GOVERN DE LES ILLES BALEARS

D.G.T.I.C.

Estándar de desarrollo de aplicaciones del Govern de les Illes Balears

Versión 4.6

Fecha Revisión: 20/12/05

Indice de contenidos

Introducción

A. Las aplicaciones se desarrollarán siguiendo los siguientes estándares publicados por la *Direcció General de Tecnologia i Comunicacions*:
   - METRICA versión 3
   - Estándar de desarrollo de aplicaciones del *Govern de les Illes Balears*
   - Estándar de *interface* de usuario (libro de estilo)

B. Las aplicaciones deberán desarrollar los módulos mediante aplicaciones distribuidas a tres niveles (interfaz, lógica y datos).

C. El software de base a utilizar será el que se detalla a continuación:

Modelo tres niveles		
Área	**Producto**	**Tecnología**
Interfaz de usuario	Tomcat 5.0.28	Servlets 2.2 JSP 1.1
Lógica de aplicación	Jboss 3.2.7-caib1[1] JVM: j2sdk-1.4.2.10	EJB 2.0
Base de datos	Oracle 9.2.0.5	JDBC ANSI-SQL

Aplicaciones Lotus Domino		
Área	**Producto**	**Tecnología**
Aplicación	Domino Server 6.5.5 Lotus Notes 6.5.1	Lotus Script
Flujo de Procesos	Domino Workflow 6.5.1	Lotus Script
Almacén de documentos	Document Manager 6.5.1	Lotus Script

En función de criterios de mantenimiento y disponibilidad de versiones y con el objetivo de mejorar el servicio ofrecido a las c*onsellerias*, el Centro de Proceso de Datos de la DGTIC se reserva la facultad de actualizar las versiones del software aquí reflejadas por otras superiores en el momento de la puesta en producción.

D. El hardware sobre el que se implantará la solución será el siguiente, haciendo especial hincapié en el hecho de que dicho hardware no será de utilización exclusiva, sino compartida con numerosas aplicaciones de las *consellerias*:

Producto	**Hardware**
Jboss 3.2.6 + Tomcat 4.1.18	Intel Xeon S.O. Red Hat Enterprise Linux 3 RAM: 2 GB Almacenamiento: 72 GB
Oracle 9.2.0.3	Intel Xeon

1 La versión de Jboss tiene que ser la proporcionada por la DGTIC y puede descargarse de http://dgtic.caib.es/estandards/index.html

	S.O. Red Hat Enterprise Linux 3 RAM: 2 GB Almacenamiento: 200 GB
Lotus Domino Domino.doc Domino.Workflow	Intel Xeon Windows 2000 server RAM: 2 GB Almaceenamiento: 500 GB

E. El producto final y las actualizaciones se entregarán según el formulario estándar de cuadernos de carga estandarizados por la DGTIC (ver apartado Entrega de aplicaciones para paso a producción).

F. El sistema deberá venir acompañado de los siguientes informes:
   - Estudio de consumos de cada módulo software: CPU, memoria, disco y ancho de banda de red.
   - Estudio de la concurrencia en el acceso a datos y módulos software: elementos críticos, bloqueos entre usuarios y situaciones de *dead-lock*.
   - Manual de procedimientos de operación y copias de seguridad
   - Manual de usuario

G. El sistema deberá cumplir las medidas de seguridad designadas en el R.D. 994/1999, de 11 de junio, por el que se aprueba el Reglamento de Medidas de Seguridad de los ficheros automatizados que contengan datos de carácter personal.

NOTA:	La versión de Jboss modificada por la DGTIC y las últimas versiones de los estándares están disponibles en http://dgtic.caib.es/estandards/index.html

Estándar de desarrollo de aplicaciones Java

Incluye el conjunto de normas a verificar por las aplicaciones del *Govern de les Illes Balears*.

El **procedimiento de puesta en producción** de nuevas aplicaciones es el siguiente:

1. Solicitud de un código de aplicación a suport@caib.es
2. Desarrollo de la aplicación
3. Solicitud de instalación de la aplicación en el entorno de pruebas de la DGTIC, enviando una petición a suport@caib.es
4. Validación y test por parte de los usuarios en el entorno de pruebas de la DGTIC
5. Solicitud de instalación de la aplicación en el entorno de producción, enviando una petición a suport@caib.es

El **índice de contenidos**, con un *abstract* para cada capítulo, es el siguiente:

1. Solicitud de código de aplicación:

   Será necesario solicitar un código antes de comenzar el desarrollo de una nueva aplicación. Es la base para la nomenclatura de todos los objetos de la aplicación.
2. Nomenclatura de objetos de base de datos:

   Incluye toda la normativa de los objetos del esquema de base de datos de la aplicación a desarrollar.
3. Directorios y nomenclatura de archivos de los fuentes de la aplicación:

   Localización de los fuentes en el servidor principal.
4. Nomenclatura de aplicaciones J2EE:

   Normativa para el desarrollo de la aplicación *web*.
5. Entrega de aplicaciones para el paso a producción:

   Qué hace falta entregar, a nivel de archivos ejecutables, fuentes, *scripts* DDL de creación de base de datos y documentación, tanto a la hora de la entrega inicial, como en cada entrega posterior para actualizaciones.

1. SOLICITUD DE CÓDIGO DE APLICACIÓN

Antes de comenzar el desarrollo de una nueva aplicación es necesario solicitar un código a la DGTIC (*Direcció General de Tecnologies i Comunicacions*). La petición se hará enviando un correo a la dirección suport@caib.es indicando **necesariamente**, la siguiente información:

- Petición de asignación de código de aplicación
- Nombre y descripción de la nueva aplicación
- Fecha de la comisión de informática
- *Conselleria* y dirección general
- Persona de contacto en la *conselleria* (nombre y teléfono)

Como respuesta, el personal de la DGTIC enviará un correo a la dirección remitente con la siguiente información:

- Código asignado a la aplicación
- Prefijo para los nombres de los objetos
- Usuario propietario de las tablas, usuario del *pool* de conexiones y base de datos de pruebas
- Versiones de software a utilizar
- Estándar de nomenclatura a seguir y plantilla del cuaderno de carga, para enviar las peticiones de instalación a suport@caib.es

2. NOMENCLATURA DE OBJETOS DE BASE DE DATOS

2.1. Consideraciones generales

El código de aplicación y su prefijo habrán sido facilitados previamente por el *Centre de Procés de Dades* de la DGTIC (ver apartado Solicitud de código de aplicación).

