

**Universidad Autónoma Metropolitana**  
**Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación

Recopilador de configuración y estado de un manejador de base de datos relacional

**Alumna: Guadalupe Becerril Velázquez**  
**Matrícula: 204242119**

**Alumno: Iván Ageo Gutiérrez Limón**  
**Matrícula: 204202347**

Trimestre Lectivo: 2013 Primavera  
Fecha: 16 de julio de 2013

Asesor: M. en C. Hugo Pablo Leyva  
Categoría: Profesor Titular en la División de Ciencias Básicas e Ingeniería de la UAM  
Azcapotzalco.

## Tabla de Contenido

Objetivo General .....	3
Objetivos Específicos .....	3
Resumen .....	3
Introducción .....	4
Descripción técnica .....	4
Especificación Técnica.....	7
Instalación .....	8
MySQL v5.2.....	9
Requerimientos .....	9
Proceso de Instalación para el Sistema Operativo Windows XP edición familiar v2002 service pack 3 .....	9
Proceso de Instalación para Ubuntu 13.04 GNU/Linux.....	12
PostgreSQL 9.2 y PostgreSQL 8.4.17.....	14
Requerimientos .....	14
Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 .....	14
Proceso de Instalación para Ubuntu 13.04 GNU/Linux.....	19
Oracle database 11g express edition release v11.2.0.2.0 .....	26
Requerimientos .....	26
Proceso de Instalación para el Sistema Operativo Windows XP edición familiar v2002 service pack 3 .....	26
Proceso de Instalación para Ubuntu 13.04 GNU/Linux.....	31
DB2 express-c v1.01 y DB2 express-c v10.5.....	34
Requerimientos .....	34
Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 y Ubuntu 13.04 GNU/Linux.....	35
NetBeans IDE 7.3.1.....	35
Requerimientos .....	35
Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 y Ubuntu 13.04 GNU/Linux.....	36
iTextpdf v5.3.4.....	37

Desarrollo del proyecto.....	37
Pasos para instalar este proyecto en NetBeans IDE 7.3.1 en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 .....	37
Pasos para instalar este proyecto en NetBeans IDE 7.3.1 en Ubuntu 13.04 GNU/Linux .....	44
Manual de Usuario .....	47
Introducción .....	47
Objetivos .....	47
Guía de uso.....	47
Apéndice .....	54
Usuarios de las Bases de Datos .....	54
Base de Datos utilizada para las pruebas.....	59
Script de la base de datos collections de MySQL.....	59
Script de la base de datos collections de PosrtgreSQL.....	61
Script de la base de datos collections de DB2.....	62
Script de la base de datos collections de Oracle .....	64
Estructura de la base de datos de prueba “collections” .....	66
Salida de una corrida de la aplicación .....	67
Código fuente .....	79
Conclusiones .....	92
Referencias y Bibliografía.....	92

## Objetivo General

Desarrollar un programa que permita visualizar un reporte de la información de la configuración y estado de los manejadores de bases de datos relacionales: MySQL[1], PostgreSQL[2], DB2[3] y Oracle[4] de forma fácil.

## Objetivos Específicos

- Investigar los diferentes parámetros de configuración para cada uno de los manejadores de bases de datos relacionales mencionados anteriormente.
- Crear un módulo de conexión a las bases de datos relacionales para los diferentes manejadores.
- Crear un módulo de recopilación de información del contenido y parámetros de configuración para los diferentes manejadores de bases de datos relacionales.
- Crear un módulo de reportes donde se visualice la información recopilada del módulo anterior.
- Unir los módulos antes mencionados.

## Resumen

En la actualidad, las organizaciones a menudo emplean un sin número de plataformas de base de datos en su arquitectura de sistemas de información.

No es raro ver a medianas y grandes empresas utilizando de tres a cuatro diferentes paquetes de RDBMS[5] en donde es necesario obtener cierta información indispensable tanto de los parámetros de cada manejador de base de datos que se está empleando como de su esquema general, es decir, de la información contenida que en éste se encuentra.

Pero cada manejador de base de datos tiene diferentes características de configuración y de igual forma diferentes comandos para poder tener acceso a la información contenida. Por esta razón se considera de gran utilidad contar con una aplicación que permita obtener información detallada de la base de datos, es decir, sobre sus campos, registros, tablas, valores de configuración tal como el espacio en disco utilizado, la ruta o path, parámetros de seguridad y toda información que necesita conocer la empresa u organización para cada uno de los manejadores de bases de datos relacionales que utiliza. Por otra parte, dicha información se visualizará en un archivo con formato PDF (por sus

siglas en inglés: *Portable Document Format*, Formato de Documento Portátil) por lo que podrá servir también como un respaldo de la misma.

## **Introducción**

En cualquier actividad productiva en la que estemos inmersos, día a día debemos tomar decisiones que indicarán el rumbo de nuestra empresa, ya sea hacia el éxito o al fracaso. Pero para tomar una decisión lo más acertada posible es necesario basarnos en información que esté organizada y controlada como se encuentra en una base de datos.

Para poder usar las bases de datos debemos conocer una herramienta que nos permita almacenar y consultar información, estas herramientas son llamadas Sistemas Manejadores de Bases de Datos Relacionales (RDBMS).

Actualmente es necesario contar con un programa o con una aplicación que nos permita conocer de manera sencilla los parámetros de configuración y el esquema general de una base de datos, independientemente del manejador de base de datos que se esté utilizando. Debido a que comúnmente se requiere conocer la estructura de cada base de datos para que el usuario pueda interpretar la información o manipularla de acuerdo a las necesidades que tenga.

Es aquí donde entra en funcionamiento este proyecto, en el cual se desarrollará un programa que permitirá mostrar un reporte donde se contendrán informes detallados acerca de los parámetros de configuración y del esquema general de los diferentes manejadores de bases de datos relacionales tales como: MySQL, PostgreSQL, DB2 y Oracle.

## **Descripción técnica**

Se cuenta con un módulo de conexión el cual se encarga de conectar a la base de datos existente. Este módulo de conexión funciona independientemente de en qué RDBMS se encuentre la base de datos.

También contiene un módulo de recopilación, el cual realiza el análisis y la recolección de la información de la configuración y esquema general de las bases de datos según los estándares específicos de cada manejador. Esto se realiza con un programa que está codificado en JAVA[6].

De igual modo, se cuenta con un módulo de reportes, el cual permite mostrar la información del módulo de recopilación en formato PDF.

Para mayor claridad, este proceso se muestra en la Figura 1.

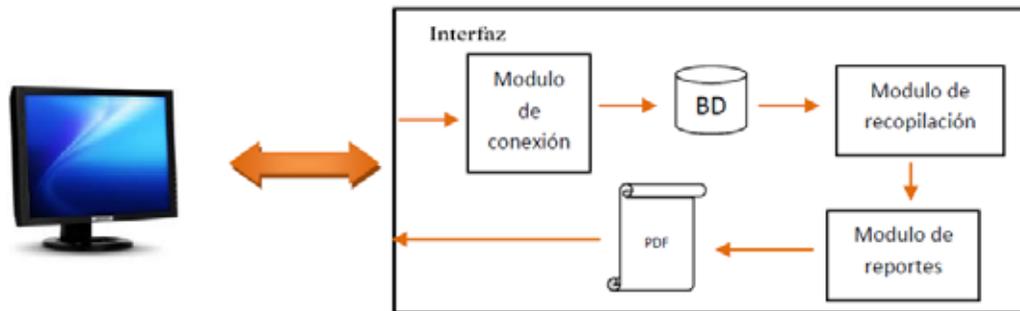


Figura 1. Diagrama de proceso

Cabe mencionar que las variables de configuración varían para cada uno de los manejadores de bases de datos relacionales que se usan para este proyecto, pero en términos generales, las variables de configuración que se muestran en el reporte son: la ruta al directorio que hospeda al fichero específico de servidor, información sobre usuarios e información sobre los procesos del servidor por ejemplo: el número de identificación o id para el usuario, el nombre del usuario y el host desde el que se conecta la base de datos predeterminada, la ruta o path y los parámetros de seguridad.

Las variables de estado generales que se muestran en el reporte son: las bases de datos, las tablas que contiene cada base de datos, la estructura de campos de las tablas, información descriptiva de una tabla como: las diferentes columnas, filas y los índices. De forma más explícita:

El resultado obtenido muestra:

- nombre de la tabla.
- tipo de tablas
- formato de almacenamiento de la fila
- número de filas
- promedio de bytes usados por las filas de la tabla
- tamaño actual en bytes de fichero de tabla

- tamaño máximo en bytes que puede alcanzar el archivo.
- tamaño actual en bytes del archivo de índice.
- número de bytes en el archivo de datos que no se utilizan.
- hora que se creó la tabla.
- hora de la última modificación.
- última vez que la tabla fue reparada o revisada.
- comentarios aparecidos en la creación de la tabla.

Para las diferentes columnas de una tabla:

- nombre de la columna.
- el tipo de columna.
- si la columna puede contener valores nulos.
- si la columna está indexada.
- el valor por defecto.
- información extra de la columna:
- los privilegios de la columna.

Para los índices que contiene una tabla:

- el nombre de la tabla que contiene el índice.
- si puede contener valores duplicados.
- el nombre del índice.
- el número de columnas en el índice.
- el nombre de la columna.
- orden de clasificación de la columna dentro del índice.
- el número de valores únicos en el índice.

Información sobre las variables de estado del servidor:

- número de conexiones de un cliente canceladas
- número de intentos fallidos de conexión al servidor.
- número de intentos realizados para conectarse al servidor.
- número de tablas temporales creadas mientras se procesan las consultas.
- número de errores ocurridos mientras se procesan filas.
- número de errores actuales.

- número de filas introducidas.
- número de solicitudes necesarias para suprimir una fila de una tabla.
- número de solicitudes para actualizar una fila.
- número de solicitudes para insertar una fila.
- número de bloques en uso en la cache de índice.
- número de solicitudes para leer un bloque desde la cache.
- número de lecturas físicas de los bloques de índice desde el disco.
- número de requisitos para escribir un bloque en la cache de índice.
- número de escrituras físicas de los bloques de índices de disco.
- número máximo de conexiones que se han abierto simultáneamente.
- número de bloques en la cache de clave que han sido modificados pero aun no han sido vaciados del disco.
- número de archivos abiertos.
- número de flujos abiertos.
- número de tablas abiertas.
- total de tablas que han sido abiertas.
- número de consultas que ha recibido el servidor.

Las variables de estado que se mencionaron son las principales, sin embargo también se muestran las variables propias de cada manejador.

## **Especificación Técnica**

La arquitectura que utilizamos es la de Cliente-Servidor en 2 niveles, ya que se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

Se usó dicha arquitectura ya que en este caso, la aplicación se ejecutará esporádicamente, es decir, sólo cuando el cliente lo solicite.

La arquitectura Cliente-Servidor en 2 niveles se muestra en la Figura 2.



Figura 2. Arquitectura Cliente-Servidor en 2 niveles

Los manejadores de Bases de Datos Relacionales que están soportados por este proyecto son:

- MySQL
- PostgreSQL
- DB2
- Oracle

Las pruebas del programa se realizaron en Windows XP de 32 bits y superior y GNU/Linux Ubuntu 13.04 x86.

El lenguaje de programación elegido para la elaboración de este proyecto es Java, por su seguridad, fácil portabilidad y también porque permite la ejecución de un mismo programa en múltiples sistemas operativos. Las pruebas consistieron en realizar reportes de la configuración del servidor de base de datos, así como la información del contenido de la misma.

