

History of tasks

a) Change the color of the header buttons when selected to deep red

Esta tarea no tuvo complicación alguna, pues una vez desplegada la aplicación en local y con la ayuda del inspeccionador de elementos, bastó con buscar la parte que se quería cambiar y modificar la línea correspondiente en el .less.

b) Change the background color of the header menu to light green

Al igual que en la tarea anterior, no tuvimos ningún tipo de problema. Bastó con encontrar la línea correspondiente e ir probando a cambiar los valores de los colores por los deseados.

c) Change the message of the Welcome page to “Welcome to the PSG2 Petclinic”.

Esta tarea no tuvo ninguna complicación debido a la simplicidad de la tarea. Bastó con localizar el archivo que contenía los mensajes y modificar el mensaje de Welcome por el indicado en la tarea.

d) Add a remove functionality to the pets of each owner

j) Add delete support to all main domain objects (owners, pets, veterinarians...)

(EXTRA CRITERIA)

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.

e) Add a translation of the messages in Spanish

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.

e) 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.

f) Create a Pet Hotel functionality, allowing clients to submit details about which pet they want to book a room for, as well as the start and finish dates for the booking. (EXTRA CRITERIA)

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.

g) Change the overall style of the website, modifying the logos, fonts and colors of the pages. It should use brownish tones. (EXTRA CRITERIA)

Pese a que esta tarea no es muy compleja en la parte de código, solo teniendo que modificar los archivos `less` que contienen los estilos de la web y modificando la cabecera para cambiar el logo. La complicación que nos surgió fue conseguir una buena combinación de tanto colores como de un logo que representase bien la web.

Al final decidimos mantener el estilo que tenía antes la web combinando varias tonalidades de marrón y seleccionar un logo muy representativo de la web, manteniéndolo en las mismas posiciones que el logo antiguo. Por último variamos la imagen principal de la web ya que una foto con fondo blanco no encajaba con el estilo que queríamos obtener.

h) Add creation and editing support to veterinarians. (EXTRA CRITERIA)

Esta tarea nos resultó sencilla excepto por el hecho de que no conseguimos en principio añadir especialidades para los veterinarios. Tras tener creadas las vistas y los controladores, decidimos que sólo se pudieran crear y editar veterinarios sin tener en cuenta las especialidades de los mismos. Es decir, al crear sólo tendríamos que poner el nombre y los apellidos y el campo de la especialidad iría vacío. Para el otro caso, el de editar, había que tener en cuenta que en el caso de que un veterinario ya tuviera una o varias especialidades asignadas (de la base de datos) deberíamos reasignarles dichas especialidades nuevamente en el controlador POST (cogiendo las ya tenía de la base de datos y seteando de nuevo el campo) dado que siempre le llegaban nulas del formulario.

Tras investigar un poco más sobre el tema, nos dimos cuenta de que lo hacía falta era un conversor similar al que ya hay en el proyecto (PetTypeFormatter). Luego, lo creamos (SpecialtyFormatter) y resolvimos el problema. Finalmente, pudimos añadir especialidades a los veterinarios tanto en el proceso de creación como en la edición.

EXTRA CRITERIA. Ventajas y desventajas de estrategia de rama

Ventajas:

1. Capacidad de poder trabajar cada miembro del grupo en distintas tareas al mismo tiempo.
2. Poder trabajar en una misma versión del proyecto de manera individual.
3. En el caso de que algún miembro del grupo realice alguna modificación independiente a nuestra tarea, bastaría con hacer un pull para incluir dicha modificación.

Desventajas:

1. Es necesaria mucha coordinación a la hora de subir una modificación al repositorio remoto, así como al hacer un merge entre las distintas versiones de cada uno.
2. Es importante tener en cuenta sobre qué rama se está trabajando, pues podríamos estar haciendo una modificación sobre la rama que no corresponde sin darnos cuenta.