

## **Taller de Diseño de Páginas - Thymeleaf**

### Computación en Internet

#### **Objetivos:**

- Aplicar el patrón MVC en aplicaciones Spring.
- Diseñar e implementar interfaces de usuario utilizando Thymeleaf.

Se debe realizar una aplicación Web utilizando Thymeleaf. Debe incluir repositorios SpringData con una base de datos H2 y gestionar las entidades utilizadas previamente en el taller 1 acorde a las siguientes actividades.

#### **Actividades:**

1. Crear una copia del proyecto Spring Boot del taller 1 de pruebas y adicionar las dependencias adecuadas para el proyecto Web con seguridad.
2. (0.8) Realizar una pantalla para inicio de sesión de los usuarios. Debe implementar el servicio y repositorio que realice las validaciones necesarias utilizando la tabla USERR. Es necesario crear al menos un usuario de cada tipo para las pruebas. Los tipos son administrador y operador. El administrador puede gestionar las personas y los tipos de documentos, y los operadores los documentos e instancias de documentos.
3. (1.6) Realizar las pantallas para guardar personas, tipos de documentos, documentos e instancias de documentos utilizando los servicios implementados previamente, incluyendo las validaciones necesarias mostrar en la interfaz. En el caso de la persona, debe poder escoger el tipo de documento obligatoriamente. En el caso del documento, debe poder escoger la persona y el tipo de documento obligatoriamente (no es necesario asignar el followup). En el caso de la instancia del documento, debe poder escoger la persona y el tipo de documento obligatoriamente (no es necesario asignar el estado).
4.
  - Para cada implementación, se evalúa lo siguiente (personas, tipos de documentos, documentos e instancias de documentos):
    - a. (0.13) Plantilla incluyendo despliegue de mensajes de error.
    - b. (0.13) Controlador incluyendo validación de errores.
    - c. (0.07) Entidad modificada para validar la adición utilizando grupos.
    - d. (0.07) Conexión adecuada entre la plantilla y el controlador.
5. (1.6) Realizar las pantallas para editar personas, tipos de documentos, documentos e instancias de documentos utilizando los servicios implementados previamente, incluyendo las validaciones necesarias mostradas en la interfaz. En el caso de la persona, debe poder escoger el tipo de documento obligatoriamente. En el caso del documento, debe poder escoger la persona y el tipo de documento obligatoriamente (no es necesario asignar el followup). En el caso de la instancia del documento, debe poder escoger la persona y el tipo de documento obligatoriamente (no es necesario asignar el estado). Para cada implementación se evalúa igual que en el punto anterior.
6. (1.0) Realizar las pantallas para consultar personas, tipos de documentos, documentos e instancias de documentos. En el caso en que la entidad depende de otra, debe tener el parámetro en el path para poder seleccionar la entidad (Ejemplo, el id de la institución para mostrar los dispositivos, estados y tipos de la institución).
  - Para cada implementación, se evalúa lo siguiente (personas, tipos de documentos, documentos e instancias de documentos)
    - a. (0.15) plantilla que muestre la información.
    - b. (0.15) Controlador que permitan cargar la información a mostrar.
    - c. (0.2) Acceso a la adición, edición y entidades asociadas o padre \* (si aplica).
    - d. (0.1) Conexión adecuada entre la plantilla y el controlador.

\* El acceso a entidades asociadas o padre, implica poder ir desde una entidad a su lista de elementos asociados y volver a la entidad a la que pertenece. Suponiendo que se gestionará la institución y el campus, debería ser posible desde la consulta de instituciones acceder a los campus de esa institución y viceversa, desde el campus acceder a la consulta de la institución.