

# **Badulake Inc**

Leandro Vega Palma, Mario Muñoz Mesa, César Muñoz Reinoso,  
José Cruz López



# Índice

<b>Índice</b>	<b>2</b>
1. Entidad Ficticia	<b>3</b>
2. Estructura de almacenamiento.	<b>4</b>
3. Ubicar las copias del control file	<b>4</b>
4. TableSpaces, DataFiles y Redo log files	<b>5</b>
Redo log files	5
Creación de la base de datos	6
Ejecutamos los scripts necesarios para construir las vistas del diccionario de datos, sinónimos y paquetes PL/SQL	7
Creación de los tablespaces y datafiles	7
Tablas necesarias	8
Sentencias necesarias para su creación	9
5. Estructura de Roles	<b>10</b>
Permisos:	11
Sentencias necesarias:	12
6. Perfiles	<b>14</b>
Sentencias:	14
7. Política de Backups	<b>15</b>
Backup Frío	15
Backup Caliente	15
Export	16
Bibliografía	<b>17</b>
Anexo	<b>17</b>
Configuración de VirtualBox	17
Contenido del pfile initBadudb.ora	17
Iniciar la BD	19

# 1. Entidad Ficticia

Nuestra entidad ficticia es una cadena de supermercados, **Badulake Inc.** El SGBD se utilizará para gestionar las ventas, los productos disponibles y los fondos disponibles en las cuentas bancarias.

La entidad tendrá un total de 100 trabajadores repartidos en varios establecimientos.

La base de datos será utilizada por los trabajadores de Badulake Inc, dado que al ser una cadena de supermercados sobredimensionados, no tiene sentido que un cliente cualquiera tenga acceso a la información de la BD.

Los **ejecutivos** se encargan de la administración general de la empresa. Este departamento actualmente está formado por el CEO, Apu.

El departamento dedicado a la **gerencia** que es en torno al 19% de la empresa. Este consta de 19 empleados, los familiares de Apu. Son los encargados de la gestión de recursos humanos, nóminas, contratos de proveedores...

Luego están los **encargados** los cuales los hay de dos tipos:

- Los trabajadores de **desarrollo**, que hacen pruebas con el sistema e introducen los nuevos avances tecnológicos para la compañía, el equipo está formado por 8 trabajadores, ubicados en la oficina central.
- Los **informáticos**, que constan 2 empleados, encargados de la administración de la BD.

Los **usuarios de aplicaciones** podrán utilizar las herramientas proporcionadas por la empresa y estos trabajan con los clientes. Este grupo está formado por 70 trabajadores, los cuales son **reponedores y cajeros**.

Sus privilegios dentro del sistema se verán con más detalle en la sección de **Roles y Perfiles**.

## 2. Estructura de almacenamiento.

Teniendo en cuenta los recursos de nuestra entidad y sus necesidades, consideramos que con dos controladoras se podría cumplir con los objetivos propuestos. Se utilizarían las siguientes controladoras de disco:

- 2 controladoras SATA3
- 1 controlador LTO-9.

Como dispositivos de almacenamiento usaremos:

- 1 HDD 500gb para el SGBD junto con sus ficheros asociados (ficheros de control, ficheros redo logs...); y los tablespaces.
- 1 HDD 1tb para almacenar copias: del fichero de control, de los redo logs; y el tablespace del entorno de explotación.
- 1 cinta magnética LTO-9 de 45Tb para las copias de seguridad.

En la cinta magnética tendríamos, por las propias características de la cinta, una organización de ficheros secuencial física. Se elige una cinta magnética para backups por el alto tiempo de vida y bajo costo.

En los HDD tendríamos una organización de ficheros de acceso directo por la alta velocidad de lectura/escritura que proporciona. Se elige un disco mecánico HDD y no NVMe por la gran cantidad de lecturas y escrituras que requerirá el SGBD.

