

Estándares  
de desarrollo de aplicaciones del GOIB  
**Base de datos**



G CONSELLERIA  
O ADMINISTRACIONS  
I PÚBLIQUES I  
B MODERNITZACIÓ  
/ DIRECCIÓ GENERAL  
MODERNITZACIÓ I  
ADMINISTRACIÓ DIGITAL

Palma, diciembre de 2020

## Índice

<b>HISTORIAL DE VERSIONES.....</b>	<b>3</b>
<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
<b>2. BASE DE DATOS Y VERSIONES SOPORTADAS.....</b>	<b>5</b>
<b>3. NOMENCLATURA.....</b>	<b>6</b>
3.1. Aplicaciones.....	6
3.2. Objetos de bases de datos.....	6
3.3. Tablespaces.....	6
3.4. Tablas y vistas.....	7
3.5. Secuencias.....	8
3.6. Restricciones (constraints).....	8
3.7. Otras restricciones.....	8
3.8. Índices.....	9
3.9. Roles.....	9
3.10. Procedimientos, funciones, paquetes y resto de objetos.....	9
3.11. Sinónimos.....	10
<b>4. RESTRICCIONES Y RECOMENDACIONES.....</b>	<b>11</b>
4.1. Accesos a las bases de datos.....	11
4.2. Tipos de datos.....	11
4.3. Codificación.....	11
4.4. Deadlocks.....	11
4.5. Particionamiento de tablas.....	11
<b>5. REPRESENTACIÓN DE LA LÓGICA DE NEGOCIO.....</b>	<b>12</b>
5.1. Database Links.....	12
5.2. Jobs.....	12
5.3. Sinónimos.....	12
5.4. Disparadores (Triggers).....	12
<b>6. PROBLEMAS FRECUENTES.....</b>	<b>13</b>
6.1. Nomenclatura de objetos.....	13
6.2. Objetos de tablespaces incorrectos.....	13
6.3. Uso de campos byte en lugar de char.....	13
6.4. Asignación de privilegios incorrectos.....	13
6.5. Objetos con código dentro del script SQL.....	14
<b>7. ENVÍO DEL CUADERNO DE CARGA Y EJECUCIÓN DE SCRIPTS.....</b>	<b>15</b>

## Historial de versiones

Fecha	Versión	Descripción	Autor
05/04/16	7.0	Revisión de estándares	DGDT
28/05/19	7.5	Revisión de estándares	DGMAD
17/01/20	9.0	Cambio formato documento Revisión de estándares	DGMAD
03/12/20	9.5	Aclaraciones	DGMAD

## 1. INTRODUCCIÓN

Este documento describe el estándar de base de datos que se debe seguir para el desarrollo de aplicaciones que se instalarán en los servidores de la *Dirección General de Modernización y Administración Digital* (DGMAD).

Las principales diferencias respecto de la versión anterior del documento son las siguientes:

1. Actualización de las bases de datos soportadas.
2. Apartado Representación de la lógica de negocio.
3. Apartado Problemas Frecuentes, inclusión codificación ficheros bases de datos.

Para más información sobre el procedimiento a seguir para enviar peticiones (CAIS), consultar el documento "Estándar de implantación de aplicaciones".

### **Nota:**

**Este documento es una traducción del documento original en catalán. En caso de conflicto prevalecerá la versión catalana del mismo.**

## 2. Base de datos y versiones soportadas

Las aplicaciones deberán estar preparadas para funcionar sobre una base de datos **Oracle 12.2**, concretamente sobre la versión 19c (release 12.2.0.3) que será la utilizada para todas las nuevas aplicaciones.

Mejoras de la versión:

- Soporte a largo plazo por parte del proveedor (hasta 2026).
- Actualización directa desde las versiones 11<sup>g</sup> (release 2), 12<sup>c</sup> y 18<sup>c</sup>.
- Indexación automática.
- Redireccionamiento DML de Active Data Guard.
- Tablas con particiones híbridas.
- Cuarentena de consultas.
- Compatibilidad con JSON.

