 <b>Generalitat de Catalunya</b> <b>Centre de Telecomunicacions</b> <b>i Tecnologies de la Informació</b>	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 1 / 15

## 0. INTRODUCCIÓ

### 0.1 Resum i context

En el cicle de vida d'un projecte J2EE empresarial cal que tot l'equip estigui prou informat i tingui coneixement complet de quines són les normes, guies i bones pràctiques a seguir així com saber gestionar-les. Per tal efecte ha d'existir una consciència comuna que el compliment de les normes i la gestió de la qualitat han de ser una pràctica contínua.

### 0.2 Objectiu

Recollir les normes i millors pràctiques que apliquen al desenvolupament i desplegament d'aplicacions J2EE a la Generalitat de Catalunya.

Mitigar els riscos associats a un mal desenvolupament i l'ús de males pràctiques als projectes desenvolupats per o a la Generalitat de Catalunya en la tecnologia J2EE,.

### 0.3 Àmbit i vigència

Aquesta norma va dirigida a:

- Desenvolupadors, amb l'objectiu que coneguin com han de realitzar-se els desenvolupaments de les aplicacions segons la normativa del Centre de Telecomunicacions i Tecnologies de la Informació.
- Personal d'explotació, amb l'objectiu que s'unifiquin les polítiques relacionades amb la infraestructura i les seves configuracions
- Revisors o auditors, per realitzar les tasques de revisió del compliment del normatiu

Totes les normes recollides són d'aplicació al programari desenvolupat a o per la Generalitat de Catalunya en J2EE independentment del lloc on es desplegui l'aplicació o solució.

El seguiment d'aquestes normes no s'aplica a:

- Aplicacions ja desplegades sense manteniment
- Paquets o llibreries de tercers, encara que es disposi del seu codi Font. Exemple: SAP, Alfresco, Liferay, etc. En cas que es tracti d'un manteniment d'una aplicació ja existent s'hauria de planificar una fase prèvia de redisseny de l'aplicació per al seguiment de les normes.

En aquests casos s'hauran de considerar aquelles normes que afecten a la forma dels desplegaments (separació contingut estàtic i dinàmic, configuració dels fitxers de desplegament, etc.), però no les normes que afecten directament al codi font.

Qualsevol incompliment amb aquest estàndard, haurà de ser justificat per escrit en el Document de Descripció d'Arquitectura (DA) i aprovat per la Unitat d'Arquitectura del Centre de Telecomunicacions i Tecnologies de la Informació.

### 0.4 Responsable document


Unitat d'arquitectura corporativa, estàndards i qualitat. Àrea de Solucions.

### 0.5 Normes referenciades

La present norma dona cobertura als aspectes recollits a l'estàndard J2EE publicat per Sun i les seves propostes reconegudes a tota la comunitat.

### 0.6 Definicions

- AJAX : Asynchronous Javascript And Xml
- JSONP: JavaScript Object Notation with Padding
- CORS : Cross-Origin Resource Sharing

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	NORMA		NOR0009
	Desenvolupament en J2EE		
	N.versió:03 D. actualització: 04/03/2014		Pàg. 2 / 15

- JSP : JavaServer Pages
- JSF : JavaServer Faces
- SSL : Secure Socket Layer
- JMS : Java Message Service
- Tipus de norma:
  - Obligatòria 
  - Condicional 
  - Recomanada 

## 1. ESTÀNDARDS J2EE

Els desenvolupaments a la Generalitat es basen en les millors pràctiques acceptades per la comunitat i els diferents patrons de desenvolupament que milloren el manteniment i la traçabilitat del desenvolupament.

La referència a les millors pràctiques en el desenvolupament d'aplicacions Java empresarials es pot trobar en el següent enllaç: <http://docs.oracle.com/javaee>

## 2. ARQUITECTURA FÍSICA DE LES APLICACIONS J2EE

Les aplicacions web desplegades a qualsevol infraestructura de la Generalitat ho faran en una arquitectura de 3 capes:

1. **Capa de Presentació:** servidors web (Apache). En aquests Apaches és on ha de residir el contingut estàtic de l'aplicació (html, javascript, imatges, fitxers d'estils css, etc.)
2. **Capa de Negoci:** servidors J2EE. En aquests residirà el contingut de presentació dinàmic de l'aplicació (jsp, jsf...) i la lògica de presentació, de negoci i accés a dades (.class, .jar, fitxers de configuració que requereixen, etc.). Aquí és on estaria desplegada una aplicació Canigó (Framework J2EE Corporatiu de la Generalitat).
3. **Capa d'Accés a Dades.** En aquesta capa residiran les dades de l'aplicació, en general la base de dades.

## 3. FRAMEWORK CANIGÓ

Canigó és el framework de referència per al desenvolupament d'aplicacions J2EE a la Generalitat. Està format per una sèrie de llibreries de referència àmpliament utilitzades per la comunitat i per altres desenvolupades a mida per donar resposta a necessitats habituals dins dels àmbits (diferents connectors amb altres sistemes i solucions, ...).


