

PROYECTO FINAL
BDD empleada para administrar ventas de autos -Parte 2

El siguiente paso a desarrollar en este documento es la creación de los objetos de base de datos necesarios en cada uno de los nodos de la BDD.

1.1. CONFIGURACIÓN DE SERVICIOS DE RED.

1.1.1. Conexiones de Red.

Las máquinas físicas que participarán en la construcción de la base de datos distribuida deberán conectarse a una misma red de tal forma que sea posible establecer una comunicación bidireccional. Para efectos del proyecto final, ambos equipos pueden conectarse a una misma red empleando alguna de las siguientes técnicas:

- A través de una Red Wifi. Se cuenta un Router que asigna IPs, ya sea de algún servicio proveedor de Internet, o de algún dispositivo de red tipo Router. Ejemplo: Dispositivos TP-LINK, etc.
- A través del uso de un Celular que actúa como Router. En algunos casos es posible hacer uso de la funcionalidad “Compartir Internet” que ofrecen los teléfonos celulares. El uso del plan de datos se deshabilita, las máquinas se conectan a la Red WiFi que expone el celular y por lo tanto se crea una red en la que ambas máquinas serán visibles. Se recomienda dejar esta opción como último recurso ya que el enlace generalmente es lento e inestable.
- A través de la tarjeta Ethernet. Alguno o ambos equipos pueden estar conectados al Router empleando un cable de red Ethernet.
- A través de una conexión directa entre los equipos empleando un cable de red. En la siguiente sección se explica la forma en la que se puede realizar la configuración para conectar 2 equipos a través de un cable de red (conexión máquina – máquina).
- A través del uso de una VPN. Es posible conectar 2 máquinas desde ubicaciones remotas a través de la configuración de una VPN. Existen algunas soluciones gratuitas que pueden ser útiles en especial para este tipo de proyectos. La conexión no suele ser tan rápida, pero puede ser útil para trabajar en equipo desde ubicaciones remotas. Cada integrante del equipo puede estar trabajando desde su casa u oficina. La única desventaja de esta opción es que se requiere de una conexión a internet la cual puede ser no viable en especial para realizar la entrega del proyecto, pero muy útil para su desarrollo. Hamachi VPN es un servicio que permite implementar esta idea. En caso de requerirse, los siguientes enlaces permiten configurar una VPN en una máquina Linux:
 - <https://www.vpn.net/>
 - <https://tecadmin.net/install-and-configure-logmein-hamachi-on-rhelcentos-56/>

1.1.1.1. Conexión máquina – máquina empleando un cable de red.

El siguiente procedimiento describe los pasos a seguir para conectar 2 equipos a través de un cable de red.

A. Ejecutar el comando `ip a` para identificar los nombres de las tarjetas de red

```
[jorge@jrc-ora-pc network-scripts]$ ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00
2: p4p1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:00:00:00
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 08:00:27:00:00:00
```

En este ejemplo, se muestran 3 dispositivos. Generalmente la tarjeta Ethernet se distingue de la tarjeta inalámbrica por la primera letra (W = Wireless). La tarjeta de red lo (loopback) hace referencia a una tarjeta de red “virtual”. Por lo tanto, para este ejemplo, en nombre de la tarjeta Ethernet es **p4p1**.

B. Ejecutar el comando `nmcli` para crear una conexión. Considerar los valores de los siguientes parámetros:

- La nomenclatura para nombrar la conexión es `<nombre_tarjeta>-bdd`.
- Asignar una IP en algún segmento válido. Por ejemplo `192.168.1.1` a una máquina y `192.168.1.2` a la otra máquina. En caso de ser necesario, modificar por otros valores.
- Indicar el valor de la dirección IP Gateway, asegurarse de asignarla al mismo segmento. Por ejemplo: `192.168.1.254`
- Indicar el nombre de la tarjeta Ethernet. Por ejemplo `p4p1`

Considerando lo anterior, el comando a ejecutar para la tarjeta de red **p4p1** será:

```
sudo nmcli con add type ethernet con-name p4p1-bdd \
ifname p4p1 ip4 192.168.1.1/24 gw4 192.168.1.254
```

Este comando creará un archivo llamado `ifcfg-<nombre_tarjeta>-bdd` en el directorio `/etc/sysconfig/network-scripts`.

C. Abrir el archivo y verificar que los parámetros indicados en el comando sean los correctos. En contenido del archivo será similar al siguiente ejemplo.