## Instalación

En este capítulo se mostrará la forma de instalar cada uno de los programas necesarios para que el sistema funcione de manera correcta. Como primera instancia, se mostrarán los requerimientos mínimos para poder ejecutar cada manejador y también la forma de cómo se instalan cada uno de ellos, es decir, de MySQL, PostgreSQL, Oracle y DB2.

## MySQL v5.2

### Requerimientos

- Windows evoXPecial 3 v2010 service pack 3 de 32 bits o superior.
- Ubuntu-13.04-desktop-i386
- Soporte para protocolo TCP/IP.
- Una copia de la distribución binaria de MySQL para Windows, que se puede descargar de <http://dev.mysql.com/ddownloads/>
- Una herramienta capaz de leer ficheros *.zip*, para descomprimir el fichero de distribución.
- Suficiente espacio en disco duro para descomprimir, instalar, y crear las bases de datos de acuerdo a sus requisitos.

### Proceso de Instalación para el Sistema Operativo Windows XP edición familiar v2002 service pack 3

- Elegimos el paquete de instalación. En nuestro caso elegimos MySQL Workbench 5.2.46 ya que es una aplicación de diseño de base de datos visual que se puede utilizar para diseñar de manera eficiente, gestionar y documentar esquemas de base de datos.
- Ejecutamos el programa que descargamos, como se indica en la Figura 3



Figura 3

- Se muestra la ventana de Bienvenida, y para continuar con la instalación se da clic en el botón “Next”

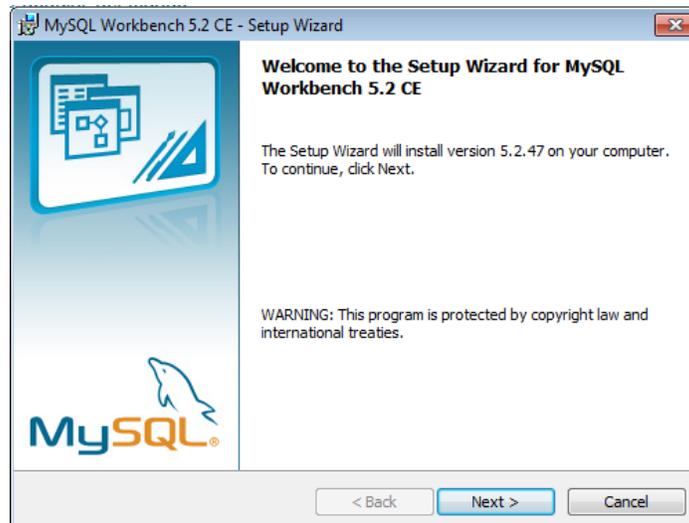


Figura 4

- El siguiente paso muestra la ruta donde se instala el programa

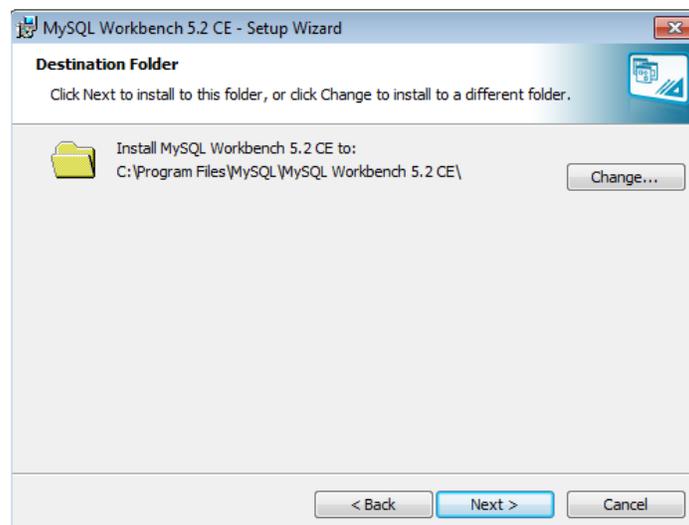


Figura 5

- Se elige el tipo de instalación, en nuestro caso "Complete"

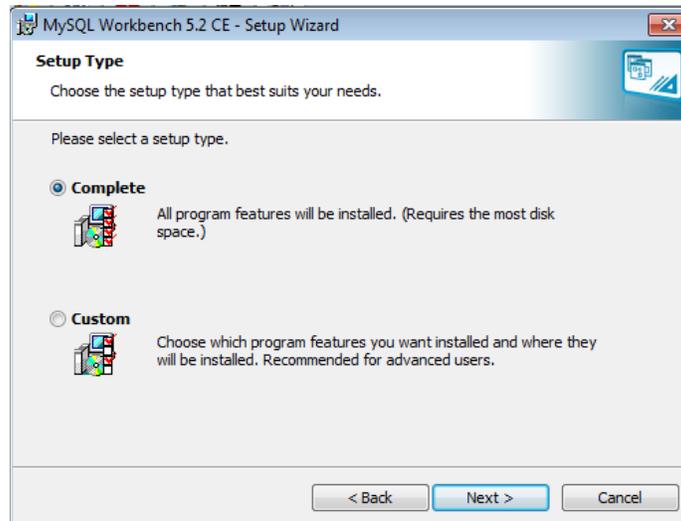


Figura 6

- La instalación del programa se inicia presionando el botón “*Install*”.

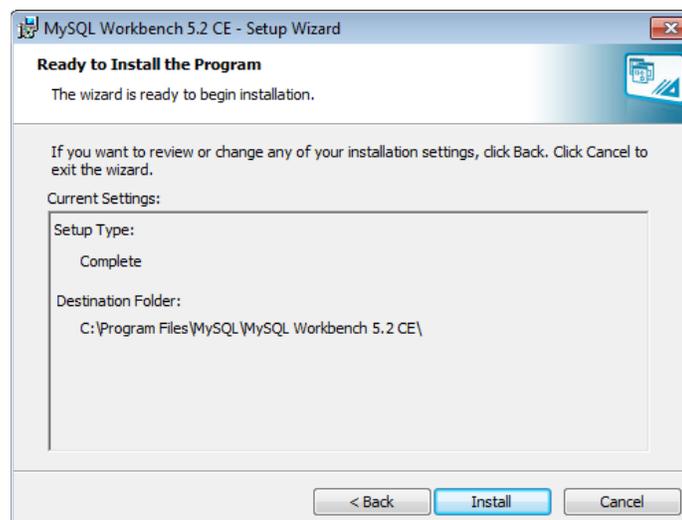


Figura 7

- En la siguiente figura se muestra el momento en el que se está procesando la instalación hasta completarla.



Figura 8

- Por último, muestra la pantalla de instalación finalizada.



Figura 9

## Proceso de Instalación para Ubuntu 13.04 GNU/Linux

- Elegimos el paquete de instalación. En nuestro caso elegimos MySQL Workbench 5.2.46
- Descargamos el MySQL Workbench 5.2.46 desde <http://dev.mysql.com/downloads/tools/workbench/5.2.html#downloads>



Figura 10

- Damos clic en *No thanks, just start my download.*



Figura 11

- Se descarga el siguiente archivo *mysql-workbench-gpl-5.2.46-1ubu1204-i386.deb.*

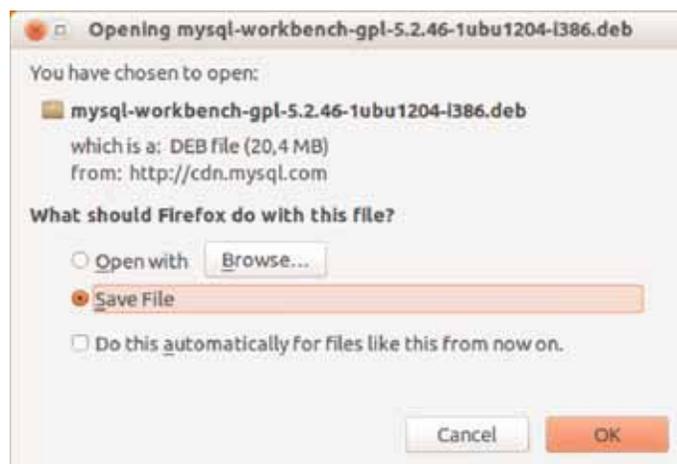


Figura 12

- Instalamos algunas dependencias que necesitaremos.
  - `sudo apt-get install python-paramiko mysql-client python-pysqlite2 mysql-common python-gdbm mysql-client-5.5 libdbi-perl libdbd-mysql-perl mysql-client-core-5.5 liblprc-perl libnet-daemon-perl`
- Descargamos el binario para libtemplate0 y lo instalamos.
  - `wget -c http://launchpadlibrarian.net/83494419/libtemplate0_1.0-1_i386.deb`
  - `sudo dpkg -i libtemplate0_1.0-1_i386.deb`
- Instalamos el MySQL WorkBench 5.2.46
  - `sudo dpkg -i mysql-workbench-gpl-5.2.46-1ubu1204-i386.deb`
- Y ya posteriormente podremos ejecutarlo

## PostgreSQL 9.2 y PostgreSQL 8.4.17

### Requerimientos

- Windows evoXPecial 3 v2010 service pack 3 de 32 bits o superior.
- Ubuntu-13.04-desktop-i386
- Suficiente espacio en disco duro para descomprimir, instalar, y crear las bases de datos de acuerdo a sus requisitos.
- Para obtener PostgreSQL 9.2 y PostgreSQL 8.4.17 es necesario visitar su sitio oficial, el cual es: <http://postgresql.org/> . Una vez descargado el paquete, lo ejecutamos y procedemos con la instalación.

### Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3

- Aparte de instalar PostgreSQL 9.2, también es recomendable instalar un cliente gráfico que nos facilitará la interacción con el servidor. Un cliente gráfico recomendable es pgAdmin III, que funciona muy bien.
  1. En el primer paso de la instalación aparece la pantalla de bienvenida en la cual únicamente tendremos que dar clic en el botón “*Siguiente*”, tal como se muestra en la Figura 13



Figura 13

2. En el siguiente paso elegimos el directorio en donde PostgreSQL será instalado, y damos clic en el botón “*Siguiente*”

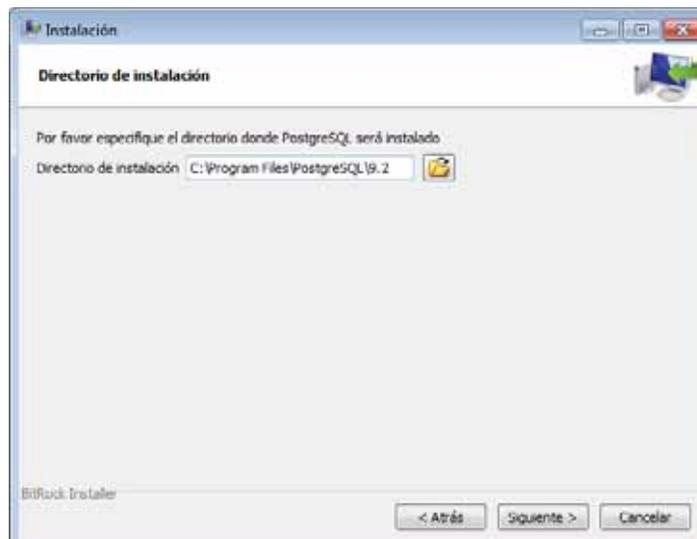


Figura 14

3. A continuación elegiremos el directorio dentro del cual se van a almacenar nuestros datos, observe la Figura 15 y de nueva cuenta damos clic en el botón “*Siguiente*”.