### 3. Ubicar las copias del control file

Vamos a tener copias multiplexadas del fichero de control. Se repartirán en dos discos y en cada disco se tendrán dos copias para tener redundancia y prevenir ante errores I/O, corrupción de fichero... etc. Debemos especificar el nombre del archivo y su ubicación en el parámetro CONTROL\_FILES del fichero de parámetros. Una vez añadidas las rutas de los control files al fichero de parámetros el propio SGBD se encargará de realizar los cambios oportunos para que todas las copias estén actualizadas. El proceso que hemos seguido ha sido el siguiente:

1. Creamos un directorio en cada disco donde estarán ubicados los ficheros de control.

```
mkdir -p /database/app/control_files  
mkdir -p /mnt/disk_c/oracle/Badudb/control_files/
```

2. Creamos un nuevo fichero de parámetros a partir de una plantilla. En este caso, vamos a usar el spfile de la base de datos que hemos usado en las prácticas y lo modificaremos según nuestros requisitos. Para eso, nos conectamos a la base de datos y con el comando mostrado a continuación, exportamos el archivo binario SPFILE creando un archivo PFILE, ya legible, en la ruta especificada.

```
cp $ORACLE_BASE/admin/oradba/pfile/init.ora.252022164630  
/database/app/oracle/dbs/initBadudb.ora
```

3. Cerramos la base de datos:

```
shutdown immediate;
```

4. Abrimos el PFILE que acabamos de crear y editamos el parámetro CONTROL\_FILES. Este cambio se aplicará la siguiente vez que iniciemos la base de datos.

```
control_files=(/database/app/oracle/oradata/Badudb/Badudb_control01.ctl,  
/database/app/oracle/oradata/Badudb/Badudb_control02.ctl,  
/mnt/disk_c/oracle/Badudb/control_files/Badudb_control03.ctl,  
/mnt/disk_c/oracle/Badudb/control_files/Badudb_control04.ctl)
```

5. Si posteriormente quisiéramos consultar el directorio de los control files nos bastaría con ejecutar:

```
SQLPlus> select name from v$controlfile;
```

## 4. TableSpaces, DataFiles y Redo log files

### Redo log files

Para calcular los grupos y ficheros de redo log estimamos unas 16 horas de servicio por día, 7 días de servicio por semana, 488000 operaciones por semana y que cada redo log file pueda almacenar 5000 operaciones en 200MB:

$$\frac{488000 \text{ ops}}{\text{semana}} \cdot \frac{200 \text{ MB}}{5000 \text{ ops}} \cdot \frac{1 \text{ semana}}{7 \text{ días}} \cdot \frac{1 \text{ día}}{16 \text{ horas}} \approx \frac{175 \text{ MB}}{\text{hora}}$$

Si suponemos que en momentos de máximo uso se realizan más del doble de operaciones por hora, unos 175MB/30min, y que consideramos una frecuencia razonable de cambio de grupos del redo log file en esos picos de uso es de unos 20 minutos, con unos redo log files de 175MB tendríamos el tamaño suficiente para gestionar la situación.

Si programamos un log switch cada 30 min, dada la capacidad de nuestro fichero de redo log, tendríamos tiempo suficiente para almacenar una copia de la información en otra ubicación antes de que el fichero vuelva a ser usado. Además, en momentos de máximo uso, no se llenaría en menos de 20min. Consideramos por estos motivos que el mínimo que requiere Oracle [\[1\]](#), dos grupos, es suficiente en este caso.

Dado que puede crecer el número de operaciones por el propio crecimiento de la empresa se establece un número generoso para máximo número de grupos de redo log files. Esto lo haremos dándole el valor 10 al parámetro MAXLOGFILES cuando creemos nuestra base de datos.

Para evitar errores de lectura/escritura y archivos corruptos, multiplexaremos los ficheros de cada grupo en cuatro ubicaciones diferentes, dos en cada disco.