Les aplicacions J2EE desenvolupades per la Generalitat de Catalunya han d'utilitzar el Framework Canigó. Per a nous desenvolupaments la versió a utilitzar ha de ser la última, actualment la versió 3.0.

El framework es basa principalment en el patró MVC (Model-Vista-Controlador), on les diferents responsabilitats (presentació, lògica de negoci, ...) s'abstreuen en diferents tipus de components que treballen plegats per donar resposta a una necessitat però amb un grau de desacoblament suficient per facilitar el seu manteniment.

Per a més informació consultar <http://canigo.ctti.gencat.cat/>

## 4. QUALITAT DEL DESENVOLUPAMENT

La qualitat és un aspecte cabdal en els desenvolupaments, siguin J2EE o no.

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 3 / 15

El CTTI disposa de diferents eines per assegurar la qualitat dels desenvolupaments i es troben englobades al voltant de la solució SIC (Servei d'Integració Contínua). El SIC incorpora diferents funcionalitats relacionades amb el cicle de vida d'una aplicació:

- **Repositori de codi:** basat en Subversion (SVN), on s'haurà de repositar el codi font de totes les aplicacions, independentment de la tecnologia o de que facin servir altres funcionalitats del servei.
- **Construcció de l'aplicació (Jenkins):** permet fer la construcció del projecte reposat al subversion (sempre que l'anàlisi de la qualitat del codi i les proves unitàries hagin estat satisfactoris)
  - o **Execució de proves unitàries:** en el procés de construcció de l'aplicació s'executen les proves unitàries i en resulten dos informes, un per als testos passats i erroris, i un altre amb el percentatge de cobertura de codi.
  - o **Anàlisi de la qualitat del codi:** es realitza un anàlisi del codi envers diversos paràmetres (complexitat de mètodes, deute tècnic, regles de codificació, implementació de bones pràctiques...). Actualment s'estan utilitzant dues eines:
    - Cast (pendent d'integració amb la oficina de qualitat)
    - Sonar (a extingir)
  - o **Generació dels artefactes:** es generen els diferents artefactes estàtics i dinàmics que es desplegaran al servidor d'aplicacions i al servidor web.
- **Desplegament automàtic (Jenkins):** fa el desplegament dels artefactes construïts en servidors d'aplicacions Java (Weblogic, Tomcat, JBoss). És fa al primer entorn del que disposi l'aplicació (normalment Integració o Preproducció).

Per aconseguir fer el desplegament automàtic, que implica fer la compilació i construcció del projecte, aquests s'han d'haver construït amb Maven2.

Els servidors de destí hauran de tenir habilitat un usuari "deployer" per poder fer el desplegament automàtic de l'aplicació. Els serveis PaaS dels CPD Cloud tindran habilitats els mecanismes necessaris per a que això es pugui dur a terme.

És obligatori que el codi de tots els desenvolupaments que es facin per a la Generalitat es desin al repositori de codi del SIC.

## 5. CONFIGURACIÓ DE LES APLICACIONS

Per a desplegar una aplicació J2EE a les infraestructures de la Generalitat només caldrà acordar el codi de l'aplicació.

`<codi_aplicació>`


A partir d'aquí, serà el CPD qui proveirà les variables més comunes que permetran a l'aplicació interactuar amb la resta de serveis.

- 1) A la les JAVA\_OPTS d'arrencada de la JVM se li passarà el paràmetre "conf\_dir", on s'informarà la ruta del fitxer de propietats comunes per a aquell servidor.

`-Dconf_dir=/serveis/conf`

Aquesta variable es pot recuperar amb `System.getProperty("conf_dir")`

- 2) A la ruta `<conf_dir>/server.properties` es trobarà el fitxer que definirà les propietats comunes a totes les aplicacions que corrin en aquell servidor

 <b>Generalitat de Catalunya</b> <b>Centre de Telecomunicacions</b> <b>i Tecnologies de la Informació</b>	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 4 / 15

Aquí estarà la variable `<entorn>`, tot i que per compatibilitat amb desenvolupaments antics, aquesta variable encara es proporcionarà com a variable del sistema.

```
<codi_servei>
<smtp_host>
<smtp_port>
<proxy_host>
<proxy_port>
```


3) A la ruta `<conf_dir>/<entorn>/<codi_servei>/<codi_aplicació>/app.properties` es trobaran totes aquelles variables específiques per a una aplicació.

```
<db_pool>
<datasource>
<db_user>
<db_password>
```

Com que el codi d'aplicació estarà acordat entre desenvolupament i CPD, sistemes desarà el fitxer `app.properties` a la ruta corresponent i aquí es podran informar les variables que apliquen a l'aplicació.