Figura 15

4. En el paso siguiente colocaremos una contraseña para el superusuario de PostgreSQL, la cual debemos recordar ya que la utilizaremos cada vez que hagamos uso de este manejador. Para avanzar al siguiente paso damos clic en el botón “*Siguiente*”.

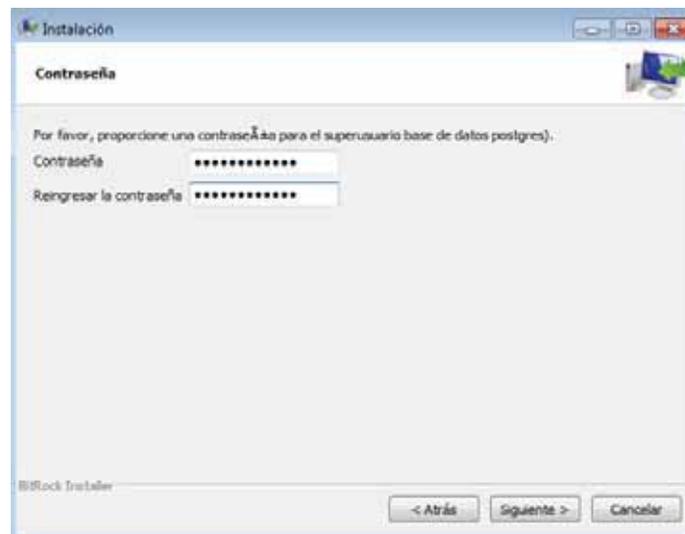


Figura 16

5. En el siguiente paso debemos seleccionar el número de puerto en el que el servidor debería escuchar, por ejemplo el puerto 5432, como se indica en la Figura 17

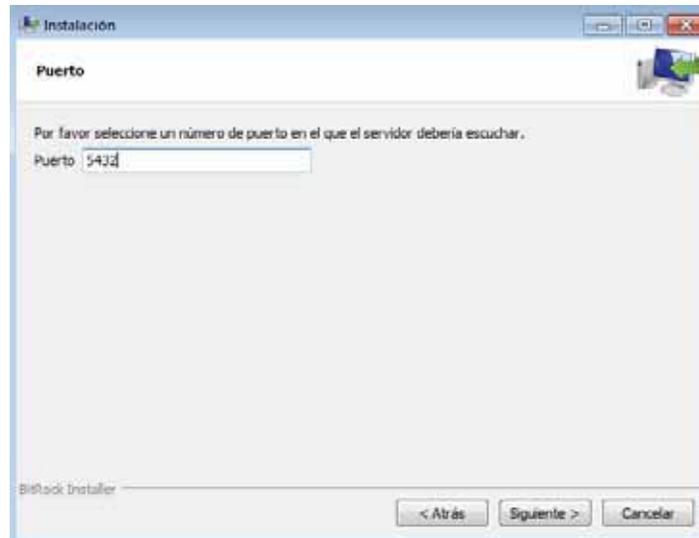


Figura 17

6. En el siguiente punto elegimos el tipo de configuración regional usada por el nuevo clúster de la base de datos, en nuestro caso elegimos la Configuración Regional por Defecto.

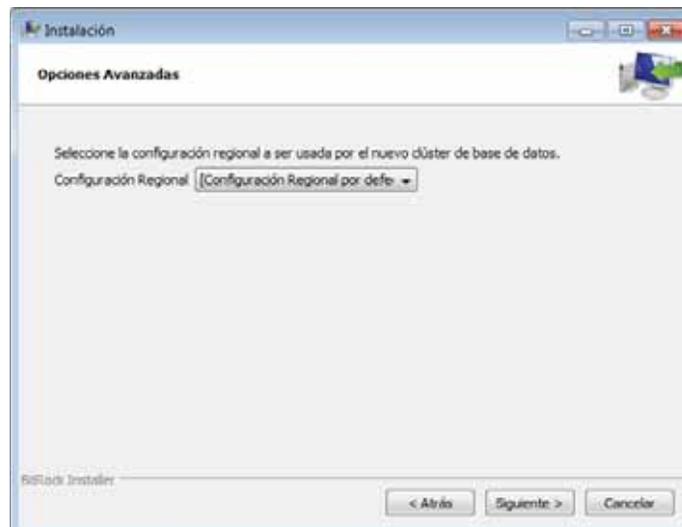


Figura 18

7. Al llegar a este paso, el programa ya está listo para iniciar la instalación de PostgreSQL en nuestra computadora, únicamente damos clic en "Siguiente" para completar el proceso de instalación.

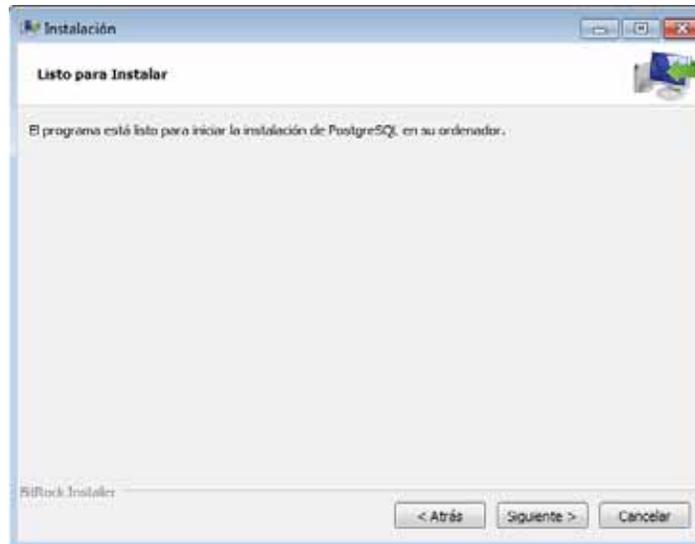


Figura 19

8. La siguiente ventana nos ejemplifica como se está procesando la instalación.

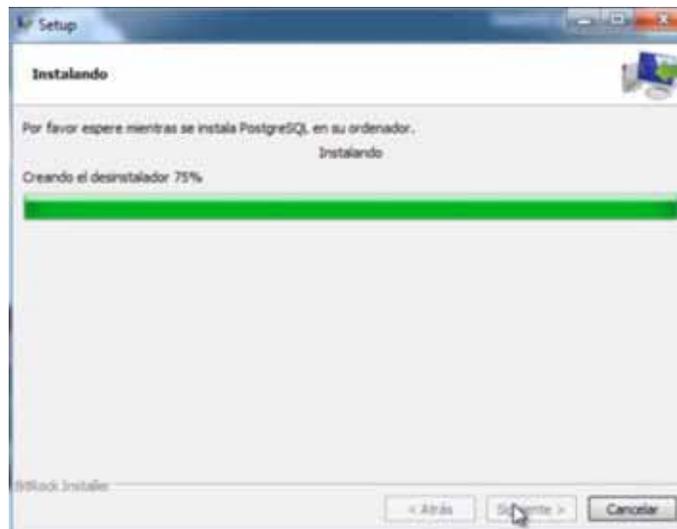


Figura 20

9. Por último, la instalación queda completada.



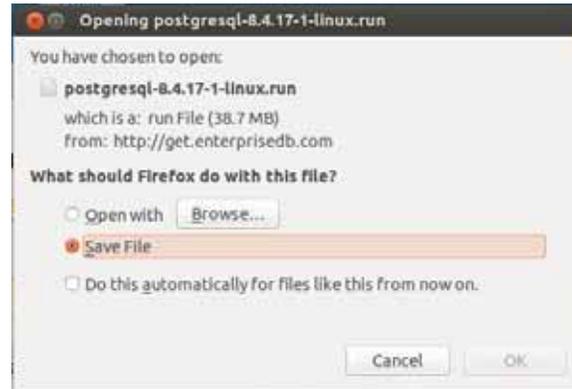
Figura 21

## Proceso de Instalación para Ubuntu 13.04 GNU/Linux

- Lo primero que tenemos que hacer es descargar la versión del programa de instalación que corresponde con nuestro sistema. En nuestro caso nos descargamos desde <http://www.enterprisedb.com/products/pgdownload.do> la versión PostgreSQL 8.4.17



- Guardamos el archivo que se descargó.



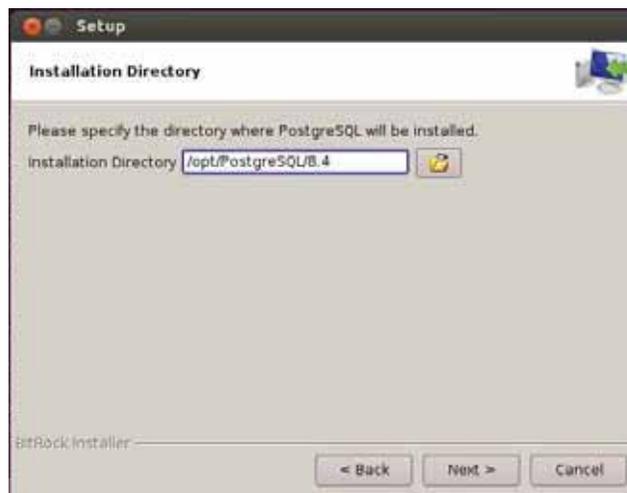
- Una vez descargado lo tendremos que ejecutar como administrador en nuestro sistema. 'root' en Ubuntu 13.04.

```
root@ivan-Aspire-one: /tmp
root@ivan-Aspire-one:/tmp# ls
hsperfdata_ivan
jna
lurbe04e.tmp
OSL_PIPE_1000_SingleOfficeIPC_fb8a21ad2cf89f0454f75d4cdaeb3db
postgresql-8.4.17-1-linux.run
pulse-PKdhtXMr18n
ssh-nKUKDwf7sveh
unity_support_test.0
root@ivan-Aspire-one:/tmp# chmod +x postgresql-8.4.1-1-linux.bin
chmod: cannot access 'postgresql-8.4.1-1-linux.bin': No such file or directory
root@ivan-Aspire-one:/tmp# chmod +x postgresql-8.4.17-1-linux.run
root@ivan-Aspire-one:/tmp# ls
hsperfdata_ivan
jna
lurbe04e.tmp
OSL_PIPE_1000_SingleOfficeIPC_fb8a21ad2cf89f0454f75d4cdaeb3db
postgresql-8.4.17-1-linux.run
pulse-PKdhtXMr18n
ssh-nKUKDwf7sveh
unity_support_test.0
root@ivan-Aspire-one:/tmp# chmod +x postgresql-8.4.17-1-linux.run
root@ivan-Aspire-one:/tmp# ./postgresql-8.4.17-1-linux.run
```

- La primera pantalla que se nos muestra es la bienvenida al instalador de PostgreSQL 8.4.17. A partir de ahora tendremos que pulsar "Next" cada vez que queramos avanzar al siguiente paso.