Todos los objetos de base de datos de una aplicación serán propiedad (tendrán como *owner*) de un mismo usuario de base de datos que deberá coincidir con el código de aplicación, previamente asignado por la DGTIC.

Invariablemente, todos estos objetos empezarán por un prefijo de tres letras, representativas de la aplicación, seguidas de un guión bajo (_).

Cada vez que se haga referencia a este prefijo, de ahora en adelante, utilizaremos como ejemplo el literal '**APL**_'.

2.2. Nomenclatura de tablas y vistas

Las tablas, después del prefijo, se identificarán con un nombre representativo de la entidad a la que corresponden de, como máximo, 6 caracteres: **APL_XXXXXX**

Ejemplos:

APL_CLIENT

APL_NOTA

Nota 1: evitar los nombres largos, combinación de los dos nombres de tabla origen, para las tablas resultantes de las relaciones N:M. Coger sólo las tres primeras letras de los dos nombres de tabla originales.

Nota 2: en general, los nombres de tabla compuestos tienen que formarse preferentemente con el formato **AABBCC** o **AAABBB** (mismo número de letras pegadas de cada palabra del nombre compuesto).

Ejemplo: tabla resultante de una relación N:M entre APL_CLIENT y APL_NOTA

Incorrecto:	APL_CLIENT_NOTA
Correcto:	APL_CLINOT

2.3. Nomenclatura de campos (columnas)

Los nombres de columna de cada tabla empezarán por las tres primeras letras del nombre de ésta como prefijo, seguidos del nombre correspondiente a la propia columna, que, como máximo, podrá ser de seis caracteres.

Ejemplos de columnas de la tabla APL_CLIENT:

CLI_CODI

CLI_NOM

CLI_ DOMICI

- *Campos especiales:*

  Se recomienda, para los nombres de columnas que correspondan al identificador de la tabla, que el nombre de la columna sea **CODI** (ejemplo: CLI_CODI), excepto en los casos que no se trate de un código generado, sino de un concepto particular (ejemplo: CLI_NIF).

  Para las columnas correspondientes a claves extranjeras, el nombre tendrá que ser representativo de la tabla y columna a la cual hacen referencia (ejemplo: CLI_CODNOT, como nombre de una columna de la tabla APL_CLIENT que hace referencia a la columna CÓDI de la tabla APL_NOTA).

2.4. Nomenclatura de secuencias

Seguirán al patrón **APL_SEQXXX**, dónde XXX son las tres primeras letras del nombre representativo de la tabla para la cual se crea la secuencia.

Ejemplo:

APL_SEQCLI para el contador del código de la tabla APL_CLIENT.

2.5. Nomenclatura de triggers

Seguirán al patrón **APL_XXX_YYYYYY**, dónde XXX son las tres primeras letras del nombre representativo de la tabla a la que se asocia el *trigger*, y YYYYYY es un nombre representativo del propio *trigger* de, como máximo, seis caracteres.

Ejemplo:

APL_CLI_ALTAP

2.6. Nomenclatura de constraints

- *Primary key*

  Seguirán el patrón **APL_XXX_PK**, dónde XXX son las tres primeras letras del nombre representativo de la tabla.

  Ejemplo:

  APL_CLI_PK

- *Foreing key*

  Seguirán al patrón **APL_XXXYYY_FK**, dónde XXX son las tres primeras letras del nombre representativo de la tabla origen y YYY las tres primeras letras del nombre representativo de la tabla referenciada.

  Ejemplo:

  APL_CLIILL_FK (clave extranjera de la tabla cliente, hacia una tabla APL_ILLA)

- Constraints *particulares*

  Seguirán al patrón **APL_XXXYYY_ZZZ**, dónde XXX son las tres primeras letras del nombre representativo de la tabla, YYY (opcional) un nombre que haga referencia a lo que hace la *constraint* y ZZZ un literal que se refiere al tipo de *constraint* de que se trata.

  Ejemplo:

  APL_CLI_UNI

  APL_ILLNOM_DOM

2.7. Nomenclatura de índices

Los índices siguen la misma nomenclatura que la *constraint* correspondiente, seguida del sufijo '**_I** '.

Ejemplos:

APL_CLI_PK_I

APL_CLIILL_FK_I

Nota: no se trata de una norma obligatoria, sino de obedecer la pauta que sigue el propio Oracle cuando genera los índices de forma automatizada. En los casos particulares, es suficiente que el nombre del índice sea representativo de su función, y verifique las normas de prefijo y nombres compuestos.

2.8. Nomenclatura de dominios, procedimientos, funciones, packages i roles

En estos casos, la nomenclatura es más libre, siempre que se siga la norma de empezar cada nombre por el prefijo de la aplicación, y que el nombre del objeto sea el más simple y representativo posible.

Ejemplos:

Dominios	APL_NOMILLA
Roles	APL_CONSULTA
	APL_INTRODUCCIO
	APL_ADMINISTRACIO
Packages	APL_GESTIOCLI

2.9. Normas referentes a las tablas comunes

Hay una serie de tablas especiales que utilizan las herramientas Oracle como *Designer/2000* o *Developer/2000*. Los ejemplos más claros son:

CG_REF_CODES

CG_FORM_HELP

CG_CODE_CONTROL

El criterio seguido en todas las bases de datos del *Govern de les Illes Balears* es el de mantener una sola versión pública de cada tabla - propiedad de *SYSTEM* - que contenga los datos de todas las aplicaciones. Para las aplicaciones desarrolladas que las utilicen, será necesario proporcionar las sentencias DML correspondientes (INSERT) con el fin de introducir los datos particulares de la aplicación en la tabla común correspondiente.

2.10. Normas referentes a sinónimos

La utilización del prefijo particular de la aplicación hace que cada nombre de objeto sea único dentro de la base de datos. Eso permite que todos los objetos de cada aplicación tengan asignados los correspondientes sinónimos públicos. Es necesario adjuntar los *scripts* de creación de estos sinónimos públicos para las tablas, vistas, secuencias, procedimientos, funciones y *packages* de la aplicación.

Ejemplo:

```
CREATE PUBLIC SYNONYM APL_CLIENT FOR NOMUSU.APL_CLIENT
```

2.11. Acceso a la base de datos. Normas referentes a los privilegios de acceso (GRANTS)

Para el acceso a base de datos, deberá definirse un *pool* de conexiones. El usuario del *pool* deberá seguir la nomenclatura WWW_xxxxxx, donde xxxxxx tiene que coincidir con el código de aplicación.