En cuanto a las aplicaciones ya existentes, desde la DGMAD se está trabajando junto con las diferentes consejerías en la migración a versiones superiores puesto que, algunas de ellas, como las versiones 9i o la 10g, dejaron de tener soporte con el consiguiente riesgo de seguridad y estabilidad que ello conlleva.

A continuación se muestra una tabla de drivers JDBC y versiones JDK que deberá ser tenida en cuenta para las migraciones a versiones superiores.

BD Oracle	Versiones JDBC y JDK
19c	ojdbc10.jar con JDK10, JDK11 ojdbc8.jar con JDK8, JDK9 y JDK11
18c	ojdbc8.jar con JDK8, JDK9, JDK10 y JDK11
12c Release 2 (12.2)	ojdbc8.jar con JDK8
12c Release 1 (12.1)	ojdbc7.jar con JDK7 y JDK8 ojdbc6.jar con JDK6
11g Release 2 (11.2)	ojdbc6.jar con JDK6, JDK7 y JDK8 (Nota: JDK7 y JDK8 sólo están soportadas en 11.2.0.3 y 11.2.0.4) ojdbc5.jar con JDK5

La necesidad de utilización de otras bases de datos (como PostgreSQL) deberá venir motivada y ser aprobada por parte del responsable del Área de Administración de Bases de Datos debido a las limitaciones de soporte y alta disponibilidad.

## 3. Nomenclatura

### 3.1. Aplicaciones

Todas las aplicaciones deben disponer de un código de aplicación y un prefijo que habrán sido facilitados previamente por el *Centre de Procés de Dades* de la DGMAD (ver documento “Estándares de implantación de aplicaciones”, Capítulo 2. “Solicitud de código de aplicación”).

Todos los objetos de base de datos de una aplicación serán propiedad de un mismo usuario de base de datos, que deberá coincidir con el código de aplicación previamente asignado por la DGMAD.

Asimismo, todos los objetos de base de datos empezarán por un prefijo de tres letras, también asignado por la DGMAD, seguidos de un guion bajo (\_).

En los ejemplos incluidos en este documento, cada vez que se haga referencia al código de aplicación se utilizará como ejemplo el literal ‘**APLICACION**’, y como prefijo el literal ‘**APL**’.

### 3.2. Objetos de bases de datos

El nombre de los objetos será como máximo de 30 caracteres, y solo pueden incluir los caracteres A-Z, a-z , 0-9 y guion bajo (\_).

La creación de los objetos no puede incluir comillas en la definición del nombre del objeto.

### 3.3. Tablespaces

En general, todos los objetos del usuario de base de datos se ubicaran en el tablespace de nombre igual que el usuario (tablespace por defecto del usuario).

Esto es válido para aplicaciones de tamaño reducido. Si la aplicación hace un uso intensivo de índices, es recomendable utilizar tablespaces separados para datos e índices. Si la aplicación contiene LOBs, por cuestiones de rendimiento y administración, es obligatorio almacenarlos en un tablespace separado del resto de objetos.

En el momento de solicitar el código de aplicación se tiene que indicar si la aplicación utiliza LOBs y si se quieren tener datos e índices separados.

En el caso de tener los objetos en dos o más tablespaces (datos, índices y LOBs) los nombres dependerán del tipo de aplicación, como se indica en la siguiente tabla.

Tipo de aplicación	Tablespace de datos	Tablespace de índices	Tablespace de LOB
Aplicación de tamaño reducido	APLICACION	APLICACION	No permitidos
Aplicación de tamaño reducido con LOB	APLICACION	APLICACION	APLICACION_LOB
Aplicación con datos e índices separados	APLICACION_DADES	APLICACION_INDEX	No permitidos
Aplicación con datos e índices separados y LOB	APLICACION_DADES	APLICACION_INDEX	APLICACION_LOB

Para separar los objetos en un mayor número de tablespaces se deberá consultar con el Área de Administración de Bases de Datos de la DGMAD.

