

Algunas Notas de Apoyo para las Prácticas

Autor: Sergio Ilarri Artigas.

Fecha de creación: 26 de marzo de 2018.

Última actualización: 24 de febrero de 2020.

Contribuyentes: Sergio Ilarri Artigas.

Se incluyen a continuación algunos ejemplos básicos que pueden servir de inspiración para resolver problemas relacionados con las prácticas, así como algunos comentarios sobre diversos SGBD. Este es un documento únicamente de apoyo y puede contener información desactualizada (aunque se intenta actualizar periódicamente cuando se detecta algún aspecto que ha quedado obsoleto). En caso de duda, o para cualquier sugerencia, por favor contactad con el profesor.

Contenido

Ejemplo de uso de VARRAY en Oracle	2
Notas sobre Oracle.....	4
Notas sobre MySQL	10
Notas sobre PostgreSQL.....	14
Notas generales sobre Caché	18
Notas sobre IBM DB2	20
Notas sobre Hibernate	21
Configuración de la red en las máquinas virtuales (VirtualBox)	24
Configuración del teclado en las máquinas virtuales	28
Modificar el tamaño del espacio de swap en la máquina virtual.....	29
Notas sobre Linux.....	30
Uso de Java en Sistema de 32 bits.....	31
Notas sobre Db4o	32

Ejemplo de uso de VARRAY en Oracle

```
CREATE OR REPLACE TYPE listaNombresTipo IS VARRAY(99999) OF
VARCHAR2(100);

/

CREATE TABLE Persona
    (dni NUMBER PRIMARY KEY,
    nombresDeAmigos listaNombresTipo);

INSERT INTO Persona VALUES (999999999, listaNombresTipo('Pedro', 'Laura'));

SELECT * FROM Persona;

SELECT nombresDeAmigos FROM Persona;


CREATE OR REPLACE PROCEDURE incorporarAmigo(dni IN NUMBER, nombreAmigo
IN VARCHAR2) IS
    variableAmigos listaNombresTipo := listaNombresTipo();
BEGIN
    EXECUTE IMMEDIATE
    'SELECT nombresDeAmigos FROM Persona WHERE dni = :1 FOR UPDATE OF
    nombresDeAmigos'
        INTO variableAmigos USING dni;
    variableAmigos.EXTEND(1); -- O, simplemente: variableAmigos.EXTEND;
    variableAmigos(variableAmigos.LAST) := nombreAmigo;
    EXECUTE IMMEDIATE
    'UPDATE Persona SET nombresDeAmigos = :1 WHERE dni = :2'
        USING variableAmigos, dni;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN --Si no hay una persona con ese DNI, se da de alta con
        el/la amigo/a indicado/a como parámetro.
        EXECUTE IMMEDIATE
        'INSERT INTO Persona VALUES (:1, listaNombresTipo(:2))' USING dni, nombreAmigo;
    END incorporarAmigo;

/
```

```
CALL incorporarAmigo(999999999, 'Juan');
```

```
CALL incorporarAmigo(999999999, 'Lucas');
```

```
--CALL incorporarAmigo(949999999, 'Lucas'); -- DNI no existente => se lanzará  
NO_DATA_FOUND y se insertará la persona.
```

Nota: Como comentamos en clase, usar un VARRAY no es la única opción. Una alternativa es utilizar tablas anidadas. Con VARRAY es necesario actualizar todos los elementos a la vez, tal y como se indica en la documentación de Oracle:

“Each varray is stored as a single object, either inside the table of which it is a column (if the varray is less than 4KB) or outside the table but still in the same tablespace (if the varray is greater than 4KB). You must update or retrieve all elements of the varray at the same time, which is most appropriate when performing some operation on all the elements at once. But you might find it impractical to store and retrieve large numbers of elements this way.”

(https://docs.oracle.com/cd/B14117_01/appdev.101/b10807/05_colls.htm)

Notas sobre Oracle

Libros de Oracle (documentación oficial):

- Libros para Oracle 9i: http://docs.oracle.com/cd/B10500_01/nav/docindex.htm.
- Libros para Oracle 18c: <https://docs.oracle.com/en/database/oracle/database/18/books.html>.

Conexión con sqlplus al servidor de Oracle del DIIS:

- **Oracle 9i Enterprise Sparc64 (9.0.1.3.0):**
 - sqlplus <nombreUsuario>@vicious.hendrix-oracle.cps.unizar.es
 - Host: hendrix-oracle.cps.unizar.es
 - SID: vicious
- **Oracle 18 (18.0.0):**
 - sqlplus <nombreUsuario>@barret.danae04.unizar.es
 - Host: danae04.unizar.es
 - SID: barret

Para más detalles, se puede consultar la siguiente URL:

<https://diis.unizar.es/WebEst/hendrix/#soracle>

Modelo objeto/relacional en Oracle:

- Oracle9i Application Developer's Guide - Object-Relational Features, Release 2 (9.2): http://docs.oracle.com/cd/B10500_01/appdev.920/a96594/toc.htm. Documentación correspondiente en PDF: http://docs.oracle.com/cd/B10500_01/appdev.920/a96594.pdf. Documentación asociada a la última versión (Oracle 11g): http://docs.oracle.com/cd/B28359_01/appdev.111/b28371.pdf.
- Oracle9i Application Developer's Guide - Object-Relational Features, Release 2 (9.2), Chapter 9 "A Sample Application Using Object-Relational Features" (http://docs.oracle.com/cd/B10500_01/appdev.920/a96594/adobjxmp.htm).
- Se puede encontrar una explicación detallada sobre los tipos de objeto PL/SQL (tipos completos, tipos incompletos empleados como referencia hacia adelante) en http://docs.oracle.com/cd/B10500_01/appdev.920/a96624/10_objs.htm (Oracle 9.2) y http://docs.oracle.com/cd/B28359_01/appdev.111/b28371/toc.htm.
- Nótese que el uso de valores de tipo REF en Oracle no es incompatible con la definición de una restricción de integridad referencial asociada. Por ejemplo, en Oracle9i Application Developer's Guide - Object-Relational Features, Release 2 (9.2), Chapter 9 "A Sample Application Using Object-Relational Features" (http://docs.oracle.com/cd/B10500_01/appdev.920/a96594/adobjxmp.htm) se puede ver algún ejemplo. Si no se añade la restricción de integridad referencial ni una cláusula SCOPE, la columna REF permitirá referenciar cualquier objeto fila. La diferencia entre una restricción de integridad referencial y una restricción de SCOPE es que esta última no implica una dependencia con respecto al objeto apuntado (por tanto, puede darse el caso de "DANGLING REF").
- Sobre los identificadores de objetos basados en clave primaria, puede consultarse http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_7002.htm#i2159502 (Oracle 11g) y http://docs.oracle.com/cd/B10501_01/server.920/a96540/statements_73a.htm#2095

[335](#) (Oracle 9i): "The OID_clause lets you specify whether the object identifier (OID) of the object table should be system generated or should be based on the primary key of the table. The default is SYSTEM GENERATED. [...] You cannot specify OBJECT IDENTIFIER IS PRIMARY KEY unless you have already specified a PRIMARY KEY constraint for the table."

