Patrones para desarrollo JSF2 (Mojarra) + JPA + Primefaces

1. Registrar metadata en base de datos, utilizando el SDK
   1. clase\_recurso: insertar utilizando el SDK
   2. parametro: insertar con el proceso parametro\_cargar\_metadata()
   3. clase\_recurso\_parametro: insertar con el proceso clase\_recurso\_parametro\_cargar\_metadata()
   4. dominio: insertar utilizando el SDK
   5. dominio\_parametro: insertar utilizando el SDK
   6. funcion: operaciones CRUD con el proceso funcion\_cargar\_metadata()
   7. funcion\_parametro: operaciones CRUD con el proceso funcion\_cargar\_metadata()
   8. pagina: utilizando el SDK
   9. pagina\_funcion: utilizando el SDK
   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.
   1. En caso de que sea un entity class en base a una vista, se debe agregar el Id correspondiente al entity
   2. Agregar "NamedQuery"s conforme a las consultas que se requieran y son utilizadas en los BEANs.
   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”
   1. Nomenclatura: nombre del objeto del recurso + “Constants.java”
   2. Se establecen constantes de: columnas, funciones, informes, procesos y parámetros.
   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;*

1. Crear el DAO del objeto
   1. Nomenclatura: nombre del objeto + “DAO.java”
2. Crear el facade del objeto: este contiene los métodos que manipulan la base de datos
   1. Nomenclatura: operación CRUD + nombre del objeto
   2. Create
      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;

* 1. Update
     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;

* 1. Delete
  2. FindAll
  3. Procesos específicos del recurso

1. Crear el bean que utilizará la página: nombre del recurso + “Bean”
   1. Constructor
   2. Validación de inicio de sesión
      1. SessionValidate
      2. UserFunctionValidate: referenciar a la función principal de la página, utilizando las constantes definidas en la librería "com.egt.base.entity.constants"
      3. No implementar el UserFunctionValidate para "mi info" porque no se controlan roles.
   3. Managed Component Definition: definir los componentes a utilizar en la página
      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;*

* + 1. Gets y Sets de la variables (generar con netbeans)
  1. Validación de privilegios de usuario
  2. Validación de despliegue de página
  3. Métodos específicos
     1. Para utilizar JPA y JPQ se debe instanciar "EntityManagerFactory" de la clase JpaUtil
     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;*

* 1. Implementar el data model (lazy)
     1. Crear el lazy del objeto principal, en el package "model"
     2. Nomenclatura: nombre del objeto principal + “LazyList.java”
     3. Para filtros por rango de fecha se condiciona en el LazyList.
  2. Instanciar el facades
     1. del recurso principal
     2. rastroFuncionFacade y
     3. rastroFuncionParametroFacade.
  3. Métodos CRUD: se instancian los métodos definidos en el “facade.java”
     1. Crear registro
     2. Guardar registro
     3. Actualizar
     4. Eliminar
  4. Rastros de auditoria
     1. Rastro función
     2. Rastro función parámetro
        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,PersonaFormacionConstants.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;*

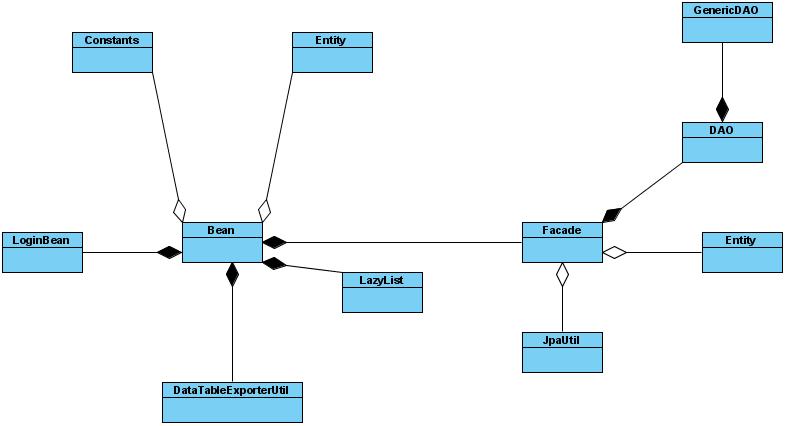
**Observacion: para los parametros tipo fecha, se debe agregar el formato con: dfDateMedium.format() y evaluar que el valor no sea nulo para aplicar el format: ej**

**persona.getFechaCertificacion() == null?null: String.valueOf(dfDateMedium.format(persona.getFechaCertificacion()))**

* 1. Procesos de exportación de datos
     1. PreProcessPDF
     2. PostProcessXLS

1. Crear las páginas
   1. Crear la página de lista del recurso: nombre del recurso + “CRUD.xhtml”
   2. Crear la página: nombre del recurso + “CreateDialog.xhtml”
   3. Crear la página: nombre del recurso + “UpdateDialog.xhtml”
   4. Crear la página: nombre del recurso + “DeleteDialog.xhtml”
   5. Crear la página: nombre del recurso + “ViewDialog.xhtml”
   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 [datos](http://es.wikipedia.org/wiki/Datos" \o "Datos)de una aplicación, la [interfaz de usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario), y la [lógica de negocio](http://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) en tres [componentes](http://es.wikipedia.org/wiki/Componentes) distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones [web](http://es.wikipedia.org/wiki/Web), donde la vista es la página [HTML](http://es.wikipedia.org/wiki/HTML) y el código que provee de datos dinámicos a la página. El modelo es el [Sistema de Gestión de Base de Datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos) y la [Lógica de negocio](http://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio), y el controlador es el responsable de recibir los eventos de entrada desde la vista. 