- En el siguiente paso tendremos que definir el directorio donde vamos a instalar todos los programas que vienen con esta versión de PostgreSQL 8.4.17. En nuestro caso, utilizaremos el valor por defecto que el programa nos sugiere



- Una vez que hemos escrito el directorio que queremos utilizar para instalar el programa, pasamos al siguiente y de paso tenemos que definir el directorio de datos en donde se crearán nuestras bases de datos. De nuevo, en nuestro caso utilizaremos el valor por defecto que el programa nos sugiere

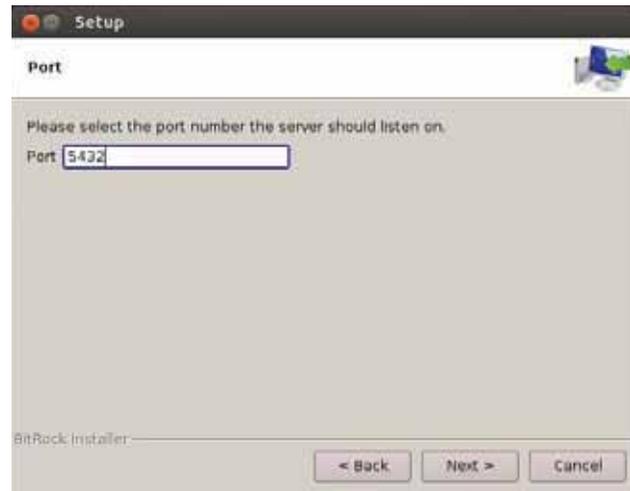


- En el siguiente paso, tenemos que definir una clave de acceso para el usuario administrador de nuestra base de datos PostgreSQL 8.4.17





- En este paso tendremos que especificar el puerto que PostgreSQL 8.4.17 utilizará para escuchar por conexiones. En nuestro caso dejamos el valor por defecto, 5432.



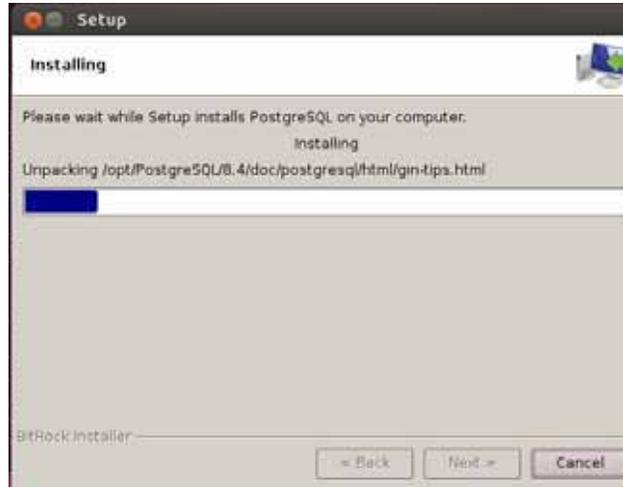
- Una vez que hemos terminado con los pasos básicos, el programa entra en la sección de opciones avanzadas. En este paso tenemos que decidir que 'locale' queremos utilizar y si queremos instalar PL/pgSQL en la base de datos template1. Te recomiendo elegir "Locale: C" e instalar PL/pgSQL en template1. (template1 es una base de datos por defecto en PostgreSQL 8.4.17 que se puede utilizar como plantilla cuando creamos bases de datos.



- Una vez que hemos terminado con todos los pasos de configuración, el programa nos informa que está listo para empezar a instalar e inicializar nuestra instalación.



- Pulsamos por última vez "Next" y esperamos a que el programa termine de instalar todo.



- Una vez terminada la instalación, podremos salir del instalador pulsando "Finish".



- Si todo el proceso de instalación ha salido bien tendríamos que tener un nuevo menú en nuestro sistema. En este menú podremos acceder a la documentación de PostgreSQL 8.4.17.



## Oracle database 11g express edition release v11.2.0.2.0

Para fines de este proyecto utilizamos la versión *Oracle Database Express Edition*, es la versión gratuita de Oracle Database. Esta versión tiene las siguientes limitaciones: sólo puede utilizar 1 procesador del servidor donde esté instalada, un máximo de 1 Gb de RAM, y tiene limitado el almacenamiento a 4 Gb de datos de usuario. Cuando se superen estas limitaciones, Oracle ofrece un sencillo proceso de actualización a otras versiones más completas, y en las que sí que hay que pagar licencias. De todas maneras es una opción muy válida para quien quiera practicar con bases de datos Oracle, o para quien necesite una base de datos limitada en cuanto a tamaño, pero fiable y eficiente, y gratuita.

### Requerimientos

- Windows evoXPecial 3 v2010 service pack 3 de 32 bits o superior.
- Ubuntu-13.04-desktop-i386
- Para la descarga de Oracle Database XE nos dirigimos al sitio oficial de Oracle <http://www.oracle.com>
- En el menú de opciones elegimos la opción DOWLOADS y se despliegan otras opciones categorizadas.
- En la categoría “Database” selecciona la versión actual del Oracle database Express Edition y pulsa sobre la opción que dice “Oracle Database 11g Express Edition”.
- Acepta los términos de la licencia.
- Selecciona el link para descargar donde dice “*Oracle Database Express Edition 11g Release 2 for Windows x32*”.
- Y por último nos pregunta dónde queremos ubicar el instalador de Oracle Database 11g XE.

### Proceso de Instalación para el Sistema Operativo Windows XP edición familiar v2002 service pack 3

1. Descomprimir el archivo descargado, luego aparece una ventana para ubicar el lugar donde se guardaran los archivos a descomprimir. Cuando se ubique la ruta, el proceso de descompresión inicia.
2. Después de descomprimir el archivo, se ha generado un nuevo archivo con el mismo nombre del archivo que descargaste.

3. Se abre la carpeta "**DISCK1**" que se encuentra dentro de ella y se ejecuta dando doble clic el archivo "**setup**".

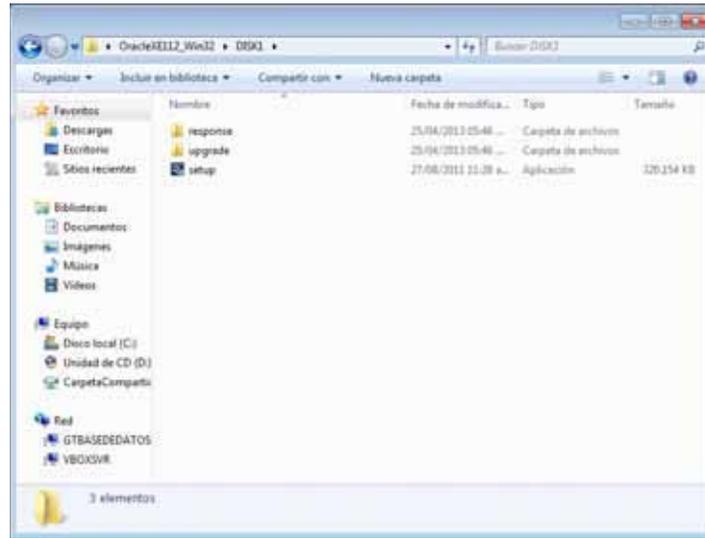


Figura 22

4. Al dar doble clic el Windows te pedirá permisos para ejecutar este archivo para que este pueda hacer cambios en el equipo, das clic en "Sí".

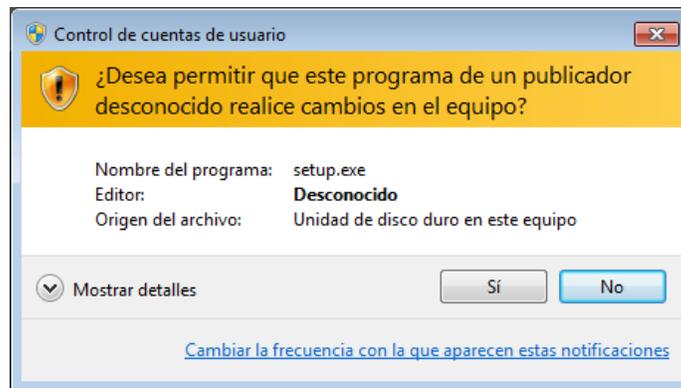


Figura 23

5. Se iniciará un proceso de descompresión para prepara e iniciar la instalación.

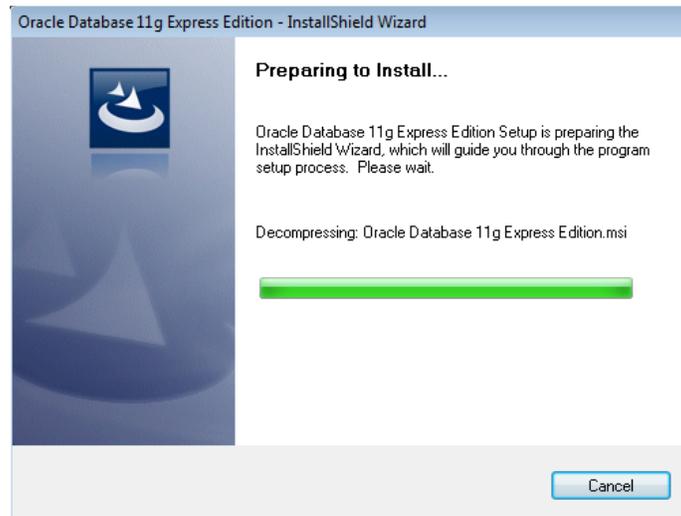


Figura 24

6. Al terminar el proceso de preparación te aparece la primera ventana del proceso de instalación dándote la bienvenida. Para continuar se selecciona "Next" o "Siguiete".

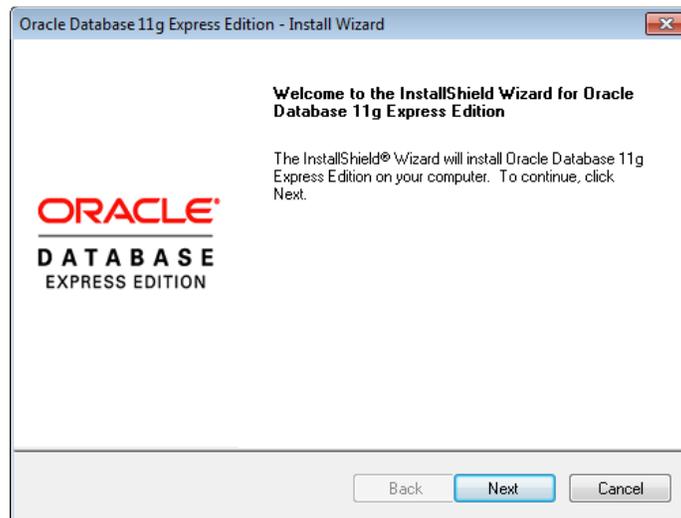


Figura 25

7. La segunda ventana muestra los acuerdos de licencia, para continuar aceptamos los términos de la licencia chequeando "I accept the terms in the license agreement" y luego en "Next" o "Siguiete".



Figura 26

8. La tercera venta te permite ubicar en que parte del equipo se instalara Oracle XE. Como se percibe en la siguiente imagen en la sección de "Destination Folder" de la parte inferior la ubicación por defecto donde instalara Oracle es en "C:" en una carpeta llamada "oraclexe". Si no deseas esta ubicación pulsa en el boton "Browse..." selecciona la ubicación deseada. Para este caso se dejara la ubicación por defecto. Para continuar se da clic en "Next" o en "Siguiete".

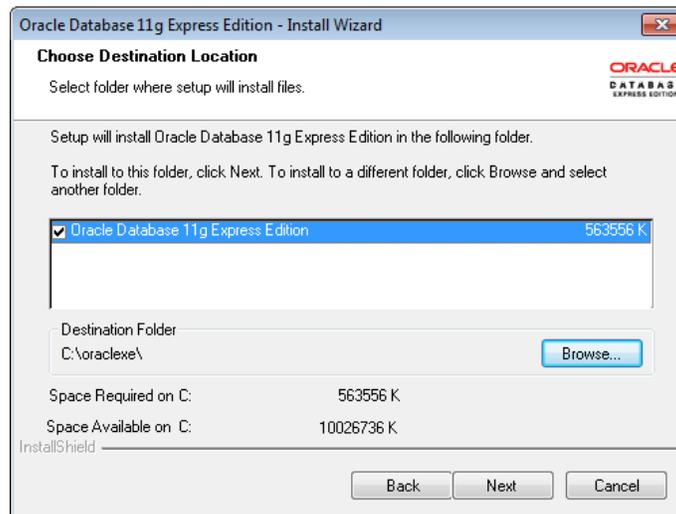


Figura 27

9. La cuarta ventana te permite dar una contraseña general para los usuarios SYS y SYSTEM de la base de datos.



Figura 28

10. La quinta ventana te muestra algunos datos con lo que se instalará Oracle. Es recomendable anotar estos datos ya que por lo general serán necesarios posteriormente. Para continuar seleccionamos en "Install".