### Creación de la base de datos

Una vez decididas las características de nuestro sistema de redo logs, procederemos a crear la base de datos. Para esto, en primer lugar, debemos asignar a la variable del sistema `ORACLE_SID` el nombre de nuestra base de datos:

```
ORACLE_SID=Badudb
export ORACLE_SID
```

Generamos el fichero de contraseñas e introducimos una:

```
orapwd file=/database/app/oracle/dbs/orapwBadudb
# Contraseña: ABD3oradba!
```

Para continuar, necesitamos poder acceder con el usuario SYSDBA.  
Además, debemos tener listo nuestro fichero de parámetros y estar en modo *STARTUP NOMOUNT*:

```
SQLPlus> STARTUP NOMOUNT PFILE= '$ORACLE_HOME/dbs/initBadudb.ora'
```

Llegados a este punto, ejecutando el siguiente comando crearemos nuestra base de datos:

```
CREATE DATABASE Badudb
USER SYS IDENTIFIED BY BaduSys
USER SYSTEM IDENTIFIED BY BaduSystem
LOGFILE GROUP 1
('/mnt/disk_c/oracle/Badudb/redo_log_files/group1log1.log',
'/mnt/disk_c/oracle/Badudb/redo_log_files/group1log2.log',
'/database/app/oracle/oradata/Badudb/group1log3.log',
'/database/app/oracle/oradata/Badudb/group1log4.log') SIZE 175M,
GROUP 2 ('/mnt/disk_c/oracle/Badudb/redo_log_files/group2log1.log',
'/mnt/disk_c/oracle/Badudb/redo_log_files/group2log2.log',
'/database/app/oracle/oradata/Badudb/group2log3.log',
'/database/app/oracle/oradata/Badudb/group2log4.log') SIZE 175M
MAXLOGFILES 10
MAXLOGMEMBERS 5
CHARACTER SET US7ASCII
EXTENT MANAGEMENT LOCAL
ENABLE PLUGGABLE DATABASE SEED
FILE_NAME_CONVERT=('/database/app/oracle/oradata/Badudb', '/database/app/
oracle/oradata/Badudb') SYSTEM DATAFILES SIZE 350M SYSAUX DATAFILES SIZE
100M
UNDO TABLESPACE baduundotbs1
DATAFILE '/database/app/oracle/oradata/Badudb/baduundotbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

Ejecutamos los scripts necesarios para construir las vistas del diccionario de datos, sinónimos y paquetes PL/SQL

En SQL\*Plus, nos conectamos a la base de datos con privilegio administrativo SYSDBA y ejecutamos [2]:

```
@?/rdbms/admin/catalog.sql
@?/rdbms/admin/catproc.sql
@?/rdbms/admin/utlrp.sql
```

En SQL\*Plus, nos conectamos a la base de datos como usuario SYSTEM y ejecutamos [2]:

```
@?/sqlplus/admin/pupbld.sql
```

## Creación de los tablespaces y datafiles

El siguiente paso sería crear los tablespaces y sus correspondientes datafiles. Crearemos tres tablespaces, cada uno para una etapa del desarrollo de nuestro sistema:

*DESARROLLO, PRUEBAS Y EXPLOTACION*. Estimamos que, con el paso del tiempo, la cantidad de datos almacenados en nuestra base de datos se incrementará. Por ese motivo, configuramos los datafiles para que su tamaño aumente conforme vaya siendo necesario. Otra cuestión que debemos tratar es la ubicación de los datafiles. Hemos decidido ubicar el datafile del tablespace de explotación en una unidad diferente para que el trabajo de desarrollo no afecte a la experiencia del usuario final. A continuación se muestran los comandos necesarios para crear los tablespaces y los datafiles:

```
CREATE TABLESPACE DESARROLLO
DATAFILE
'/database/app/oracle/oradata/Badudb/desa_data.dbf' SIZE
1024M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLESPACE PRUEBAS
DATAFILE
'/database/app/oracle/oradata/Badudb/pru_data.dbf' SIZE
1024M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

CREATE TABLESPACE EXPLOTACION
DATAFILE
'/mnt/disk_c/oracle/Badudb/table_spaces/exp_data.dbf' SIZE
1024M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

