**Change the background color of the table header when looking for owners to light grey.**

Esta tarea ha sido bastante sencilla, aunque nos hemos encontrado con un pequeño problema: al principio, se cambiaban las cabeceras de todas las tablas, no solo la de la búsqueda de propietarios. La solución que aplicamos fue crear en el fichero petclinic.less un estilo exclusivo para ella, utilizando la id de dicha tabla.

**Delete support**

Esta tarea nos generó algunos problemas al principio. Comenzamos por crear un controlador que eliminase a un propietario, con su correspondiente servicio en las clases ClinicService.java y ClinicServiceImpl.java, y el método al que llama en el repositorio OwnerRepository.java. En dicho repositorio, el método que elimina a un propietario recibía un objeto de tipo Owner como parámetro.

Tras realizar todo esto, comprobamos que, al intentar eliminar un propietario en la aplicación desplegada en local, no se realizaba correctamente.

Finalmente, nuestra solución llegó al cambiar el parámetro que recibía el método de eliminación del repositorio, al cuál ahora le pasamos el id del propietario que queremos borrar, es decir, tipo int.

Este cambio fue un éxito, y a partir de ahí, implementamos de la misma manera la funcionalidad de eliminación en Vet y en Pet.

[**Add a translation of the messages in Spanish**](https://github.com/gii-is-psg2/PSG2-1819-G1-14-L1/issues/5)

Esta tarea fue fácil de realizar a pesar de ser muy tediosa debido a la búsqueda durante todos los archivos de palabras y expresiones que se pudieran traducir para encontrarnos con la página totalmente en español.

Para realizarla no tuvimos más que usar el comando <fmt:message key=”Expresión o palabra a traducir”/> y escribir esa palabra dentro del archivo message\_es.properties con su correspondiente traducción.

Al haber expresiones que fueron acotadas en una palabra,usando la notación camelCase, estas se encuentran también expresadas en el message.properties con su equivalencia.

[**Submit details about pets**](https://github.com/gii-is-psg2/PSG2-1819-G1-14-L1/issues/9)

Esta tarea ha sido de las que más nos ha costado implementar ya que era la que más cosas englobaba ya que sigue el formato Modelo-Vista-Controlador, además de implementar una tabla nueva en la base de datos.

Comenzamos creando el controlador para que pudiera guardar la mascota, mediante su ID, en la reserva del hotel, aplicando esta actividad en las clases ClinicService.java y ClinicServiceImpl.java.

Tras esto creamos las clases correspondientes a la vista y el modelo para que se pudiera efectuar y ver este servicio en nuestra página.

Tras varias modificaciones para conseguir el despliegue de la página nos dimos cuenta de que la mayoría de los errores se encontraban en el controlador.

Al encontrar estos, les dimos solución y conseguimos el perfecto funcionamiento y despliegue de la página.