Ejemplo:

Dado el código de aplicación GESACO, el usuario del *pool* de conexiones deberá tener el nombre WWW_GESACO

Para que el usuario WWW_xxxxxx pueda utilizar los objetos del usuario propietario será necesario dar los privilegios de acceso (grants) adecuados.

Ejemplo:

Las sentencias GRANT relativas a la tabla APL_CLIENT de la aplicación GESACO podrían ser:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON APL_CLIENT TO WWW_GESACO;
```

2.12. Restricciones adicionales

No se permitirá la utilización de campos de tipo LONG.

El juego de caracteres de las bases de datos es UTF8 (NLS_CHARACTERSET=UTF8).

No se permitirá el uso de caracteres especiales en los nombres de los objetos (por ejemplo: Ñ, Ç, Á, À,Ä,...)

3. DIRECTORIOS Y NOMENCLATURA DE LOS FUENTES DE LA APLICACIÓN

En el servidor de aplicaciones de informática se guarda una copia (protegida) de todos los fuentes de cada aplicación.

Los fuentes se organizan en subdirectorios, uno por cada *conselleria* y, dentro de cada *conselleria*, un directorio por aplicación. Cada aplicación, si procede, deberá dividirse en subdirectorios que recojan los diferentes tipos de fuentes (de la aplicación J2EE, *scripts* SQL de generación de objetos Oracle y, en general, de cualquier otro tipo de fuente).

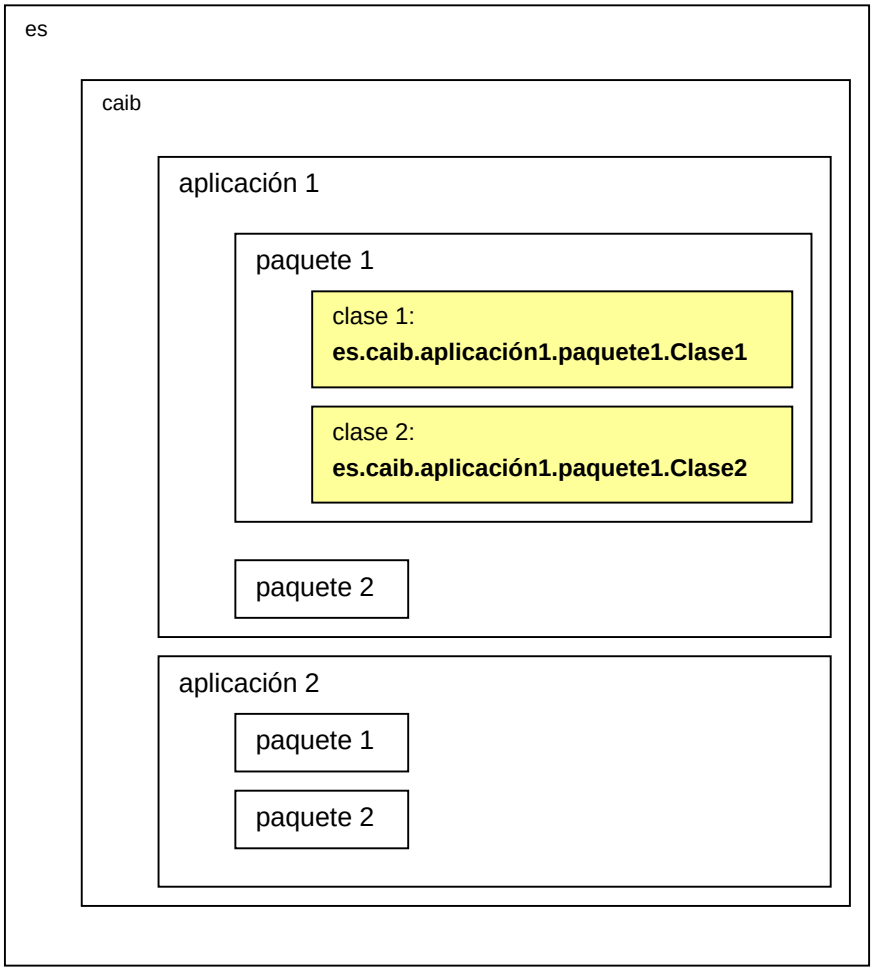
4. NOMENCLATURA DE APLICACIONES J2EE

4.1. Nomenclatura de clases

4.1.1. Jerarquía de paquetes

Las clases de objetos se estructurarán en aplicaciones y paquetes. Todas las aplicaciones y paquetes dependerán jerárquicamente del dominio de paquetes **es.caib.**

Así las clases se denominarán es.caib.*Aplicación.Paquete.Clase*



Los caracteres válidos serán aquellos definidos por el estándar Java: letras mayúsculas y minúsculas del alfabeto inglés y números en posición no inicial.

Los nombres de aplicación estarán siempre en minúsculas y deberán ser solicitados y autorizados por el *Centre de Procés de Dades* de la DGTIC (ver apartado Solicitud de código de aplicación).

Los nombres de paquete estarán siempre en minúsculas y podrán ser nombrados, dentro del paquete de aplicación, a criterio de analistas y diseñadores.

4.1.2. Nomenclatura de clases

Las clases se nombrarán con la primera letra mayúscula y el resto en minúsculas. Las clases formadas por varias palabras utilizarán mayúsculas para la inicial de cada una de ellas:

es.caib.aplicacion.paquete.Clase

es.caib.aplicacion.paquete.ClaseDeVariosVocablos

4.1.3. Nomenclatura de métodos

Los métodos se nombrarán con todas las letras minúsculas, incluida la inicial. Las clases formadas por varias palabras utilizarán mayúsculas para la inicial de las segundas palabras:

es.caib.aplicacion.paquete.Clase.metodo

es.caib.aplicacion.paquete.Clase.metodoDeVariosVocablos

4.2. Arquitectura de aplicaciones

4.2.1. Servicios de directorio del servidor de aplicaciones

El acceso al servicio de directorio (*NamingFactory*) se realizará siempre con los parámetros por defecto, asumiendo que las propiedades JNDI están correctamente configuradas.

Los servicios de directorio del servidor transaccional identificarán cada *Enterprise Java Bean* mediante su nombre jerárquico completo, debiendo acceder las clases Java a él mediante dicho nombre.

El acceso a otro tipo de servicios, tales como conexiones a base de datos o *pools* de conexiones se realizará a través de nombres jerárquicos dependientes de la jerarquía de la aplicación.

