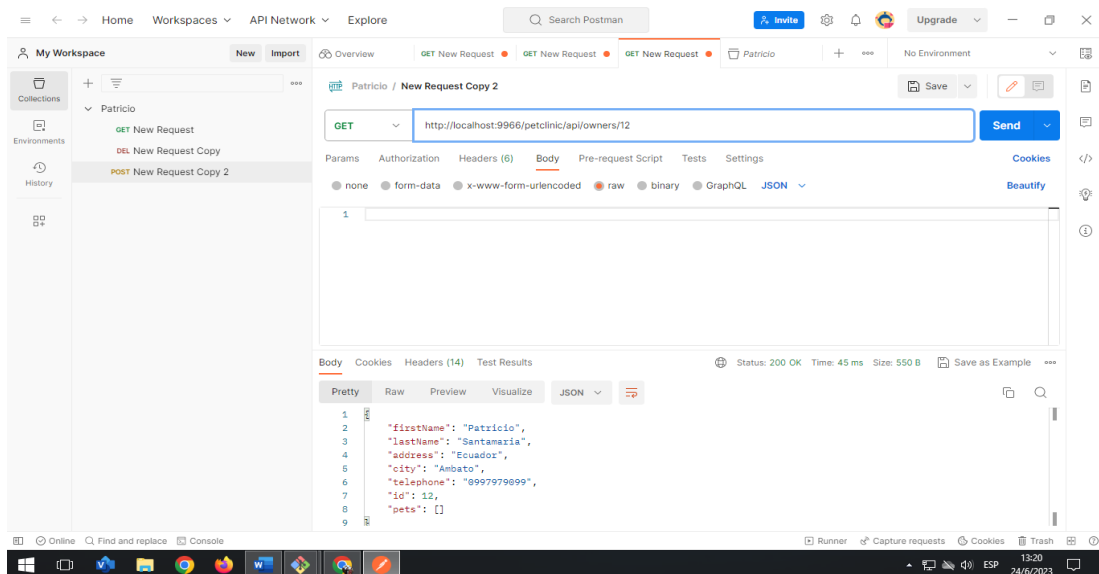


Figura 7

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Para realizar una actualización de propietarios use el método put, el sistema realiza la actualización sin embargo me devuelve como respuesta un 204 no content

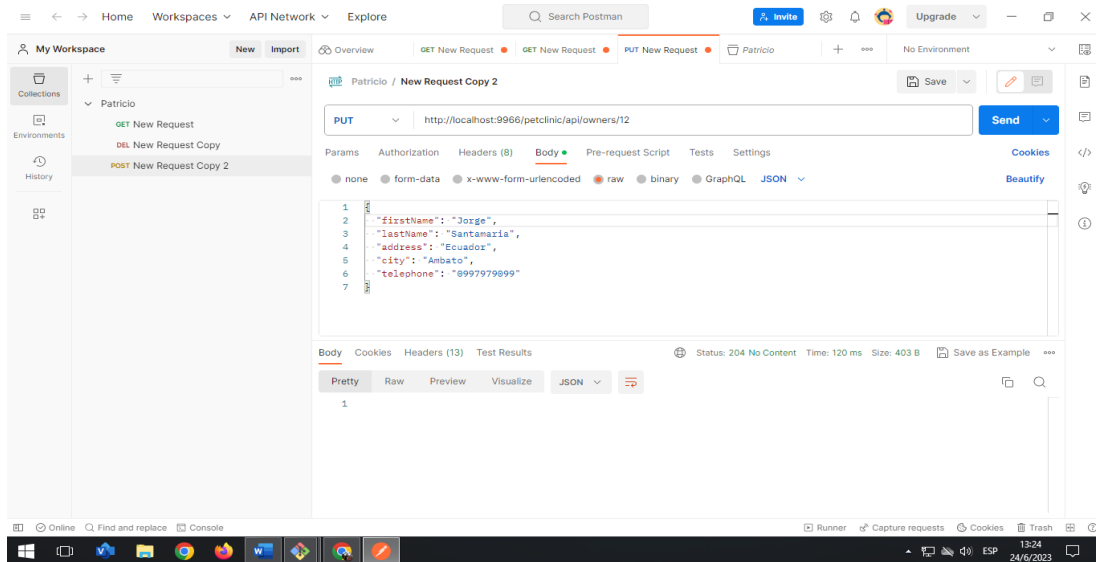


Figura 8

Para realizar una eliminación de propietarios use el método delete, el sistema realiza la eliminación correctamente sin embargo me devuelve como respuesta un 204 no content

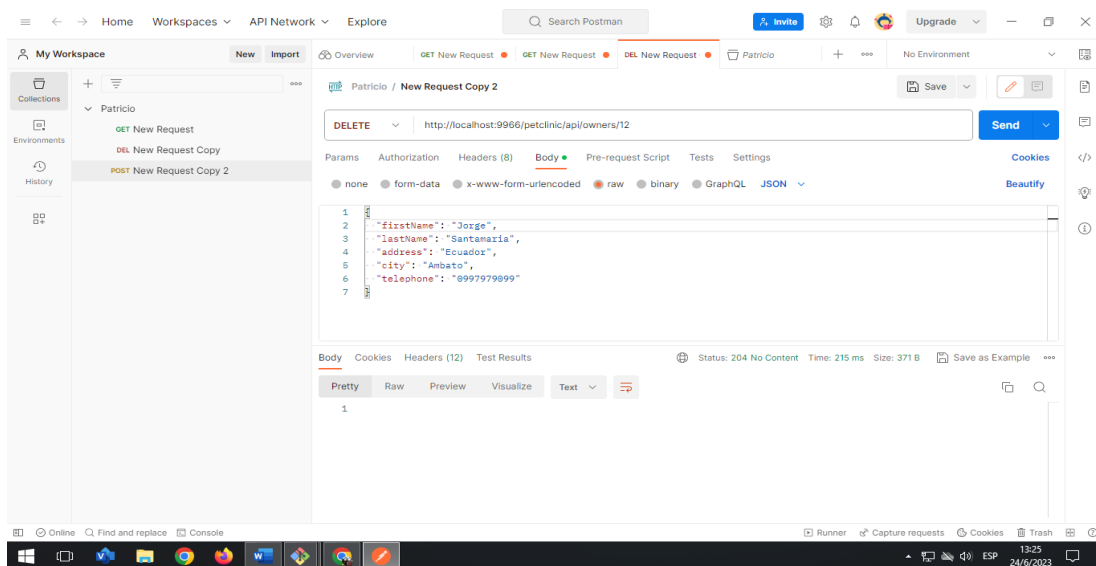


Figura 9

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

También hice una búsqueda de un propietario ya eliminado teniendo como respuesta un 204 not found