### 3.4. Tablas y vistas

Seguirán el patrón **APL\_XXX**.

Donde XXX es un nombre representativo de la entidad a la que corresponde.

Ejemplos:

- APL\_CLIENT
- APL\_FACTURA

En las tablas resultantes de una relación N:M, el nombre de la tabla contendrá el nombre (o parte de él) de cada una de las tablas.

Ejemplo: tabla resultante de una relación N:M entre APL\_CLIENT y APL\_FACTURA:

- APL\_CLIENT\_FACTURA

#### 3.4.1. Tablas con campos LOB

Para crear tablas que tengan algún campo de tipo LOB (BLOB, CLOB o NCLOB) se seguirá el siguiente formato (en el ejemplo se crea la tabla APL\_TABLA):

```
CREATE TABLE APL_TABLA
(
    campo1 tipo_campo1,
    campo2 CLOB,
    campo3 BLOB)
TABLESPACE APLICACION_DADES
LOB (campo2) STORE AS APL_TABLA_campo2_LOB
(TABLESPACE APLICACION_LOB
INDEX APL_TABLA_campo2_LOB_I)
LOB (campo3) STORE AS APL_TABLA_campo3_LOB
(TABLESPACE APLICACION_LOB
INDEX APL_TABLA_campo3_LOB_I);
```

donde,

APL\_TABLA  
campo2

representa el nombre de la tabla  
representa un campo de tipo CLOB

campo3	representa un campo de tipo BLOB
APLICACION_DADES	representa el nombre del tablespace de DATOS
APLICACION_LOB	representa el nombre del tablespace de LOB

El nombre del objeto LOB (cláusula STORE AS) y de su índice asociado se forma añadiendo al nombre de la tabla el nombre del campo LOB, seguido del sufijo `_LOB` (o `_LOB_I` para el índice del LOB). Si el nombre resultante supera los 30 caracteres, se tiene que reducir el nombre del campo o de la tabla, pero siempre manteniendo el prefijo de la aplicación (APL\_) y el sufijo (`_LOB` o `_LOB_I`).

### 3.4.2. Columnas

Los nombres de columna de las tabla son libres, no es necesario que empiecen por las tres letras identificativas del nombre de la tabla. La única restricción es la limitación del tamaño máximo a 30 caracteres.

## 3.5. Secuencias

Seguirán al patrón **APL\_XXX\_SEQ**.

Donde XXX es un nombre representativo de la tabla o campo para la cual se crea la secuencia.

Ejemplo:

- APL\_CLIENT\_SEQ: para la secuencia del código de la tabla APL\_CLIENT.

## 3.6. Restricciones (constraints)

### 3.6.1. Clave Primaria

Seguirán el patrón **APL\_XXX\_PK**

Donde XXX indica el nombre de la tabla para la cual se crea la clave primaria.

Ejemplo:

- APL\_CLIENT\_PK

### 3.6.2. Clave Extranjera

Seguirán al patrón **APL\_XXX\_YYY\_FK**

Donde XXX indica el nombre de la tabla de origen. YYY indica el nombre de la tabla referenciada.

Ejemplo: clave extranjera de la tabla APL\_CLIENT hacia la tabla APL\_ILLA

- APL\_CLIENT\_ILLA\_FK

## 3.7. Otras restricciones

Seguirán al patrón **APL\_XXX\_YYY\_ZZ**



Donde XXX indica el nombre de la tabla, YYY es un nombre representativo del campo o campos afectados y ZZ es el sufijo que indica lo que hace la restricción, que puede tomar uno de estos valores:

- \_UK: para claves únicas (UNIQUE).
- \_CK: para restricciones de comprobación (CHECK)
- \_NN: para restricciones no nulas (NOT NULL).