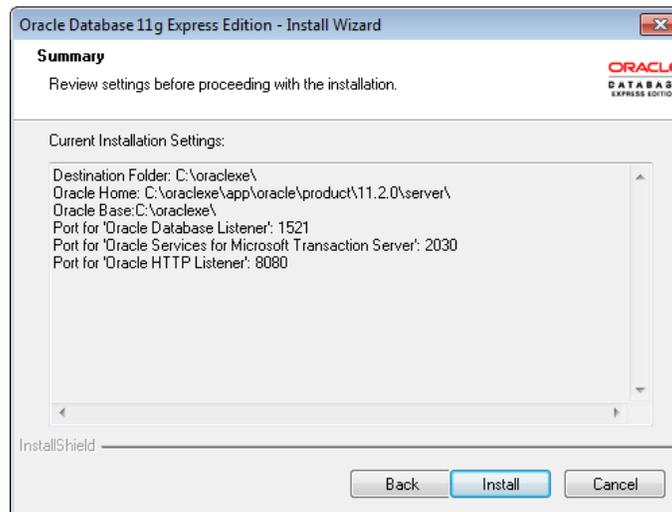


Figura 29

11. Al presionar en *"Install"*, el proceso de instalación de Oracle inicia. Este es un proceso que dura unos minutos.

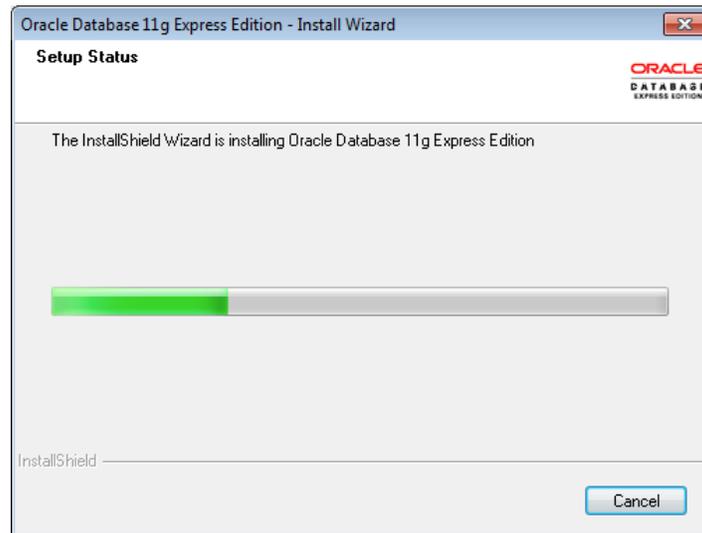


Figura 30

12. Al terminar el proceso de instalación te aparece una última ventana donde confirma la satisfacción con la que se completo la instalación. Para terminar seleccionamos *"Finish"*.

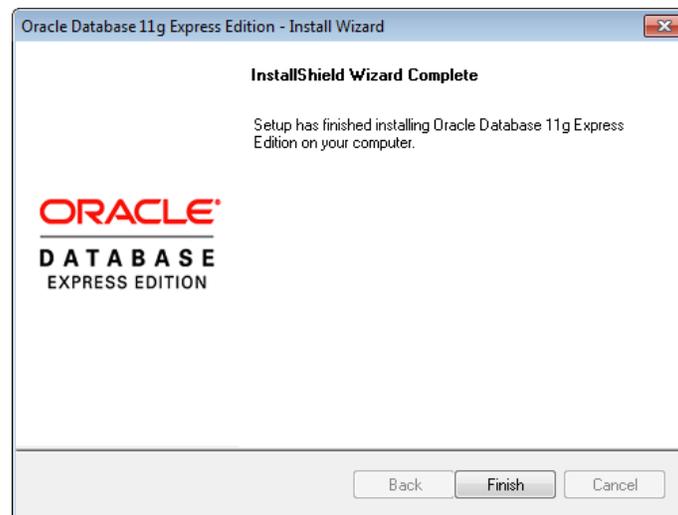


Figura 31

## Proceso de Instalación para Ubuntu 13.04 GNU/Linux

Bueno, la instalación de Oracle 11gR2 es un poco complicada en un sistema tan moderno como Ubuntu 13.04, desafortunadamente Oracle no actualiza tan rápido sus dependencias de paquetes, ya

que se mantiene dentro de los estándares de los sistemas comerciales. Pero es posible adecuar algunos atributos de compilación para poder instalarlo.

1. Ejecutar desde terminal:

**Crear usuario y grupos:**

```
$ sudo groupadd nobody
$ sudo groupadd dba
$ sudo useradd -g dba -G dba,sudo oracle -d /home/oracle -s /bin/bash
$ sudo passwd oracle
$ sudo mkdir /home/oracle
$ sudo mkdir /opt/oracle
$ sudo chown oracle:dba /home/oracle /opt/oracle
```

2. **Crear perfil de usuario:**

```
$ sudo -u oracle vi /home/oracle/.profile
===== ARCHIVO =====
export ORACLE_BASE=/opt/oracle/app/product/11.2.0
export ORACLE_HOME=$ORACLE_BASE/db_1
export PATH=$ORACLE_HOME/bin:$PATH
=====
```

3. **Crear directorios para Oracle:**

```
$ sudo -u oracle mkdir -p /opt/oracle/app/product/11.2.0/db_1
```

4. **Crear oraInst.loc:**

```
$ sudo vi /etc/oraInst.loc
===== ARCHIVO =====
inventory_loc=/opt/oracle/oraInventory
inst_group=dba
=====
```

5. **ConFigurar parámetros de sistema operativo:**

```
$ sudo vi /etc/security/limits.conf
===== ARCHIVO =====
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
=====

$ sudo vi /etc/sysctl.conf
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
```

```
net.core.wmem_max = 1048586
```

```
=====
```

```
$ sudo systemctl -p
```

6. **Instalar paquetes adicionales y trucos para Ubuntu:**

```
$ sudo apt-get install libstdc++5 libelf1 libelf-dev libaio1 libaio-dev sysstat elfutils  
lesstif2 libmotif4 libmotif-dev libtool numactl expat make unixodbc unixodbc-dev  
make gcc
```

```
$ sudo ln -s /usr/bin/awk /bin/awk
```

```
$ sudo ln -s /usr/bin/basename /bin/basename
```

```
$ sudo ln -s /usr/lib/x86_64-linux-gnu /usr/lib64
```

```
$ sudo ln -s /lib/x86_64-linux-gnu/libgcc_s.so.1 /lib64/libgcc_s.so.1
```

7. **Iniciar el instalador de Oracle:**

```
$ xhost +
```

```
$ su - oracle
```

```
$/software/Disk1/runInstaller
```

8. Una vez iniciado el instalador, sigue los pasos Windows (siguiente, siguiente, siguiente...) y en el momento de realizar el ligado de las librerías **aparecerá un error sobre el archivo “ins\_emagent.mk”**. Antes de reintentar, realiza las siguientes modificaciones.

```
$ cd $ORACLE_HOME
```

```
$ vi sysman/lib/ins_emagent.mk
```

```
reemplaza las líneas:
```

```
$(SYSMANBIN)emdctl:
```

```
$(MK_EMAGENT_NMECTL)
```

```
por:
```

```
$(SYSMANBIN)emdctl:
```

```
$(MK_EMAGENT_NMECTL) -lnnz11
```

```
$ vi rdbms/lib/env_rdbms.mk
```

```
reemplaza las líneas:
```

```
KFED_LINKLINE=$(LINK) $(SOMAIN) $(SSKFEDED) $(SKFEDPT) \
```

```
por:
```

```
KFED_LINKLINE=$(LINK) $(SOMAIN) -Wl,--no-as-needed $(SSKFEDED)
```

```
$(SKFEDPT) \
```

```
reemplaza las líneas:
```

```
KFOD_LINKLINE=$(LINK) $(SOMAIN) $(SSKFODED) $(SKFODPT) $(KFODOBJ) \
```

```
por:
```

```
KFOD_LINKLINE=$(LINK) $(SOMAIN) -Wl,--no-as-needed $(SSKFODED)
```

```
$(SKFODPT) $(KFODOBJ) \
```

```
reemplaza las líneas:
```

```
KFNDG_LINKLINE=$(LINK) $(SOMAIN) $(SSKFNDGED) $(SKFNDGPT)
```

```
$(KFNDGOBJ) \
```

```
por:
```

```
KFNDG_LINKLINE=$(LINK) $(SOMAIN) -Wl,--no-as-needed $(SSKFNDGED)
```

```
$(SKFNDGPT) $(KFNDGOBJ) \
```

```
reemplaza las líneas:
```

```
AMDU_LINKLINE=$(LINK) $(SOMAIN) $(SSKFMUED) $(SKFMUPT) \
```

```
por:
```

```
AMDU_LINKLINE=$(LINK) $(SOMAIN) -Wl,--no-as-needed $(SSKFMUED)
```

```
$(SKFMUPT) \
```

```
reemplaza las líneas:
```

```
ORACLE_KERNEL_LIBS=$(LLIBSKGTR) $(LLIBPERFSRV) $(NAETTOBJS)
```

```
$(CONFIG) \
```

```
$(SPOBJS) $(LLIBSERVER) $(LLIBODM) $(LLIBSAGE) \
```

```

por:
ORACLE_KERNEL_LIBS=$(LLIBSKGTR) $(LLIBPERFSRV) $(NAETTOBJS)
$(CONFIG)\
$(SPOBJS) -Wl,--no-as-needed $(LLIBSERVER) $(LLIBODM) $(LLIBSAGE) \
$ vi bin/genorasdksh
reemplaza las líneas:
$LD $LD_RUNTIME $LD_OPT $LD_OUT $LD_SELF_CONTAINED $BASE_LIB \
por:
$LD $LD_RUNTIME -Wl,--no-as-needed $LD_OPT $LD_OUT
$LD_SELF_CONTAINED $BASE_LIB \
$ vi srvmlib/ins_srvmlib.mk
reemplaza las líneas:
GETCRSHOME_LINKLINE=$(LINK) $(LDPATHFLAG)$(RDBMSLIB) $(CDEBUG)
$(OPT) $(GETCRSHOME_OBJ1) \
$(OCRLIBS_DEFAULT) $(LLIBCLNTSH) $(LINKLDLIBS)
por:
GETCRSHOME_LINKLINE=$(LINK) $(LDPATHFLAG)$(RDBMSLIB) $(CDEBUG)
$(OPT) $(GETCRSHOME_OBJ1) \
-Wl,--no-as-needed $(OCRLIBS_DEFAULT) $(LLIBCLNTSH) $(LINKLDLIBS)
$ vi network/lib/env_network.mk
reemplaza las líneas:
TNSLSNR_LINKLINE=$(LINK) $(TNSLSNR_OFILES) $(LINKTTLIBS) $(LLIBONS) \
por:
TNSLSNR_LINKLINE=$(LINK) $(TNSLSNR_OFILES) -Wl,--no-as-
needed $(LINKTTLIBS) $(LLIBONS) \

```

Una vez realizadas estas modificaciones reintentas desde el instalador y debe terminar todo bien.