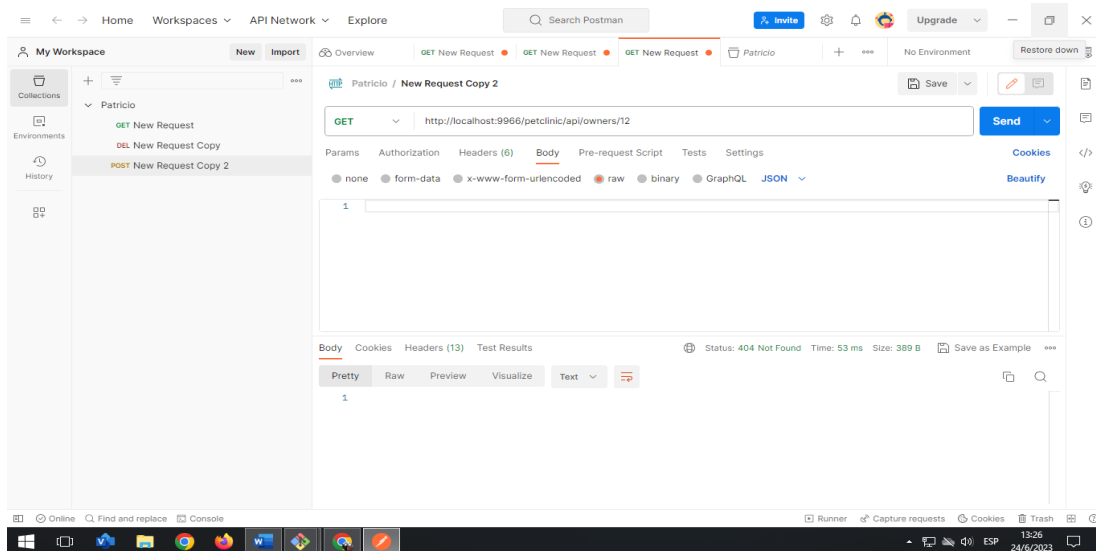


Figura 10

Estos son los endpoints que se identificaron de especialidades para realizar las pruebas en la herramienta Postman

GET /api/specialties/{specialtyId} URL:

http://localhost:9966/petclinic/api/specialties/{specialtyId} Obtiene una especialidad por su ID.

PUT /api/specialties/{specialtyId} URL:

http://localhost:9966/petclinic/api/specialties/{specialtyId} Actualiza una especialidad por su ID.

DELETE /api/specialties/{specialtyId} URL:

http://localhost:9966/petclinic/api/specialties/{specialtyId} Elimina una especialidad por su ID.

GET /api/specialties URL: http://localhost:9966/petclinic/api/specialties Enumera todas las especialidades.

POST /api/specialties URL: http://localhost:9966/petclinic/api/specialties Crea una nueva especialidad.

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Para realizar búsquedas de especialidades use el método Get, cuando realice la búsqueda obtuve un 200 de respuesta

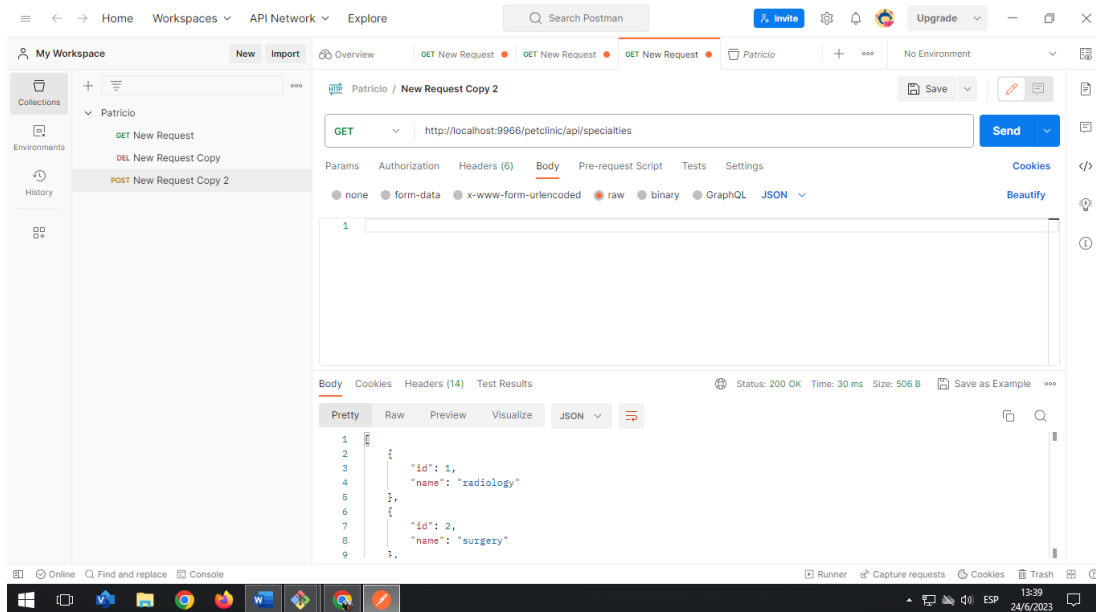


Figura 11

Para realizar búsquedas de especialidades por su id use el método Get, cuando realice la búsqueda obtuve un 200 de respuesta

