

PROJECTE FINAL DE CARRERA

Alumne: Jorge Casanovas Hernández

Titulació: Enginyeria Informàtica

Consultor: Josep Maria Camps Riba

Data: 16/01/2012



Presentació Aplicació de
Gestió: GCANEC

Memòria del projecte

Índex

1

- Introducció

2

- OpenFrame

3

- Escenari

4

- Base de Dades

5

- Aplicació

Introducció

- ⇒ Desenvolupament d'una Aplicació de Gestió per mantenir les versions dels paquets que formen els productes de software que es distribueixen als usuaris de l'Ajuntament de Barcelona.
- ⇒ Utilitzar entorn de Desenvolupament del l'IMI (Institut Municipal d'Informàtica), mitjançant el framework OpenFrame.
- ⇒ El desenvolupament es duu a terme segons les normatives de desenvolupament de l'IMI.

OpenFrame

Característiques arquitectura del framework:

- ⇒ De cost reduït.
- ⇒ Flexible i escalable.
- ⇒ Oberta, basada en estàndards i no lligada a cap proveïdor.
- ⇒ Fàcil d'evolucionar, ampliar i adaptara a les necessitats.
- ⇒ Fiable, estable i provada.
- ⇒ D'alt rendiment.

OpenFrame

Característiques principals:

- ⇒ Programació Orientada a Interfícies: Oferir mitjançant interfícies l'accés a la implementació.
- ⇒ Configuració declarativa: Configurar tots els serveis i elements de l'aplicació de forma declarativa sense afectar el codi.
- ⇒ Solució Oberta: Poder afegir i intercanviar qualsevol peça amb un cost molt reduït.
- ⇒ Simplificar la complexitat inherent a J2EE.
- ⇒ Oferir components de desenvolupament.
- ⇒ Proporcionar eines de suport per facilitar el desenvolupament.

OpenFrame

Components base principals i característiques:

- ⇒ **Struts:** Controlador principal ja implementat, gestió Automàtica dels formularis amb refresc entre pantalles, gestió dels errors, internacionalització de l'aplicació (multiidioma), llibreria de tags per utilitzar a les vistes, etc.
- ⇒ **Spring:** Contenedor centralitzat d'objectes i serveis, configuració d'objectes fora del codi de l'aplicació, facilita best practices com ara programar contra interfícies en lloc de contra classes, desacoblament de serveis, lògica basant-se en POJOS's, etc.
- ⇒ **Hibernate:** Mapeig objecte-relacional flexible, persistència d'objectes de manera transparent, llenguatge de querys HQL, configuració mitjançant fitxers xml, catxé, etc..

Escenari

L'aplicació està dividida en dues parts ben diferenciades:

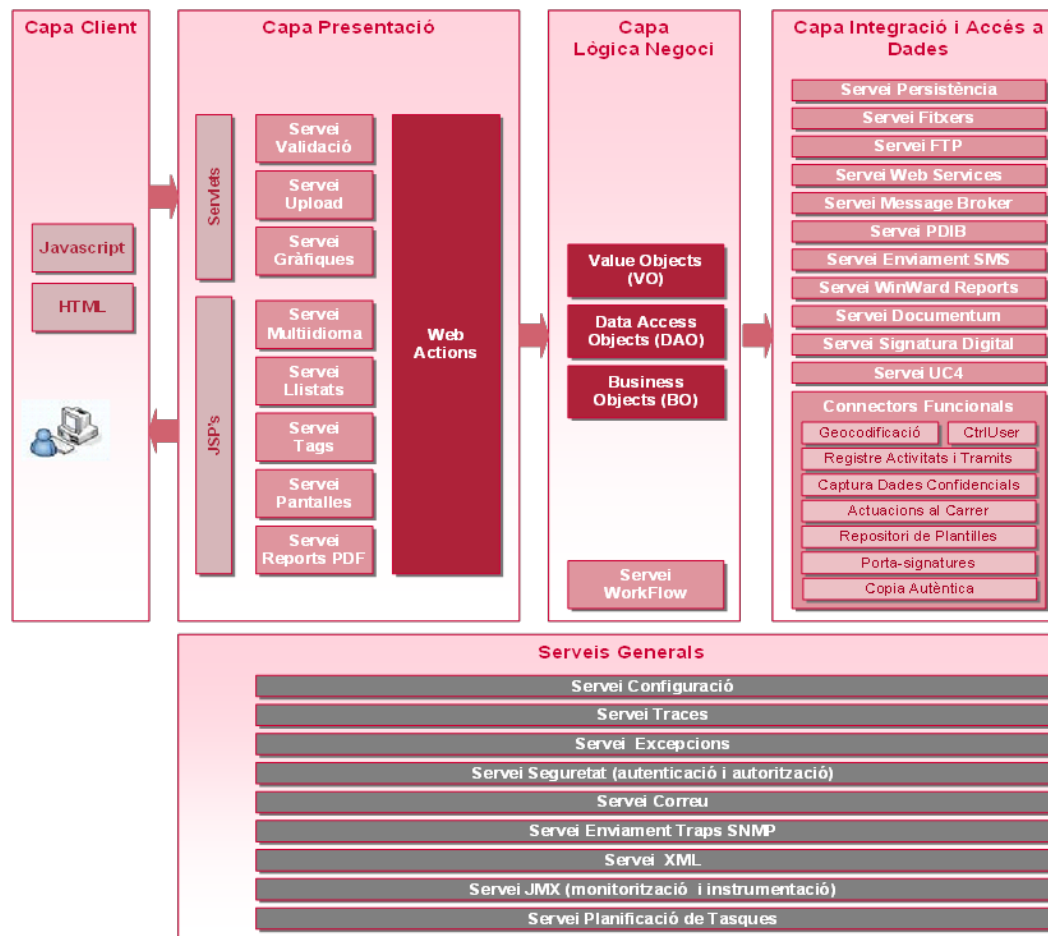
- ⇒ Base de Dades: Esquema de Base de Dades amb les taules i dades necessàries per al funcionament de l'aplicació. L'esquema estarà format per set taules: Peticions, paquets, productes, grups, contexts, estats i peticions_paquets.
- ⇒ Aplicació: L'aplicació estarà formada per quatre projectes : gcanecn, gcanecnBussines, gcanecnCommon i gcanecnWeb

Base de Dades: Taules

- ⇒ **Peticions:** Peticions de nous productes o d'actualitzacions de productes existents.
- ⇒ **Productes:** Productes que es distribueixen als usuaris.
- ⇒ **Paquets:** Paquets que formen cadascun dels productes.
- ⇒ **Estats:** Diferents estats en els quals es pot trobar una petició.
- **Grups:** Grups, formats per usuaris, als quals s'assignen els paquets.
- **Contexts:** Contexts on es distribueixen els paquets de software (Ofimàtica, Desenvolupament, Tècnics, etc..).
- **Peticions_Paquets:** Paquets que formen cadascuna de les peticions.

Aplicació

Segueix un arquitectura en 3 capes:



Aplicació: Capa Integració i Accés a Dades

Servei de Persistència basat en Hibernate:

⇒ Creació dels artefactes d'Hibernate (hibernate.cfg.xml, fitxers de mappings *.hbm.xml i VOs): Hem generat un per cada taula de la Base de Dades.

```
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
    <property name="hibernate.connection.url">jdbc:oracle:thin:@172.16.17.49:1541:CORDE1</property>
    <property name="hibernate.connection.username">gcanec_u</property>
    <property name="hibernate.connection.password">gcanec_p</property>
    <property name="hibernate.dialect">org.hibernate.dialect.OracleDialect</property>

    <!-- Application Mapping Files -->

    <mapping resource="hibernate/mappings/Peticions.hbm.xml" />
    <mapping resource="hibernate/mappings/Productes.hbm.xml" />
    <mapping resource="hibernate/mappings/Paquets.hbm.xml" />
    <mapping resource="hibernate/mappings/Peticions_Paquets.hbm.xml" />
    <mapping resource="hibernate/mappings/Grups.hbm.xml" />
    <mapping resource="hibernate/mappings/Grups_Novell.hbm.xml" />

    <!-- END OF Application Mapping Files -->
  </session-factory>
</hibernate-configuration>
```

Aplicació: Capa Negoci

⇒ Value Objects: Objectes de transferència que s'utilitzen per a la comunicació entre les 3 capes i representen les dades de les **Entitats de Negoci**. Cada Entitat es crea en un fitxer EntitatVo.java.

Package	Projecte	VO'S
es.bcn.gca.nec.common. model	gcanecnCommon	PeticionsVO.java ProductesVO.java PaquetsVO.java GrupsVO.java EstatsVO.java ContextsVO.java

Aplicació: Capa Negoci

⇒ BOs (Business Objects): Són POJOS en els que la lògica de negoci es programa en els seus mètodes. El nom dels fitxers són **EntitatBO.java** per a les interfícies i **EntitatBOImpl.java** per a la implementació.