## DB2 express-c v1.01 y DB2 express-c v10.5

DB2 Express-C es la versión gratuita soportada por la comunidad de DB2 que permite desarrollar, implementar y distribuir aplicaciones que no usen las características avanzadas de las versiones comerciales de DB2. Esta versión de DB2 puede ser concebida como el núcleo de DB2, las diferentes ediciones incluyen las características de Express-C más funcionalidades específicas.

### Requerimientos

- Windows evoXPecial 3 v2010 service pack 3 de 32 bits o superior.
- Ubuntu-13.04-desktop-i386
- Descargar el paquete de la siguiente página, y una vez descargado lo ejecutamos: [www.ibm.com/db2/express](http://www.ibm.com/db2/express)

## Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 y Ubuntu 13.04 GNU/Linux

- Se utilizó la versión Db2 express-c v10.1 para el Sistema Operativo Windows XP edición familiar v2002 service pack 3 y la versión Db2 express-c v10.5 para Ubuntu 13.04.
- Para instalar DB2 Express-C v10.1 en Windows, debes ser un usuario con los permisos suficientes para poder instalar aplicaciones sobre el Sistema Operativo.
- Existen muchos métodos para instalar DB2 Express-C, el método más fácil es usar el Wizard de instalación de DB2 (DB2 Installation Wizard) de interface gráfica (GUI-based). Luego de bajar y descomprimir la imagen DB2 Express-C v10.1, podemos seguir con el Wizard que nos guiará en la instalación:
  - ✓ Windows: ejecuta el archivo setup.exe en el directorio EXP/image
  - ✓ Linux: ejecuta el comando db2setup (en consola) en el directorio exp/disk1
- DB2 Express-C es muy fácil de instalar siguiendo las instrucciones del DB2 installation wizard. En la mayoría de casos, las configuraciones por defecto son suficientes, por lo que lo único que debe hacer es aceptar la licencia, clic en el botón “Next” varias veces, y clic en el botón “Finish”. Después de poco tiempo tu instalación estará completada y DB2 está listo para ser usado.

### NetBeans IDE 7.3.1

#### Requerimientos

- El Java SE Development Kit (JDK) 6 Update 26 (o posterior) o JDK 7 Update 10 (o superior) es necesario para instalar el IDE NetBeans 7.3.1[7]. Puede descargar la última versión de JDK 7 de la <http://www.oracle.com/technetwork/java/javase/downloads>
- El PHP y C / C + + paquetes de NetBeans sólo requieren el Java Runtime Environment (JRE) 6 a instalar y ejecutar. Sin embargo, si usted está planeando utilizar cualquiera de las funciones de Java, necesita JDK 7.
- El JavaFX SDK 2.2.5 es necesario para utilizar JavaFX 2.2.5 Las características del IDE NetBeans 7.3.1. Puede descargar el JavaFX SDK 2.2.5 del sitio de descarga de JavaFX .
  - Nota:
    - JavaFX 2.2.4 viene con el último JDK 7 para Windows, Mac y Linux.

- Si instala el JDK 7 Update 10 o superior a ubicaciones predeterminadas, el IDE NetBeans puede crear una plataforma JavaFX Java habilitado automáticamente.
- Para utilizar las funciones de JavaFX en el IDE NetBeans, le recomendamos actualizar su versión de JDK JDK 7 Update 10 o superior.

## **Proceso de Instalación en el Sistema Operativo Windows XP edición familiar v2002 service pack 3 y Ubuntu 13.04 GNU/Linux**

Estas instrucciones de instalación se aplican a todas las plataformas soportadas.

1. Obtener el software de la página oficial de NetBeans la cuál es: <http://netbeans.org/downloads/> y descargamos la versión completa.
2. En Windows, el archivo ejecutable instalador tiene la extensión .exe. Haga doble clic en el archivo de instalación para ejecutarlo.
3. Para las plataformas Solaris y Linux, el archivo instalador tiene la extensión .sh . Para estas plataformas, es necesario que los archivos ejecutables de instalación utilizando el siguiente comando: `chmod + x < instalador-file-name >`
4. Si ha descargado el Java EE, usted puede personalizar su instalación. Realice los siguientes pasos en la página de bienvenida del asistente de instalación:
  - a. Haga clic en Personalizar.
  - b. En el cuadro de diálogo de instalación personalizada, realice sus selecciones.
  - c. Haga clic en Aceptar.
5. En la página de bienvenida del asistente de instalación, haga clic en Siguiente.
6. En la página Contrato de licencia, revise el contrato de licencia, haga clic en la casilla de aceptación, y haga clic en Siguiente.
7. En la página Contrato de licencia JUnit, decida si desea instalar JUnit y haga clic en la opción adecuada, haga clic en Siguiente.
8. En la página de instalación de NetBeans IDE, haga lo siguiente:
  - a. Acepte el directorio de instalación predeterminado para el IDE NetBeans o especifique otro directorio.
  - b. Acepte la instalación JDK predeterminado para usar con el IDE NetBeans o seleccione una instalación diferente de la lista desplegable. Si el asistente de instalación no encuentra una instalación de JDK compatible para usar con el IDE NetBeans, su JDK no está instalado en la ubicación predeterminada. En este caso, especifique la ruta de acceso a un JDK instalado y haga clic en Siguiente, o cancele la instalación actual. Después de instalar la versión requerida JDK puede reiniciar la instalación.

9. Si la página de Enterprise Server Open Source Edition 4.0 la instalación se abre, acepte el directorio de instalación predeterminado o especifique otra ubicación de la instalación.
10. En la página Resumen, haga lo siguiente:
  - a. Compruebe que la lista de los componentes que desea instalar es correcta.
  - b. Seleccione la casilla de verificación Comprobar actualizaciones si desea comprobar el Centro de actualizaciones de los posibles cambios y tener la biblioteca JUnit instalado durante la instalación.
  - c. Compruebe que dispone de espacio suficiente en el sistema para la instalación.
11. Haga clic en Instalar para comenzar la instalación.
12. En la página Instalación completada haga clic en Finalizar.

## **iTextpdf v5.3.4**

- También necesitaremos librerías tales como la llamada iTextpdf, la cual permite a los desarrolladores generar dinámicamente documentos en formato PDF. Esta librería la elegimos debido a que con ella se pueden generar documentos con entradas escritas por el usuario a través de variables, el contenido del documento puede ser personalizado, y se pueden generar documentos a partir de Bases de Datos, entre otras características. Para obtener esta librería es necesario visitar la página oficial: <http://itextpdf.com>.
- Una vez descargada la librería la guardamos en una carpeta para su futura utilización.

## **Desarrollo del proyecto**

### **Pasos para instalar este proyecto en NetBeans IDE 7.3.1 en el Sistema Operativo Windows XP edición familiar v2002 service pack 3**

Iniciamos NetBeans IDE 7.3.1. En el menú “File” elegimos la opción “New Project”, como se muestra en la siguiente pantalla (Figura 32):

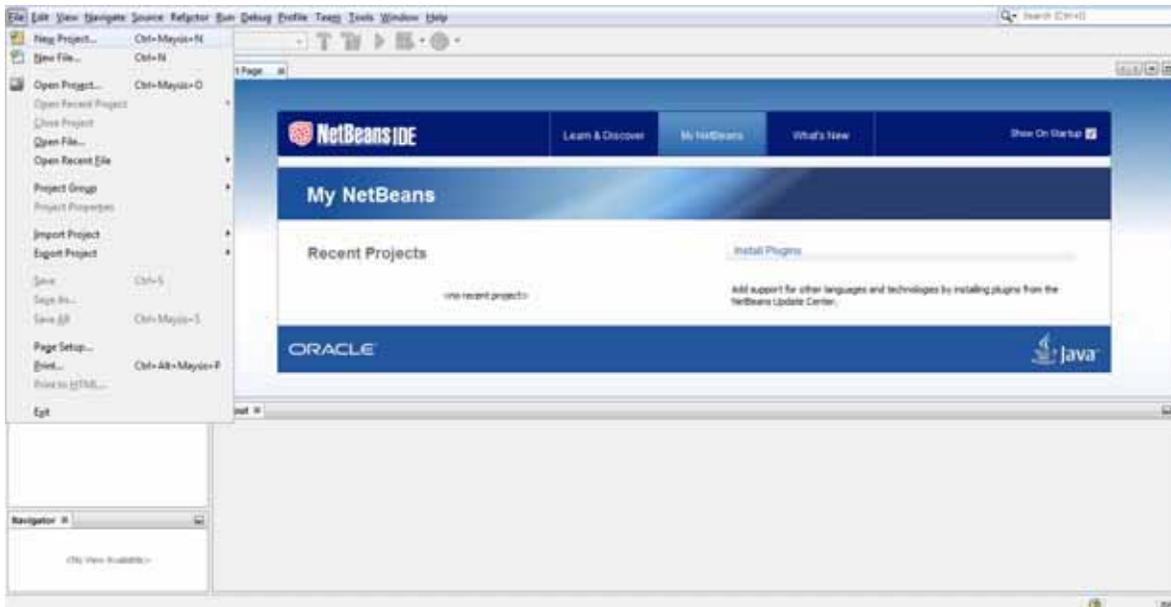


Figura 32

En la ventana que se desplegó elegimos “Java” del apartado “Categories” y Proyectos elegimos “Java Project with Existing Sources” y damos clic en el botón “Next”, como se ilustra en la Figura 33.

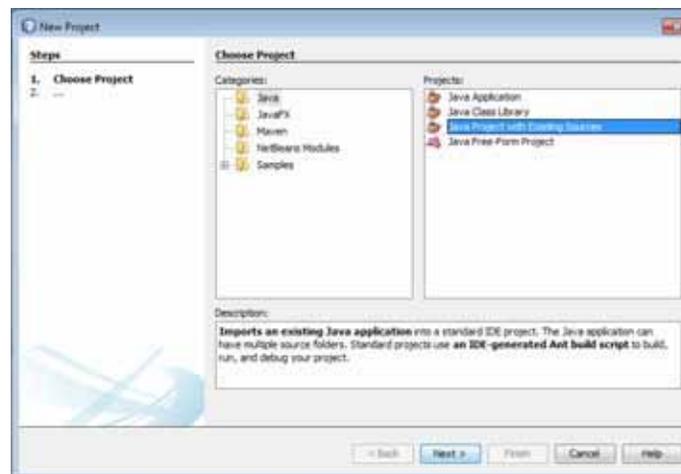


Figura 33

En el siguiente paso, elegiremos el nombre del proyecto y el directorio en el cual será guardado, observe la Figura 34 y dé clic en el botón “Next”.



Figura 34

Agregamos las fuentes existentes para poder utilizar el sistema, para ello, elegimos la opción “AddFolder...”