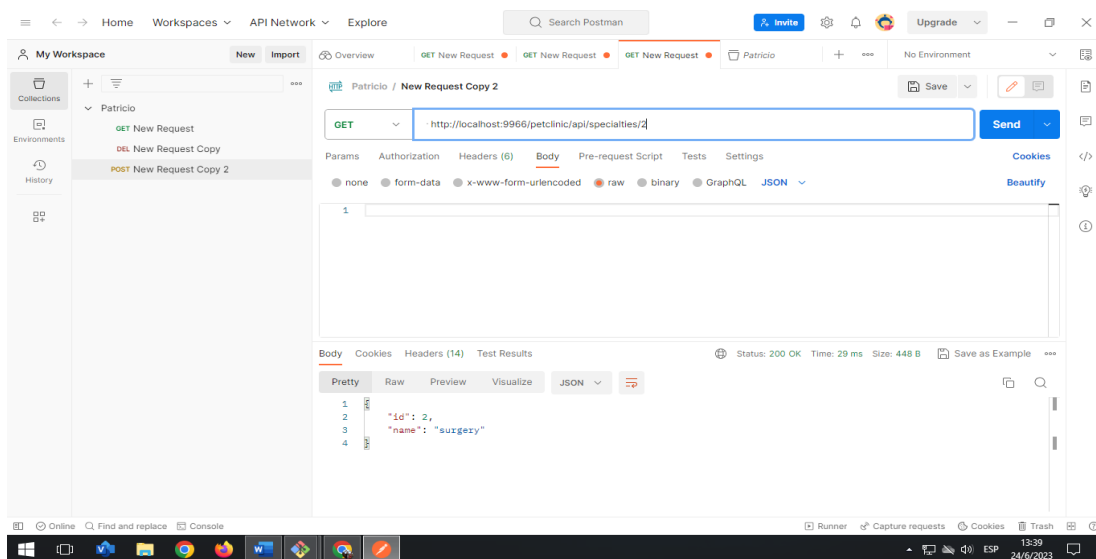


Figura 12



Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Para realizar una actualización de especialidades use el método put, el sistema realiza la actualización correctamente sin embargo me devuelve como respuesta un 204 no content

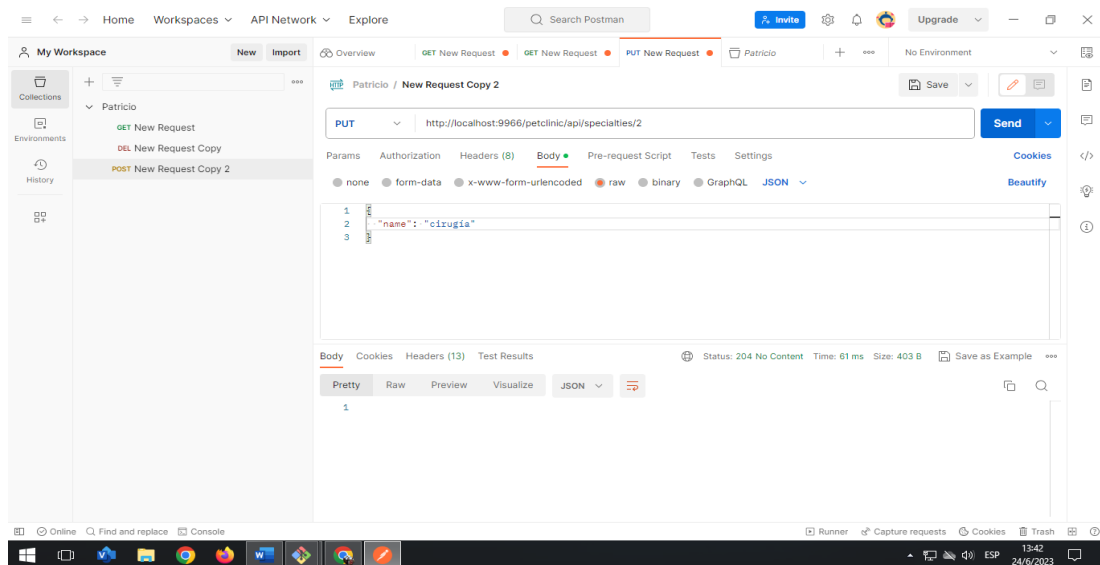


Figura 13

Para agregar una nueva especialidad se usó el método Post y teniendo como respuesta un 201, esto quiere decir que la operación es correcta

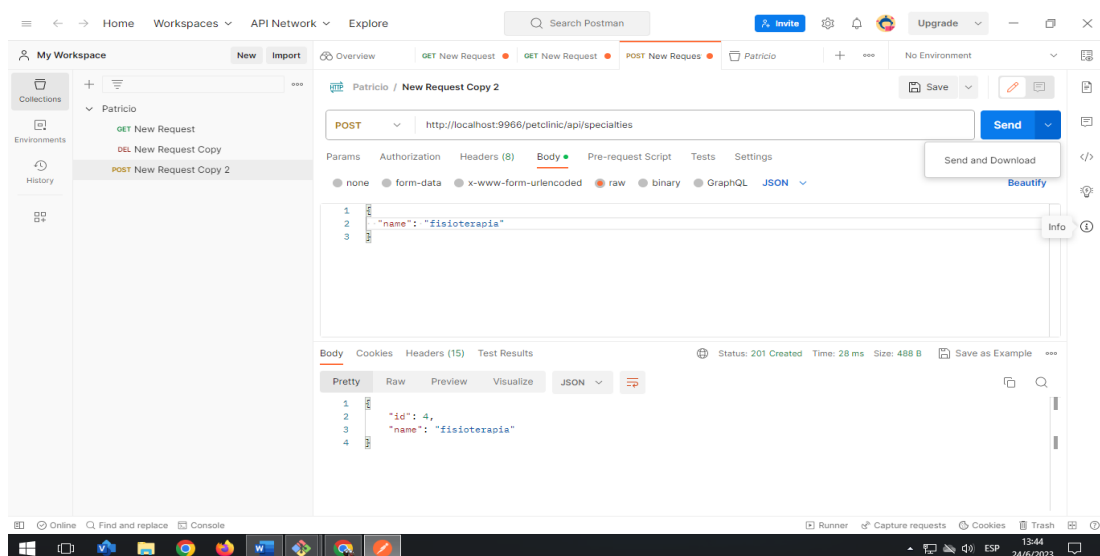


Figura 14

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Para realizar una eliminación de especialidades use el método delete, el sistema realiza la eliminación correctamente sin embargo me devuelve como respuesta un 204 no content