## 6. CONVENCIIONS

### 6.1 Convencions de les Classes

6.1.1  El nom dels packages ha de seguir la notació 'cat.gencat....'.

El nom dels packages ha de seguir la notació:

```
cat.gencat.<ambit>.<nomaplicacio>.nomsubpackage...
```

o

```
cat.gencat.<nomaplicacio>.nomsubpackage...
```

en cas de que l'aplicació sigui d'ús corporatiu.

A continuació de 'nomaplicacio' es poden aniar nous subpackages per estructurar les classes segons la seva responsabilitat, funcionalitat, etc.

Exemples:

```
cat.gencat.ctti.facturacio.model
cat.gencat.ptop.pipex.control
```

### 6.2 Convencions de Disseny i Codificació de les Classes


6.2.1  No es permet de forma general el canvi de propietats de sistema.


Canvis en les propietats de sistema poden afectar a d'altres aplicacions desplegades en el mateix servidor.

Ex: `System.setProperty("javax.net.ssl.trustStore", "/path/trustStore");`

En cas d'incloure llibreries de tercers a l'aplicació s'hauria de validar que no es realitza aquesta pràctica gens recomanada.







### 6.3 Estructura dels Directoris de Desplegament

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 5 / 15

6.3.1  Els desplegaments s'han de definir tenint com a principal premissa la separació del contingut estàtic i el contingut dinàmic en 2 artefactes diferenciats.






6.3.2 Directoris del Contingut Estàtic i Dinàmic

6.3.2.1  Estructura de directoris d'una aplicació J2EE (mòdul EAR).

	META-INF			Directorio específic on s'inclouran els fitxers <ul style="list-style-type: none"> <li>application.xml</li> <li>weblogic-application.xml (en cas que el servidor d'aplicacions sigui BEA WLS)</li> <li>Altres fitxers específics del servidor d'aplicacions</li> </ul>
		Mòdul 1		Mòdul WAR / JAR / RAR. En cas que sigui un WAR ha de complir l'estructura especificada a 6.3.1.2
		Mòdul 2		Idem
		...		...
		APP-INF		
			lib	Directorio de llibreries (jar) comunes a tots els mòduls. Aquest serà creat de forma automàtica per Maven.



6.3.2.2  Estructura de directoris de la part dinàmica de l'aplicació (mòdul WAR)


La part dinàmica ha de ser estructurada de la següent forma:



	WEB-INF			Directorio específic on s'inclouran els fitxers <ul style="list-style-type: none"> <li>web.xml</li> <li>weblogic.xml (en cas que el servidor d'aplicacions sigui BEA WLS)</li> <li>Altres fitxers específics del servidor d'aplicacions</li> </ul>
		jsp		Directorio de pàgines JSP
		classes		Directorio de classes Java
			config	Subdirectorio de fitxers de configuració de l'aplicació. Es poden crear subdirectoris per estructurar-lo.  En aquest directorio es troben els fitxers de configuració aplicables a tots els entorns. Per aquells paràmetres que tinguin valors diferents per entorn es crearan diferents subcarpetes.
		lib		Directorio de llibreries (jar). Aquest serà creat de forma automàtica per Maven.
welcome.jsp				Pàgina inicial de l'aplicació

6.3.2.3  Estructura de directoris de la part estàtica de l'aplicació

La part estàtica ha de ser estructurada com segueix:

	html	Directorio de fitxers HTML, HTM
	css	Directorio de fitxers d'estil CSS

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 6 / 15





	scripts	Directori de fitxers javascript
	images	Directori d'imatges (.png, .gif, .jpg, ...)

### 6.3.3 Directoris de Fitxers de Traces i altres fitxers permanents entre desplegaments

#### 6.3.3.1 Ubicació dels fitxers de logs i de tota informació que no es vulgui perdre en cas de fer un redesplegament

En cas que es faci un redesplegament de l'aplicació, tot el contingut previ que hi havia al servidor s'esborra i es torna a publicar. Això implica que aquells fitxers que no es troben pròpiament a l'empaquetat i s'han generat en la pròpia execució de l'aplicació s'hauran perdut (logs, arxius generats per l'aplicació, etc.). Per evitar-ho cal utilitzar un directori extern a l'aplicació, en el directori indicat a l'apartat "5. CONFIGURACIÓ DE LES APLICACIONS"

Utilitzar la següent estructura:

	logs	Subdirectorí on s'han de deixar els fitxers de traces (obligatori)
	out	Directorí on es poden ubicar els fitxers que s'hagin d'enviar o s'hagin generat de forma dinàmica (fitxers XML de sortida, ...)
	in	Directorí on ubicar aquells fitxers que hagin estat rebuts per l'aplicació
	...	En cas necessari, l'aplicació pot definir altres subdirectoris

#### 6.3.3.2 S'ha de fer servir un nom de fitxer de log diferent per cada instància del clúster

Per tal d'evitar problemes de bloquejos en els fitxers de logs, cal que les aplicacions configuren un fitxer diferent a cada instància.