Package	Projecte	VO'S
es.bcn.gca.nec.common.model .bo	gcanecnCommon	PeticionsBO.java ProductesBO.java PaquetsBO.java GrupsBO.java EstatsBO.java ContextsBO.java

Aplicació: Capa Negoci

Package	Projecte	VO'S
es.bcn.gca.nec.common.model.bo.impl	gcanecnBusiness	PeticionsBOImp.java ProductesBOImp.java PaquetsBOImp.java GrupsBOImp.java EstatsBOImp.java ContextsBOImp.java

⇒ Per aconseguir això s'han d'afegir els beans de negoci al fitxer **applicationContext.xml** del projecte **gcanecnBusiness** al directori **/resources/spring**.

Aplicació: Capa Presentació

⇒ Servei de Pantalles:

- Permet la creació de pantalles mitjançant l'acoblament de vèries parts.
- Permet definir herència entre pantalles.
- Permet definir diferents tipus de pantalles: vertical, portal, horitzontal.

Implementació	Fitxer	Funció
Struts	struts.congif.xml	Configurar l'ús de plantilles
Tiles	tiles.definitions.xml	Definició de pàgines

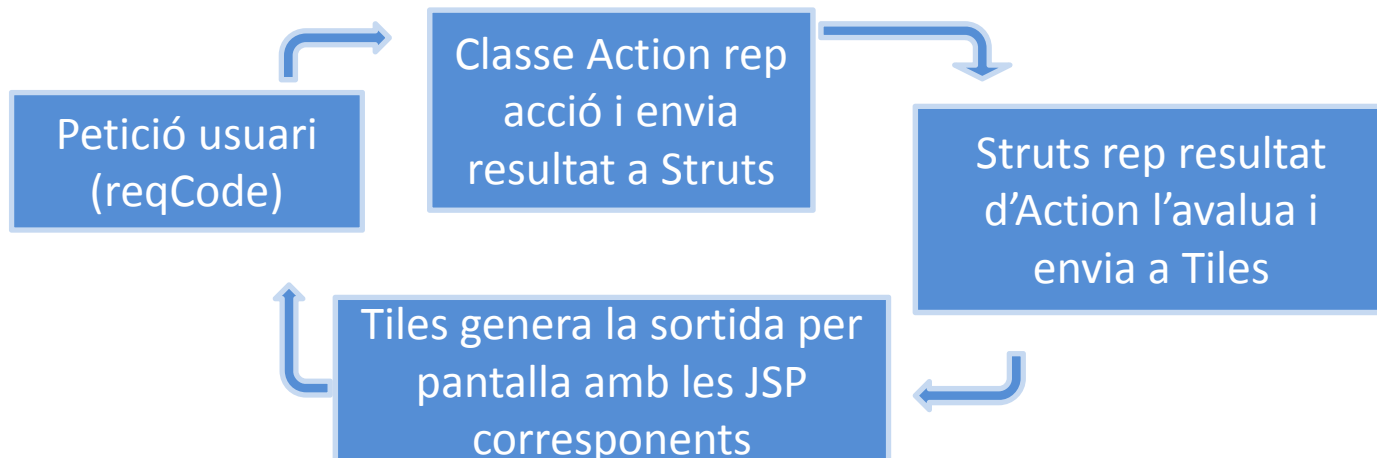
Aplicació: Capa Presentació

- Per a cada cas d'ús (Peticions, Productes, Paquets i Grups), s'han creat dues pantalles diferents. La primera mostra formulari de cerca i un llistat i la segona un formulari de creació/edició/consulta.
- La primera pantalla està formada per dos JSP.
- La segona pantalla està formada per una sola JSP que es mostrarà, mitjançant el Servei de Tags, d'una manera diferent segons si es creació, edició o consulta.
- Creació dels POJOS (DTO) per treballar amb la Capa de Persistència. D'aquesta manera no es treballa directament amb els VO.
- Creació dels fitxers de configuració per a les Action.
- Es creen les classes Action per a cada cas d'ús (Peticions, Productes, Paquets i Grups), per dur a terme les diferents accions que es poden realitzar.