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.1.1
PREFIX=24
GATEWAY=192.168.1.254
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=p4p1-bdd
UUID=8b9d3647-437b-4a96-b1f9-bcc8cdbcea9f
DEVICE=p4p1
ONBOOT=yes
```

- D. Ejecutar el comando en la máquina 2 con sus respectivos valores.
- E. Conectar un cable de red entre ambas máquinas (no es necesario que sea un cable cruzado).
- F. Ejecutar el comando `nmtui` para realizar la activación (similar al comando ejecutado en prácticas anteriores), seleccionar la opción "Activate a connection" o
- G. Observar que aparecerá en la lista de conexiones el nombre de la conexión configurada anteriormente, en este caso **p4p1-bdd**. Seleccionar esta conexión y seleccionar la opción "Activate" del lado derecho.
- H. Activar la conexión en la otra máquina.
- I. Finalmente, realizar un `ping` entre ambos equipos para verificar que la comunicación bidireccional funciona correctamente.

Posiblemente cada vez que se inicie el equipo se deberá realizar la activación de esta conexión antes de iniciar a trabajar en el proyecto.

1.1.2. Configuración del archivo `/etc/hosts`

- Recordar configurar este archivo para evitar problemas de conectividad. Se recomienda configurar ambas IPs en ambos nodos. La idea es que en los archivos de configuración de Oracle se empleará siempre el nombre de host (hostname) del sitio para evitar la configuración de IPs de forma directa.
- El único punto de configuración de IPs (mapeo hostname <-> IP) se realizará en este archivo.
- Como primer paso ejecutar el comando `ip a` para verificar que la IP obtenida sea la misma a configurar en este archivo.
- En este documento se consideran los siguientes nodos como ejemplo:

Nodo 1:

- o Hostname: `jrc-ora-pc.fi.unam`
- o IP: `192.168.0.4`

Nodo 2:

- o Hostname: `arc-ora-pc.fi.unam`
- o IP: `192.168.0.51`

- Las siguientes imágenes muestran la configuración del archivo:

- o Configuración en `jrc-ora-pc`

```
#maquina local
127.0.0.1      jrc-ora-pc.fi.unam      jrc-ora-pc
192.168.0.4    jrc-ora-pc.fi.unam      jrc-ora-pc

#maquina remota
192.168.0.51   arc-ora-pc.fi.unam      arc-ora-pc
```

- Configuración en arc-ora-pc

```
#maquina local
127.0.0.1          arc-ora-pc.fi.unam      arc-ora-pc
192.168.0.51      arc-ora-pc.fi.unam      arc-ora-pc

#maquina remota
192.168.0.4        jrc-ora-pc.fi.unam      jrc-ora-pc

```

- Configurar el archivo `/etc/hosts` de forma similar para ambos nodos.
- No olvidar editar el archivo `/etc/hostname` para configurar el nombre del host.
- Verificar la comunicación bidireccional ejecutando en comando `ping` empleando los nombres de host :

Ejemplo:

- `ping jrc-ora-pc` desde `arc-ora-pc`
- `ping arc-ora-pc` desde `jrc-ora-pc`

Nota: Si las maquinas no están configuradas con una IP fija, cada vez que se prenda el equipo se deberá hacer una validación de IPs, y en caso de haber cambiado, se deberán actualizar las IPs en ambos nodos.

1.1.3. Revisión del archivo `/etc/profile`.

- Revisar la configuración del archivo, verificar que todas las variables tengan valor adecuado, en especial `ORACLE_SID`, `ORACLE_HOSTNAME`.

Ejemplo:

Configuración para `arc-ora-pc`

```
#variables para Oracle 12c
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=arc-ora-pc.fi.unam
export ORACLE_UNQNAME=arcbd
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/12.1.0.2/db_1
export ORACLE_SID=arcbd
export TNS_ADMIN=$ORACLE_HOME/network/admin
export PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib

```

1.1.4. Configuración del archivo `listener.ora`

- En caso de ser necesario, actualizar el valor de "Hostname" en este archivo con base al valor asignado en `/etc/hosts` para ambos nodos.
- En el siguiente ejemplo se muestra la configuración del archivo en el nodo `arc-ora-pc` el cual apunta a localhost (127.0.0.1). Realizar lo mismo para el otro nodo.