- Sobre el soporte de colecciones de tipos de datos en Oracle, es interesante mirar el siguiente enlace:
http://docs.oracle.com/cd/B28359_01/appdev.111/b28371/adobjcol.htm (Oracle 11g), http://docs.oracle.com/cd/B10501_01/appdev.920/a96624/05_colls.htm (Oracle 9i).
- Para el acceso a información almacenada en una tabla sustituible de Oracle, es interesante tener en cuenta la funcionalidad ofrecida por la función TREAT. Se puede consultar información acerca de esta función en http://docs.oracle.com/cd/B28359_01/server.111/b28286/functions206.htm (Oracle 11g) y http://docs.oracle.com/cd/B10501_01/server.920/a96540/functions152a.htm (Oracle 9i). Ejemplo de sentencia recogida en el manual: SELECT name, TREAT(VALUE(p AS employee_t).salary salary FROM persons p;

Metadatos en Oracle:

- Algunos elementos útiles para consultar el catálogo/diccionario de la BD:
 - Prefijos de las vistas del diccionario de datos: USER (visión del usuario, lo que está en el esquema), ALL (visión del usuario expandida, lo que puede acceder el usuario), DBA (visión del administrador del sistema, lo que está en cualquier esquema). Se puede consultar la documentación para Oracle 11g (http://docs.oracle.com/cd/B28359_01/server.111/b28318/datadicti.htm) y Oracle 9i (http://docs.oracle.com/cd/B10500_01/server.920/a96524/c05dicti.htm).
 - Tabla USER_CONSTRAINTS / ALL_CONSTRAINTS: documentación para Oracle 11g (http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_5230.htm), http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_1044.htm) y para Oracle 9i (http://docs.oracle.com/cd/B10500_01/server.920/a96536/ch2517.htm, http://docs.oracle.com/cd/B10500_01/server.920/a96536/ch229.htm).
 - Tabla USER_OBJECTS (sinónimo OBJ) / ALL_OBJECTS: documentación para Oracle 11g (http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_5340.htm), http://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_1145.htm) y para Oracle 9i (http://docs.oracle.com/cd/B10500_01/server.920/a96536/ch2580.htm, http://docs.oracle.com/cd/B10500_01/server.920/a96536/ch290.htm).
- Lectura que puede resultaros de interés: "Reverse Engineering a Data Model Using the Oracle Data Dictionary", Eve Andersson (<http://www.eveandersson.com/writing/data-model-reverse-engineering>).

Bases de datos distribuidas en Oracle:

- Database Links en Oracle: documentación para Oracle 9i (http://docs.oracle.com/cd/B10501_01/server.920/a96521/ds_admin.htm) y Oracle 11g (http://docs.oracle.com/cd/E11882_01/server.112/e25494/ds_concepts002.htm).
- Managing a Distributed Database: documentación para Oracle 9i (http://docs.oracle.com/cd/B10501_01/server.920/a96521/ds_admin.htm) y Oracle 11g (http://docs.oracle.com/cd/E11882_01/server.112/e25494/ds_admin.htm).

Ejemplo de uso de vararray en Oracle (enlace externo):

- http://www.java2s.com/Tutorial/Oracle/0520_Collections/LoadingaTablewithaVARRAYintINSERTVALUEswithConstants.htm

Otras notas sobre Oracle:

- Si incluíis comentarios en el código con /* ... */ tenéis que aseguráros de dejar un espacio en blanco detrás del primer asterisco: "You must enter a space after the slash-asterisk(/*) beginning a comment, otherwise the comment is treated as a command, and the slash is interpreted as an execute command, executing any command in the SQL*Plus buffer.". Para más detalles, podéis consultar http://docs.oracle.com/cd/B10500_01/server.920/a90842/ch6.htm.
- Al utilizar SQL*Plus en hendrix es probable que al pulsar la tecla de borrado os aparezca ^? en pantalla en lugar de realizarse el borrado deseado. Para solucionarlo, podéis ejecutar lo siguiente desde sqlplus (donde dice "[TECLA-DE-BORRADO]" no tenéis que escribir esa cadena de caracteres, sino pulsar la tecla de borrado): !stty erase [TECLA-DE-BORRADO]
- Los esquemas en Oracle están asociados a usuarios ("Oracle Database automatically creates a schema when you create a user"). Podéis consultar la documentación de Oracle 9i (http://docs.oracle.com/cd/B10501_01/server.920/a96540/statements_614a.htm) y Oracle 11g (http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_6014.htm).
- Si al definir una vista o un trigger nos dice que se ha definido con errores de compilación y, sin embargo, no nos muestra ningún error al ejecutar "SHOW ERRORS", siempre es posible intentar utilizar la sintaxis completa de SHOW ERRORS para que nos muestre específicamente los errores ligados a ese objeto. Por ejemplo: "SHOW errors VIEW miVista". Para más detalles, puede consultarse el uso de SHOW en el manual de Oracle: versión 9i (http://docs.oracle.com/cd/B10501_01/server.920/a90842/ch13.htm#1013398) y versión 11g (http://docs.oracle.com/cd/B28359_01/server.111/b31189/ch12041.htm).
- Para formatear la salida de SQL*Plus es interesante consultar http://docs.oracle.com/cd/B28359_01/server.111/b31189/ch6.htm (Oracle 11g) y http://docs.oracle.com/cd/B10501_01/server.920/a90842/ch7.htm (Oracle 9i). En concreto, para fijar el tamaño de línea podéis utilizar SET LINESIZE (ejemplo: SET LINESIZE 60). También podéis fijar el tamaño utilizado para mostrar los valores de las columnas utilizando "COLUMN format" (ejemplo: COLUMN owner FORMAT A10).
- Puede resultar útil la utilización del comando "spool" para redireccionar los resultados en SQL PLUS. Podéis encontrar información en http://docs.oracle.com/cd/B10501_01/server.920/a90842/ch13.htm#1013669 (Oracle

- 9i) y http://docs.oracle.com/cd/B28359_01/server.111/b31189/ch12043.htm (Oracle 11g).
- Es posible añadir comentarios en Oracle para que queden recogidos como metadatos en el catálogo/diccionario de la base de datos. Para ello, se utiliza el comando "COMMENT". Para más detalles, se puede consultar http://docs.oracle.com/cd/B10501_01/server.920/a96540/statements_410a.htm#SQLRF01109 (Oracle 9i) y http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_4009.htm (Oracle 11g). Ejemplo: COMMENT ON COLUMN empleado.id_proy IS 'nombre abreviado del proyecto';
 - Si queremos utilizar comandos que muestren información en pantalla, como por ejemplo dbms_output.put_line('Ejemplo de mensaje');, habilitar la salida inmediata con SET SERVEROUTPUT ON (o dbms_output.enable(buffer_size => NULL);).

Driver JDBC para Oracle:

- Nombre del driver: oracle.jdbc.driver.OracleDriver
- Cadena de conexión: jdbc:oracle:thin:@hendrix-oracle.cps.unizar.es:1521:vicious

Cambio de password de Oracle en hendrix:

- El administrador ha puesto en marcha un interfaz web que permite a los usuarios cambiar su password de Oracle (sin necesidad de conocer el actual): <https://webdiis.unizar.es/cgi-bin/oracle-pw>

Para más detalles, se puede consultar la siguiente URL:

<https://diis.unizar.es/WebEst/hendrix/#soracle>

Tareas asíncronas en Oracle:

- Someter subprogramas para su ejecución asíncrona: *DBMS_JOB* (http://docs.oracle.com/cd/B10500_01/appdev.920/a96612/d_job.htm).
- Ver los trabajos en funcionamiento: vista *DBA_JOBS* (http://docs.oracle.com/cd/A91202_01/901_doc/server.901/a90117/jobq.htm, "Managing Job Queues").
- A partir de Oracle 10g es aconsejable utilizar *DBMS_SCHEDULER* en lugar de *DBMS_JOB* (http://docs.oracle.com/cd/B19306_01/server.102/b14231/jobtosched.htm, "Moving from DBMS_JOB to DBMS_SCHEDULER").
- Answers to "Why are my jobs not running?": <https://community.oracle.com/thread/648581>.

Instalación de Oracle en la Máquina Virtual Turnkey Linux de 32 bits:

Nota: Oracle XE 11g no está disponible para Linux de 32 bits, sólo para Linux de 64 bits.