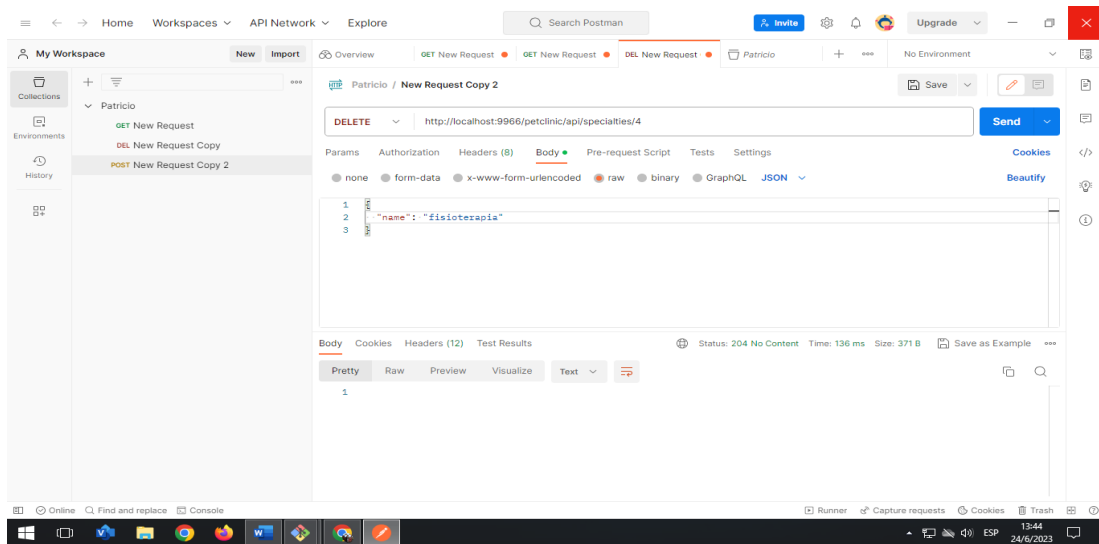


Figura 15

En esta entidad se está respetando las relaciones de las tablas de base de datos

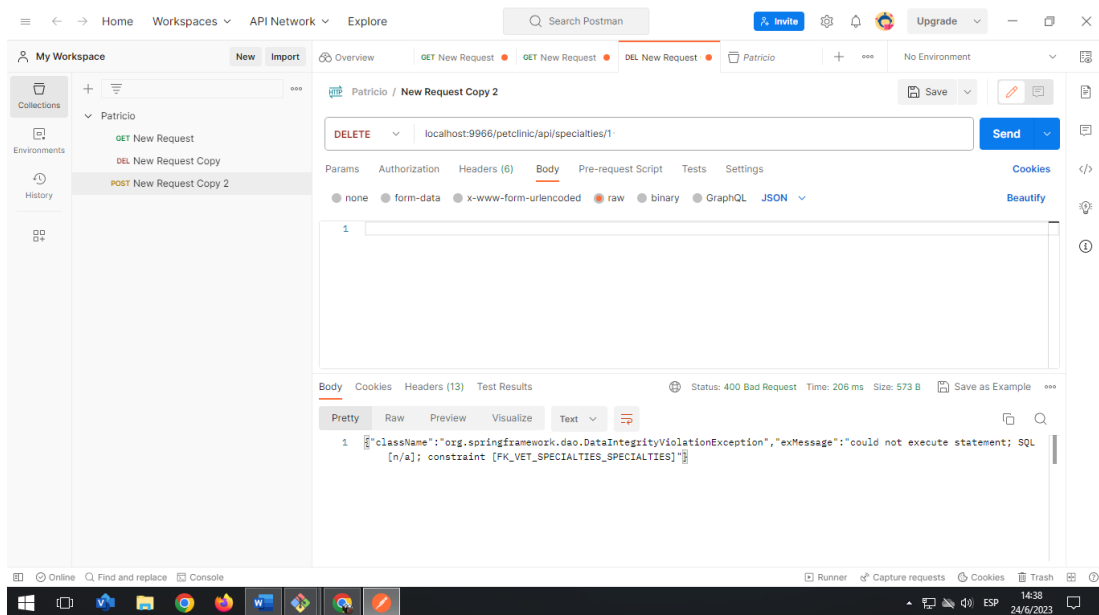


Figura 16

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Como conclusiones se puede sacar lo siguiente:

- Se realizaron pruebas de las entidades propietarios y especialidades
- La funcionalidad básica de crud de estas entidades esta correcta
- La integridad referencial esta correcta
- No devuelve bien las entidades al modificar y al eliminar (devuelve 204 sin objeto) pero si realiza la operación
- El openApi no está bien formado

## Pruebas de Carga

Para realizar las pruebas de carga utilicé la herramienta Jmeter y tomé como ejemplo la entidad pets

En el programa se hizo la prueba de carga con 1000 usuarios y un tiempo de rampa de 300 segundos para realizar peticiones get a la entidad pets

Ejemplo:

- Número de usuarios: 1000
- Periodo de tiempo de rampa: 300 segundos (5 minutos)

Con esta configuración, JMeter agregará 1000 usuarios durante un período de 5 minutos, lo que significa que aproximadamente se agregarán 3.3 usuarios por segundo