## Tablas necesarias

El grueso de la base de datos se resume en 3 tipos de tablas, las cuales se replican en cada uno de los entornos de los que dispone la empresa. Estas tablas y su estructura son:

### Producto:

- ID codigo producto (int)**
- Stock (int)**
- Precio (double)**



**Ventas:**

IDVenta (int)  
Producto (varchar)  
Cantidad (int)

**Cuentas:**

IDtransacción (int)  
Cuantía (int)  
Tipo (varchar)

Las cuales se replican en los tres entornos de la empresa: Explotación, desarrollo y pruebas.

## Sentencias necesarias para su creación

```
CREATE TABLE producto_explotacion(  
    idcod NUMBER(9) PRIMARY KEY,  
    stock NUMBER(4),  
    precio NUMBER (6,2))  
    TABLESPACE explotacion  
;  
  
CREATE TABLE producto_pruebas(  
    idcod NUMBER(9) PRIMARY KEY,  
    stock NUMBER(4),  
    precio NUMBER (6,2))  
    TABLESPACE pruebas  
;  
  
CREATE TABLE producto_desarrollo(  
    idcod NUMBER(9) PRIMARY KEY,  
    stock NUMBER(4),  
    precio NUMBER (6,2))  
    TABLESPACE desarrollo  
;  
  
CREATE TABLE venta_explotacion(  
    idtransaccion NUMBER(9) PRIMARY KEY,  
    idproducto NUMBER(9) FOREIGN KEY REFERENCES producto_explotacion(idcod),  
    cantidad NUMBER(4),  
    tipo VARCHAR(20))  
    TABLESPACE explotacion  
;
```

```
    idven NUMBER(9) PRIMARY KEY,  
    producto VARCHAR2(20),  
    cantidad NUMBER (4))  
    TABLESPACE explotacion  
;  
  
CREATE TABLE venta_pruebas(  
    idven NUMBER(9) PRIMARY KEY,  
    producto VARCHAR2(20),  
    cantidad NUMBER (4))  
    TABLESPACE pruebas  
;  
  
CREATE TABLE venta_desarrollo(  
    idven NUMBER(9) PRIMARY KEY,  
    producto VARCHAR2(20),  
    cantidad NUMBER (4))  
    TABLESPACE desarrollo  
;  
  
CREATE TABLE cuentas_explotacion(  
    idtran NUMBER(9) PRIMARY KEY,  
    cuantia NUMBER(4),  
    tipo VARCHAR2 (20))  
    TABLESPACE explotacion  
;  
  
CREATE TABLE cuentas_pruebas(  
    idtran NUMBER(9) PRIMARY KEY,  
    cuantia NUMBER(4),  
    tipo VARCHAR2 (20))  
    TABLESPACE pruebas  
;  
  
CREATE TABLE cuentas_desarrollo(  
    idtran NUMBER(9) PRIMARY KEY,  
    cuantia NUMBER(4),  
    tipo VARCHAR2 (20))  
    TABLESPACE desarrollo  
;
```

## 5. Estructura de Roles

Los roles de los que dispone el SGBD son los siguientes:

**APU** -> EJECUTIVO

**MANAGER** -> GERENCIAL

**ENCARGADO** -> DESARROLLO | INFORMÁTICA

**REPONEDOR/CAJERO** -> USUARIO DE APP

Los distintos entornos de nuestra empresa, cabe destacar que el de gestión está replicado en cada uno de los otros tres, aunque el que realmente se utiliza es el de explotación.

- Explotación
  - **Productos a la venta siempre.**
- Gestión
  - Ventas
  - Cuentas
- Pruebas
  - Nuevos productos que se empiezan a vender nuevos.
- Desarrollo
  - Proveedores que ofrecen nuevos productos.