Por tanto, habría que jugar con la versión de Oracle XE 10 para 32 bits:

https://oss.oracle.com/debian/dists/unstable/non-free/binary-i386/oracle-xe_10.2.0.1-1.1_i386.deb

Algunos comandos e ideas que pueden resultar de utilidad:

```
apt-get install bc
dpkg --install oracle-xe_10.2.0.1-1.1_i386.deb
/etc/init.d/oracle-xe configure
```

Otros que podrían resultaros de utilidad:

```
apt-get install libaio1
apt-get upgrade libaio
apt-get upgrade glibc
```

<<

Specify the HTTP port that will be used for Oracle Application Express [8080]: 8081

Specify a port that will be used for the database listener [1521]:

Specify a password to be used for database accounts. Note that the same password will be used for SYS and SYSTEM. Oracle recommends the use of different passwords for each database account. This can be done after initial configuration: ORA-PASS1

Do you want Oracle Database 11g Express Edition to be started on boot (y/n) [y]: n

Starting Oracle Net Listener...Done

Configuring database...Done

Starting Oracle Database 11g Express Edition instance...Done

Installation completed successfully.

>>

```
vi /usr/lib/oracle/xe/app/oracle/product/10.2.0/server/bin/nls_lang.sh
```

(

reemplazar:

```
if [[ -n "$LC_ALL" ]]; then
    locale=$LC_ALL
elif [[ -n "$LANG" ]]; then
    locale=$LANG
else
    locale=
fi
```

por:

```
if [ -n "$LC_ALL" ]; then
    locale=$LC_ALL
elif [ -n "$LANG" ]; then
    locale=$LANG
else
    locale=
```



```
fi  
)
```

vi /etc/profile:

```
(
```

añadir:

```
. /usr/lib/oracle/xe/app/oracle/product/10.2.0/server/bin/oracle_env.sh  
export PATH  
)
```

vi .bashrc

```
(
```

reemplazar:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

por:

```
PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  
)
```

./etc/profile

Conectarse como root (2 alternativas):

```
sqlplus sys/ORA-PASS1 as sysdba
```

```
sqlplus sys as sysdba
```

Ejemplo de test al conectarse: *select * from v\$version;*

A partir de ahí ya se debería poder hacer lo que se quiera: crear usuarios, crear bases de datos, etc.

Algunas instrucciones básicas para arrancar, parar, comprobar el estado, etc., del servidor:

```
/etc/init.d/oracle-xe start
```

```
/etc/init.d/oracle-xe stop
```

```
/etc/init.d/oracle-xe restart
```

```
/etc/init.d/oracle-xe status
```

Pero antes de nada hay que habilitar el servidor:

```
/etc/init.d/oracle-xe enable
```

De todos modos, se recomienda consultar los detalles y la documentación. Asimismo, deben revisarse las posibles restricciones de uso recogidas en la licencia correspondiente.

Notas sobre MySQL

Instrucciones de MySQL en Solaris (para lanzar y configurar MySQL en máquinas del laboratorio):

```
BASEDIR=$HOME/usr/web  
MYDIR=$BASEDIR/mysql  
mkdir -p $MYDIR
```

1.- Asegurarse de que se tiene mysql/mysqld en el PATH

```
hendrix01:~/ type mysql  
mysql is /opt/csw/bin/mysql  
hendrix01:~/ type mysqld_safe  
mysqld_safe is /opt/csw/bin/mysqld_safe
```

2.- Crear \$HOME/.my.cnf

Ajustar (al menos cambiar 6969 por otro valor, por ejemplo, basado en el UID, que puede obtenerse con "id <nombreUsuario>"; hay que tener en cuenta las limitaciones de rango de puertos que puedan existir):

```
$ cat > ~/.my.cnf <<-EOF  
[client]  
#password      = my_password  
port           = 6969  
socket         = /tmp/mysql.sock.6969  
[mysqld]  
port           = 6969  
socket         = /tmp/mysql.sock.6969  
#Since MySQL 5.5, dot not use "skip-locking" but "skip-external-locking".  
skip-external-locking  
#skip-networking  
#bind_address = 127.0.0.1  
# Start logging  
log  
EOF
```

(Nota: el EOF indica el final del fichero, lógicamente no debe escribirse en el fichero de configuración)

3.- Crear directorios

```
mkdir $MYDIR  
cd $MYDIR  
mkdir -p var bin ; chmod 700 var
```

4.- Ejecutar

(cambiar "foobar" por el password deseado)

PW=foobar

Instalar la BD:

```
mysql_install_db --datadir="${MYDIR}"/var
```

Lanzar el servidor:

```
nohup mysqld_safe --datadir="${MYDIR}"/var >/dev/null 2>&1 &  
(o mysqld_safe --datadir="${MYDIR}"/var &)
```

Asignar un password para root:

```
mysqladmin -u root password "$PW"
```

(

Nota: si el servidor MySQL está en remoto, entonces tenemos que escribir:

```
mysqladmin -u root -h `hostname` password "$PW"
```

Si está en local, podemos escribir:

```
mysqladmin -u root -h localhost password "$PW"
```

Pero no hace falta, basta con escribir:

```
mysqladmin -u root password "$PW"  
)
```

Nota: si quieres ver posibles mensajes en pantalla, entonces ejecuta lo siguiente para lanzar el servidor:

```
mysqld_safe --datadir="${MYDIR}"/var &
```

(

En este último caso, puedes obtener algo del estilo de:

```
hendrix01:~/usr/web/mysql/ mysqld_safe --datadir="${MYDIR}"/var  
)
```

5.- Trabajar

```
mysql -u root -p
```

(y luego introducir el password)

6.- Matar mysqld

```
mysqladmin -u root -p shutdown
```

(
para las siguientes sesiones, para arrancar el servidor, basta con ejecutar:

```
nohup mysqld_safe --datadir="${MYDIR}"/var >/dev/null 2>&1 &
```

o si se quiere ver la salida:

```
mysqld_safe --datadir="${MYDIR}"/var &
```

suponiendo que hemos guardado todo lo necesario en el .bashrc y que nuestro SHELL es /bin/bash
)

Notas adicionales:

- Las variables BASEDIR y MYDIR habría que almacenarlas en el fichero .bashrc para que estén disponibles en sesiones futuras.
- Los directorios anteriores son para Solaris (hendrix). Si estáis en Linux (máquinas locales de los laboratorios) necesitaréis cambiar los directorios de forma adecuada.
- En el fichero de configuración, es posible quitar el "#" que está comentando la línea de password e incluir ahí el password de cliente que se quiere utilizar; si no se hace esto, entonces conectarse directamente con "mysql -u root" no será posible y habrá que introducir "mysql -u -p" y luego indicar el password. No obstante, hay que tener en cuenta que no hay que declarar el password antes de ejecutar mysqladmin en el paso 4, ya que mysqladmin intentará conectarse con el password que esté definido en el fichero .my.cnf (si lo hay): dicho password no funcionará, ya que no se ha establecido todavía. La secuencia correcta de pasos, por tanto, sería: 1) no indicar inicialmente el password en el fichero .my.cnf; 2) ejecutar mysqladmin para fijar el password; y 3) ahora, si queréis, podéis definir el password que habéis fijado con mysqladmin en el fichero .my.cnf (esto es simplemente para que no tengáis que introducirlo cada vez).

Algunas notas sobre MySQL:

- Crear una instancia de BD: CREATE DATABASE databasename. Sintaxis: <http://dev.mysql.com/doc/refman/5.6/en/create-database.html>.
- En versiones previas a la 8.0.16, MySQL deja especificar *CHECK constraints*, pero las ignora. Para más detalles, se recomienda consultar el manual de referencia (ver <https://dev.mysql.com/doc/refman/8.0/en/create-table-check-constraints.html>).
- Con respecto a la restricción de integridad referencial, aunque siempre podemos especificar claves ajenas, la integridad referencial sólo se ofrece con el gestor de almacenamiento InnoDB. De acuerdo con el manual de referencia (ver <https://dev.mysql.com/doc/refman/8.0/en/ansi-diff-foreign-keys.html>): "For storage engines that do not support foreign keys (such as [MyISAM](#)), MySQL Server parses and ignores foreign key specifications."
- Dos gestores de almacenamiento populares en MySQL son InnoDB y MyISAM (ver <https://dev.mysql.com/doc/refman/8.0/en/create-table.html>, apartado "Storage Engines"): "InnoDB: Transaction-safe tables with row locking and foreign keys. The default storage engine for new tables. [...] MyISAM: The binary portable storage engine that is primarily used for read-only or read-mostly workloads.". Existe una opción ENGINE (en CREATE TABLE) para especificar el motor de almacenamiento para la tabla.

