Patrones para desarrollo JSF2 (Mojarra) + JPA + Primefaces

1. Registrar metadata en base de datos, utilizando el SDK
   1.1. clase_recurso: insertar utilizando el SDK
   1.2. parametro: insertar con el proceso parametro_cargar_metadata()
   1.3. clase_recurso_parametro: insertar con el proceso clase_recurso_parametro_cargar_metadata()
   1.4. dominio: insertar utilizando el SDK
   1.5. dominio_parametro: insertar utilizando el SDK
   1.6. funcion: operaciones CRUD con el proceso funcion_cargar_metadata()
   1.7. funcion_parametro: operaciones CRUD con el proceso funcion_cargar_metadata()
   1.8. pagina: utilizando el SDK
   1.9. pagina_funcion: utilizando el SDK
   1.10. opcion_menu: utilizando el SDK

2. Crear el entity class del recurso principal en el paquete "entity": utilizar el generador de NetBeans para generar entity.
   2.1. En caso de que sea un entity class en base a una vista, se debe agregar el Id correspondiente al entity
   2.2. Agregar "NamedQuery"s conforme a las consultas que se requieran y son utilizadas en los BEANs.
   2.3. Agregar el SequenceGenerator y GeneratedValue para el id del recurso: esto sirve para crear los registros y el id se genera en base a una secuencia de la base de datos.

3. Crear las constantes del recurso en el paquete “constants”
   3.1. Nomenclatura: nombre del objeto del recurso + “Constants.java”
   3.2. Se establecen constantes de: columnas, funciones, informes, procesos y parámetros.
   3.3. Utilizar los siguientes scripts para obtener los valores constantes a partir de la metadata.

```
select  'public static final String COLUMNA_'||upper(codigo_parametro) ||' = "'|| lower
(codigo_parametro)||'";'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
      and cr.codigo_clase_recurso = 'resolucion_ministerial'
order by fp.id_funcion_parametro;
```

```
select 'public static final long FUNCION_'||upper(codigo_funcion) || ' =
'||id_funcion::varchar||'L;'
from  funcion f
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where cr.codigo_clase_recurso = 'resolucion_ministerial'
order by f.id_funcion;

select 'public static final String INFORME_FUNCION_'||upper(codigo_funcion) || ' = '||
cr.codigo_clase_recurso ||'_'|| replace( REPLACE(lower(f.codigo_funcion),'emitir_','')
,'_'||cr.codigo_clase_recurso,'')|| ';'
from  funcion f
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where cr.codigo_clase_recurso = 'resolucion_ministerial'
    and f.numero_tipo_funcion = 22
order by f.id_funcion;

select 'public static final String PROCESO_FUNCION_'||upper(codigo_funcion) || ' = '||
cr.codigo_clase_recurso ||'_'|| replace( REPLACE(lower(f.codigo_funcion),'emitir_','')
,'_'||cr.codigo_clase_recurso,'')|| ';'
from  funcion f
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where cr.codigo_clase_recurso = 'resolucion_ministerial'
    and f.numero_tipo_funcion = 13
order by f.id_funcion;

select 'public static final long PARAMETRO_'||upper(codigo_parametro) ||' = '||
p.id_parametro::varchar ||'L;'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
    and cr.codigo_clase_recurso = 'resolucion_ministerial'
order by fp.id_funcion_parametro;
```

4. Crear el DAO del objeto
   4.1. Nomenclatura: nombre del objeto + “DAO.java”

5. Crear el facade del objeto: este contiene los métodos que manipulan la base de datos
   5.1. Nomenclatura: operación CRUD + nombre del objeto
   5.2. Create
      5.2.1. Reemplazar “personaFormacion” por el nombre del recurso.

```
select  'personaFormacion.set'||replace(initcap(p.codigo_parametro),'_','')
||'(personaFormacion.get'|| replace(initcap(p.codigo_parametro),'_','')||'());'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
```

```
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
  and cr.codigo_clase_recurso = 'persona_formacion'
order by fp.id_funcion_parametro;
```

5.3. Update

5.3.1. Reemplazar persistedPersonaFormacion por “persisted”+nombre del recurso

```
select
'persistedPersonaFormacion.set'||replace(initcap(p.codigo_parametro),'_','')
||'(personaFormacion.get'|| replace(initcap(p.codigo_parametro),'_','')||'());'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
  and cr.codigo_clase_recurso = 'persona_formacion'
order by fp.id_funcion_parametro;
```