### Permisos:

**rol\_acceso\_producto\_explotacion** -> Todos (productos de EXPLOTACIÓN)

**rol\_acceso\_gestion\_explotacion** -> manager, apu (ventas cuentas de EXPLOTACION)

**rol\_acceso\_pruebas** (producto + gestión) -> reponedor, encargado, manager (producto de PRUEBAS)

**rol\_acceso\_desarrollo** (producto + gestión) -> reponedor, encargado, apu

**rol\_interno** -> encargado, apu, manager

**rol\_usuario**

- rol\_acceso\_producto\_explotacion

**rol\_encargado**

- **rol\_desarrollo**
  - rol\_usuario
  - rol\_acceso\_pruebas
- rol\_acceso\_desarrollo
- **rol\_informatica** -> DBA
  - rol\_desarrollo
  - rol\_interno (DBA)

**rol\_gerencial**

- rol\_acceso\_pruebas
- rol\_acceso\_gestion\_explotacion
- rol\_usuario

#### **rol\_ejecutivo**

- rol\_gerencia
- rol\_acceso\_desarrollo

### Sentencias necesarias:

```
CREATE ROLE rol_acceso_producto_explotacion ;
GRANT SELECT, INSERT, UPDATE, DELETE ON producto_explotacion TO
rol_acceso_producto_explotacion;
```

```
CREATE ROLE rol_acceso_gestion_explotacion ;
GRANT SELECT, INSERT, UPDATE, DELETE ON venta_explotacion TO
rol_acceso_gestion_explotacion;
GRANT SELECT, INSERT, UPDATE, DELETE ON cuentas_explotacion TO
rol_acceso_gestion_explotacion;
```

```
CREATE ROLE rol_acceso_pruebas ;
GRANT SELECT, INSERT, UPDATE, DELETE ON producto_pruebas TO
rol_acceso_pruebas;
GRANT SELECT, INSERT, UPDATE, DELETE ON venta_pruebas TO
rol_acceso_pruebas;
GRANT SELECT, INSERT, UPDATE, DELETE ON cuentas_pruebas TO
rol_acceso_pruebas;
```

```
CREATE ROLE rol_acceso_desarrollo ;
GRANT SELECT, INSERT, UPDATE, DELETE ON producto_desarrollo TO
rol_acceso_desarrollo;
GRANT SELECT, INSERT, UPDATE, DELETE ON venta_desarrollo TO
rol_acceso_desarrollo;
GRANT SELECT, INSERT, UPDATE, DELETE ON cuentas_desarrollo TO
rol_acceso_desarrollo;
```

```
CREATE ROLE rol_usuario ;
GRANT CREATE SESSION, ALTER SESSION TO rol_usuario;
GRANT rol_acceso_producto_explotacion TO rol_usuario;
```

```
CREATE ROLE rol_desarrollo;  
GRANT rol_usuario TO rol_desarrollo;  
GRANT rol_acceso_pruebas TO rol_desarrollo;  
GRANT rol_acceso_desarrollo TO rol_desarrollo;
```

```
GRANT CREATE ANY TABLE, CREATE ANY VIEW, DROP ANY VIEW, CREATE ANY  
PROCEDURE, ALTER ANY PROCEDURE, DROP ANY PROCEDURE, CREATE ANY TRIGGER,  
ALTER ANY TRIGGER, DROP ANY TRIGGER, CREATE ANY SEQUENCE, DROP ANY  
SEQUENCE, CREATE ANY SYNONYM, DROP ANY SYNONYM, DEBUG ANY PROCEDURE TO  
rol_desarrollo;
```

```
CREATE ROLE rol_informatica;  
GRANT rol_desarrollo TO rol_informatica;  
GRANT DBA TO rol_informatica;
```

```
CREATE ROLE rol_gerencia;  
GRANT rol_acceso_pruebas TO rol_gerencia;  
GRANT rol_acceso_gestion_explotacion TO rol_gerencia;  
GRANT rol_usuario TO rol_gerencia;
```

```
CREATE ROLE rol_ejecutivo;  
GRANT rol_gerencia TO rol_ejecutivo;  
GRANT rol_acceso_desarrollo TO rol_ejecutivo;
```