- No es posible ejecutar sentencias de control de transacciones desde dentro de un trigger (<https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html>): "The trigger cannot use statements that explicitly or implicitly begin or end a transaction, such as START TRANSACTION, COMMIT, or ROLLBACK. (ROLLBACK to SAVEPOINT is permitted because it does not end a transaction.).". El mecanismo que habilita MySQL para devolver errores (que supondrán que la transacción se aborte) es el uso de SIGNAL (ver <https://dev.mysql.com/doc/refman/8.0/en/signal.html>), por ejemplo: SIGNAL sqlstate '45000' set message_text = "¡Esto es un error!".
- Los manuales de referencia de MySQL están disponibles en <https://dev.mysql.com/doc/> y además la interfaz web facilita conmutar cada página de ayuda de una versión a otra, mediante la selección de la versión deseada en un desplegable.
- Una forma de terminar el servidor de MySQL (que había recomendado inicialmente nuestro administrador) es ejecutar los siguientes comandos:

```
ps -ef | grep mysqld  
kill -9 $PID
```

donde *\$PID* es el identificador del proceso correspondiente al servidor de MySQL (devuelto por "grep").

Sin embargo, en general, este método no se considera seguro ni recomendable. Por un lado, supone terminar el servidor de forma abrupta (puede ser un problema porque puede haber estructuras de datos y ficheros en uso, por ejemplo). Por otro lado, el "vigilante" (*watchdog*) de *mysqld_safe* detecta que "se ha caído" el servidor y vuelve a lanzar otro proceso servidor de MySQL (por eso tenemos que hacer varios "kill" de varios procesos *mysqld* distintos hasta que al final lo logramos). Por ello, he modificado las instrucciones que inicialmente proporcionó el administrador de sistemas. Para parar el servidor de forma "limpia" podemos ejecutar *mysqladmin -u root -p shutdown*.

Driver JDBC para MySQL:

- Nombre del driver: com.mysql.jdbc.Driver
- Cadena de conexión: jdbc:mysql://hendrix01.cps.unizar.es:<numPuerto>/<nombreBD>

Notas sobre PostgreSQL

Instrucciones de PostgreSQL en Solaris (para lanzar y configurar PostgreSQL en máquinas del laboratorio):

```
BASEDIR=$HOME/usr/web  
PGDIR=$BASEDIR/postgresql  
mkdir -p $PGDIR
```

1.- Asegurarse de tener psql en el path

```
$ type psql
```

(la salida será "psql is /usr/bin/psql" o "psql is /opt/csw/postgresql/bin/psql")

En caso de no estar, puede faltar "csw" en ~/.software

2.- Crear el fichero de configuracion y los directorios necesarios

```
cd $PGDIR  
mkdir etc  
mkdir -p var/pgdata bin ; chmod 700 var/pgdata  
cp /opt/csw/etc/postgresql.conf.csw $PGDIR/etc/postgresql.conf
```

Editar \$PGDIR/etc/postgresql.conf

```
PGDATA -> $PGDIR/var/pgdata
```

3.- Inicializar la base de datos

```
SERVEROPTS=""  
. $PGDIR/etc/postgresql.conf  
$PGINIT -D $PGDATA
```

4.- Asignar un puerto para el servidor de PostgreSQL (lo puedes incluir en el shell script de inicio para que siempre esté disponible):

```
export PGPORT=<numPuerto>
```

5.- Arrancar la base de datos

```
$PGCTL -D $PGDATA -l $PGDATA/postgresql.log start $SERVEROPTS
```

(comprobar en el fichero de log que no ha habido problemas: "more \$PGDATA/postgresql.log")

6.- Trabajar

Ejemplo

```
$ psql -l
      List of databases
  Name | Owner | Encoding
-----+-----+-----
postgres | spd | LATIN1
template0 | spd | LATIN1
template1 | spd | LATIN1
(3 rows)

$ psql postgres
```

(Nota: para salir \q, para ayuda \?)

7.- Parar la base de datos

```
$PGCTL -D $PGDATA -l $PGDATA/postgresql.log stop
```

(
para las siguientes sesiones, para arrancar el servidor, basta con ejecutar:

```
$PGCTL -D $PGDATA -l $PGDATA/postgresql.log start $SERVEROPTS
```

suponiendo que hemos guardado todo lo necesario en el .bashrc y que nuestro SHELL es /bin/bash
)

Notas adicionales:

- Las variables BASEDIR y PGDIR habría que almacenarlas en el fichero .bashrc para que estén disponibles en sesiones futuras.
- Las sentencias "SERVEROPTS="" y ". \$PGDIR/etc/postgresql.conf" se pueden almacenar en el fichero .bashrc para que estén disponibles en sesiones futuras.
- Los directorios anteriores son para Solaris (hendrix). Si estáis en Linux (máquinas locales de los laboratorios) necesitaréis cambiar los directorios de forma adecuada. Si pg_ctl (necesario para arrancar y parar PostgreSQL) no estuviera disponible, tendríais que utilizar Solaris (hendrix-ssh).
- Cada alumno/a tiene que utilizar un número de puerto distinto, para evitar conflictos: podéis utilizar algo basado en el NIP o en el identificador de usuario devuelto por "id <nombreUsuario>" (siempre teniendo en cuenta el rango de puertos permitido). Si obtenéis un error diciendo que el puerto indicado está fuera del rango permitido, podéis introducir un número de puerto distinto "al azar". En general, debéis comprobar en el fichero de log, tras arrancar el servidor, que no ha habido problemas al abrir el puerto.
- En algunos casos hemos observado el siguiente aviso en el fichero de log: "LOG: could not bind IPv6 socket: Address already in use HINT: Is another postmaster already running on port 5432? If not, wait a few seconds and retry.". Si posteriormente aparece "LOG: database system is ready to accept connections" en el fichero de log, entonces no hay problema: no es un error producido por intentar utilizar un puerto que ya está

en uso. Al parecer se debe a que se intenta utilizar IPv6 primero (y falla) y luego IPv4 (y tiene éxito).

Algunas notas sobre PostgreSQL:

- La versión del servidor de PostgreSQL disponible en el laboratorio la podéis obtener ejecutando "select version();" desde el prompt de psql. Esto es importante para poder consultar el manual correspondiente a nuestra versión (y así poder verificar con precisión, por ejemplo, el soporte para orientación a objetos con el que contamos). De forma similar, la versión del cliente de PostgreSQL la podemos obtener ejecutando "psql --version".
- Crear un esquema de BD: CREATE SCHEMA schemaname. Sintaxis: <http://www.postgresql.org/docs/8.3/static/sql-createschema.html>.
- Es interesante mirar el documento siguiente: *Chapter 5. Data Definition, Section 5.8 Inheritance* de PostgreSQL 8.1 (<http://www.postgresql.org/docs/8.1/static/ddl-inherit.html>); documentación asociada a la última versión 9.2 en <http://www.postgresql.org/docs/9.2/static/ddl-inherit.html>.
- Todo lo que escribáis en el prompt de PostgreSQL se encola hasta que se finaliza con un ";" (por tanto, nuestra entrada puede constar de múltiples líneas). Lo que ocurre es que si nos equivocamos en la entrada podemos terminar teniendo en el buffer (por descuido) algo que no deseamos (llevando posteriormente a mensajes de error inesperados). Los siguientes comandos son útiles para controlar el buffer: \p (muestra el contenido del buffer), \e (edita el contenido del buffer con el editor definido en la variable de entorno EDITOR), y \r (limpia el buffer).