**Nota:** para las restricciones NOT NULL no es necesario definir el nombre si se hacen en línea (cuando se define el nombre del campo en la sentencia de creación de tabla). Para el resto de restricciones sí que se tiene que definir el nombre siguiendo el patrón.

Ejemplos:

- APL\_CLIENT\_NIF\_UK: clave única para el campo NIF de la tabla APL\_CLIENT
- APL\_CLIENT\_SEXE\_CK: comprobación del campo SEXE de la tabla APL\_CLIENT
- APL\_CLIENT\_EDAT\_NN: el campo EDAT de la tabla APL\_CLIENT no puede ser nulo

### 3.8. Índices

Seguirán al patrón **APL\_XXX\_YYY\_I**

En general, los índices siguen la misma nomenclatura que la constraint correspondiente, seguida del sufijo '**I**'.

Para el resto de índices, XXX indica el nombre de la tabla e YYY el nombre del campo a indexar.

Ejemplos:

- APL\_CLIENT\_PK\_I: índice para la clave primaria
- APL\_CLIENT\_ILLA\_FK\_I: índice para la clave extranjera
- APL\_CLIENT\_NUM\_SS\_I: índice para el campo NUM\_SS

### 3.9. Roles

Seguirán al patrón **APL\_XXX**

Donde XXX es un nombre representativo del rol.

Ejemplos:

- APL\_CONSULTA
- APL\_MANTENIMENT
- APL\_ADMINISTRACIO

### 3.10. Procedimientos, funciones, paquetes y resto de objetos

Generalmente, no se permite el uso de procedimientos (procedure), funciones (function), ni paquetes (packages).

En los casos que su uso esté justificado, 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.

Aunque no es obligatorio, se recomienda utilizar un sufijo para cada tipo de objeto a fin de identificarlos rápidamente. Se proponen los siguientes:

Procedimientos	APL_XXX_PR
Funciones	APL_XXX_FN
Paquetes	APL_XXX_PQ

### 3.11. Sinónimos

No está permitida la creación de sinónimos públicos. Los sinónimos deben quedar asociados al esquema del usuario propietario de la aplicación a partir del cual deberán crearse. Si se tiene que sustituir un sinónimo, se tiene que hacer antes el DROP SYNONYM y luego volverlo a crear. No está permitido utilizar la opción de CREATE OR REPLACE.

En cualquier caso, deja de ser necesario la creación de sinónimos para poder referenciar los objetos del usuario APLICACION desde el usuario WWW\_APLICACIÓN. En su lugar, además de conceder los grants desde el usuario APLICACION al usuario WWW\_APLICACIÓN, se añadirá la siguiente línea al fichero de configuración del datasource "APLICACION-ds.xml":

```
<new-connection-sql>  
  BEGIN  
    EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA=APLICACION';  
  END;  
</new-connection-sql>
```

## 4. Restricciones y recomendaciones

### 4.1. Accesos a las bases de datos

Para el acceso a la base de datos deberá definirse un pool de conexiones. El usuario del pool de conexiones seguirá la nomenclatura WWW\_APLICACION, donde APLICACION coincide con el código de aplicación asignado por la DGMAD.

Para que el usuario WWW\_APLICACION pueda utilizar los objetos del usuario propietario (usuario APLICACION) será necesario dar los privilegios de acceso (grants) adecuados. Los permisos no se darán directamente a usuarios finales. Se darán solamente a roles o a usuarios WWW\_APLICACION.

Los únicos permisos que se permiten asignar a roles o usuarios son: SELECT, INSERT, UPDATE, DELETE, EXECUTE.

Ejemplo: las sentencias GRANT relativas a la tabla APL\_CLIENT de la aplicación APLICACION podrían ser:

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

### 4.2. Tipos de datos

- No se permitirá la utilización de campos de tipo LONG.
- En el caso de crear campos de tipo VARCHAR2 o CHAR, al indicar el número de caracteres se prestará atención a que no ponga "BYTE".