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

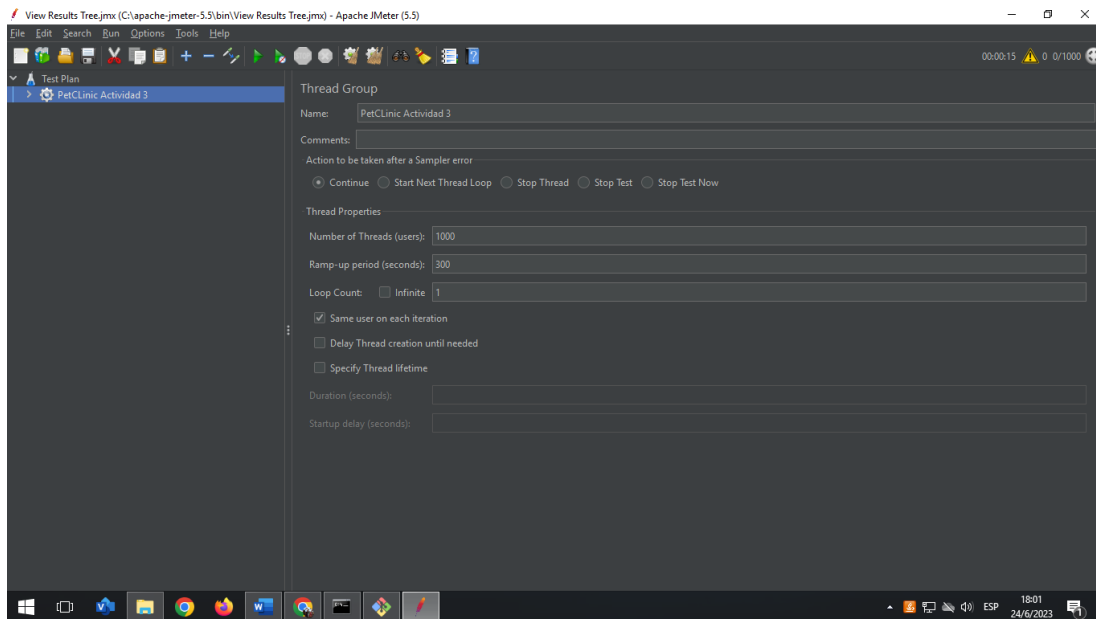


Figura 17

Después se procedió a configurar el protocolo, el servidor, el puerto y el path

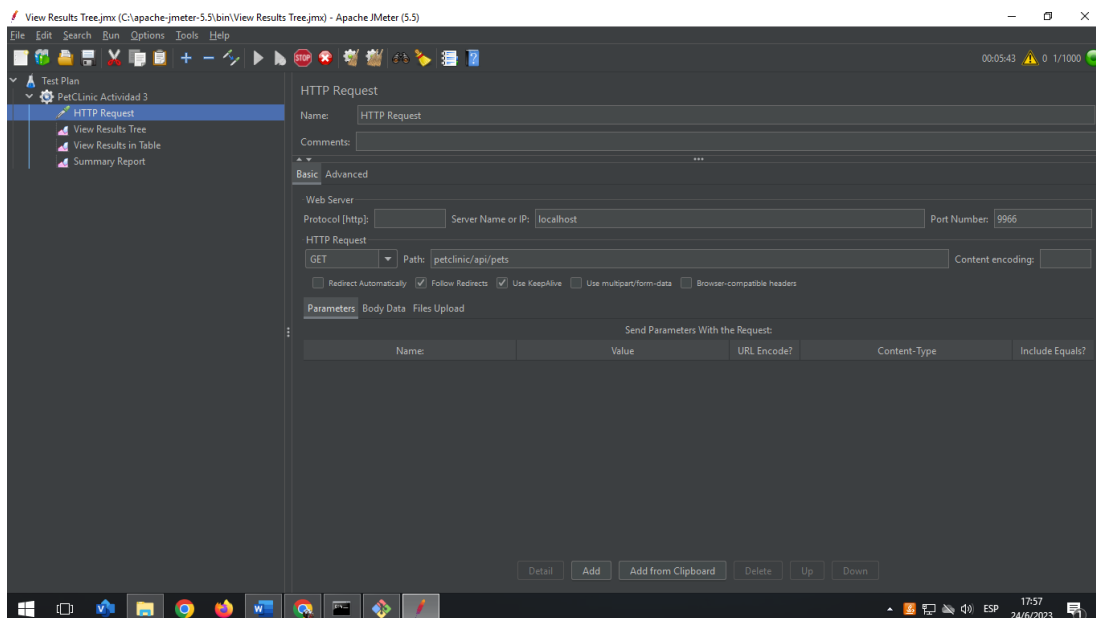
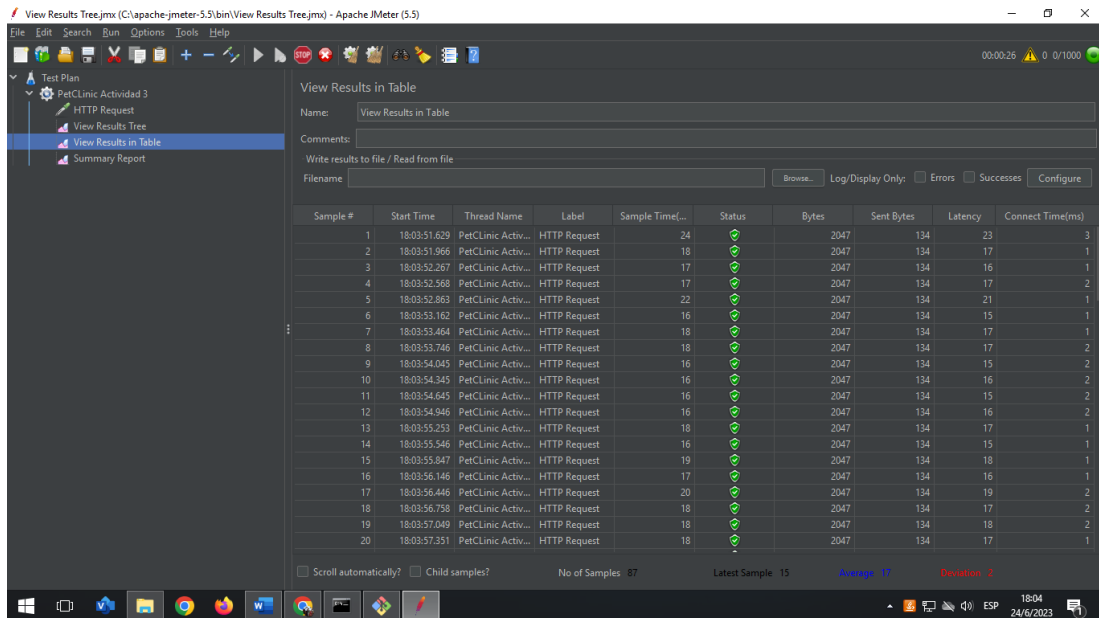


Figura 18

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Después se procedió a ejecutar la prueba de carga y se obtuvieron los siguientes resultados



Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	18:03:51.629	PetClinic Activ...	HTTP Request	24	✓	2047	134	23	3
2	18:03:51.966	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	1
3	18:03:52.267	PetClinic Activ...	HTTP Request	17	✓	2047	134	16	1
4	18:03:52.568	PetClinic Activ...	HTTP Request	17	✓	2047	134	17	2
5	18:03:52.863	PetClinic Activ...	HTTP Request	22	✓	2047	134	21	1
6	18:03:53.162	PetClinic Activ...	HTTP Request	16	✓	2047	134	15	1
7	18:03:53.464	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	1
8	18:03:53.746	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	2
9	18:03:54.045	PetClinic Activ...	HTTP Request	16	✓	2047	134	15	2
10	18:03:54.345	PetClinic Activ...	HTTP Request	16	✓	2047	134	16	2
11	18:03:54.645	PetClinic Activ...	HTTP Request	16	✓	2047	134	15	2
12	18:03:54.946	PetClinic Activ...	HTTP Request	16	✓	2047	134	16	2
13	18:03:55.253	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	1
14	18:03:55.546	PetClinic Activ...	HTTP Request	16	✓	2047	134	15	1
15	18:03:55.847	PetClinic Activ...	HTTP Request	19	✓	2047	134	18	1
16	18:03:56.146	PetClinic Activ...	HTTP Request	17	✓	2047	134	16	1
17	18:03:56.446	PetClinic Activ...	HTTP Request	20	✓	2047	134	19	2
18	18:03:56.758	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	2
19	18:03:57.049	PetClinic Activ...	HTTP Request	18	✓	2047	134	18	2
20	18:03:57.351	PetClinic Activ...	HTTP Request	18	✓	2047	134	17	1

Figura 19

## Front End

Se rellenaron los campos de first y last name con un solo carácter y sale el mensaje “El nombre debe tener al menos 2 caracteres de longitud”, en esta parte el sistema esta correcto

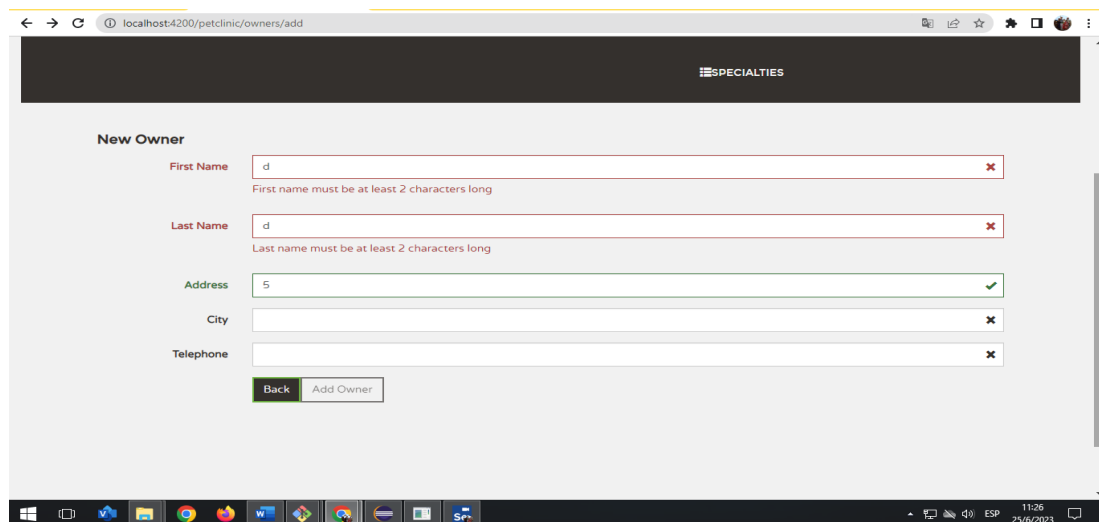


Figura 20

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Un error que se ha encontrado en la interfaz de usuario es que al escribir números en los campos del formulario donde solo se debe admitir letras el sistema no arroja ningún mensaje