En l'apartat de **Convencions de Traces d'Aplicació** s'explica com configurar els logs per a especificar el nom de la instància del servidor d'aplicacions en el nom del fitxer a on s'escriuran els logs.

### 6.3.4 Directoris de Configuració de l'Aplicació


#### 6.3.4.1 Estructura de directoris de configuració de l'aplicació segons l'entorn




Una aplicació, a més de fer ús de JNDI per accedir a recursos, pot utilitzar també paràmetres propis de l'aplicació (exemples: correu electrònic de l'administrador, cicle en el que ha d'executar un daemon, etc.). Els valors d'aquests paràmetres poden variar en cada entorn.

Com gestionar els paràmetres en cada entorn és un aspecte força important i que no s'ha de descuidar. No es permet el desplegament de paquets diferents per cada entorn si no un únic paquet de desplegament que ha de ser vàlid per tots els entorns: l'aplicació ha de seleccionar quins fitxers de propietats carregar en funció de la variable de sistema <entorn> que s'habilita a tal efecte, amb els valors "int", "pre" o "pro", tal i com s'explica en els següents apartats (per a Canigó 2 i 3) i en l'apartat 5 d'aquest document.

#### 6.3.4.2 Estructura i format dels arxius de configuració per a Canigó 2.x

Els fitxers de configuració han de residir al directori 'WEB-INF/classes/config'. Si determinats paràmetres tenen valors que depenen de l'entorn, s'ubicaran en fitxers sota un dels següents subdirectoris:

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	NORMA		NOR0009
	Desenvolupament en J2EE		
	N.versió:03 D. actualització: 04/03/2014		Pàg. 7 / 15


	int	Ubicació dels fitxers amb informació de configuració aplicable a l'entorn d'integració
	pre	Ubicació dels fitxers amb informació de configuració aplicable a l'entorn de preproducció
	pro	Ubicació dels fitxers amb informació de configuració aplicable a l'entorn de producció

#### 6.3.4.3 Estructura i format dels arxius de configuració per a Canigó 3.x


Els fitxers de configuració han de residir al directori "WEB-INF/classes/config/props". Si determinats paràmetres tenen valors que depenen de l'entorn, les propietats dependents s'hauran d'informar seguint el format de "multiproperties" adoptat per aquesta versió:

*.mail.host=smtp.gencat.cat	Propietat genèrica en cas de no trobar cap d'específica per a l'entorn on es troba desplegada l'aplicació. En aquest cas l'entorn d'integració seleccionaria aquesta propietat ja que no hi ha definida cap exclusiva per al seu entorn.
pro.mail.host=smtp.pro.gencat.cat	Propietat exclusiva per a l'entorn de Producció
pre.mail.host=smtp.pre.gencat.cat	Propietat exclusiva per a l'entorn de Preproducció

## 6.4 Convencions d'Urls i pàgines

6.4.1  Utilitzar com a pàgina 'welcome' (pàgina d'inici) una pàgina que redirigeixi a la url principal de l'aplicació.

Per independitzar el coneixement per part de l'usuari o de qualsevol enllaç extern de quina és la url lògica principal de l'aplicació, és obligatori definir una pàgina de 'welcome' al fitxer 'web.xml'

6.4.2  Nom i contingut de la pàgina "welcome" per a Canigó 2.x.

Aquesta pàgina s'ha d'anomenar 'welcome.jsp' i tenir un contingut com el mostrat a continuació:


```
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<c:redirect url="home.do">
```

NOTA: Es pot escollir l'ús d'una pàgina estàtica html 'welcome.html' que simuli el mateix comportament.

Segons l'estructura de directoris de desplegament definida, les pàgines JSP han de residir al subdirector 'WEB-INF/jsp'. En el cas de la pàgina 'welcome.jsp' s'ha d'ubicar a l'arrel de l'aplicació Web per a que funcioni correctament. Per tant, la pàgina de benvinguda 'welcome.jsp' és l'única pàgina que pot residir directament a l'arrel de l'aplicació Web.


Configurar al fitxer 'web.xml' el següent contingut:

```
<welcome-file-list>
<welcome-file>welcome.jsp</welcome-file>
</welcome-file-list>
```

6.4.3  Nom i contingut de la pàgina "welcome" per a Canigó 3.x.

Aquesta pàgina s'ha d'anomenar "index.html" i tenir un contingut com el mostrat a continuació:

```
<!DOCTYPE html>
<html>
```

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 8 / 15

```
<head>
<meta http-equiv="Refresh" content="0;url=AppJava/views/index.jsf?set-locale=ca_ES"
/>
<title>redirigint ...</title>
</head>
<body>
    redirigint ...
</body>
</html>
```

#### 6.4.4 Usar plantilles per la composició i reutilització de pantalles.

L'ús de plantilles en la creació de pantalles d'aplicacions web simplifica el seu desenvolupament i fomenta la reutilització de blocs de pantalles d'ús comú. Alguns d'aquests blocs d'ús freqüent en el desenvolupament de pantalles web són: els menús, capçaleres o peus de pàgines entre altres.