Aplicació: Capa Presentació

Projecte	POJOS	Configuració	Actions
gcanecnWeb	PeticionsTO.java ProductesTO.java PaquetsTO.java GrupsTO.java	action.servlet.peticions.xml action.servlet.productes.xml action.servlet.paquets.xml action.servlet.grups.xml	PeticionsAc.java ProductesAc.java PaquetsAc.java GrupsAc.java

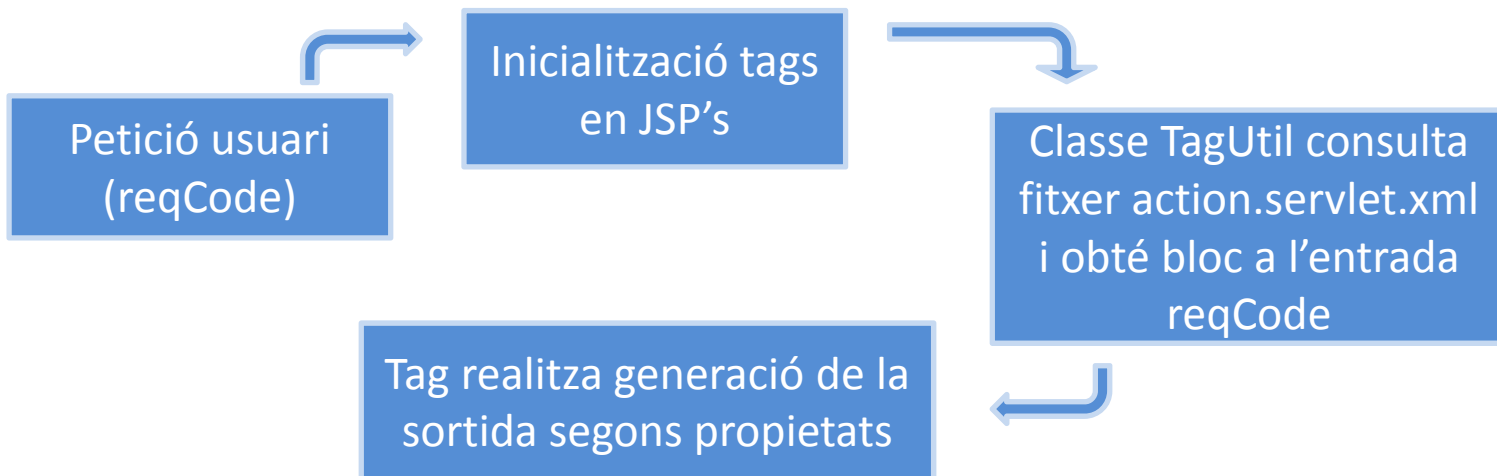
- El circuit és el següent:



Aplicació: Capa Presentació

⇒ Servei de Presentació amb Tags:

- Crear un pàgina única per a presentar formularis per a diferents propòsits.
- Marcar camp requerits automàticament en la pàgina (Servei Validació).
- Conversions automàtiques de dades.



Aplicació: Capa Presentació

- Els tags es configuren als fitxers de configuració 'action-servlet'.
- Cadascun dels camps d'un formulari han de tenir el seu propi Tag.
- Els formularis de creació, edició i consulta es mostren mitjançant la mateixa JSP, però es mostren diferent, segons la següent property.
- El value 'E,E,E' es per cadascun dels resultats (creació,edició,consulta).
- Els valors són: E(Editable), H(Ocult), S(No editable), etc..

```
<bean parent="textFieldTag">  
  <property name="styleId" value="petTiquet" />  
  <property name="mode" value="E,E,E" />  
  <property name="layout" value="true" />  
  <property name="key" value="forms.peticioForm.field.petTiquet" />  
</bean>
```

Aplicació: Capa Presentació

⇒ Servei de Llistats:

- Permet ordenar per columna de manera ascendent o descendent
 - Paginació dels resultats i navegació per pàgines.
 - Exportació automàtica del llistat a PDF o Excel.
-
- Es generen llistats per mostrar els resultats. Els llistats són els de Peticions, Paquets, Productes, etc..
 - Es generen llistats per mostrar els resultats als combos dels formularis de creació d'entitats. Els llistats són els d'Estats, Contexts, Productes i Grups.
 - Es configuren en els fitxers de configuració 'action-servlet', mitjançant la property 'valueListActionHelper'.