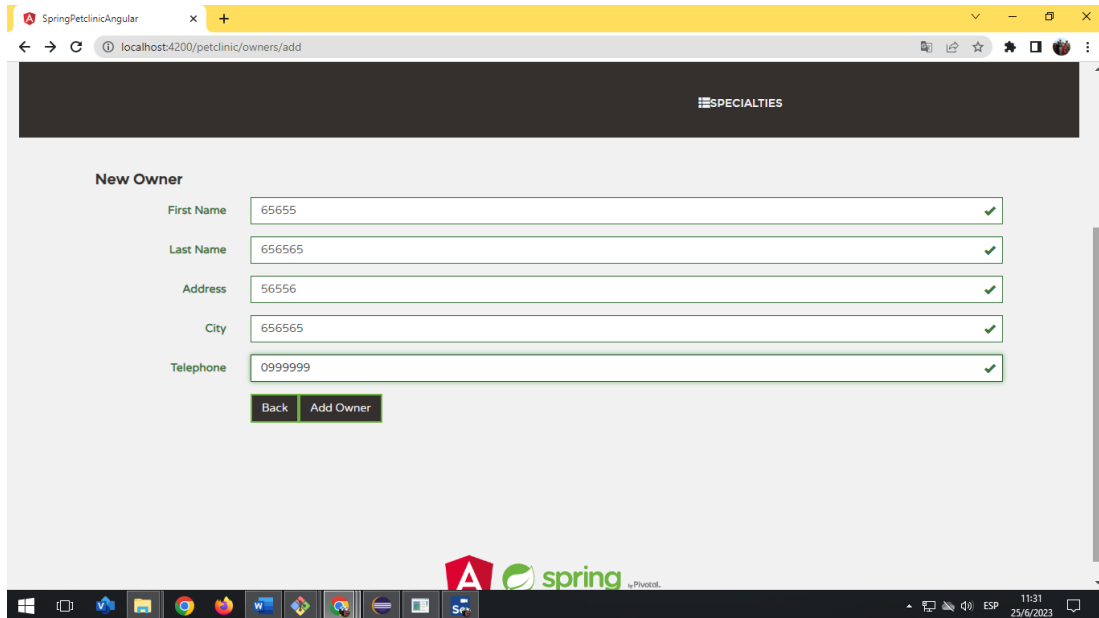


Figura 21

También se utilizó Selenium Ide para realizar las pruebas en el front

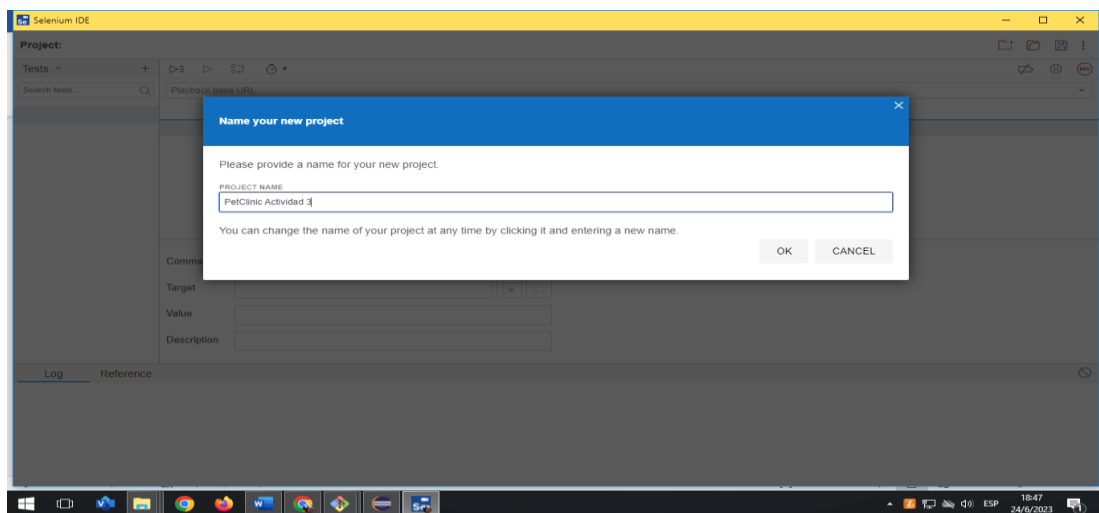


Figura 22

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Después configuré la URL base, en la configuración del proyecto establecí la URL base: "http://localhost:4200/petclinic/owners/add". Esto asegurará que todas las pruebas se realicen en esa URL específica.

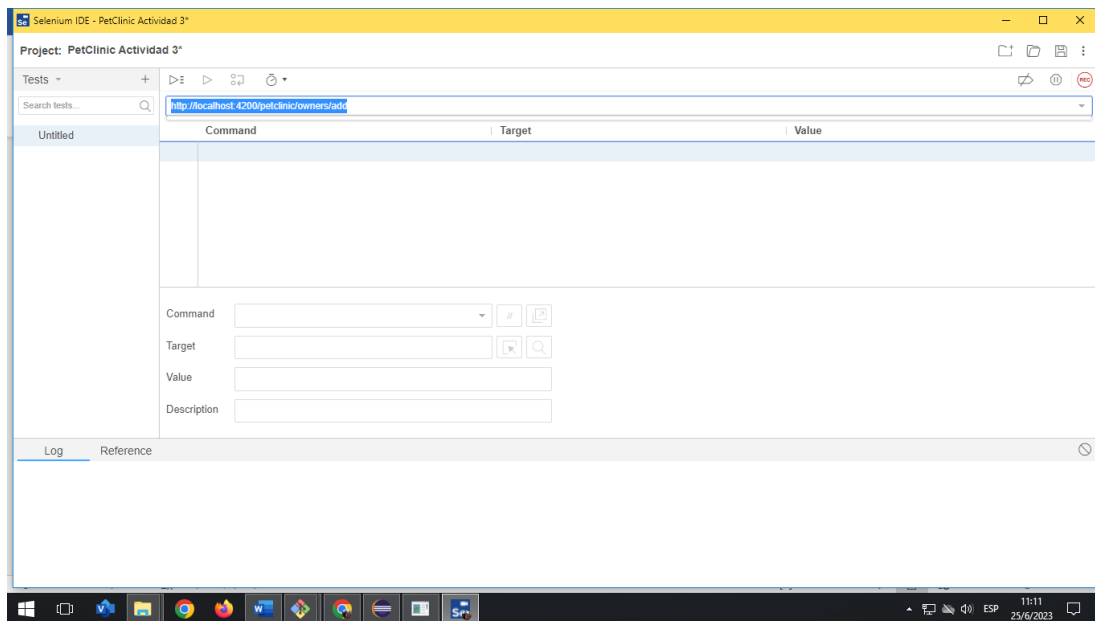


Figura 23

El siguiente paso fue grabar una prueba y realizar las acciones que se deseaba probar en el sitio web

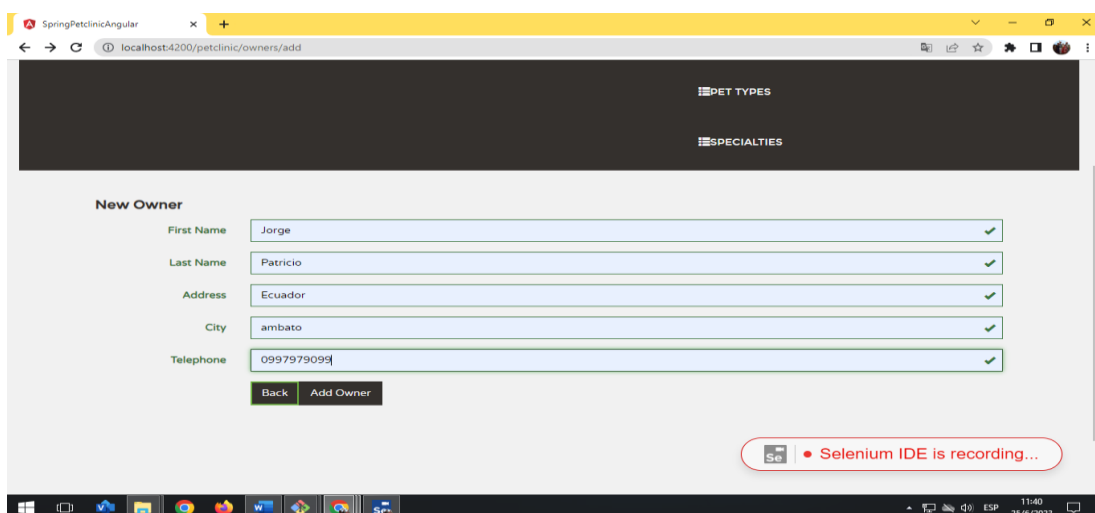


Figura 24

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Escribí todo correctamente y el sistema me permitió guardar un propietario exitosamente