El Framework Canigó segons la seva versió utilitza diferents tecnologies per oferir aquesta composició de pantalles. Per a la versió 3 del Framework s'utilitza *Facelets*, mentre que a les versions anteriors s'utilitza *Tiles*.

En la construcció d'aplicacions web modernes es tendeix a la creació d'aplicacions basades en la separació del backend i del frontend, de manera que el backend exposa serveis (REST) i el client és una aplicació web basada en tecnologia estàtica (html/css/javascript) que es comunica amb el backend mitjançant AJAX o JSONP o CORS que "pinten" la informació al client. El concepte de plantilles també existeix en aquesta arquitectura d'aplicacions i és perfectament vàlid i recomanable el seu ús.

En el cas d'aplicacions basades en una única pàgina (SPA – Single Page Applications) aquest concepte també aplica.

#### 6.4.5 Els missatges mostrats per pantalla han d'estar preparats per múltiples idiomes.

No es poden escriure directament al codi o a les pantalles els missatges de text a mostrar als usuaris. S'han d'utilitzar claus pels missatges i mitjançant les funcionalitats d'internacionalització que ofereix Java resoldre aquests literals segons el idioma que estigui utilitzant l'usuari.


Un exemple de codi d'una pantalla seria:

```
<tr>
    <td align="right"><fmt:message key="jsp.login.user" /></td>
    <td><input class="textLogin" type='text' name='j_username'
value="usuari_admin1"></td>
</tr>
<tr>
    <td align="right"><fmt:message key="jsp.login.password" /></td>
    <td><input class="textLogin" type='password' name='j_password'
value="password1"></td>
</tr>
```

I dels fitxers de propietats:

```
Fitxer applicationResources_es.properties (pels literals en castellà)
#login.jsp
jsp.login.user=Nombre del usuari
jsp.login.password=Contraseña
```




 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	NORMA		NOR0009
	Desenvolupament en J2EE		
	N.versió:03 D. actualització: 04/03/2014		Pàg. 9 / 15

```


Fitxer applicationResources_ca.properties (pels literals en català)
#login.jsp
jsp.login.user=Nom d'usuari
jsp.login.password=Paraula clau

```

En el cas de desenvolupar aplicacions com les comentades anteriorment on es separa el backend i el frontend, en aquest últim cas existeixen estratègies d'internacionalització i localització però basades en tecnologia de client (fitxers JSON, ..., amb JQuery, per exemple, o AngularJS)


**6.4.6**  Les validacions de dades poden realitzar-se a client, però forçosament també han d'existir a servidor.


En les pàgines on existeixin formularis per l'entrada de dades (Data Entry), el fet que incorporin validacions no exigeix que hagin d'executar-se també a servidor. D'aquesta manera s'eviten possibles entrades de dades no vàlides per altres vies que no sigui el corresponent formulari web.

**6.4.7**  Definir el scope més apropiat per cada vista.


Cada pàgina (vista) pot requerir mantenir el seu estat durant diferents peticions del client al servidor. A Canigó 3 (JSF) existeix el concepte de scope associat al backing bean que gestiona la vista, i és per aquest que s'ha de configurar el scope més restrictiu. L'objectiu és minimitzar el consum de memòria mantenint l'estat del que sigui estrictament necessari.

## 6.5 Convencions d'Accés a la Base de Dades

**6.5.1**  No es permet l'ús d'accessos directes JDBC a base de dades sense pool de connexions. No es permet l'ús d'accessos JDBC de forma directa a la base de dades, ni la definició de fitxers que continguin informació de l'usuari i password de connexió a la base de dades. Només es permet l'ús de pools de connexions definides mitjançant JNDI al Servidor d'Aplicacions i els seus corresponents datasources.

**6.5.2**  No construir sentències SQL mitjançant concatenació de variables. No es permet executar sentències SQL que han estat construïdes mitjançant la concatenació de variables. Aquestes crides s'han de fer mitjançant l'ús de *PreparedStatement* o bé altres solucions proporcionades pel servei de persistència del Framework Canigó.


Al executar aquest tipus de sentències es poden produir atacs d'injecció de codi a la Base de dades (*SQLInjection*) i provocaria un greu problema de seguretat per a les aplicacions.

**6.5.3**  Independitzar el màxim possible les sentències d'accés a bbdd de la tecnologia subjacent.

El llenguatge d'accés preferent per l'accés a bbdd ha de ser JPQL (JPA), HQL (Hibernate, però podria ser una altre ORM) i SQL per aquest ordre. Quan més independent de la tecnologia base utilitzada siguin els accessos, més fàcil serà en un futur poder canviar aquestes tecnologies. Per exemple canviar d'Oracle a MySQL o d'Hibernate a Toplink Essentials.