Ejemplo:

es.caib.aplicación.db.presid

es.caib.aplicacion.db.BaseDeDatos

es.caib.aplicacion.db.PoolDeConexiones

4.2.2. Acceso a bases de datos

- El acceso a bases de datos se realizará a través de los objetos RMI recuperados del servicio de directorio. Dicho acceso se realizará a través de la jerarquía es.caib.codigoaplicacion.db, donde c*odigoaplicacion* indica el código asignado por la DGTIC a la aplicación.

  Dentro de esta jerarquía se encontrará un objeto para cada conexión a base de datos definida en la aplicación. La base de datos seguirá los criterios de nomenclatura de las clases de objeto:

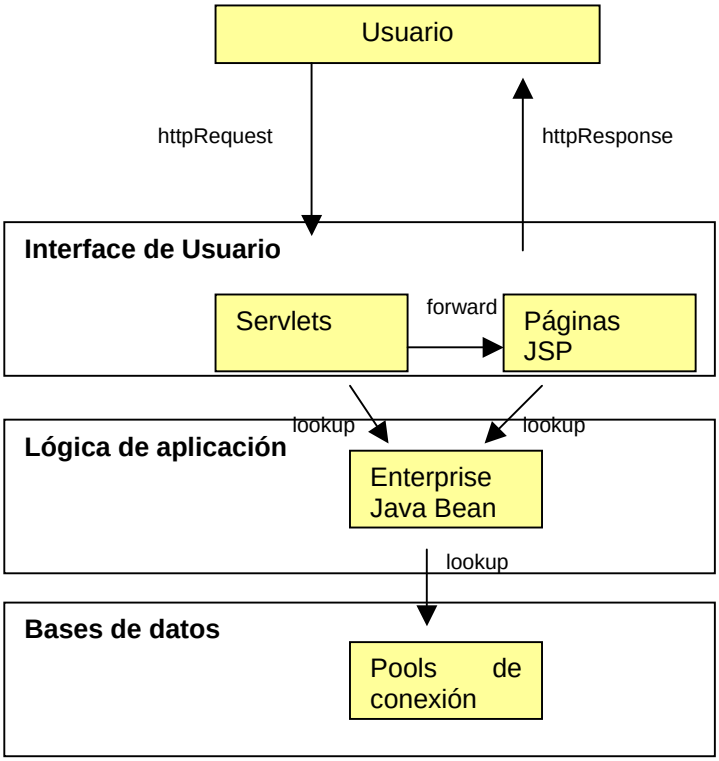
  es.caib.aplicacion.db.Presidencia

  es.caib.aplicacion.db.Defecto

  es.caib.aplicacion.db.RecursosHumanos
- El usuario del *pool* de conexiones deberá seguir la nomenclatura WWW_xxxxxx, donde xxxxxx tiene que coincidir con el código de aplicación.
- El acceso a la base de datos debe hacerse utilizando cliente *thin*, no OCI.

4.2.3. Módulos JSP, Servlets y Enterprise Java Beans.

La arquitectura de la aplicación deberá ser la siguiente, si bien se admiten ligeras variantes:

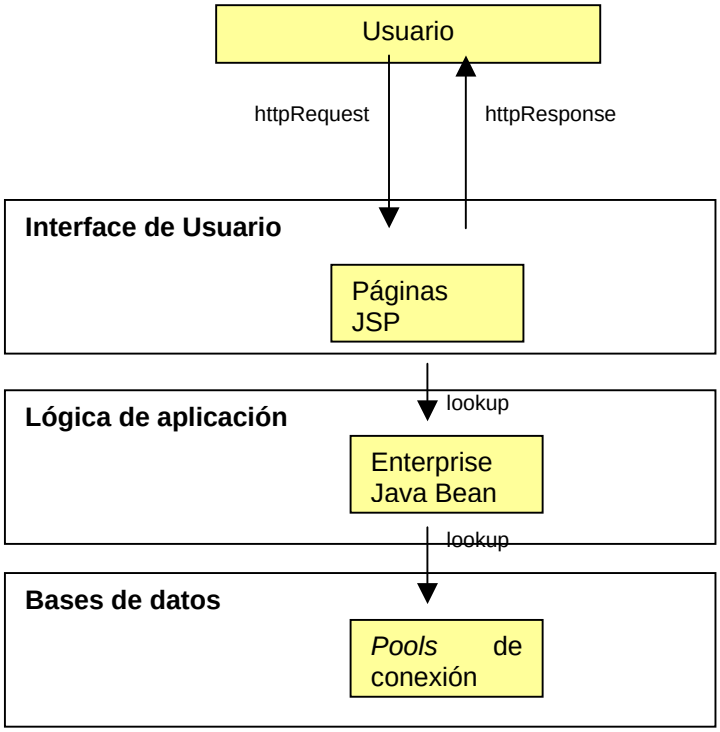


Normalmente, la petición del usuario será recogida por un *servlet*, el cual localizará el *Enterprise Java Bean* adecuado a través del método *lookup* del servicio de directorio y le solicitará las acciones pertinentes. Es muy importante remarcar que bajo ninguna circunstancia ni el *servlet* ni las páginas JSP deberán acceder de forma directa a los *pools* de conexión a la base de datos. Toda operación contra bases de datos deberá ser canalizada a través de los *EJBs*.

Dicho EJB realizará las operaciones necesarias a través de los *pools* de conexión a la base de datos y devolverá información al s*ervlet*.

Este *servlet* analizará la respuesta y la redirigirá a la página JSP correspondiente, la cual realizará las funciones de representación del formulario adecuado a mostrar.

En condiciones excepcionales, cuando la lógica del proceso a generar sea prácticamente nula y no se tenga que mostrar una página u otra en función de los datos introducidos, se permitirá que el usuario envíe la petición http directamente a una página JSP. En este caso el diagrama es el siguiente:



4.3. Seguridad de aplicaciones

Todos los aspectos relativos a identificación y autorización de los usuarios a *servlets*, *JSPs* o *EJBs* serán gestionados de forma externa a las aplicaciones, desde el entorno de administración de la plataforma J2EE, por lo que no se debe codificar dentro de *servlets*, *JSPs* o *EJBs* ninguna regla o criterio de autenticación.

Sí pueden estar codificados dentro de la aplicación aspectos relativos a cómo se presenta el *interface* de usuario.

En caso de que la aplicación J2EE requiera restringir el acceso a los recursos mediante un usuario y *password* deberán configurarse los siguientes elementos:

<security-constraint>
<login-config>
<security-role>

4.3.1. Elemento <login-config>

El método utilizado para autentificar el usuario deberá ser **BASIC** (utilizar la autenticación del *browser*) y el nombre de *realm* especificado **Govern de les Illes Balears**. No deberá utilizarse el tag <form_login_config>.

Ejemplo:

```
<login-config>
   <auth-method>BASIC</auth-method>
   <realm-name>Govern de les Illes Balears</realm-name>
 </login-config>
```

4.3.2. Elemento <security-role>

En el fichero **web.xml** (y ejb-jar.xml) se deberán definir uno o varios roles para la aplicación, con sus respectivas descripciones.

Ejemplo:

```
<security-role>
   <description> … descripción …</description>
   <role-name>APL_ XXXXXX</role-name>
  </security-role>
```

Para poder integrar la seguridad definida a nivel de aplicación con el sistema de seguridad de la CAIB será necesario que los nombres de roles definidos en el fichero web.xml estén estandarizados según las normas de la DGTIC.

Para el caso de una aplicación con prefijo APL_ el nombre especificado con el tag <role-name> debe ser APL_XXXXXX, donde XXXXXX debe ser un nombre lo más simple y representativo posible.

Ejemplos de nombres de roles:

APL_CONSULTA
APL_INTRODUCCIO
APL_ADMINISTRACIO

4.3.3. Elemento <security-constraint>

Se deberá utilizar en caso de tener que definir privilegios de acceso para una colección de recursos. Deberán especificarse los roles que tendrán acceso a los recursos protegidos.

4.3.4. Protección de EJBs

Es necesario proteger los *EJBs* de manera que ningún usuario anónimo pueda ejecutarlos, salvo que los *EJBs* deban ser públicos. Para protegerlos hay que poner *security constraints* a los *EJBs* con el tag method-permission el fichero ejb-jar.xml

Ejemplo:

```
<security-role>
<role-name>nombre_de_rol</role-name>
</security-role>
   <method-permission>
    <role-name>nombre_de_rol </role-name>
    <method>
     <ejb-name>nombre_de_EJB</ejb-name>
     <method-name>método</method-name>
     <method-params>
      <method-param>parámetro1</method-param>
      <method-param>parámetro2</method-param>
     </method-params>
</method>
```

4.3.5. Declaración de dominios de seguridad en JBoss (Elemento <security-domain>)

El acceso a los recursos protegidos deberá hacerse dentro del siguiente dominio de seguridad (*Security Domain*):

java:/jaas/seycon

4.4. Nombres de aplicación

Para evitar problemas de coincidencias de nombres a la hora de desplegar las aplicaciones en el servidor J2EE, los nombres de aplicación (fichero *.ear) y de aplicación web (fichero *.war) deberán definirse de la siguiente forma:

- El nombre del fichero ***.ear** deberá coincidir con el nombre (**código**) de aplicación proporcionado por la DGTIC.

Ejemplo:

Si el código de aplicación es GESACO, el nombre del fichero *.ear deberá ser gesaco.ear

- Para la nomenclatura de los ficheros ***.war** se considerarán dos posibilidades:
  - Si la aplicación tiene un único fichero *.war, éste deberá tener el mismo nombre que el fichero *.ear
  - Si la aplicación tiene varios ficheros *.war, los nombres de estos deberán estar precedidos por los tres caracteres de **prefijo** de aplicación seguidos de _.

  Ejemplo:

  Si el prefijo de la aplicación GESACO es ACO, los nombres de ficheros *.war deberán ser aco_xxxxxx, donde xxxxxx será un nombre lo más simple y representativo posible.

El código de aplicación y su prefijo habrán sido facilitados previamente por el Centro de Proceso de Datos de la DGTIC.

4.5. Nombres de EJBs

Para la nomenclatura de los ficheros ***.jar** se considerarán dos posibilidades:

- Si la aplicación tiene un único fichero *.jar, éste deberá tener el mismo nombre que el fichero *.ear
- Si la aplicación tiene varios ficheros *.jar, los nombres de estos deberán ser:

  es_caib_nombreaplicación_nombrejar.jar

  donde nombreaplicación deberá coincidir con el código de aplicación facilitado por el *Centre de Procés de Dades* de la DGTIC (ver apartado Solicitud de código de aplicación).

Ejemplos:

Dada una aplicación con código GESACO, prefijo ACO y varios ficheros *.jar, los nombres de ficheros *.jar deberán ser es_caib_gesaco_xxxxxx, donde xxxxxx será un nombre lo más simple y representativo posible.

Dada una aplicación con código GESACO, prefijo ACO y un único fichero *.jar, el nombre del fichero deberá ser gesaco.jar.

4.6. Context root

- En caso de tener un único *context root*, éste deberá coincidir con el código de la aplicación.
- Si la aplicación tiene un *frontoffice* (público) y un *backoffice* (privado), el ear deberá contener dos war, y la nomenclatura del *context root* será:
  - Nombre de la aplicación seguido de la palabra *front* ('nombreaplicación'front) para el *context root* del *frontoffice.*
  - Nombre de la aplicación seguido de la palabra *back* ('nombreaplicación'back) para el *context root* del *backoffice.*

4.7. Restricciones adicionales

- Las aplicaciones deberán utilizar el juego de caracteres UTF-8:

  <?xml version=... encoding="**UTF-8**" ?>
- NO se permitirá la utilización de librerías de terceros - *struts, lucene* u otros -, salvo las incluidas en la versión de Jboss modificada por la DGTIC, disponible en http://dgtic.caib.es/estandards/index.html.
- NO deberán utilizarse e*ntity beans*.
- Los *beans* serán preferentemente *stateless session beans*.
- Los *stateful session beans* deberán implementar adecuadamente los métodos *activate* y *passivate* al efecto de minimizar el consumo de memoria y recursos.
- Todo acceso a un recurso localizable via JNDI debe estar referenciado de forma relativa a **java:comp**

1. Ejemplo **WebLogic**:

   Contenido del fichero **weblogic.xml**:

```
<weblogic-web-app>
  <reference-descriptor>
    <ejb-reference-description>
      <ejb-ref-name>ejb/PuntEntradaEJB</ejb-ref-name>
      <jndi-name>es.caib.seycon.ejb.PuntEntradaEJB</jndi-name>
    </ejb-reference-description>
  </reference-descriptor>
</weblogic-web-app>
```

   El fichero **web.xml** deberá contener:

```
<ejb-ref>
    <ejb-ref-name>ejb/PuntEntradaEJB</ejb-ref-name>
    <ejb-ref-type>Session</ejb-ref-type>
    <home>es.caib.seycon.ejb.PuntEntradaEJBHome</home>
    <remote>es.caib.seycon.ebj.PuntEntradaEJB</remote>
  </ejb-ref>
```

   Dentro de la **clase Java**, el lookup del EJB deberá hacerse de la siguiente forma:
   new javax.naming.InitialContext ().lookup ("java:comp/ejb/PuntEntradaEJB");

2. Ejemplo **JBoss**:

   Contenido del fichero **web.xml**:

```
<web-app>
  <servlet>
    <servlet-name>AServlet</servlet-name>
    <servlet-class>AServlet</servlet-class>
  </servlet>
  <!-- JDBC DataSources (java:comp/env/jdbc) -->
  <resource-ref>
    <description>The default DS</description>
    <res-ref-name>jdbc/DefaultDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
  <!-- JavaMail Connection Factories (java:comp/env/mail) -->
  <resource-ref>
    <description>Default Mail</description>
    <res-ref-name>mail/DefaultMail</res-ref-name>
    <res-type>javax.mail.Session</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
  <!-- JMS Connection Factories (java:comp/env/jms) -->
  <resource-ref>
```

```
        <description>Default QueueFactory</description>
        <res-ref-name>jms/QueFactory</res-ref-name>
        <res-type>javax.jms.QueueConnectionFactory</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>
</web-app>
```

Contenido del fichero **jboss-web.xml**:

```
<jboss-web>
    <resource-ref>
        <res-ref-name>jdbc/DefaultDS</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <jndi-name>java:/DefaultDS</jndi-name>
    </resource-ref>
    <resource-ref>
        <res-ref-name>mail/DefaultMail</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <jndi-name>java:/Mail</jndi-name>
    </resource-ref>
    <resource-ref>
        <res-ref-name>jms/QueFactory</res-ref-name>
        <res-type>javax.jms.QueueConnectionFactory</res-type>
        <jndi-name>QueueConnectionFactory</jndi-name>
    </resource-ref>
</jboss-web>
```

NOTA:	Utilizar funcionalidad estándar J2EE, prescindiendo totalmente de características no incluidas en el estándar.

5. ENTREGA DE APLICACIONES PARA PASO A PRODUCCIÓN

Los elementos que hay que entregar para la instalación de una aplicación en preproducción o producción son los siguientes:

5.1. Scripts de generación de los objetos de base de datos Oracle (DDLs)

Tienen que contener todas las sentencias DDL necesarias para crear el esquema completo de base de datos correspondiente a la aplicación. Se comprobará que verifiquen las normas de nomenclatura y seguridad que se especifican en este documento.

Las sentencias DDL deben clasificarse en diferentes archivos según el siguiente criterio:

- Creación de tablas: Sólo contendrá las sentencias de creación de tablas. El archivo tendrá extensión .tab.
- Creación de *constraints*: Sólo contendrá las sentencias de creación de *constraints*. El archivo tendrá extensión .con.
- Creación de índices: Sólo contendrá las sentencias de creación de índices. El archivo tendrá extensión .ind.
- Creación de secuencias: Sólo contendrá las sentencias de creación de secuencias. El archivo tendrá extensión .seq.
- Creación de *triggers*: Sólo contendrá las sentencias de creación de *triggers*. El archivo tendrá extensión .trg.
- Creación de procedimientos y funciones: Sólo contendrá las sentencias de creación de procedimientos y funciones. El archivo tendrá extensión .pro.
- Creación de paquetes: Sólo contendrá las sentencias de creación de paquetes. El archivo tendrá extensión .pck.
- Otras: El archivo tendrá extensión .sql.

NOTA:
Tiene que incluirse una estimación del tamaño necesario del *tablespace* o *tablespaces* requeridos por la aplicación.

5.2. Creación de sinónimos

Creación de los sinónimos públicos de todos los objetos de la aplicación.

5.3. Asignación de permisos (GRANTS) al usuario del pool de conexiones

Adjuntar los *scripts* de asignación de permisos sobre los objetos del propietario de la aplicación al usuario del *pool* de conexiones, WWW_xxxxxx.

5.4. Aplicación J2EE (fichero ear)

Adjuntar el fichero ear que contiene la aplicación.

5.5. Roles de la aplicación J2EE

En caso de que la aplicación esté protegida con uno o varios roles, deberá indicarse su creación, incluyendo el nombre de cada uno de los roles y su descripción.

5.6. Fuentes de la aplicación

Se deberán entregar siempre los archivos fuente de la aplicación. NO se pasará a producción ningún programa que no adjunte los archivos fuente actualizados.

5.7. Información adicional

Las instrucciones de instalación deben incluir toda la información necesaria para la correcta configuración de la aplicación sobre el servidor de aplicaciones.

5.8. Documentación

Deberán entregarse, siempre, como mínimo:

- Manual de instalación, operación y mantenimiento.
- Manual de usuario.
- Información técnica referente a los datos de carácter personal usados en la aplicación.

5.9. Actualizaciones de aplicaciones en producción

A menudo, durante un cierto periodo de adaptación, las aplicaciones que ya se han pasado a producción van sufriendo modificaciones, tanto en los ejecutables como a nivel de base de datos.

Tendrán que entregarse:

- Instrucciones precisas del procedimiento de actualización.
- Los nuevos ejecutables (fichero ear), juntamente con los fuentes actualizados.
- Si hay modificaciones en la base de datos (cambios de estructura de una tabla, nuevas tablas, nuevos procedimientos, etc.), tendrán que incluirse, siempre que sea necesario:
  - Todas las sentencias de creación/borrado de sinónimos que puedan afectar a los cambios a realizar.
  - Sentencias de *grants* de permisos a los *roles*/usuarios de la aplicación para la correcta utilización de los nuevos objetos.
  - Cualquier otro tipo de sentencia de mantenimiento necesaria para mantener el entorno de operación en un estado completamente válido.

Este tipo de sentencias correspondientes a los elementos de seguridad como sinónimos y permisos no suelen tenerse en cuenta, pero son indispensables para el correcto funcionamiento de la aplicación. El personal de sistemas que mantiene la aplicación no tiene por qué tener que conocer su estructura ni sus elementos, sino encargarse simplemente de que se ejecuten con corrección los procedimientos de instalación/actualización entregados por el equipo encargado del desarrollo. Es pues responsabilidad del desarrollador incluir los procedimientos para mantener este tipo de elementos del entorno de seguridad.

Estándar de desarrollo de Aplicaciones Domino

1. NOMENCLATURA DE OBJETOS

Las bases de datos se estructuran en la siguiente jerarquía de directorios:

- Servidor
  - Conselleria
    - Dirección General
      - Base de Datos

Respecto a las bibliotecas de Lotus Domino, cada aplicación podrá tener sus propias bibliotecas, estableciéndose la siguiente relación:

- Biblioteca => Aplicación o Conselleria
- Salas de archivo => Diferentes departamentos o ubicaciones
- Portafolios => Expediente
- Documentos => Documentos del expediente

Los nombres de los objetos seguirán la siguiente norma, con independencia de los ALIAS que se muestren al usuario:

Tipo de objeto	Nomenclatura
Formulario	fmXXXXXX máximo de 8 letras, comenzando en mayúscula y siguiendo en minúsuculas: frmExpedi
Campo	flXXXXXX máximo de 8 letras, comenzando en mayúscula y siguiendo en minúsculas: fldNif, fldTelefo
Vista	vwXXXXXX máximo de 8 letras, comenzando en mayúscula y siguiendo en minúsculas
Marco	frXXXXXX máximo de 8 letras, comenzando en mayúscula y siguiendo en minúsculas
Agente	agXXXXXX máximo de 8 letras, comenzando en mayúscula y siguiendo en minúsculas
Funciones Java y LotusScript	Los métodos se nombrarán con todas las letras minúsculas, incluida la inicial. Las clases formadas por varias palabras utilizarán mayúsculas para la inicial de las segundas palabras

Desde el punto de vista de separación de datos y aplicaciones, el desarrollador entregará la aplicación en forma de plantilla, independientemente de las bases de datos necesarias derivadas de esta.

2. ARQUITECTURA DE APLICACIONES

2.1. Aspectos de seguridad

Todos los accesos deben ser controlados mediante el uso de Listas de Control de Acceso (ACLs) y definición de roles. No se introduciran usuarios en las ACL's, sólo grupos de usuarios. Estos grupos deberán especificarse en el cuaderno de carga, para que se creen en el names. La nomenclatura de los grupos debe ser: xxx_nombregrupo, dónde xxx es el prefijo que se habrá especificado desde la DGTIC.

En aquellos casos en que se considere necesario se implementará la firma digital de los documentos, sin que esta sea necesaria con carácter general.

2.2. Acceso mediante navegador web y cliente notes

Las aplicaciones deben ser perfectamente funcionales independientemente del cliente utilizado. Se admitirán ciertas restricciones en la implementación web, tales como la firma digital.

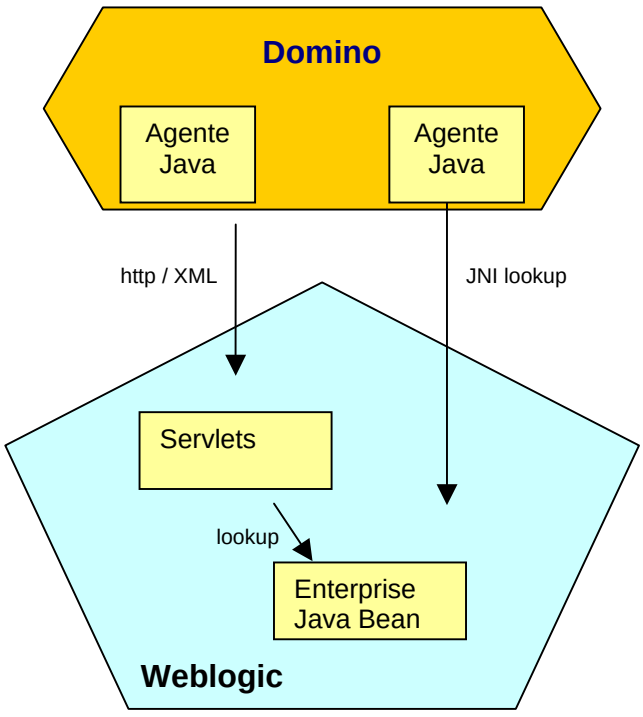
2.3. Acceso transaccional