Driver JDBC para PostgreSQL:

- Nombre del driver: org.postgresql.Driver
- Cadena de conexión: jdbc:postgresql://hendrix-ssh.cps.unizar.es:<puerto>/postgres
- Para el acceso a PostgreSQL a través de JDBC, por defecto se realiza una conexión validada. Esto requeriría establecer un certificado para que el cliente pueda autenticar al servidor. No obstante, podemos realizar una conexión no validada especificando el parámetro de conexión sslfactory=org.postgresql.ssl.NonValidatingFactory; por ejemplo, la URL de conexión podría ser `jdbc:postgresql://hendrix-ssh.cps.unizar.es:<puerto>/postgres?sslfactory=org.postgresql.ssl.NonValidatingFactory` y (donde *puerto* es el número de puerto donde está escuchando el servidor). Para más detalles, puede consultarse la documentación de JDBC para PostgreSQL (<http://jdbc.postgresql.org/>), y en concreto el Capítulo 4.A "Using SSL"- "Configuring the Client" (<http://jdbc.postgresql.org/documentation/head/ssl-client.html>).

Como se indica en dicha página:

Unlike psql and other libpq based programs the JDBC driver does server certificate validation by default. This means that when establishing a SSL connection the JDBC driver will validate the server's identity preventing "man in the middle" attacks. It does this by checking that the server certificate is signed by a trusted authority. If you have a

certificate signed by a global certificate authority (CA), there is nothing further to do because Java comes with copies of the most common CA's certificates. If you are dealing with a self-signed certificate though, you need to make this available to the Java client to enable it to validate the server's certificate.

[...]

In some situations it may not be possible to configure your Java environment to make the server certificate available, for example in an applet. For a large scale deployment it would be best to get a certificate signed by recognized certificate authority, but that is not always an option. The JDBC driver provides an option to establish a SSL connection without doing any validation, but please understand the risk involved before enabling this option.

A non-validating connection is established via a custom SSLSocketFactory class that is provided with the driver. Setting the connection URL parameter sslfactory=org.postgresql.ssl.NonValidatingFactory will turn off all SSL validation.

Bases de datos distribuidas en PostgreSQL:

- *dblink*: <http://www.postgresql.org/docs/9.3/static/dblink.html> (PostgreSQL 9.3).
- *dblink_connect*: <http://www.postgresql.org/docs/9.3/static/contrib-dblink-connect.html> (PostgreSQL 9.3).

Notas generales sobre Caché

El siguiente texto es aplicable en caso de que se haya instalado Caché en el laboratorio para el curso académico vigente.

- Antes de utilizar Caché, cada grupo debe pedir al profesor un usuario y un password para conectarse (el password deberá cambiarse la primera vez que se realice una conexión al servidor de Caché).
- Para usar Caché:

- Vía csession:

Entrar por ssh en hendrix.

Asegurarse de tener en \$HOME/.software la palabra "CACHE".

Ejecutar:

`csession CACHE`

- Vía HTTP:

Conectarse desde un cliente web a:

`http://hendrix.cps.unizar.es:57772/csp/sys/UtilHome.csp`

- Via Cache Studio:

Servidor preferido -> `hendrix01.cps.unizar.es`

- Caché Terminal ("csession CACHE"):

- Path al comando csession:

`/usr/local/CACHE/bin/csession`

- Entrar al intérprete de SQL:

`DO $SYSTEM.SQL.Shell()`

- Salir del intérprete de SQL:

`quit`

- Salir del terminal:

`Halt`

- Caché Studio:

- Start the New Class wizard by selecting File > New > General tab.
- Select Class > Add > Property to start the New Property wizard.
- Build > Compile.
- System Management Portal:
 - Browse SQL Schemas ([Home] > [SQL] > [Schemas])
 - Browse Classes ([Home] > [System] > [Classes])
- El espacio de nombres asignado al alumnado de la asignatura se denomina ASIGNATURA. Para evitar interferencias, se recomienda utilizar esquemas distintos dentro de dicho espacio de nombres.
- Caché Studio es un cliente de Caché disponible sólo para Windows. Está instalado en el laboratorio. Para utilizarlo, tenéis que arrancar los equipos en Windows.
- Documentación de Caché:
<http://docs.intersystems.com/cache20122/csp/docbook/DocBook.UI.Page.cls>.
- "Using Caché ObjectScript" (versión 2012.2):
<http://docs.intersystems.com/documentation/cache/20122/pdfs/GCOS.pdf>.

Descarga de Caché como alumno/a:

Información proporcionada por José Tomás Salvador (InterSystems Iberia):

<<

deben bajarse la licencia gratuita disponible en la web corporativa.

<https://developer.intersystems.com/start/>

Siempre está disponible la última versión liberada.

Es una instalación completa, con todas las características habilitadas. La única restricción es que es para un solo usuario.

>>

Enlaces sobre Caché:

- [Descarga de Caché](#)
- [Página de InterSystems \(.com\)](#)
- [InterSystems' Video Library](#)
- [Globals en la página de InterSystems](#)
- [Página de InterSystems \(.es\)](#)

Notas sobre IBM DB2

Se indican algunos enlaces de interés:

- DB2 Express-C database server. DB2 Express-C for Linux:
https://www14.software.ibm.com/webapp/iwm/web/pick.do?source=swg-db2expressc&S_CMP=ECDDWW01&S_TACT=DB2x_BDU.
- DB2 Express-C 10.1 (Linux 32 bit) https://www-01.ibm.com/marketing/iwm/iwm/web/pick.do?source=swg-db2expressc&lang=es_ES&S_CMP=ECDDWW01&S_TACT=AC_04_ES
- IBM Data Studio: <http://www-03.ibm.com/software/products/es/data-studio>.
- Getting started with DB2 for Linux, UNIX, and Windows:
<http://www.ibm.com/developerworks/data/newto/db2luw-getstarted.html>.
- Getting Started with DB2 Express-C (PDF Book):
ftp://ftp.software.ibm.com/software/data/db2/express-c/wiki/GettingStartedwithDB2Express-C_V95-en.pdf.
- DB2 Express-C 10.1: <https://jazz.net/downloads/DB2/releases/10.1>

Notas sobre Hibernate

Ejemplo de fichero persistence.xml:

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence\_2\_0.xsd version="1.0">

  <persistence-unit name="EjemploDeUnidadDePersistencia" transaction-
type="RESOURCE_LOCAL">

    <provider>org.hibernate.ejb.HibernatePersistence</provider>

    <class>entidades.Cliente</class>

    <properties>

      <property name="hibernate.show_sql" value="true"/>

      <property name="hibernate.dialect" value="org.hibernate.dialect.Oracle9iDialect"/>

      <property name="javax.persistence.jdbc.driver"
value="oracle.jdbc.driver.OracleDriver"/>

      <property name="javax.persistence.jdbc.url" value="jdbc:oracle:thin:@hendrix-
oracle.cps.unizar.es:1521:vicious"/>

      <property name="javax.persistence.jdbc.user" value="pon el nombre de usuario"/>

      <property name="javax.persistence.jdbc.password" value="pon el password"/>

    </properties>

  </persistence-unit>

</persistence>
```

Esquema de un ejemplo de consulta con Criteria API y el operador IN (subconsulta):

```
/*
 * Supongamos que existe una clase Cliente que tiene un atributo "apellidos" y una clase
ApellidoCliente que tiene un atributo "apellido".
 * Queremos obtener todos los clientes siempre y cuando el valor de su atributo "apellidos"
coincida con el valor de "apellido" en alguna instancia de ApellidoCliente.
 * Es decir, en SQL queremos algo de este estilo:
 *
 * select * from CLIENTE_E where APELLIDOS in (select APELLIDO from APELLIDO_E);
 *
 * (suponiendo ciertas correspondencias entre el modelo de clases en Java y las tablas de la BD
(por ejemplo, que la tabla donde se almacenan los datos de los
 * clientes se llama CLIENTE_E).
```

*/