A continuación, se presenta una interpretación de cada paso:

1. Abrir en <http://localhost:4200/petclinic/owners/add>: La prueba ha abierto correctamente la URL especificada.
2. Establecer el tamaño de la ventana en 1382x744: La prueba ha ajustado el tamaño de la ventana del navegador a las dimensiones especificadas.
3. Hacer clic en el elemento con el identificador 'firstName': La prueba ha realizado con éxito un clic en el elemento especificado.
4. Escribir en el elemento con el identificador 'firstName' el valor 'Jorge': La prueba ha ingresado el valor 'Jorge' en el elemento especificado.
5. Hacer clic en el elemento con el identificador 'lastName': La prueba ha realizado con éxito un clic en el elemento especificado.
6. Escribir en el elemento con el identificador 'lastName' el valor 'Patricio': La prueba ha ingresado el valor 'Patricio' en el elemento especificado.
7. Hacer clic en el elemento con el identificador 'address': La prueba ha realizado con éxito un clic en el elemento especificado.
8. Escribir en el elemento con el identificador 'address' el valor 'Ecuador': La prueba ha ingresado el valor 'Ecuador' en el elemento especificado.
9. Hacer clic en el elemento con el identificador 'city': La prueba ha realizado con éxito un clic en el elemento especificado.
10. Escribir en el elemento con el identificador 'city' el valor 'ambato': La prueba ha ingresado el valor 'ambato' en el elemento especificado.
11. Hacer clic en el elemento con el identificador 'telephone': La prueba ha realizado con éxito un clic en el elemento especificado.
12. Hacer clic en el elemento con el identificador 'telephone': La prueba ha realizado con éxito un clic en el elemento especificado.
13. Hacer doble clic en el elemento con el identificador 'telephone': La prueba ha realizado con éxito un doble clic en el elemento especificado.
14. Escribir en el elemento con el identificador 'telephone' el valor '0997979099': La prueba ha ingresado el valor '0997979099' en el elemento especificado.
15. Hacer clic en el elemento con el selector CSS '.btn:nth-child(2)': La prueba ha realizado con éxito un clic en el elemento especificado.



Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

Otro gran error de este sistema es que al agregar una nueva mascota me permite ingresar números como nombre

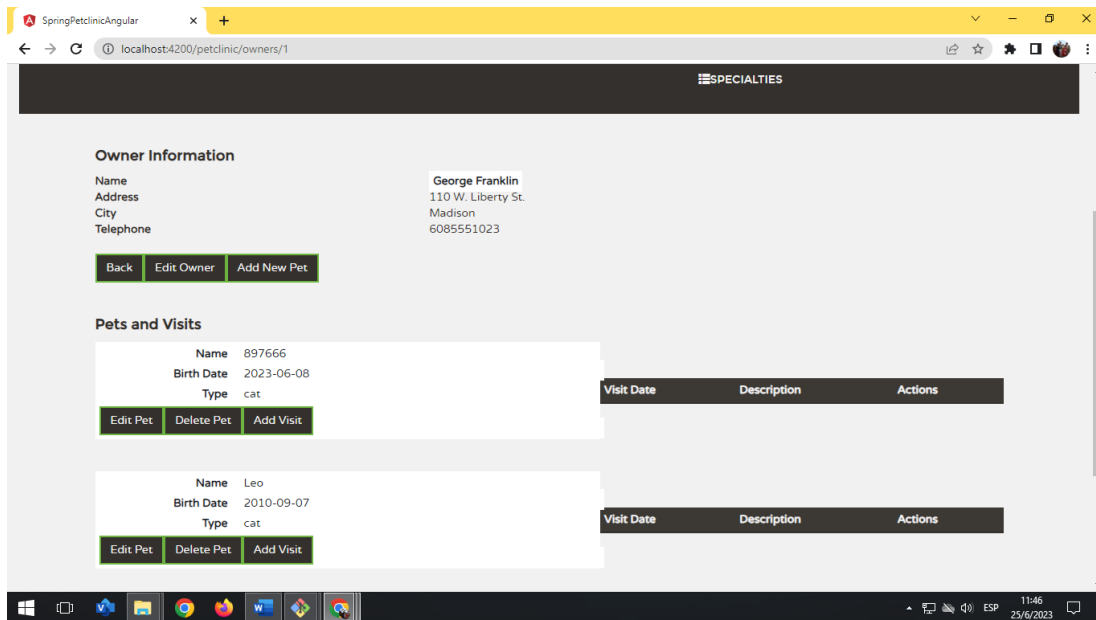


Figura 25

El sistema hace las operaciones crud de una forma incorrecta, guarda los datos, pero lo hace mal al permitir guardar números en donde solo se debería permitir guardar letras

## Conclusiones:

En conclusión, realizar pruebas tanto en el backend como en el frontend es crucial para garantizar la calidad, funcionamiento adecuado y rendimiento óptimo del software, identificando y solucionando errores de manera temprana, validando la integración de componentes y ahorrando tiempo y costos en el proceso de desarrollo.

Asignatura	Datos del alumno	Fecha
Plataformas de Desarrollo de Software	Apellidos: Santamaría Cherrez	25/6/2023
	Nombre: Jorge Patricio	

## Rúbrica

Automatización de pruebas de aceptación, funcionales, de sistema y de carga	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	Correcta automatización de pruebas de aceptación	2	20 %
Criterio 2	Correcta automatización de pruebas funcionales (interfaz de usuario web)	2	20 %
Criterio 3	Correcta automatización de pruebas de sistema ( <i>back-end</i> , API REST)	2	20 %
Criterio 4	Correcta automatización de pruebas de carga	1	10 %
Criterio 5	Memoria técnica: exposición del proceso y elección de plataformas	2	20 %
Criterio 6	Memoria técnica: exposición del aprendizaje	1	10 %
		<b>10</b>	<b>100 %</b>