```
LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = arc-ora-pc.fi.unam)(PORT = 1521))
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
  )
)
```

- Levantar el listener en ambos nodos.

1.1.5. Configuración del archivo `tnsnames.ora`

Configurar el archivo `tnsnames.ora` en ambos nodos. El archivo deberá contener la definición de 5 servicios: contenedor ROOT local, 2 PDBs locales y 2 PDBs remotos. Emplear `netmgr` para completar la configuración en cada Nodo.

Ejemplo:

Archivo `tnsnames.ora` en `arc-ora-pc`

```

ARCBD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = arc-ora-pc.fi.unam) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = arcbd.fi.unam)
    )
  )

ARCBD_S1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = arc-ora-pc.fi.unam) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = arcbd_s1.fi.unam)
    )
  )

ARCBD_S2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = arc-ora-pc.fi.unam) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = arcbd_s2.fi.unam)
    )
  )

JRCBD_S1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = jrc-ora-pc.fi.unam) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = jrcbd_s1.fi.unam)
    )
  )

JRCBD_S2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = jrc-ora-pc.fi.unam) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = jrcbd_s2.fi.unam)
    )
  )

LISTENER_ARCBD =
  (ADDRESS = (PROTOCOL = TCP) (HOST = arc-ora-pc.fi.unam) (PORT = 1521))

```

- Observar en la configuración anterior. El hostname y el nombre de servicio local se marca con azul, el host y el nombre de servicio remoto se marcan con rojo.
- Para verificar que la comunicación entre ambos nodos sea correcta, se puede emplear el comando `tnsping` (Ejecutar en ambos nodos).
- Nota: No olvidar levantar las instancias y revisar que las PDBs estén en status OPEN.

Ejemplo:

```

tnsping ARCBD_S1
tnsping ARCBD_S2
tnsping JRCBD_S1
tnsping JRCBD_S2

```

1.2. CREACIÓN DE OBJETOS EN LA BDD

1.2.1. Reglas generales.

Antes de iniciar con la creación y ejecución de scripts SQL, considerar las siguientes reglas:

- Los scripts NO deberán ser creados con el usuario `oracle` del sistema operativo.
- Los scripts NO deberán ser ejecutados con el usuario `oracle` del sistema operativo.
- Los scripts NO deberán ser ejecutados con el usuario `SYS` o `SYSTEM` a menos que se traten de sentencias que lo requieran.
- Todos los scripts deberán hacer uso de la sentencia `connect` para cambiarse de usuario o de PDB. Lo anterior implica que al entrar a `Sql *Plus` se debe emplear la instrucción `sqlplus /nolog`.

Omitir cualquiera de estas reglas causará puntos menos en la calificación.

1.2.2. Creación de usuarios.

Crear un solo script SQL `s-01-autos-usuario.sql`. Este script se deberá realizar las siguientes acciones:

- Eliminar al usuario `autos_bdd` en caso de existir.
- Crear un usuario llamado `autos_bdd`
- Otorgar privilegios para crear sesión, tablas, secuencias, procedimientos, vistas, sinónimos, y ligas. No otorgar más de los permisos requeridos.

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Eliminación y creación del usuario.

declare
v_count number(1,0);
begin
select count(*) into v_count
from dba_users
where lower(username) = 'autos_bdd';
if v_count > 0 then
execute immediate 'drop user autos bdd cascade';
else
dbms_output.put_line('El usuario autos bdd no existe');
end if;
end;
/

Prompt Creando al usuario autos_bdd
--Completar
```

- No olvidar el encabezado a cada script como se muestra en la figura anterior.
- Crear un script llamado `s-01-autos-main-usuario.sql`. Este script deberá conectarse a los 4 nodos y ejecutar el script anterior.

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Creación de usuarios en los 4 nodos

clear screen
whenever sqlerror exit rollback;
set serveroutput on

Prompt Iniciando creacion/eliminacion de usuarios.
accept syspass char prompt 'Proporcione el password de sys: ' hide

prompt =====
prompt Creando usuario en jrcbd_s1
prompt =====
connect sys/ &syspass@jrcbd_s1 as sysdba
```

```
@s-01-autos-usuario.sql
```

```
prompt =====
prompt Creando usuario en jrcbd_s2
prompt =====
connect sys/ &&syspass@jrcbd_s2 as sysdba
@s-01-autos-usuario.sql
```

```
prompt =====
prompt Creando usuario en arcdb_s1
prompt =====
connect sys/ &&syspass@arcdb_s1 as sysdba
@s-01-autos-usuario.sql
```

```
prompt =====
prompt Creando usuario en arcdb_s2
prompt =====
connect sys/ &&syspass@arcdb_s2 as sysdba
@s-01-autos-usuario.sql
```