```
EntityManagerFactory entityManagerFactory
= Persistence.createEntityManagerFactory("EjemploDeUnidadDePersistencia");
EntityManager em = entityManagerFactory.createEntityManager();
CriteriaBuilder cb = em.getCriteriaBuilder();

CriteriaQuery<Cliente> cq = cb.createQuery(Cliente.class);
Root<Cliente> from = cq.from(Cliente.class);
Path<Object> path_a_apellidos_de_cliente = from.get("apellidos");
CriteriaQuery<Cliente> select = cq.select(from);

Subquery<ApellidoCliente> subconsulta = cq.subquery(entidades.ApellidoCliente.class);
Root fromApellido = subconsulta.from(entidades.ApellidoCliente.class);
subconsulta.select(fromApellido.get("apellido"));

select.where(cb.in(path_a_apellidos_de_cliente).value(subconsulta));
TypedQuery<Cliente> q = em.createQuery(select);

List<Cliente> results = q.getResultList();
for (Cliente clienteActual:results)
{
    System.out.println("Id. del cliente: " + clienteActual.getIdCliente());
    System.out.println("Nombre del cliente: " + clienteActual.getNombre());
    System.out.println("Apellidos del cliente: " + clienteActual.getApellidos());
    System.out.println("Tipo del cliente: " + clienteActual.getTipoCliente());
}