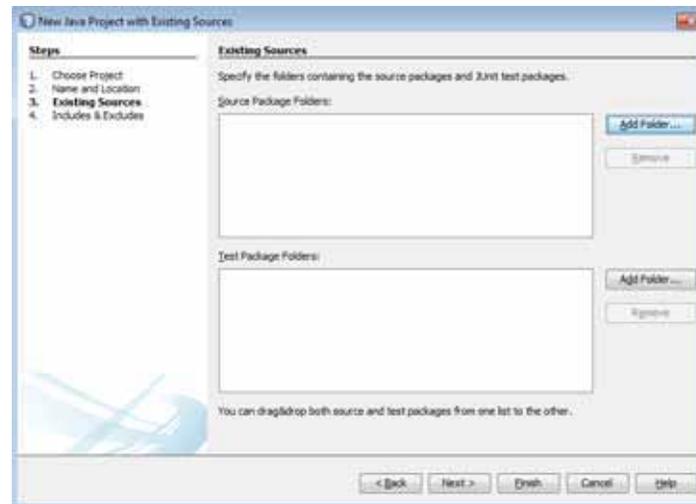


Figura 35

En la ventana que se abrirá, elegimos la carpeta donde está alojado el sistema y abrimos la carpeta para añadirla al proyecto, en la Figura 36 se muestra un ejemplo.

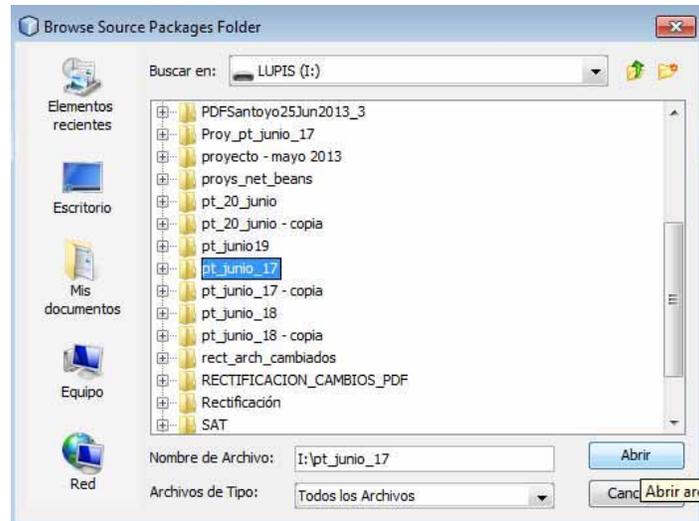


Figura 36

Y para terminar de agregar el proyecto únicamente damos clic en “*Finish*”, la Figura 37 ilustra este paso.

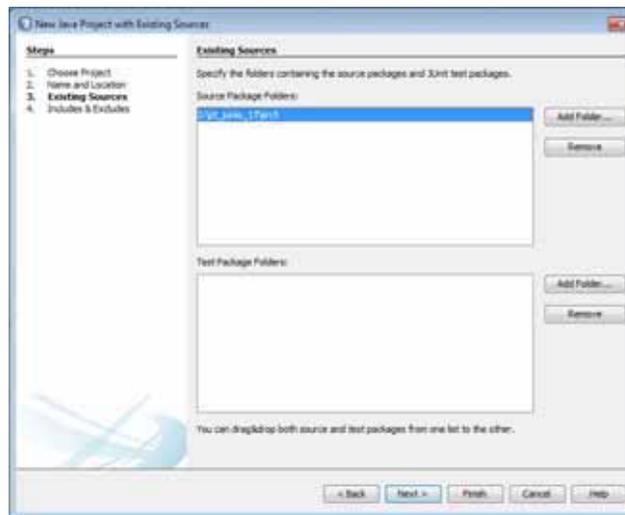


Figura 37

Ya que hayamos agregado el proyecto tendremos que agregarle las librerías necesarias para que el sistema pueda funcionar.

Las primeras librerías que elegiremos son las librerías propias de cada manejador, para ello nos posicionamos en la carpeta donde está el proyecto y damos clic derecho sobre la carpeta “*Libraries*” y elegimos “*Add Library*”, Figura 38.

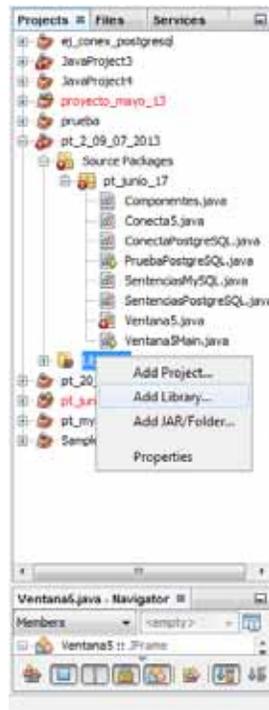


Figura 38

En la ventana que se abrirá elegimos las librerías propias de cada manejador, Figura 39

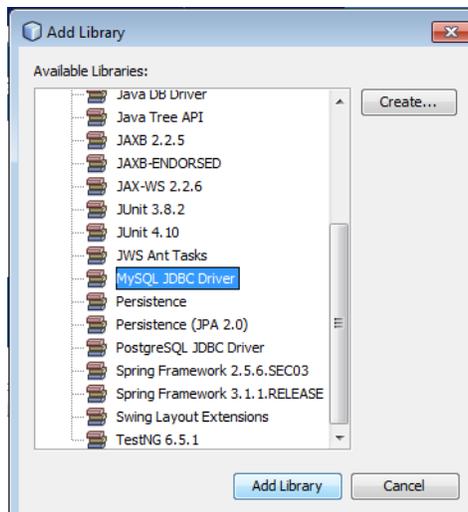


Figura 39

Ya que hayamos elegido las librerías, agregamos los “jar” de la biblioteca, para esto damos clic derecho sobre “Libraries” y elegimos la opción “Add JAR/Folder”, justo como indica la Figura 40.

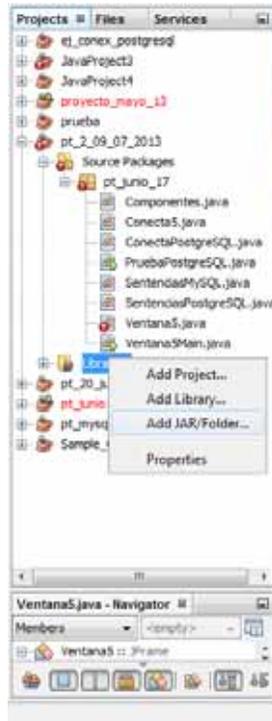


Figura 40

Elegiremos la ubicación donde tenemos guardados los “*jar*” de la biblioteca *iText*, y los agregamos todos al proyecto, observe la Figura 41. También agregaremos los “*jar*” necesarios para que pueda funcionar DB2 correctamente, así como lo indica la Figura 42.

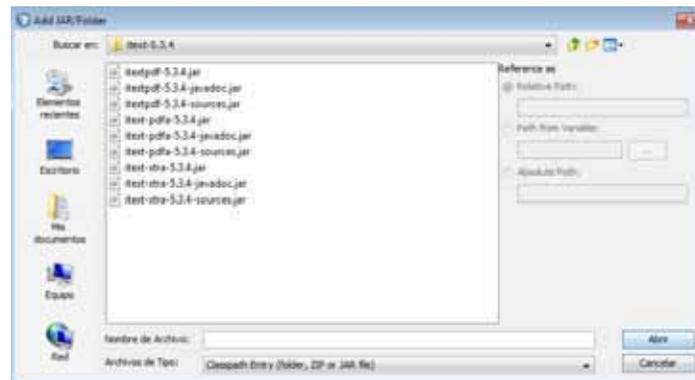


Figura 41

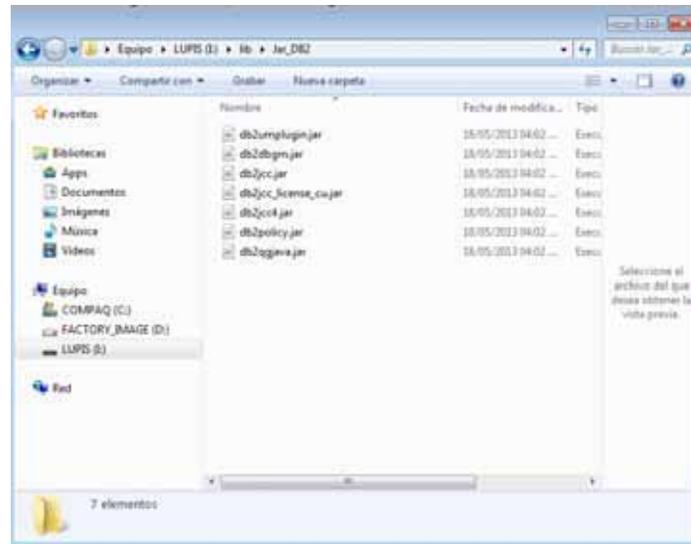


Figura 42

Si se agregaron de manera correcta los “jar”, podremos observarlos en el paquete de “Libraries”, tal como se observa en la Figura 43.

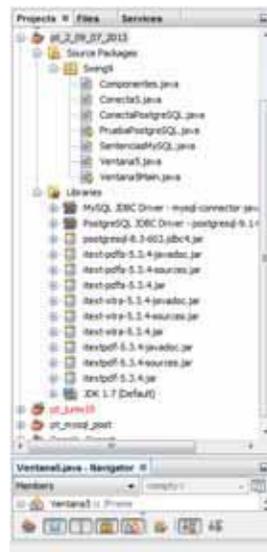
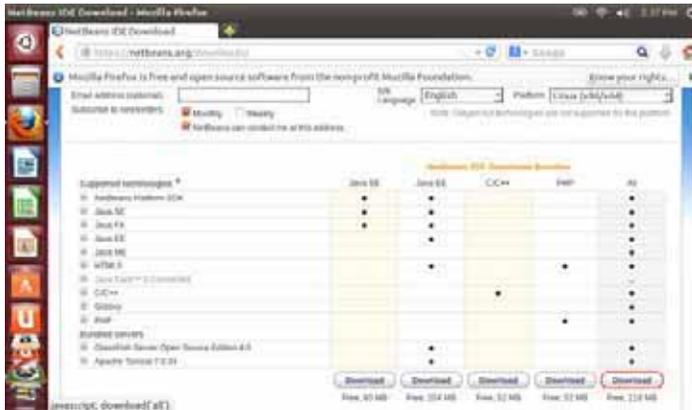


Figura 43

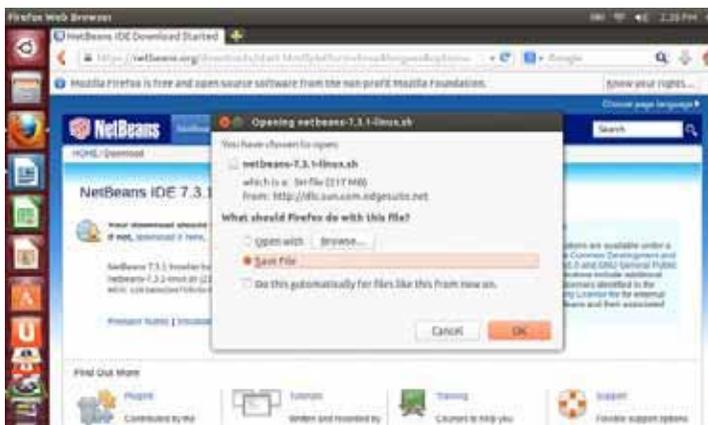
Ya que hayamos llegado a éste punto, el proyecto no deberá contener ningún error.