**6.5.4**  Definir transaccionalitat a nivell de mètode de negoci.

La transaccionalitat de l'aplicació ha d'implementar-se a nivell de mètode de negoci donat que cadascun d'aquests mètodes pot implicar diverses operacions a bbdd. Convencions d'ús servei de missatgeria (JMS)

 <b>Generalitat de Catalunya</b> <b>Centre de Telecomunicacions</b> <b>i Tecnologies de la Informació</b>	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 10 / 15

#### 6.5.5 Revisió dels paràmetres de configuració del servei.

La configuració final d'una cua o tòpic JMS depèn de l'ús que faci l'aplicació, però sempre cal revisar els següents paràmetres per adequar els seus valors a les necessitats reals de l'aplicació.

- Persistència del missatge. La persistència del missatge garanteix l'entrega del missatge però consumeix més recursos del servidor. Així doncs la persistència només s'ha de fer en cas que la garantia d'entrega dels missatges sigui un requeriment funcional.
- Ús de cues/tòpics distribuïdes. L'ús de recursos distribuïts garanteix l'entrega a tots els membres subscrits als missatges dintre d'un *cluster*.

#### 6.5.6 Nomenclatura de definició dels JMSServer i les cues i tòpics.

Tipus recurs	Subtipus	Nomenclatura	Exemple
JMS	JMSServer	jms/Server<NomAplicació>	jms/ServerTramits
JMS	Tòpic	jms/Topic<NomAplicació>	jms/TopicTramits
JMS	Tòpic distribuït	jms/dTopic<NomAplicació>	jms/dTopicTramits
JMS	Cua	jms/Queue<NomAplicació>	jms/QueueTramits
JMS	Cua distribuïda	jms/dQueue<NomAplicació>	jms/dQueueTramits

### 6.6 Convencions de Traces (Logs)

#### 6.6.1 Nomenclatura dels noms de fitxers de traces.

El nom dels fitxers de traces generats pels aplicatius ha de tenir el següent format

`<AAAAAA>.[<Nom-Servidor>].log.<YYYYMMDD>`

On:

- 'AAAAAA' és l'identificador o codi de l'aplicació.
- Nom-Servidor s'utilitzarà únicament en els casos en el que l'aplicació es desplegui en un clúster. En aquest cas identificarà el nom del servidor des del qual s'ha generat el fitxer.
- 'YYYYMMDD' és la data de generació del fitxer.

Configuració:


```
<appender name="file" class="org.apache.log4j.DailyRollingFileAppender">
  <param name="DatePattern" value="'.'yyyyMMdd"/>
  <param name="File" value="/export/AppJavaDades/GDO/logs/
GDO.{$weblogic.Name}.log"/>
  <param name="Append" value="true"/>
</appender>
```

#### 6.6.2 Ubicació del fitxer de traces.

`[directori de l'aplicació]/logs`

#### 6.6.3 Periodicitat del fitxer de traces

El fitxer de traces ha de tenir una periodicitat diària, és a dir s'ha de crear un nou fitxer de log per aplicació cada dia, on només es recullen els logs generats durant aquell dia.

 <b>Generalitat de Catalunya</b> <b>Centre de Telecomunicacions</b> <b>i Tecnologies de la Informació</b>	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 11 / 15

#### 6.6.4 Format del fitxer de traces

Les dades s'han d'emmagatzemar en format de text pla segons la següent especificació:

`<Text> <Data> <Nivell> [<Pid>] <Classe> - <Missatge>`

On cada element representa aquesta informació:

*Text:* Text fix, comú per a totes les traces de log

*Data:* Hora completa i data en la que es va generar la traça en el format yyyyMMdd HH:mm:ss,SSS

*Nivell:* El nivell pot ser:

- **DEBUG.** Missatge de traçabilitat, normalment associat al desenvolupament. En producció no hauria d'aparèixer cap missatge d'aquest nivell.
- **INFO.** Missatge informatiu del que fa l'aplicació.
- **WARN.** Missatge d'alerta. Per exemple pot ser un avís que ens estem quedant sense connexions.
- **ERROR.** Missatge d'error (pot ser tant funcional com tècnic).
- **FATAL.** Missatge d'error molt greu.

*Pid:* Identificador del procés (és el procés Java que executa l'aplicació)

*Classe:* Informació de la classe que ha produït el missatge. Ha de ser fully qualified, és a dir ha d'indicar la ruta completa dels packages. Exemple: `cat.gencat.tramits.Tramitador`

*Missatge:* Missatge de la traça. Si és una excepció contindrà el detall complet d'aquesta.

La configuració del layout de log4j que s'ha de realitzar per produir aquestes traces de log és la següent:

```
<layout class="org.apache.log4j.PatternLayout">
  <param name="ConversionPattern" value="Message: %d{dd MM yyyy HH:mm:ss,SSS}
  %-5p [%t] %C - %m" />
  <param name="Throwable" value="textException"/>
  <param name="ExceptionMaxLength" value="2048"/>
</layout>
```

En el cas que es vulgui utilitzar el servei de visualització de traces per pantalla incorporat en el Framework Canigó, caldrà generar també un fitxer de log en format XML.