em.close();
entityManagerFactory.close();
```

Notas sobre Hibernate:

- Al bajarse Hibernate, la distribución de Hibernate incluye la versión de antlr que necesita (por ejemplo, para Hibernate 4.1.12 es la versión 2.7.7 de antlr).
- Como ejemplo, se indica el enlace de descarga de la versión de Hibernate 4.1.12: <http://sourceforge.net/projects/hibernate/files/hibernate4/4.1.12.Final/hibernate-release-4.1.12.Final.zip>. En caso de duda, bajar este fichero, ya que contiene las librerías necesarias. En cualquier caso, es recomendable considerar la última versión estable de Hibernate.
- Explicación de los posibles valores de `hibernate.hbm2ddl.auto` en Stack Overflow: <http://stackoverflow.com/questions/438146/hibernate-hbm2ddl-auto-possible-values-and-what-they-do>.
- Para la creación automática del esquema podéis incluir la siguiente propiedad para la unidad de persistencia en el fichero `persistence.xml`: `<property name="hibernate.hbm2ddl.auto" value="create"/>`.

- Si además deseáis información acerca de las sentencias SQL ejecutadas, debéis incluir también la siguiente propiedad: `<property name="hibernate.show_sql" value="true"/>`.
- En algunos casos, puede ser interesante especificar en el fichero de configuración (persistence.xml) el dialecto de SQL a utilizar (por ejemplo, con `<property name="hibernate.dialect" value="org.hibernate.dialect.Oracle9iDialect"/>`).
- Tutorial de Java EE 6 sobre el uso del Criteria API:
<http://docs.oracle.com/javaee/6/tutorial/doc/gjiv.html>.
- Criteria API para Java EE 6:
<http://docs.oracle.com/javaee/6/api/index.html?javax/persistence/criteria/package-summary.html>.

Configuración de la red en las máquinas virtuales (VirtualBox)

Necesitas configurarla como NAT con mapeo de puertos.

Para el mapeo de puertos en VirtualBox:

Network->Adapter 1->Enable Network Adapter
Attached to NAT
despliegas "Advanced" te aparece un botón "Port Forwarding".

Ahí puedes añadir, por ejemplo:

Name: sshd
Protocol: TCP
Host IP: La del host. Alternativamente puedes usar 127.0.0.1 y así no expones la máquina virtual al exterior.

Host Port: En el caso de ssh, para evitar conflicto con el del sistema, puedes poner por ejemplo 2222.

Guest IP: La de la máquina virtual
Guest Port: 22

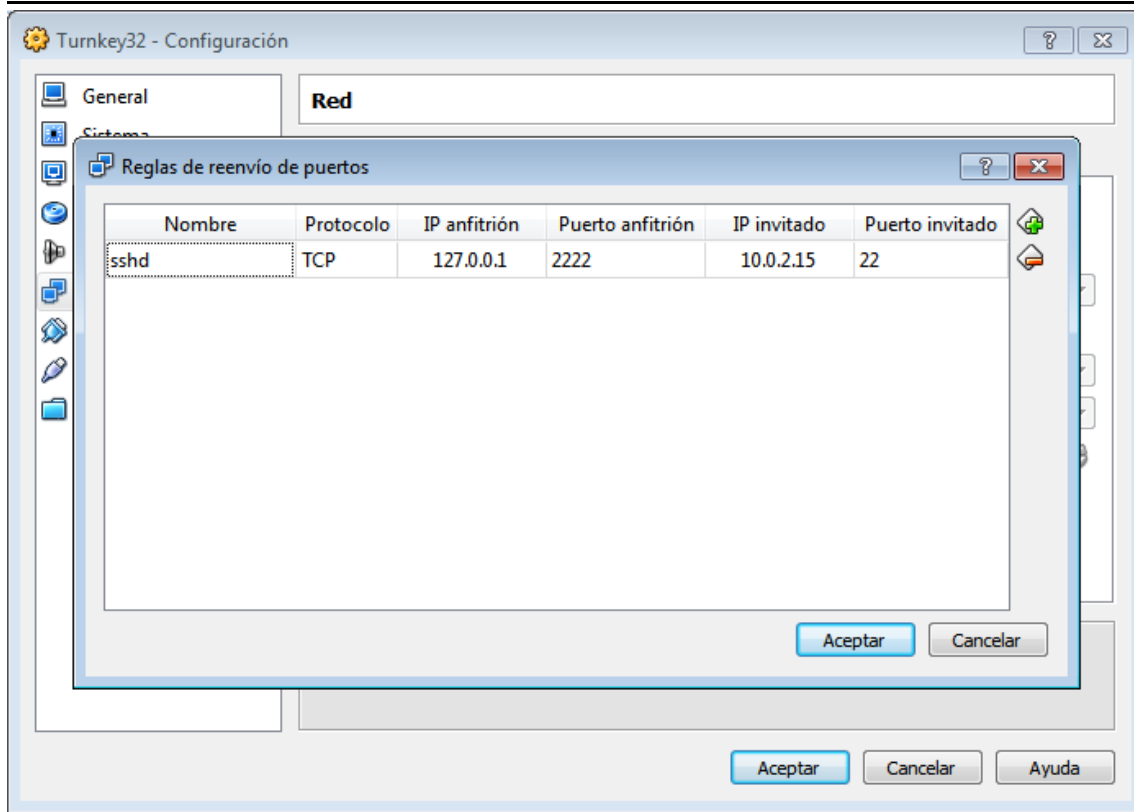
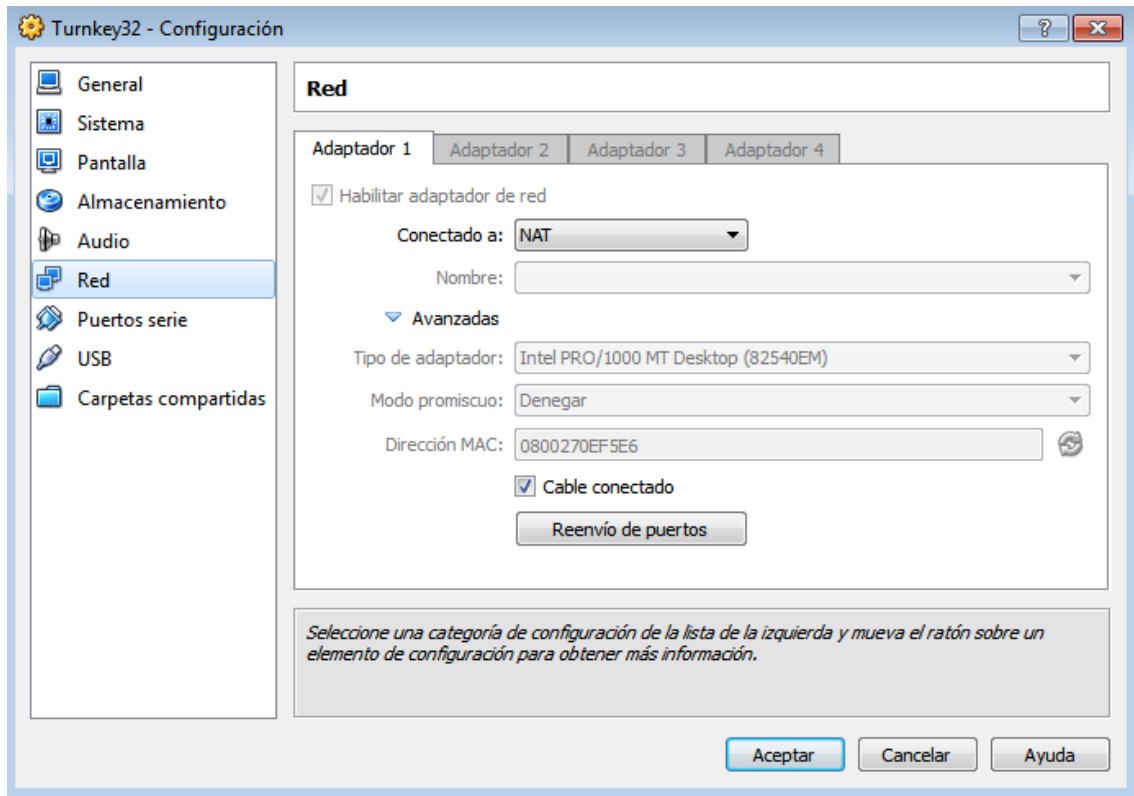
A partir de ese momento, para conectarte a la máquina virtual ya podrás hacer, por ejemplo:

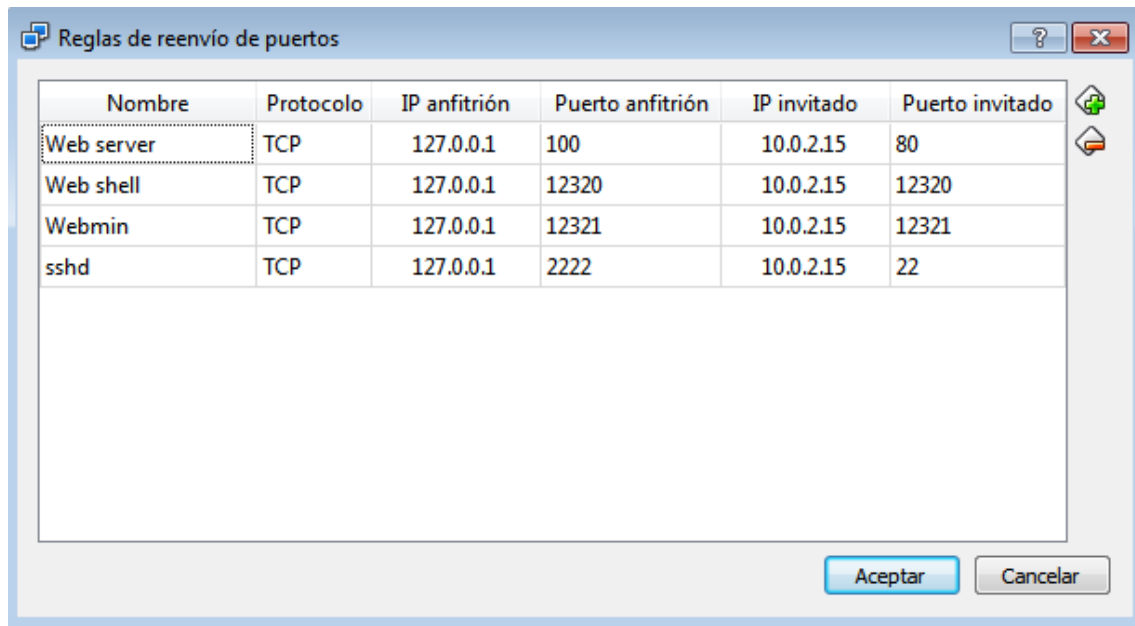
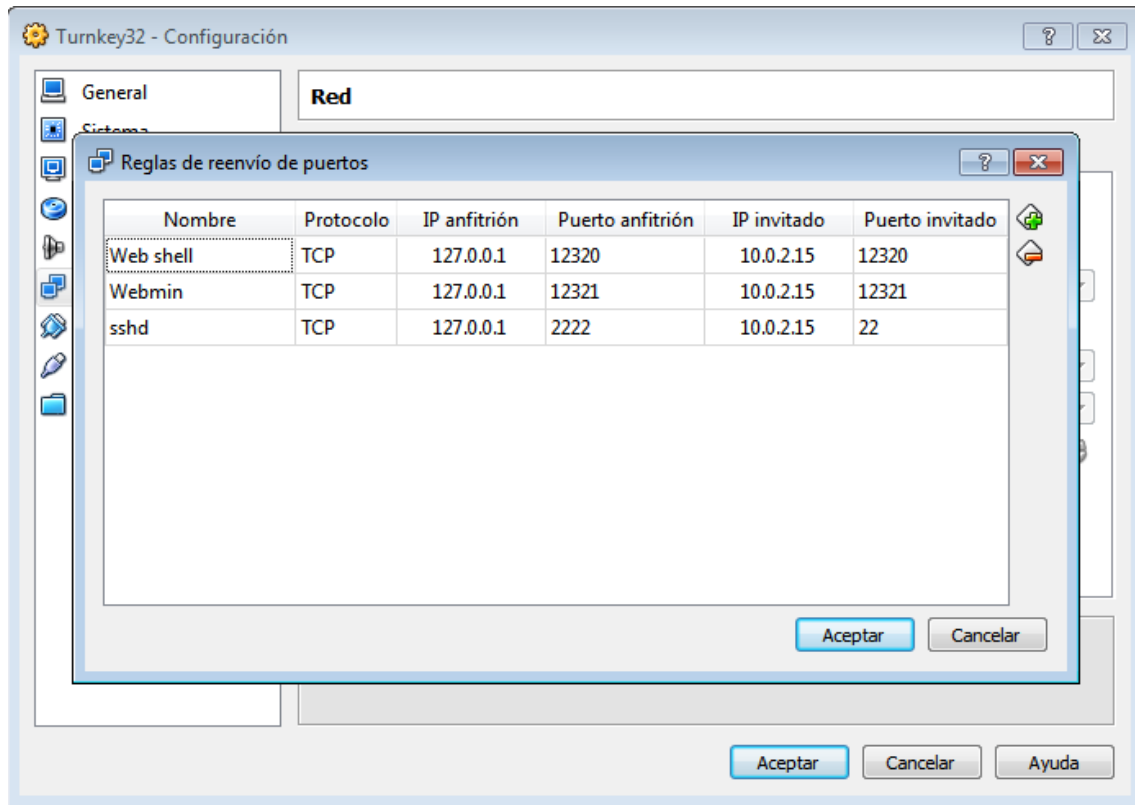
```
ssh -p 2222 localhost
```

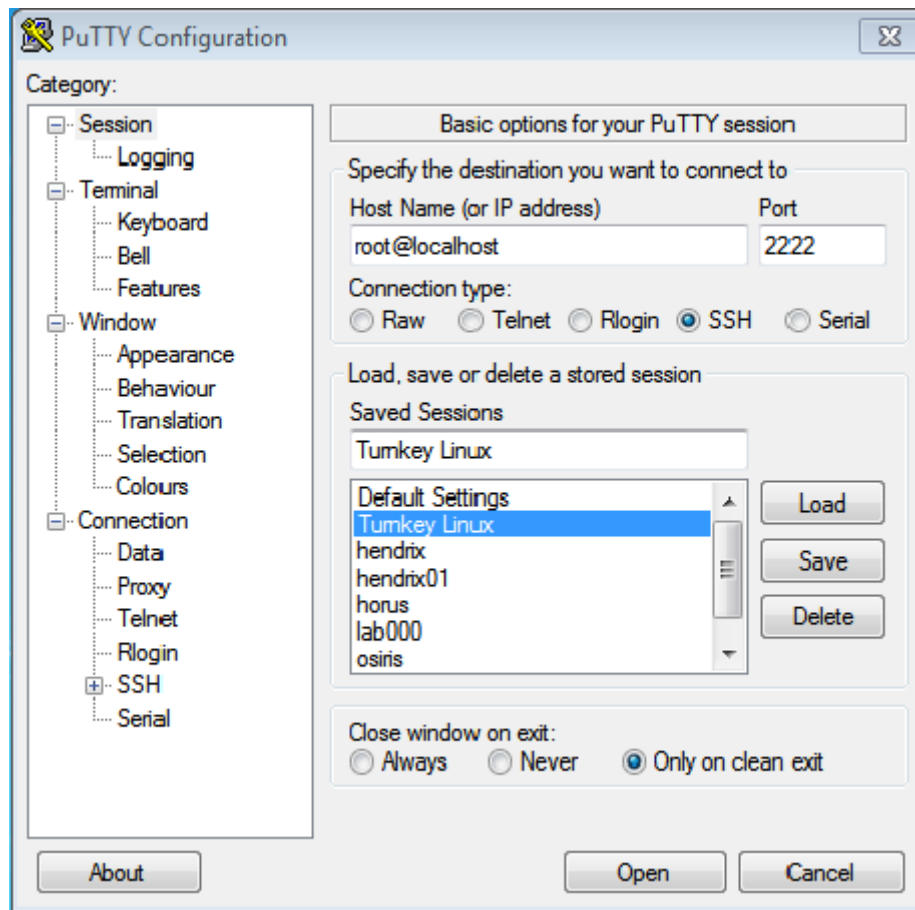
Para otros servicios, habrá que poner los puertos correspondientes, en lugar de 2222. Al no haber corriendo servicios similares en el host, se puede poner el mismo puerto en Host Port y en Guest Port.

Agradecemos estas instrucciones a nuestro administrador de sistemas (J.A. Gutiérrez), que habló con nosotros sobre la mejor forma de trabajar con VirtualBox en el laboratorio.

Algunas capturas de pantalla de ejemplo, a modo de orientación:







Configuración del teclado en las máquinas virtuales

Algunos comandos que os pueden ayudar a configurar el teclado... (elegid los aplicables):

