UNNOBA
Programación Orientada a Objetos
2018

Práctica 6: JSF

Presentación y objetivos

Se desea repasar en forma práctica los conceptos de JSF, frameworks y estructua de aplicaciones web explicados en la teoría.
El objetivo de esta actividad es implementar la funcionalidades de alta, baja y modificación para entidades del trabajo de cursada.

Materiales y recursos

Para el desarrollo de la actividad deberán utilizarse los siguientes materiales y recursos:
Virtual:

- Clases 4, 5 y 6.
- Prácticas previas.
- Foro de actividades prácticas.

Actividades

Se debe implementar las funcionalidades de ABM para las entidades ODS y ActividadSimple.

Vista

- Para cada entidad (ODS, ActividadSimple) crear un directorio dentro del directorio webpages.
    - Crear el directorio objetivos
    - Crear el directorio actividades_simples.
- Cada directorio deberá contener los archivos index.xtml, new.xhtml y edit.xhtml.
- En el archivo new.xhtml, deberá contener el formulario de Alta de la entidad correspondiente.
- El archiv edit.xhtml, también contendrá un formulario pero en este caso para editar los datos de una entidad existente.
- En el archivo index.xhtml debe listar todas las intancias existentes de la entidad correspondiente.
- Utilizar templates para armar las vistas. Crear archivo layout.xhtm dentro de la carpeta resources. Los archivos new.xhtml, index.xhtml y edit.xhtml deben componer el template mencionado.
    - Agregar al template un menú que nos permita navegar por las vistas de los recursos de la aplicación. En principio solo lo utilizaremos para acceder a los ABM de objetivos y actividades simples.

Controlador

- Por cada entidad deberemos implementar un backing bean como controlador para permitir realizar la operaciones de ABM.

- Crear los backings ObjetivoBacking y ActividadSimpleBacking dentro del paquete ar.edu.unnoba.poo2018.ods.controller.
- Cada backing debera tener las anotaciones @Named (o @MangedBean) y @ViewScope.
- Cada backing deberá injectar el/los DAO correspondiente/s para poder realizar operaciones de consulta y modificación sobre la base de datos. Para realizar la injeccción de dependencia de los DAO utlizar la notación @EJB (ya que al momento de definir los DAO los implementamos como Enterprise Java Beans del tipo Stateless)

Otros

- **Converts**
  - Agregar dentro del proyecto, en el paquete ar.edu.unnoba.poo2018.ods.converter la clase provista (publicada en virtual) AbstractConverter.
  - Cada entidad deberá tener su implementación de Converter, para esto por cada entidad crear una clase que extienda de AbstractConverter.
    - Cada clase debe tener la notación @FacesConverter, sobre la misma indicar el valor del atributo forClass con la clase correspondiente. (Por ejemplo para ActividadSimples, @FacesConverter(forClass=ActividadSimple.class))
    - Además cada una de las clases de converter deberá implementar el método String getDAOName(), retornando el nombre del DAO para gestionar determinada entidad. Por ejemplo para ActividadSimple la implementación del método sería “return “ActividadSimpleDao”.
  - De esta manera cada vez que JSF necesite convertir de String a su representación en Objeto y viceversa, JSF utilizará implicitamente los converters definidos según la entidad correspondiente.
    - Por ejemplo se utliza al realizar la edición de instancias de entidades o también cuando tengamos que crear instancias de entidades que se relacionan con otras entidades como es el caso de ActividadSimple.
- Todas las vistas deben estar intercionalizadas.

Criterios de valoración

Se valorará la capacidad de resolver problemas de arquitectura de aplicaciones web, el conocimiento del patrón MVC y JSF.

Entrega

La entrega podrá ser grupal con un máximo de 2 personas por grupo y se deberá entregar subiendo a Virtual un único archivo adjunto en formato “.zip”. El archivo adjunto deberá contener el código fuente de la solución planteada, notas e información adicional que considere pertinente para la correcta ejecución del programa realizado.