Para aquellas funciones que requieran una explotación o concurrencia transaccional se prevé la utilización de transacciones J2EE.

Se debe utilizar este mecanismo en procesos que involucren integración con otras aplicaciones o plataformas, o requerimientos técnicos no cubiertos por Domino, tales como la generación de contadores o claves únicas.

El mecanismo de integración de Domino con la plataforma J2EE se podrá realizar de dos formas:

1. Integración directa J2EE. Desde Domino, un agente desarrollado en Java instanciará y utilizará uno o varios *Enterprise Java Beans*, encargados de implementar la lógica transaccional.
2. Integración http/XML. Desde Domino, un agente desarrollado en Java realizará una petición http a un *servlet* que hará uso de los *EJBs* correspondiente. El encapsulado de los datos a transmitir entre Domino y la plataforma J2EE se realizará mediante documentos XML.



2.4. Acceso desde transacciones EJB

Para aquellas funciones que requieran un acceso o actualización de bases de datos Domino desde otras plataformas, se prevé la utilización de transacciones J2EE desarrolladas sobre plataforma J2EE.

El mecanismo deberá consistir en el establecimiento de conexiones IIOP entre el servidor de aplicaciones J2EE y Domino. En ningún caso, el usuario y contraseña utilizados deberán estar codificados en el programa *EJB*. Preferiblemente se utilizarán el mismo usuario y contraseñas reconocidos por el contenedor de *EJBs*, habida cuenta de que ambos están sincronizados.

Procedimiento de puesta en producción

1. INTRODUCCION

Este documento detalla las normas que debe cumplir cualquier petición de modificación o puesta en producción de aplicaciones nuevas.

2. PROCEDIMIENTO DE ENVIO

- Todas las solicitudes de paso a producción tendrán que ser enviadas a la dirección de correo suport@caib.es y deberán tener anexados los siguientes documentos:
  - Cuaderno de carga (en formato Microsoft Word 97)
  - Fichero en formato ZIP conteniendo TODOS los ficheros necesarios para realizar la instalación:
    - *Scripts* a ejecutar
    - Fichero ear
    - …
- Dicho cuaderno de carga debe tener el nombre siguiente:
  - IN*aammdd*.doc

  donde 'aa' es el año, 'mm' es el mes y 'dd' es el día de envío de la petición a la dirección de correo suport@caib.es.

- El correo que se enviará a suport@caib.es deberá indicar como **subject** o asunto:
  - *'Código_de_aplicación'*– IN*aammdd*

  donde IN*aammdd* es el nombre del cuaderno de carga y *'Código_de_aplicación'* el código asignado por la DGTIC .

La plantilla del cuaderno de carga se detalla en el apartado Ejemplo cuaderno de carga de aplicación J2EE.

3. NORMAS

- Las solicitudes de instalación tiene que cumplir todos los requerimientos especificados en el apartado Entrega de aplicaciones para paso a producción.
- Cualquier petición que no se realice a través de la cuenta de correo suport@caib.es y en los términos establecidos en este documento **NO será tenida en cuenta**.
- Los pasos a producción se harán solamente en la ventana horaria asignada a la aplicación[2]. Las solicitudes se podrán hacer en cualquier momento, pero no se las considerará hasta el día asignado.
- Si, de forma excepcional, se tiene que hacer un paso a producción fuera del horario asignado, la orden deberá tener el mismo formato que las órdenes semanales, requiriendo autorización expresa del usuario responsable de la aplicación.

[2] Consultar el día asignado con el personal de la DGTIC

- En caso de modificaciones de la base de datos (cambios en la estructura de una tabla, tablas nuevas, nuevos procedimientos, etc.) se tendrán que incluir, siempre que sea necesario:
    - Todas las sentencias de creación/borrado de sinónimos que puedan afectar a los cambios a realizar.
    - Sentencias de *grants* de permisos a los *roles* de la aplicación para la correcta utilización de los nuevos objetos.
    - Cualquier otro tipo de sentencia de mantenimiento necesaria para mantener el entorno de operación en un estado completamente válido.
- En caso de puesta en producción de aplicaciones nuevas se deberá hacer una petición PREVIA a la petición de puesta en producción definitiva, a la dirección de correo suport@caib.es indicando:
    - Petición de asignación de código de aplicación
    - Nombre y descripción de la nueva aplicación
    - *Conselleria* y dirección general
    - Persona de contacto en la *conselleria* (nombre y teléfono)

4. CUMPLIMENTACION DEL CUADERNO DE CARGA

A continuación se describe brevemente el modo de cumplimentar los campos del formulario:

4.1. Nº

Numeración de los pasos de la instalación en orden ascendente. Se debe incluir un paso para cada una de las tareas a realizar.

4.2. APLICACION

Campo en el que se indica el CODIGO de la aplicación a la que corresponde el objeto que se va a instalar o sustituir. Dicho código coincide con el asignado por la DGTIC antes de comenzar el desarrollo.

4.3. OBJETO

Nombre del objeto a instalar (script SQL con las sentencias de creación de objetos, fichero ear a desplegar, ...).

4.4. UBICACION

Este campo sólo deberá rellenarse en caso de que se trate de ubicar ficheros ejecutables o *scripts* SQL en producción. Se indicará::

- El directorio relativo al correspondiente a la aplicación, en el que se tiene que ubicar el objeto.
- Permisos especiales a asignar al archivo o directorio (en caso de que sea necesario).

4.5. MODIFICACION

Persona	Se indicará el USUARIO responsable de la solicitud en cuestión. Tanto se podrá poner el número de usuario como una abreviatura del nombre de la persona.
Fecha	Se indicará la fecha en que se encontraba LISTO PARA INSTALAR el objeto en cuestión.

4.6. OBSERVACIONES A LA INSTALACION

Un pequeño comentario de cómo se tiene que instalar el objeto en cuestión.

MUY IMPORTANTE:

Si la solicitud consiste en lanzar un *script* sobre una Base de Datos, se tendrá que indicar:

- la base de datos sobre la que se tiene ejecutar
- el usuario con el que se tiene que ejecutar el *script*

4.7. INSTALACION

Este campo no debe ser rellenado. Los datos necesarios serán introducidos por la persona que realice la instalación en producción.

ANEXOS

1. EJEMPLO CUADERNO DE CARGA DE APLICACIÓN J2EE

Cuaderno de carga

SEMANA del 9 al 13 de agosto de 2004

Puesta en producción de la aplicación REGENTBS (*Registre d'Entitats de Benestar Social*).

Nº	APLICACION	OBJETO	UBICACION	ORDEN DE INSTALACION		OBSERVACIONES A LA INSTALACION	EJECUCIÓN INSTALACION	
				Persona	Fecha		Persona	Fecha
1	REGENTBS	Objetos.sql		P. González	09/08/04	Scripts de creación de objetos de base de datos. Ejecutar con el usuario REGENTBS en la base de datos PROD2.		
2	REGENTBS	Sinonimos.sql		P. González	09/08/04	Scripts de creación de sinónimos públicos. Ejecutar con el usuario REGENTBS en la base de datos PROD2.		
3	REGENTBS	Grants.sql		P. González	09/08/04	Asignación de permisos al usuario WWW_REGENTBS. Ejecutar con el usuario REGENTBS en la base de datos PROD2.		

4	REGENTBS	regentbs.ear	Servidor de aplicaciones	P. González	09/08/04	Desplegar en el servidor de aplicaciones.		
5	REGENTBS	FuentesRegentbs .zip	Directorio de fuentes	P. González	09/08/04	Fuentes de la aplicación. Copiar en el directorio de fuentes.		
6	REGENTBS	REB_COINIF		P. González	09/08/04	Indicar una descripción del role. Crear role.		
7	REGENTBS	REB_COINMA		P. González	09/08/04	Indicar una descripción del role. Crear role		
8	REGENTBS	REB_COINME		P. González	09/08/04	Indicar una descripción del role. Crear role		
9	REGENTBS	REB_GOVERN		P. González	09/08/04	Indicar una descripción del role. Crear role		