Configuració:


```
<appender name="fileXML" class="org.apache.log4j.DailyRollingFileAppender">
  <param name="DatePattern" value="'.'yyyyMMdd"/>
  <param name="File" value="/export/AppJavaDades/GDO/logs/
  GDO.{$weblogic.Name}.xml"/>
  <param name="Append" value="true"/>
</appender>
```

L'especificació per aquest format, equivalent a l'anterior definida per a text pla, és la següent:

```
<Registre>
  <Text />
  <Data />
  <Nivell />
  <Pid />
  <Classe />
  <Missatge />
</Registre>
```

Exemple:

```
<Registre>
  <Text>Message:</Text>
  <Data>20110110 11:01:46,059</Data>
```

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 12 / 15

```

<Nivell>DEBUG</Nivell>
<Pid>[http-8080-1]</Pid>
<Classe>
cat.gencat.ctti.canigo.arch.web.jsf.filter.ErrorHandlerFilter
</Classe>
<Missatge>ENTERING the doFilter of ErrorHandlerFilter</Missatge>
</Registre>

```

La configuració del layout de log4j que s'ha de realitzar per produir aquestes traces de log és la següent:

```

<layout
class="cat.gencat.ctti.canigo.arch.core.logging.reader.xml.PatternXMLLayout">
  <param name="NodesValuePairDelim" value="="/>
  <param name="NodesVarDelim" value=";" />
  <param name="NodesConversionPattern" value="Text=Message:;Data=%d\{yyyy MM dd
HH:mm:ss\};
  Nivell=%-5p;Pid=[%t];Classe=%C;Missatge= - %m;" />
  <param name="Throwable" value="textException"/>
  <param name="ExceptionMaxLength" value="2048"/>
</layout>

```

També es pot afegir informació contextual al missatge de text, com per exemple l'identificador de l'usuari que ha generat la traça o altres camps que vulguem posteriorment cercar i agrupar en els logs. A continuació es llisten alguns d'aquests camps opcionals:

- TipusError (opcional). En cas que la traça vagi associada a un error, indicar en aquest camp el codi de l'error. Aquest codi ha de ser numèric i permetrà poder donar una solució ràpida a l'error en tant que el codi estigui catalogat i s'ofereixi una solució.
- Mòdul (opcional). Nom del mòdul de l'aplicació que ha generat la traça
- Mètode (opcional). Mètode de la classe en la que s'ha produït el missatge
- IdUsuari (opcional). En cas que la traça es generi en una aplicació amb autenticació s'indicarà aquí el codi identificador de l'usuari.
- UrlPaginaWeb (opcional). Pàgina de presentació que ha provocat la traça.

En cas que el missatge es generi en una petició Web associada a una url s'indicarà en aquest camp la ruta completa a la sol·licitud.

- TracaExcepció (opcional). En cas que el missatge sigui provocat per una excepció s'indica el missatge complet de l'excepció (com un stackTrace).
- Node (opcional). En cas que l'aplicació es trobi desplegada en un clúster, podem generar al missatge quin node del clúster ha generat el missatge.


Exemple IdUsuari:

```

<layout
class="cat.gencat.ctti.canigo.arch.core.logging.reader.xml.PatternXMLLayout">
  <param name="NodesValuePairDelim" value="="/>
  <param name="NodesVarDelim" value=";" />
  <param name="NodesConversionPattern" value="Text=Message:;Data=%d\{yyyy MM dd
HH:mm:ss\};
  Nivell=%-5p;Pid=[%t];Classe=%C;Missatge= - %m;
  IdUsuari=%X\{userId\};" />
  <param name="Throwable" value="textException"/>
  <param name="ExceptionMaxLength" value="2048"/>
</layout>

```

En la documentació del Servei de Traces del Framework Canigó s'explica com fer accessible aquesta informació contextual al fitxer de configuració de log4j.


 <b>Generalitat de Catalunya</b> <b>Centre de Telecomunicacions</b> <b>i Tecnologies de la Informació</b>	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 13 / 15

## 6.7 Convencions de Càrrega de Classes i Llibreries

6.7.1  No es permet modificar el CLASSPATH del Servidor d'Aplicacions.

## 6.8 Convencions de Seguretat

6.8.1 Convencions d'ús de SSL

6.8.1.1  Es recomana, si l'aplicació ha de ser segura, implementar el tractament de peticions SSL

- 1) El certificat CDS normalment s'instal·larà a l'accelerador SSL
- 2) Si la petició arriba pel port 80, aquesta passarà directament al balancejador i mitjançant aquest a un servidor web (Apache)
- 3) Si la petició arriba per HTTPs (port 443), hi ha una redirecció a l'accelerador d'encriptació SSL. Aquest realitzarà la desencriptació de les dades rebudes i les passarà als Apache pel port 444 i HTTP. Per tant, tot i que la comunicació inicial era HTTPs, l'aplicació rebrà la petició com si de HTTP es tractés, però pel port 444.