5.4. Delete

5.5. FindAll

5.6. Procesos específicos del recurso

6. Crear el bean que utilizará la página: nombre del recurso + “Bean”
   6.1. Constructor
   6.2. Validación de inicio de sesión
      6.2.1. SessionValidate
      6.2.2. UserFunctionValidate: referenciar a la función principal de la página, utilizando las constantes definidas en la librería "com.egt.base.entity.constants"
      6.2.3. No implementar el UserFunctionValidate para "mi info" porque no se controlan roles.
   6.3. Managed Component Definition: definir los componentes a utilizar en la página
      6.3.1. Definir las variables a utilizar en la página

-- inicializar objetos Disabled: uno por cada parametro del recurso

```
select 'private Boolean '|| replace(initcap(p.codigo_parametro),'_','') ||'Disabled;'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
and cr.codigo_clase_recurso = 'resolucion_ministerial'
order by fp.id_funcion_parametro;
```

6.3.2. Gets y Sets de la variables (generar con netbeans)

6.4. Validación de privilegios de usuario

6.5. Validación de despliegue de página

6.6. Métodos específicos

6.6.1. Para utilizar JPA y JPQ se debe instanciar "EntityManagerFactory" de la clase JpaUtil

6.6.2. Métodos de validación para los FK del recurso

-- funcion: setearObjetoDisabled

select 'this.'|| replace(initcap(p.codigo_parametro),'_','') ||'Disabled = trueorfalse;'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
and cr.codigo_clase_recurso = 'resolucion_ministerial'
order by fp.id_funcion_parametro;

6.7. Implementar el data model (lazy)

6.7.1. Crear el lazy del objeto principal, en el package "model"

6.7.2. Nomenclatura: nombre del objeto principal + “LazyList.java”

6.7.3. Para filtros por rango de fecha se condiciona en el LazyList.

6.8. Instanciar el facades

6.8.1. del recurso principal

6.8.2. rastroFuncionFacade y

6.8.3. rastroFuncionParametroFacade.

6.9. Métodos CRUD: se instancian los métodos definidos en el “facade.java”

6.9.1. Crear registro

6.9.2. Guardar registro

6.9.3. Actualizar

6.9.4. Eliminar

6.10. Rastros de auditoria

6.10.1. Rastro función

6.10.2. Rastro función parámetro

6.10.2.1. Para la definición de las constantes utilizar:

Para Create:

select
'getRastroFuncionParametroFacade().createRastroFuncionParametro(idRastroFuncion,PersonaFormacionConstants.PARAMETRO_'||upper(codigo_parametro) ||',
String.valueOf(personaFormacion.get'|| replace(initcap(p.codigo_parametro),'_','') ||'()));'

```
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
   and cr.codigo_clase_recurso = 'persona_formacion'
order by fp.id_funcion_parametro;
```

Para update:

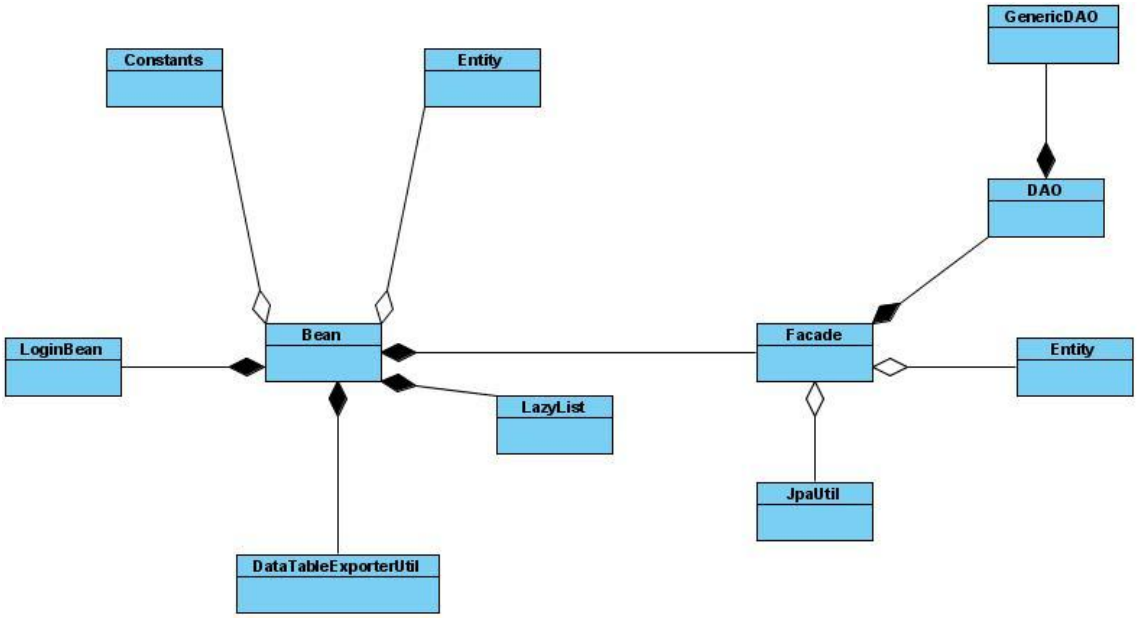
```
select
'getRastroFuncionParametroFacade().createRastroFuncionParametro(idRastroFuncion,Persona
FormacionConstants.PARAMETRO_'||upper(codigo_parametro) ||',
String.valueOf(selectedPersonaFormacion.get'|| replace(initcap(p.codigo_parametro),'_','')
||'()));'
from funcion_parametro fp
inner join parametro p on p.id_parametro = fp.id_parametro
inner join funcion f on f.id_funcion = fp.id_funcion
inner join dominio d on d.id_dominio = f.id_dominio
inner join clase_recurso cr on cr.id_clase_recurso = d.id_clase_recurso
where f.codigo_funcion like 'consultar_%'
   and cr.codigo_clase_recurso = 'persona_formacion'
order by fp.id_funcion_parametro;
```