## 6.Perfiles

Para poder limitar los recursos utilizados por los trabajadores, hemos creado perfiles para los diferentes niveles uso:

- Los usuarios de los niveles **ejecutivos** y **gerenciales** pueden tener dos conexiones simultáneas con la base de datos así como tiempo ilimitado de conexión.
- Los **informáticos** y **desarrolladores de aplicaciones** no tienen limitaciones de conexión en cuanto a cantidad ni tiempo pero si tienen un número limitado de 6 intentos de acceso. Se les obliga además a cambiar la contraseña cada 90 días.
- Los **reponedores** y **cajeros** tienen una limitación de una sesión por usuario, tiempo máximo de conexión de 8 horas así como un límite de 3 intentos de acceso. Se les obliga además a cambiar la contraseña cada 30 días.

Sentencias:

```
ALTER SYSTEM SET RESOURCE_LIMIT=TRUE;

CREATE PROFILE c##apu LIMIT
SESSIONS_PER_USER 2
CONNECT_TIME UNLIMITED;

CREATE PROFILE c##encargado LIMIT
SESSIONS_PER_USER UNLIMITED
CONNECT_TIME UNLIMITED
FAILED_LOGIN_ATTEMPTS 6
PASSWORD_LIFE_TIME 90;

CREATE PROFILE c##reponedor LIMIT
SESSIONS_PER_USER 1
CONNECT_TIME 480
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LIFE_TIME 30;
```

## 7. Política de Backups

### Backup Frío

Se realizarán 4 copias en frío en cada cambio de estación (19 de Marzo, 21 de Junio, 19 de Septiembre, 21 de Diciembre), fuera del horario laboral, a una hora de baja actividad de la DB que será escogida después de analizar las tendencias de la misma. En concreto los archivos que se copiarán serán:

- Datafiles
- Controlfiles  
(ALTER DATABASE BACKUP CONTROLFILE TO 'oracle/backup/control.bkp';) -> otra forma con comandos nativos de la BD.
- Password files
- Parameter files
- Redologs

- Crear carpeta /mnt/backups/backups\_frio

```
(Fuera de la BD) $ mkdir /mnt/backups/backups_frio
```

```
SHUTDOWN IMMEDIATE;  
$ cp -rvp $ORACLE_BASE /mnt/backups/backups_frio #carpeta oracle  
$ cp -rvp /mnt/disk_c/oracle/ /mnt/backups/backups_frio #carpeta oracle  
$ cp -rvp /databases/app/oraInventory /mnt/backups/backups_frio #carpeta  
oraInventory  
STARTUP
```

### Backup Caliente

Se realizan copias calientes cada fin de semana, exactamente los domingos a las 3 a.m.

- Crear carpeta /mnt/backups/backups\_caliente

```
(Fuera de la BD) $ mkdir /mnt/backups/backups_caliente
```

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;  
ALTER DATABASE ARCHIVELOG; #Habilitar base de datos para el modo archivado  
ALTER DATABASE OPEN;
```

```
ALTER TABLESPACE users BEGIN BACKUP;  
$ cp -rvp $ORACLE_BASE/oradata/Badudb/*.dbf  
/mnt/backups/backups_caliente  
$ cp -rvp /mnt/disk_c/oracle/Badudb/table_spaces/exp_data.dbf  
/mnt/backups/backups_caliente  
ALTER TABLESPACE users END BACKUP;
```

## Export

```
(Fuera de la BD) $ mkdir /mnt/backups/backups_export
```

Exportación **completa** todos los fines de semana junto con los backups calientes  
En bash, fuera de la base de datos:

```
$ exp full=y constraints=Y file= /mnt/backups/backups_export/completa
```

Cuando pida el usuario y la contraseña, hay que poner:

**username: BaduSys as sysdba**  
**password:ABD3oradba!**

Exportación **lógica incremental** todos los días al acabar la jornada laboral  
En bash, fuera de la base de datos:

```
exp full=y inctype=incremental constraints=Y file=  
/mnt/backups/backups_export/incremental
```

Al igual que antes, cuando pida el usuario y la contraseña, hay que poner:

**username: BaduSys as sysdba**  
**password:ABD3oradba!**



## Bibliografía

1. [https://docs.oracle.com/cd/B19306\\_01/server.102/b14231/onlineredo.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14231/onlineredo.htm)
2. [https://docs.oracle.com/cd/E11882\\_01/server.112/e25494/create.htm#ADMIN11082](https://docs.oracle.com/cd/E11882_01/server.112/e25494/create.htm#ADMIN11082)
3. <https://www.toolbox.com/tech/oracle/question/how-to-connect-as-sysdba-in-export-040403/>
4. <https://developpaper.com/oracle-database-exp-export-data-when-there-is-no-other-choice/>

## Anexo

### Configuración de VirtualBox

Pasos, fuera del contexto de Oracle, que fueron necesarios

- Añadir los discos necesarios en Virtualbox - 2 de 8gb (el por defecto)
- Particionar los discos con la herramienta gParted (tabla de particiones tipo msdos)
- Montar los discos siguiendo los pasos detallados a continuación:
  - su root
  - mkdir /mnt/backups
  - mkdir /mnt/disk\_c
  - chown -R oracle /mnt/backups
  - chown -R oracle /mnt/disk\_c
  - nano /etc/fstab
    - Dentro de nano:
      - consultar UUID de los discos con blkid y escribir
      - UUID=... /mnt/backups ext4 defaults 1 2
      - UUID=... /mnt/disk\_c ext4 defaults 1 2

```
mkdir -p /database/app/oracle/oradata/Badudb/  
mkdir /database/app/oracle/admin/Badudb/adump  
mkdir -p /mnt/disk_c/oracle/Badudb/redo_log_files  
mkdir -p /mnt/disk_c/oracle/Badudb/control_files  
mkdir -p /mnt/disk_c/oracle/Badudb/table_spaces
```

### Contenido del pfile initBadudb.ora

```
#####
```

```

#####
# Copyright (c) 1991, 2013 by Oracle Corporation
#####
#####

#####
# Cache and I/O
#####
db_block_size=8192

#####
# File Configuration
#####
control_files=(/database/app/oracle/oradata/Badudb/Badudb_control01.ctl,
/database/app/oracle/oradata/Badudb/Badudb_control02.ctl,/mnt/disk_c/ora
cle/Badudb/control_files/Badudb_control03.ctl,/mnt/disk_c/oracle/Badudb/
control_files/Badudb_control04.ctl)

#####
# Cursors and Library Cache
#####
open_cursors=300

#####
# Database Identification
#####
db_domain="pclab.localdomain"
db_name="badudb"

#####
# NLS
#####
nls_language="SPANISH"
nls_territory="SPAIN"

#####
# Miscellaneous
#####
compatible=21.0.0
diagnostic_dest=/database/app/oracle
enable_pluggable_database=true
memory_target=1135m

#####
# Processes and Sessions
#####

```

```

processes=300

#####
# Network Registration
#####
local_listener=LISTENER_BADUDB

#####
# System Managed Undo and Rollback Segments
#####
undo_tablespace=baduundotbs1

#####
# Security and Auditing
#####
audit_file_dest="/database/app/oracle/admin/Badudb/adump"
audit_trail=db
remote_login_passwordfile=EXCLUSIVE

#####
# Shared Server
#####
dispatchers="(PROTOCOL=TCP) (SERVICE=BadudbXDB)"

```

## Iniciar la BD

### CONTRASEÑA

-----  
ABD3oradba!

### INICIAR SESIÓN

-----  
lsnrctl start // iniciar listener

sqlplus BaduSys as sysdba // iniciar sesión con sys como administrador

startup

exit

## **ACABAR SESIÓN**

---

shutdown immediate

exit

lsnrctl stop