

Url : <https://github.com/JeremyManghisi/shopjmi>

Explicación de la estructura:

Consiste en una estructura Modelo Vista Controlador, en la que en el modelo definimos los atributos a utilizar, el método clone y los getters y setters necesarios, en el controlador con el método load asignamos los datos que se mostrarán en la vista y tienen su método add, remove y edit. En la vista se mostrarán los datos utilizando JSF y PrimeFaces.

Ejemplo de repositorio:

Tenemos en la vista, la carpeta admin en la que podrás crear editar y eliminar tanto categorías, promociones y tiendas, la carpeta public dónde saldrá el modo invitado, las tres vistas de categorías, promociones y tiendas y los templates donde tendremos la estructura de la vista que se hace con grid y se define los títulos como Categorías Admin, Modo Invitado, Modo Admin....

Luego tenemos los packages, en los que tendremos shopjmi donde estará JakartaRestConfiguration.

Tendremos el package controller donde estarán los controladores que en este caso serán AbstractController, CategoríaController, Controller, IController, PromocionController y TiendaController.

Tendremos package repository en el que estará IRepository y MemoryRepository.

Tendremos package model en el que se encontrarán los modelos, en este caso tendremos Categoría, ICopyable, Model, Promoción y Tienda.

Por último tendremos el package resources en el que estará JakartaEE91Resource.

Importante a destacar el pom.xml que tendremos que modificar varias cosas, entre ellas añadir dependencias de primefaces y jakarta.

Funcionamiento de un Controlador:

Voy a elegir CategoriaController, en este caso tenemos el constructor, luego tenemos el método load, en el que utilizaremos el create y utilizaremos los setters del modelo para asignar los valores que se mostrarán en la vista, posteriormente con el this.add lo añadiremos.

Luego tendremos el método remove que comprueba que haya elementos escritos y devolverá "remove" para eliminar el elemento que queramos.

Tendremos el método edit que devuelve simplemente "edit" para poder editar el elemento.

Y tendremos el método add que devuelve "success" si se ha podido crear el elemento y también hace un autoincremento de la id para que se vaya aumentando correlativamente y luego si el elemento ya está creado entonces hará un update.

Integración del controlador con JSF:

La verdad que es muy cómodo integrarlo utilizando los métodos load, remove, edit y add.

Ficheros de configuración de enrutamiento:

Tenemos faces-config.xml, web.xml y deleted.xhtml donde se asigna el enrutamiento