```
apt-get update  
apt-get upgrade  
(update and upgrade)
```

```
apt-get install keyboard-configuration  
(select Spanish keyboard)
```

```
dpkg-reconfigure keyboard-configuration
```

```
apt-get install console-data
```

```
dpkg-reconfigure console-data  
(select "pc / qwerty / Spanish / Standard / Standard")
```

```
apt-get install console-common
```

```
install-keymap /usr/share/keymaps/i386/qwerty/es.kmap.gz
```

```
apt-get install console-setup
```

Hay que tener en cuenta que también es importante la configuración del terminal desde el que se accede. En principio es mejor entrar a la máquina virtual con un ssh en lugar de utilizar directamente la ventana que saca VirtualBox al arrancarla, ya que de ese modo nos podemos aprovechar de todas las opciones de configuración de dicho terminal (teclado, posibilidad de realizar copy+paste, etc.).

Modificar el tamaño del espacio de swap en la máquina virtual

Oracle necesita un tamaño de swap mínimo. De otro modo, la instalación fallará. Este problema os puede aparecer o no dependiendo de la disponibilidad de memoria RAM.

Podéis consultar las siguientes fuentes:

- <http://www.turnkeylinux.org/forum/support/20130813/setting-swap-partition-size>
- <http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=254>

Comandos:

```
dd if=/dev/zero of=/mnt/fichero_swap1 bs=1M count=1024  
mkswap /mnt/fichero_swap1  
swapon /mnt/fichero_swap1
```

Notas sobre Linux

- Recuerda que ejecutar un shell script (`./my_script.conf`), habiendo dado permisos de ejecución al script) creará un nuevo shell para ejecutar los comandos allí. Por el contrario, si utilizamos "sourcing" (`./my_script.conf` o `source ./my_script.conf`) se ejecutarán los comandos en el shell actual. Por tanto, debemos escoger esta segunda opción si lo que queremos es fijar el valor de ciertas variables de entorno. Para más detalles, podéis consultar la siguiente URL: <http://superuser.com/questions/176783/what-is-the-difference-between-executing-a-bash-script-and-sourcing-a-bash-script>.
- Si os conectáis de forma remota a otra máquina (por ejemplo, `hendrix-ssh`) utilizando `ssh`, es interesante utilizar la opción `-X` para redireccionar la salida gráfica (así podréis utilizar programas como `nedit`, `gedit`, etc., que tienen interfaz gráfica).

Uso de Java en Sistema de 32 bits

Si el laboratorio tiene SO de 64 bits, y utilizáis Java de 32 bits, no encontrará las librerías gráficas de 32 bits en el sistema. La solución es cambiar "java" por "java64" en ~/.software.

Ejemplo de problema si utilizáis la instalación de Java incorrecta:

Exception in thread "main" java.lang.

UnsatisfiedLinkError:

/usr/local/pkg/jdk1.6.0_35/jre/lib/i386/xawt/libmawt.so: libXtst.so.6:

cannot open shared object file: No such file or directory

at java.lang.ClassLoader\$ NativeLibrary.load(Native Method)

at java.lang.ClassLoader. loadLibrary0(ClassLoader.java: 1807)

at java.lang.ClassLoader. loadLibrary(ClassLoader.java: 1703)

at java.lang.Runtime.load0(Runtime.java:770)

at java.lang.System.load(System. java:1003)

at java.lang.ClassLoader\$ NativeLibrary.load(Native Method)

at java.lang.ClassLoader. loadLibrary0(ClassLoader.java: 1807)

at java.lang.ClassLoader. loadLibrary(ClassLoader.java: 1724)

at java.lang.Runtime. loadLibrary0(Runtime.java:823)

at java.lang.System.loadLibrary(System.java:1028)

at sun.security.action. LoadLibraryAction.run(LoadLibraryAction.java:50)

at java.security. AccessController.doPrivileged(Native Method)

at sun.awt.NativeLibLoader. loadLibraries(NativeLibLoader. java:38)

at sun.awt.DebugHelper.<clinit>(DebugHelper.java:29)

at java.awt.Component.<clinit>(Component.java:566)

Could not find the main class: ventanas.VPrincipal. Program will exit.

Notas sobre Db4o

Enlaces interesantes:

- Db4o for Java:
 - <http://supportservices.actian.com/versant/default.html>
 - <http://sourceforge.net/projects/db4o/>

Tenéis disponible en Moodle un fichero de ejemplo con código Java para acceder a Db4o: db4o-examples.zip.

- <http://www.odbms.org/wp-content/uploads/2013/11/db4o-7.10-tutorial-java.pdf>
- <http://community.versant.com/documentation/reference/db4o-8.0/java/tutorial/docs/FirstGlance.html#outline2> (enlace ya no disponible)

<https://web.archive.org/web/20140922110425/http://community.versant.com/documentation/reference/db4o-8.0/java/tutorial/docs/FirstGlance.html>

- http://community.versant.com/documentation/reference/db4o-8.0/java/reference/Content/basics/querying/query_by_example/qbe_limitations.htm (enlace ya no disponible)

https://web.archive.org/web/20131224203425/http://community.versant.com/documentation/reference/db4o-8.0/java/reference/Content/basics/querying/query_by_example/qbe_limitations.htm