Observacion: para los parametros tipo fecha, se debe agregar el formato con: dfDateMedium.format() y evaluar que el valor no sea nulo para aplicar el format: ej persona.getFechaCertificacion() == null?null: String.valueOf(dfDateMedium.format(persona.getFechaCertificacion()))

6.11. Procesos de exportación de datos
   6.11.1. PreProcessPDF
   6.11.2. PostProcessXLS

7. Crear las páginas
   7.1. Crear la página de lista del recurso: nombre del recurso + “CRUD.xhtml”
   7.2. Crear la página: nombre del recurso + “CreateDialog.xhtml”
   7.3. Crear la página: nombre del recurso + “UpdateDialog.xhtml”
   7.4. Crear la página: nombre del recurso + “DeleteDialog.xhtml”
   7.5. Crear la página: nombre del recurso + “ViewDialog.xhtml”
   7.6. En la página CRUD están incluidas las páginas create, update, delete, view y además las páginas de SessionTimeOut.xhtml y accesoDenegadoDialog.xhtml.

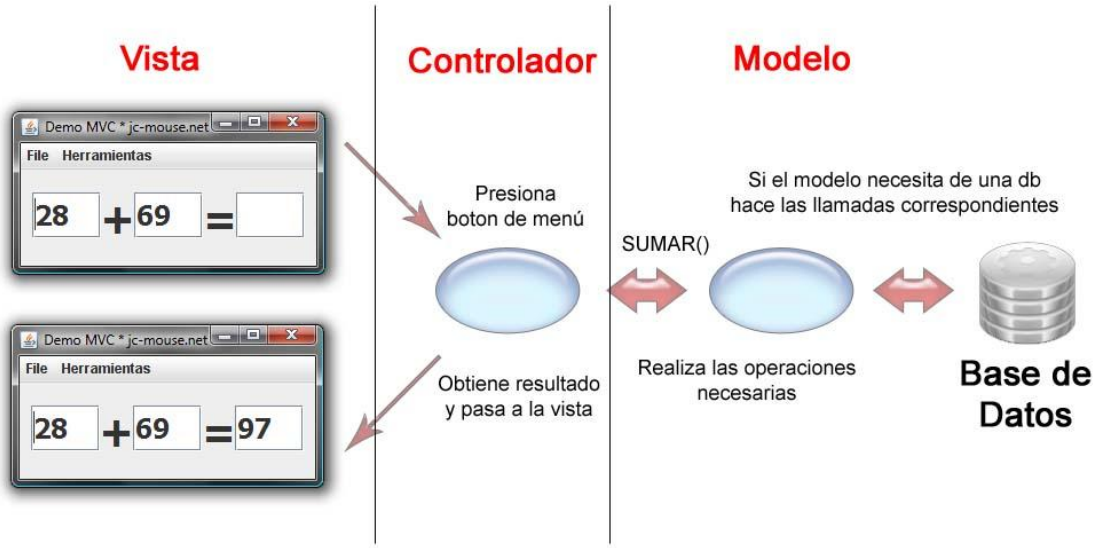
Diagrama de clases del BEAN principal



Arquitectura de la aplicación

Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datosde una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde

la vista.


Vista
Controlador
Modelo
Demo MVC * jc-mouse.net
File Herramientas
28 + 69 =
Presiona
boton de menú
Si el modelo necesita de una db
hace las llamadas correspondientes
SUMAR()
Demo MVC * jc-mouse.net
File Herramientas
28 + 69 = 97
Obtiene resultado
y pasa a la vista
Realiza las operaciones
necesarias
Base de
Datos