### 4.3. Codificación

- El juego de caracteres de las bases de datos es UTF8 (NLS\_CHARACTERSET = UTF8).
- El national character set es AL16UTF16 (NLS\_NCHAR\_CHARACTERSET = AL16UTF16).
- Las Bases de Datos tiene el valor: NLS\_LENGTH\_SEMANTICS = CHA

### 4.4. Deadlocks

Para evitar los abrazos mortales (deadlocks) que se producen al borrar registros de una tabla que tiene tablas relacionadas (tablas hijas), se recomienda crear un índice para las claves extranjeras de la tabla hija.

### 4.5. Particionamiento de tablas

Para tablas con un volumen de datos considerable y para las cuales se realicen muchas consultas por fecha, se recomienda el particionado de las mismas. Consultar antes con los administradores de base de datos de la DGMAD para ver si esto es posible y/o aplica.

## 5. Representación de la lógica de negocio

Se recomienda implementar la lógica de negocio de la aplicación en la capa de negocio, donde residen los programas que se ejecutan, y no en la capa de almacenamiento. De esta forma, la base de datos, únicamente ejerce la función de almacenamiento de los datos. Consultar el documento "Estándares. Desarrollo de aplicaciones del GOIB. Java EE" para más información.

Por lo tanto, no se permite el uso de procedimientos almacenados, paquetes, database links, jobs, triggers etc. en la base de datos. Su uso únicamente está justificado por compatibilidad con bases de datos antiguas que ya disponen de ellos y por necesidad muy motivada que deberá ser justificada por escrito.

### 5.1. Database Links

Los database links no están permitidos excepto en los siguientes casos:

- Compatibilidad con versiones antiguas que ya disponen de ellos. En tal caso, por temas de seguridad, es obligatorio que el database link sea anónimo y no tenga credenciales configuradas. Se utilizan las credenciales propias del usuario que está utilizando el DB Link.
- Aplicaciones que utilizan Pentaho. Véase documento: Estandards BBDD - database links en aplicaciones Pentaho.

### 5.2. Jobs

Los jobs no están permitidos excepto en el caso de bases de datos antiguas que ya dispongan de ellos. Sin embargo, se evitará en la medida de lo posible la implementación de nuevos jobs.

En su lugar, en caso de que esta lógica deba ser implementada en la base de datos, se recomienda la utilización de crontab. Contactar con los administradores de base de datos de la DGMAD para más información.

### 5.3. Sinónimos

Los sinónimos públicos no están permitidos y su necesidad de implementación debe estar muy justificado. Deja de ser necesaria la creación de sinónimos para referenciar los objetos del usuario APLICACION desde WWW\_APLICACION. Consultar el apartado Nomenclatura – Sinónimos para más información.

### 5.4. Disparadores (Triggers)

Los triggers no están permitidos y su necesidad de implementación en base de datos debe estar muy justificado.

## 6. Problemas frecuentes

Listado de los problemas más comunes detectados durante la validación de estándares de bases de datos, y que sería conveniente revisar antes de pedir una instalación de aplicaciones en los servidores de la DGMAD.

### 6.1. Nomenclatura de objetos

El principal problema encontrado es el referente a la nomenclatura de objetos. En general se sigue el estándar de nomenclatura para el nombre de las tablas, pero para el resto de objetos no se cumple (índices, claves primarias y extranjeras, etc.).

### 6.2. Objetos de tablespaces incorrectos

Cuando se crea un usuario con dos o más tablespaces, el tablespace por defecto del usuario es el de datos. Muchas veces se definen los índices sin indicar el nombre del tablespace de índices, por lo que se crea en el tablespace por defecto del usuario. A la hora de crear un índice o un LOB, se tiene que indicar expresamente que se cree en el tablespace correspondiente.