6.8.2 Procediment que ha de seguir una aplicació per utilitzar SSL

Canigó ofereix un mecanisme transparent per tractar amb les peticions de forma segura sota els requeriments dels Centre de Telecomunicacions i Tecnologies de la Informació.

6.8.3 Autenticació/Validació per certificats de client

El balancejador / accelerador SSL de XCAT té la capacitat de requerir certificat de client per a una IP / domini concret. Per tant, seguint els passos per a una aplicació que necessiti SSL que no disposi de domini amb certificat associat, es pot configurar per tal que el demani.

S'ha de tenir en compte que el balancejador, després de validar el certificat, passa les dades del certificat a l'aplicació.

6.8.4 Validació de certificats per servidor

En cas que es vulguin validar certificats al servidor enlloc del client, s'ha d'utilitzar el servei de validació del CATCert. Consultar la url:

<http://www.aoc.cat/Inici/SERVEIS/Signatura-electronica-i-seguretat/Validador/Que-es> ' per a més informació.


6.8.5 Crida a web services per SSL

6.8.5.1  Incloure certificats de servidor a l'aplicació

A les aplicacions que accedeixen a serveis web segurs, pot existir el dubte d'on s'han d'indicar els certificats:

- al servidor d'aplicacions
- als cacerts de la JVM
- a l'aplicació

La última opció és la correcta per a no interferir a d'altres aplicacions desplegades sobre el mateix servidor d'aplicacions. Destacar que l'ús del certificat s'ha de fer via configuració del stub client i no mitjançant propietats de sistema (Ex. `System.setProperty("javax.net.ssl.trustStore", "/path/trustStore")`) que poden afectar a d'altres aplicacions dins el mateix servidor.

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 14 / 15

## 6.9 Configuracions d'Idioma i encoding

6.9.1  Definir el encoding que utilitzarà l'aplicació a 'UTF-8' o 'ISO-8859-15'

Preferentment, les aplicacions Canigó utilitzaran l'encoding 'UTF-8'.


## 6.10 Precompilació de JSPs


6.10.1  No precompilar les JSP de les aplicacions.

Es recomana no precompilar les JSPs de les aplicacions ja que en la majoria dels casos pot afectar en el temps d'arranc dels servidors.

En cas de requerir que algunes jsp estiguin disponibles per als primers accessos a l'aplicació, s'hauran d'avaluar alternatives.

## 6.11 Convencions de Sessions


6.11.1  Tot objecte que s'emmagatzemi a la sessió HTTP ha d'implementar la interfície 'Serializable'

6.11.2  A part de tenir en compte que els objectes que s'emmagatzemen a sessió siguin serialitzables cal tenir en consideració els paràmetres de com es realitza la replicació i la persistència. Minimitzar l'us de sessió HTTP

A la sessió HTTP han d'emmagatzemar-se únicament els objectes que requereixin un temps de vida superior al de la petició HTTP, i un cop se'n pugui prescindir s'han d'eliminar. No es recomana que creixi gaire el tamany d'aquesta sessió per evitar problemes de consum de memòria en el servidor i de latència en entorns clúster on s'ha de moure aquesta informació entre els diferents nodes.

## 7. BIBLIOGRAFIA

- Portal Canigó: <http://canigo.ctti.gencat.cat>
- SIC: <http://canigo.ctti.gencat.cat/confluence/display/CANPUB/Sobre+el+SIC>

 Generalitat de Catalunya Centre de Telecomunicacions i Tecnologies de la Informació	<b>NORMA</b>	NOR0009
	<b>Desenvolupament en J2EE</b>	
	N.versió:03 D. actualització: 04/03/2014	Pàg. 15 / 15

## 8. CONTROL DOCUMENT I REGISTRE DE CANVIS

Revisió	Redactat per	Revisat per	Aprovat per	Data aprovació	Data publicació
03	Unitat d'Arquitectura, Estàndards i Qualitat		CD CTTI	30/05/2013	
03.01	Unitat d'Arquitectura, Estàndards i Qualitat		CD CTTI	04/03/2014	

Revisió	Apartat	Data Modificació	Motiu del canvi
03		28/02/2014	Actualització de la norma anterior



**Llicència Creative Commons:**

**Reconeixement – No Comercial – Compartir Igual 2.5.**

**Sou lliure de** copiar, distribuir i comunicar públicament l'obra, així com de fer-ne obres derivades, **en les següents condicions:**



**Reconeixement.** Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el licenciadador.



**No comercial.** No podeu utilitzar aquesta obra per a finalitats comercials.



**Compartir amb la mateixa llicència.** Si altereu o transformeu aquesta obra, o en genereu obres derivades, només podeu distribuir l'obra generada amb una llicència idèntica a aquesta.

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.

**Podeu trobar el text legal de la llicència a:** [Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License](https://creativecommons.org/licenses/by-nc-sa/2.5/)