```
Prompt Listo!
disconnect
```

Creación de ligas.

1.2.3. Creación de ligas

- Generar un solo script SQL llamado s-02-autos-ligas.sql
- El script deberá contener las instrucciones necesarias para crear las ligas en todos los nodos. No es necesario especificar datos de autenticación ya el usuario existe en los 4 nodos.
- La comunicación debe ser bidireccional por lo que se requieren crear 3 ligas por nodo, 12 en total.
- Emplear el nombre global de cada PDB como nombre de la liga:

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:     Creación de ligas en los 4 nodos.
clear screen
whenever sqlerror exit rollback;
Prompt =====
Prompt Creando ligas en jrcbd_s1
Prompt =====

--Ligas en la pdb jrcbd_s1
Prompt Creando ligas en jrcbd_s1
connect autos_bdd/autos_bdd@jrcbd_s1
-- PDB local
create database link jrcbd_s2.fi.unam using 'JRCBD_S2';
--PDB remotas
create database link arcdb_s1.fi.unam using 'ARCBD_S1';
Create database link arcdb_s2.fi.unam using 'ARCBD_S2';

--Ligas en la pdb jrcbd_s2
. . .

--Ligas en la pdb arcdb_s1
. . .

--Ligas en la pdb arcdb_s2
. . .

Prompt Listo!
disconnect
```

1.2.4. Creación de fragmentos.

El siguiente punto es la creación de todas las tablas que representan los fragmentos en cada sitio. Para ello, generar un archivo por cada PDB llamado s-03-autos-<pdb>-ddl.sql. Ejemplo: s-03-autos-jrc-s1-ddl.sql. Considerar los siguientes puntos.

- Se recomienda emplear Er-Studio para generar el código SQL.
- Hacer copias del modelo global y realizar los cambios correspondientes empleando la tabla de fragmentos generada en el documento anterior. **Se deberán imprimir los 5 modelos relacionales incluyendo el global y presentarlos al momento de la entrega del proyecto.**
- Las tablas se deberán crear en orden de dependencias.
- Cada tabla deberá contener la definición de sus restricciones (restricciones de llave primaria, restricciones de integridad: NOT NULL, restricciones de referencia).
- Debido al diseño de fragmentación, algunas restricciones de referencia incluidas en el script anterior no se podrán verificar de manera automática. Las únicas referencias a conservar son las llaves foráneas que provienen de catálogos y las fragmentaciones horizontales derivadas.
- Para identificar las restricciones que se deben **conservar**, revisar las reglas de fragmentación, y llenar la siguiente tabla.

Restricciones de referencia que se conservan.

Nombre de la tabla Padre	Nombre de la tabla hija	Sitio

- Las restricciones de referencia que no pueden conservarse deberán ser eliminadas desde ER -Studio conservando únicamente el campo en la tabla hija.
 - Para eliminar una restricción de referencia en ER-Studio sin eliminar el campo en la tabla hija, seleccionar la relación, presionar “Supr” o en su defecto hacer clic derecho y seleccionar “Delete Relationship”.
 - Aparecerá una ventana que solicita la confirmación de la eliminación.
 - Observar que el campo “Make Foreign Keys Native” debe ser habilitado para que el campo se conserve en la tabla hija y solo se elimine la restricción de referencia.



- Crear un solo script sql llamado s-03-autos-main-ddl.sql. El script deberá conectarse a cada PDB e invocar a los 4 archivos creados anteriormente:

Ejemplo:

```
--@Autor:           Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:     Creación de fragmentos en los 4 nodos
clear screen
whenever sqlerror exit rollback;

prompt =====
prompt Creando fragmentos para jrcbd s1
prompt =====
connect autos_bdd/autos_bdd@jrcbd_s1
@s-03-autos-jrc-s1-ddl.sql

prompt =====
prompt Creando fragmentos para jrcbd s2
prompt =====
connect autos_bdd/autos_bdd@jrcbd_s2
@s-03-autos-jrc-s2-ddl.sql

prompt =====
prompt Creando fragmentos para arcbd s1
prompt =====
connect autos_bdd/autos_bdd@arcbd_s1
@s-03-autos-arc-s1-ddl.sql

prompt =====
```

```
prompt Creando fragmentos para arc_bd_s2
prompt =====
connect autos_bdd/autos_bdd@arc_bd_s2
@s-03-autos-arc-s2-ddl.sql

Prompt Listo!
```

- Ejecutar el script.
- Continuar con la siguiente parte del proyecto.