Ejemplo:

```
CREATE INDEX APL_CLIENT_NIF_I ON APL_CLIENT(NIF) TABLESPACE
  APLICACION_INDEX;
```

### 6.3. Uso de campos byte en lugar de char

Las bases de datos de la DGMAD utilizan el NLS\_LENGTH\_SEMANTICS a CHAR. Si la empresa desarrolladora lo tiene definido en BYTES, a la hora de mandar los scripts de creación de tablas se envían con BYTE en la definición de columnas de tipo VARCHAR2. El uso de BYTE no está permitido por la DGMAD, se tiene que enviar en CHAR o sin poner nada.

Ejemplo:

- Incorrecto 

```
CREATE TABLE APL_CLIENT
(campo1 VARCHAR2(100 BYTE));
```
- Correcto 

```
CREATE TABLE APL_CLIENT
(campo1 VARCHAR2(100 CHAR));
```

### 6.4. Asignación de privilegios incorrectos

Los únicos privilegios permitidos son SELECT, INSERT, UPDATE, DELETE y EXECUTE.

No puede haber privilegios ALTER ni REFERENCES. En caso de tener que utilizarlos se tiene que consultar antes con los administradores de base de datos de la DGMAD.

Es habitual encontrar en los scripts GRANT ALTER a secuencias. Si se tiene que modificar una secuencia, se tiene que hacer enviando el correspondiente cuaderno de carga a la DGMAD con la sentencia DDL correspondiente.

### **6.5. Objetos con código dentro del script SQL**

Los scripts de base de datos se ejecutan en SQL\*Plus. Es habitual que los desarrolladores utilicen TOAD u otros programas para desarrollar, pero al enviar los scripts de base de datos estos no funcionan en el SQL\*Plus.

El error más habitual es no poner la barra (/) al final de los objetos que tienen código (disparadores, package/procedure/function con loop/declare/begin-end), ya que estos no se compilan ni ejecutan. **La barra (/) se tiene que poner después de cada objeto que contiene código.** Por ejemplo, si en un mismo archivo se envían cinco disparadores, tiene que haber una barra (/) después de la definición de cada disparador.

## 7. Envío del cuaderno de carga y ejecución de scripts

La ejecución de los scripts de base de datos tiene que seguir el procedimiento especificado en el documento “Estándar de implantación de aplicaciones”, Capítulo 3.5.1. “Scripts de generación de los objetos de base de datos”.

Se tiene que prestar especial atención a lo siguiente:

- Se tienen que enumerar los archivos (en el caso de instalaciones de muchos archivos es difícil encontrar el archivo a ejecutar) con el número correspondiente al orden de ejecución de la instalación (ejemplo: 01-tablas.sql, 02-vistas.sql, 03-procedimientos.sql, ...).
- Los nombres de archivo no tienen que incluir espacios en blanco ni caracteres especiales.
- No hay que mezclar sentencias DDL y DML en el mismo archivo
- Evitar enviar archivos muy extensos, ya que si falla algún punto del archivo, éste se continúa ejecutando hasta el final, y puede hacer que la vuelta atrás de la instalación sea muy complicada.
- En el caso de tener que ejecutar scripts en lugar de sentencias DML independientes, se tendrá que enviar cada script en un archivo a parte.
- **No enviar los scripts DML con sentencias commit.** Si la ejecución del cuaderno de carga es correcta, los administradores de la base de datos ya harán el commit correspondiente. Esto puede evitar muchos problemas ya que si se envían cinco scripts y todos hacen commit, pero el último de ellos falla, al haberse hecho commit en los scripts anteriores no es posible volver atrás la instalación completa haciendo un rollback.
- Por el contrario, **si el script se tiene que programar sí hay que enviar commit y exit al final del script.**
- En cuanto a la codificación de los ficheros, **todos los scripts de bases de datos deberán estar codificados en Unicode (UTF-8), Occidental (ISO-8859-1) o ASCII. En caso de enviar varios scripts no se pueden mezclar ficheros codificados en UTF-8 con ficheros codificados en ISO. No se acepta ficheros con BOM.**
- Para más información consultar el documento “Estándar de implantación de aplicaciones”.