

Ejercicio en Clase – Thymeleaf

Objetivo Principal:

- Diseñar e implementar interfaces de usuario utilizando Thymeleaf.

Objetivos específicos:

- Aplicar las anotaciones utilizadas para validaciones en Thymeleaf.
- Manejar el flujo entre múltiples plantillas de ingreso de datos.
- Implementar el despliegue de mensajes de error en las plantillas.
- Implementar un proceso para guardar información parcialmente.

Presentación

Para la presente actividad, se cuenta con un código base que corresponde a una posible solución del pedido realizado la semana pasada. Es decir, los cuatro formularios para adicionar y editar usuarios y citas. En caso que no haberlos realizado, se sugiere terminarlos antes de la clase y antes de continuar con este ejercicio.

Este ejercicio puede ser realizado en los grupos de cuatro personas en que van a trabajar en clase. Se sugiere que cada uno lo vaya desarrollando en su propio equipo y vayan avanzando de forma coordinada para que todos entiendan lo que están haciendo y no solamente copien lo que hace otro compañero.

Este ejercicio es la base para el ejercicio entregable en grupos de cuatro que se desarrollará en la sesión del jueves. Por tanto, debe ser terminado antes de esta sesión.

Actividad 1: Utilizando validaciones en gestiones de un solo formulario

1) Revisar las dos entidades y agregar las anotaciones necesarias para validar lo siguiente:

a) En el usuario:

- i) El nombre no debe ser vacío y debe tener al menos 2 caracteres
- ii) El correo no debe ser vacío y debe ser un correo bien formado
- iii) El tipo de usuario no debe ser vacío
- iv) La fecha de nacimiento no debe ser vacía y debe ser en el pasado
- v) El género no debe ser vacío

b) En la cita:

- i) El nombre no debe ser vacío
- ii) La fecha no debe ser vacía y debe ser hoy o en el futuro
- iii) La hora no debe ser vacía
- iv) El paciente y el doctor no deben ser vacíos

Las anotaciones que debe utilizar en las entidades se encuentran explicadas en:

<https://beanvalidation.org/2.0/spec/#builtinconstraints>

2) Modificar los controladores para que reciban el BindingResult tanto al agregar como al editar en ambas entidades (usuarios y citas) y tenga la anotación necesaria para que el elemento recibido sea validado.

Tenga en cuenta que la posición en que se coloque el BindingResult dentro de los parámetros del método afecta que funcione o no.

- 3) Agregar una validación en los controladores para que revisen si el BindingResult está reportando algún error
Tenga en cuenta que al encontrar un error debe cargar nuevamente la lista de valor y retornar a la pantalla previa. El resto se maneja modificando la vista o automáticamente.
- 4) Finalmente, debe modificar la vista para que permita visualizar el error o errores que tengas las entradas provistas por el usuario
Los mensajes se pueden mostrar junto a cada uno de los campos o se pueden mostrar en una sección de errores. Para este ejercicio, vamos a mostrarlos en cada uno de los campos. Puede utilizar un spam como el siguiente para mostrar los errores ajustándolo al campo correspondiente:

```
<span th:if="${#fields.hasErrors('name')}}"  
th:errors="*{name}" class="text-danger"></span>
```

Actividad 2: Realizando validaciones en múltiples formularios

Ahora, se propone dividir el proceso de adición de usuario en dos partes. En la primera parte, se le pedirá ingresar el username, la contraseña y la fecha de nacimiento. En la segunda parte, se le pedirá ingresar los datos restantes. Los tres datos ingresados en la primera parte no pueden ser nulos. Adicionalmente, el username debe tener al menos 3 caracteres y la contraseña 8. La fecha de nacimiento debe ser anterior a la actual.

Cuando se diligencien estos datos, se deben guardar y continuar con la segunda parte. En ambos formularios se deben hacer las validaciones inmediatamente, mostrando los errores correspondientes si es el caso.

- 1) Revisar el video disponible en:
En línea: <https://www.youtube.com/watch?v=2qd9EB0FSIE&feature=youtu.be>
Descarga: <https://home.jcmunoz.net:4443/nextcloud/index.php/s/RR6GQcfsGXkbdC8>
- 2) Revisar cómo utilizar la anotación @Validated de Spring y las anotaciones en las entidades para hacer validaciones solo de determinados campos. Es decir, como establecer grupos para los elementos validados y cómo utilizarlos.
- 3) Modificar la plantilla (template) de adición de usuarios y agregar la nueva plantilla para permitir ingresar los tres campos especificados. Se sugiere llamarlas add-user-1.html y add-user-2.html
- 4) Modificar los mappings del controlador de usuarios para soportar las dos interacciones en las dos plantillas del cliente acorde al enunciado y al video revisado.
- 5) Modificar la entidad User y el controlador para definir los campos que deban ser validados en cada paso.
- 6) Revisar cómo se puede evitar que el usuario modifique algunos campos o que los visualice, pero se sigan pasando en el formulario.