Aplicació: Capa Presentació

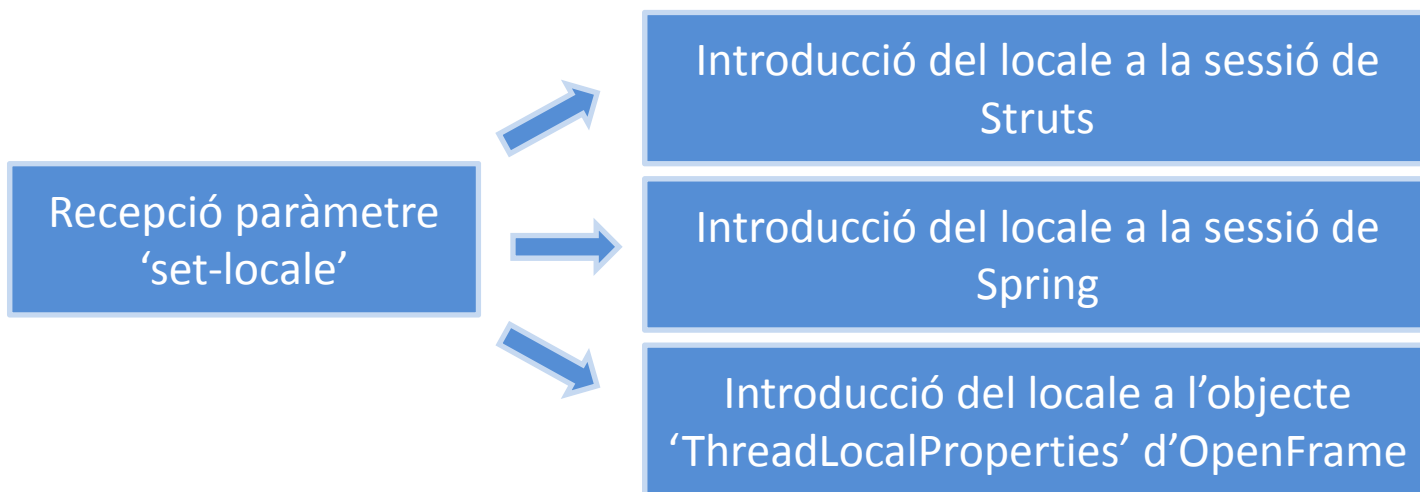
Directori	Projecte	Fitxers	Resultat
/resources/spring	gcanecnWeb	openFrame-services-web-options-lists.xml openFrame-services-web-lists.xml	Combos Llistats

```
<property name="valueListActionHelper">  
  <bean parent="valueListActionHelper">  
    <property name="listName"><value>peticioList</value></property>  
    <property name="tableId" value="PETICIONS"/>  
  </bean>  
</property>
```

Aplicació: Capa Presentació

⇒ Servei Multiidioma:

- Té com objectiu el desenvolupament d'aplicacions que presentin els seus texts i recursos en diferents llenguatges i sense que sigui necessària cap modificació del codi cada vegada que s'incorpori un nou llenguatge a l'aplicació.
- Els texts s'emmagatzemen en fitxers internacionalitzats.



Aplicació: Capa Presentació

⇒ Servei Validació:

- Comprovar que s'han introduït dades en un camp definit com obligatori.
- Comprovar que la dada introduïda es pot transformar a un tipus definit.
- Comprovar que la dada compleix amb una expressió regular o amb una fórmula.
- S'han definit validacions per a tots els formularis de l'aplicació. La validació principal es 'required' (camp requerit).
- La validació es configura als fitxers de configuració 'actionservlet', mitjançant les propietats 'validationType' i 'validatorName'.
- Les validacions es fan al costat del Servidor fent una crida AJAX.

Aplicació: Capa Presentació

- Els fitxers són els següents:

Fitxer	Funció
validation-rules.xml	Es defineixen les classes Java per al Servidor que es faran servir per els diferents tipus de validació
validation.xml	Es defineixen els tipus de validacions que podem realitzar des d'OpenFrameIML.

```
<formset>
  <form name="pro">
    <field property="prodId" depends="required">
      <arg0 key="forms.producteForm.field.prodId"/>
    </field>
    <field property="prodDesc" depends="required">
      <arg0 key="forms.producteForm.field.prodDesc"/>
    </field>
    <field property="prodDataCreacio" depends="required">
      <arg0 key="forms.producteForm.field.prodDataCreacio"/>
    </field>
  </form>
</formset>
```




Moltes gràcies

Jorge Casanovas Hernández
Estudiant d'Enginyeria en Informàtica
jcasanovash@uoc.edu