# Pasos para instalar este proyecto en NetBeans IDE 7.3.1 en Ubuntu 13.04 GNU/Linux

Descargamos netbeans de la pagina oficial



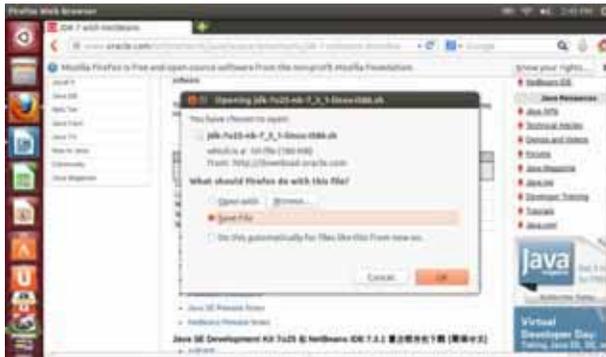
Guardamos el archivo a descargar



Descargamos el JDK



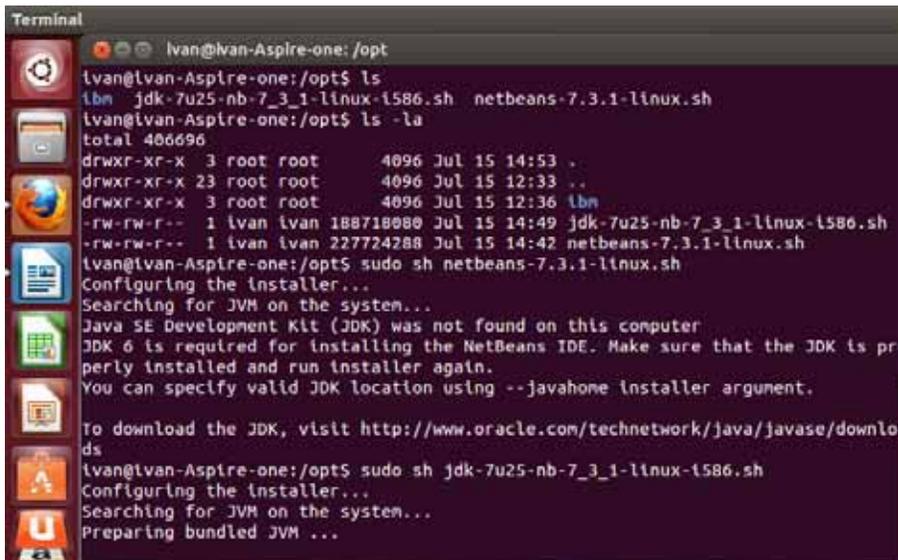
Guardamos el archivo a descargar



Nuestros archivos se han descargado exitosamente y guardado en nuestra carpeta Downloads



Descrmpimimos los archivos con el comando sudo tar -zxvf nombre-archivo



Ejecutamos

```
Terminal
ivan@ivan-Aspire-one: /opt
ivan@ivan-Aspire-one: /opt$ ls
lm  jdk-7u25-nb-7_3_1-linux-1586.sh  netbeans-7.3.1-linux.sh
ivan@ivan-Aspire-one: /opt$ ls -la
total 486696
drwxr-xr-x  3 root root    4096 Jul 15 14:53 .
drwxr-xr-x 23 root root    4096 Jul 15 12:33 ..
drwxr-xr-x  3 root root    4096 Jul 15 12:36 lm
-rw-rw-r--  1 ivan ivan 188718080 Jul 15 14:49 jdk-7u25-nb-7_3_1-linux-1586.sh
-rw-rw-r--  1 ivan ivan 227724288 Jul 15 14:42 netbeans-7.3.1-linux.sh
ivan@ivan-Aspire-one: /opt$ sudo sh netbeans-7.3.1-linux.sh
Configuring the installer...
Searching for JVM on the system...
Java SE Development Kit (JDK) was not found on this computer
JDK 6 is required for installing the NetBeans IDE. Make sure that the JDK is properly installed and run installer again.
You can specify valid JDK location using --javahome installer argument.
To download the JDK, visit http://www.oracle.com/technetwork/java/javase/downloads
ivan@ivan-Aspire-one: /opt$ sudo sh jdk-7u25-nb-7_3_1-linux-1586.sh
Configuring the installer...
Searching for JVM on the system...
Preparing bundled JVM ...
```

Se lanza el wizard



Los siguientes pasos son de igual forma a como se mostró en la instalación para Windows. Al final se muestra nuestra instalación correctamente



# Manual de Usuario

## Introducción

En este capítulo se muestra el manual de usuario del “Recopilador de configuración y estado de un manejador de base de datos relacional” con la finalidad de darle un mejor uso a nuestro sistema.

Esta aplicación contiene un programa que permite mostrar un reporte donde se contendrán informes detallados acerca de los parámetros de configuración y del esquema general de los diferentes manejadores de bases de datos relacionales tales como: MySQL, PostgreSQL, DB2 y Oracle.

## Objetivos

Permitir visualizar un reporte de la información de la configuración y estado de los manejadores de bases de datos relacionales: MySQL, PostgreSQL, DB2 y Oracle de forma fácil.

## Guía de uso

Para poder utilizar la aplicación de “Recopilador de configuración y estado de un manejador de base de datos relacional” basta con compilar y ejecutar la *Ventana5Main.java*, la Figura 43 nos muestra este proceso:

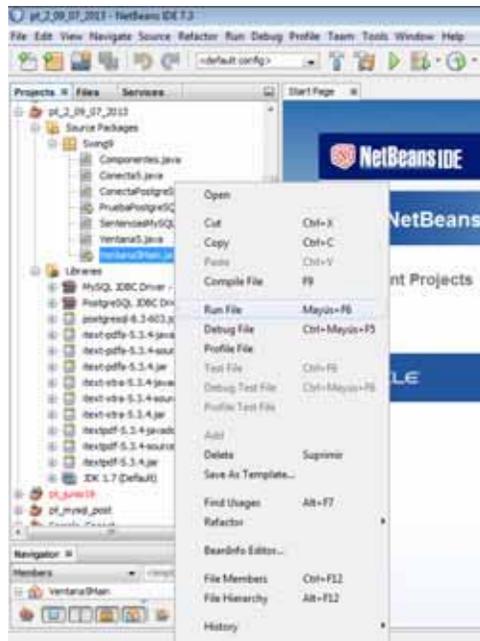


Figura 44

Ya que hayamos compilado y ejecutado el programa, aparecerá la pantalla principal de este proyecto, como indica la Figura 45

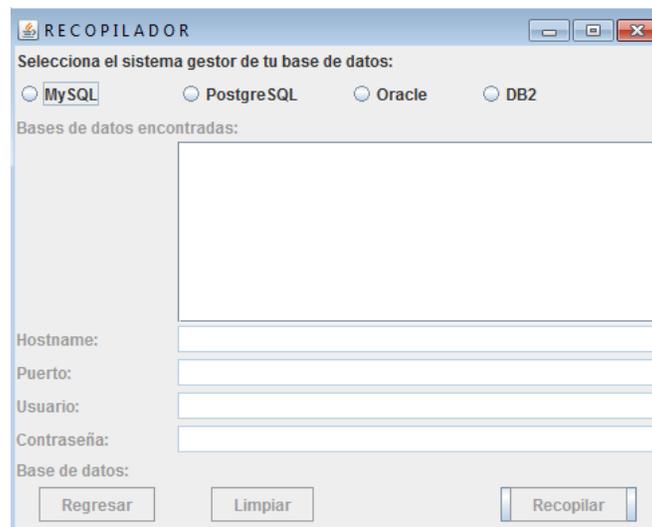


Figura 45

Antes de continuar, es necesario explicar cada uno de los componentes que integran esta aplicación, para ello observamos la Figura 46.

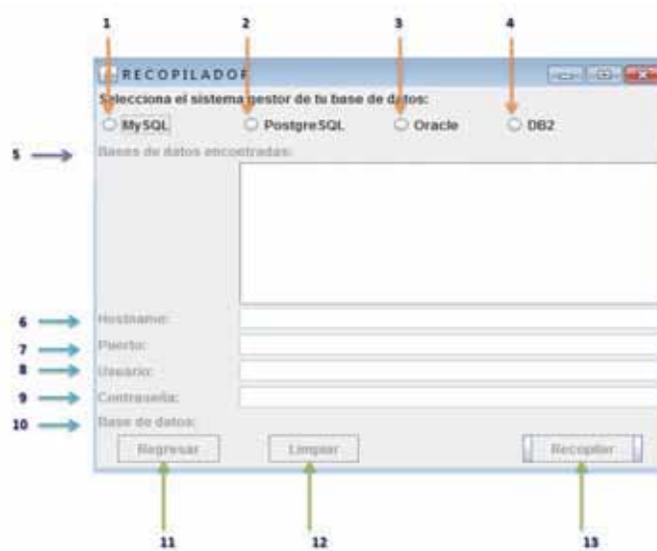


Figura 46

- Botón de radio para elegir el manejador MySQL
- Botón de radio para elegir el manejador PostgreSQL
- Botón de radio para elegir el manejador DB2
- Botón de radio para elegir el manejador Oracle
- Muestra las bases de datos encontradas en caso de que el usuario de la base de datos de MySQL tenga los privilegios necesarios para poder acceder a esta opción.
- El hostname es "localhost" si se conecta desde un script en el servidor o su nombre de dominio para conectarse remotamente. Si tiene problemas utilizando "localhost", puede en su lugar intentar con la dirección IP 127.0.0.1.
- Los puertos que utilizamos para conectar a la base de datos, son los puertos por "default" pero éstos pueden variar de acuerdo a la configuración que se tenga en cada manejador.
  - Puerto para MySQL: 3306
  - Puerto para PostgreSQL: 5432
  - Puerto para DB2: 50000
  - Puerto para Oracle: 1521
- En el campo "Usuario" se coloca el usuario de la base de datos.
- La contraseña es la que usted especificó cuándo creó el nombre de usuario.
- En este campo se indica el nombre de la base de datos.

- El botón “Regresar” se utilizará en caso de que se quiera retroceder una pantalla anterior.
- El botón “Limpiar” se utilizará en caso de que se haya tenido un error en la captura de los datos o se quieran llenar para realizar otra consulta.
- El botón “Recopilar” es el encargado de generar el reporte en documento con extensión *.pdf* en caso de que se hayan llenado de manera correcta los datos.

Ahora es momento de presentar como funciona este proyecto, para ello utilizaremos una Base de Datos Relacional llamada “*Collections*”. Los pasos que se ilustran a continuación funcionan de igual manera para los cuatro manejadores, es decir, para MySQL, PostgreSQL y Oracle.

Utilizaremos MySQL para ilustrar este proceso.

- Elegimos el botón de radio MySQL

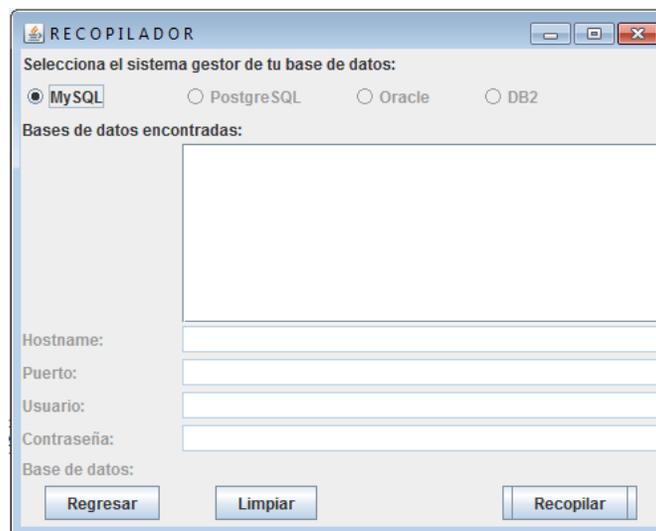


Figura 47

- En caso de que no se tengan los privilegios para poder ver la lista de las bases de datos que están alojadas en el sistema, simplemente nos aparecerá un mensaje el cual nos dice que “No hay ninguna base de datos”

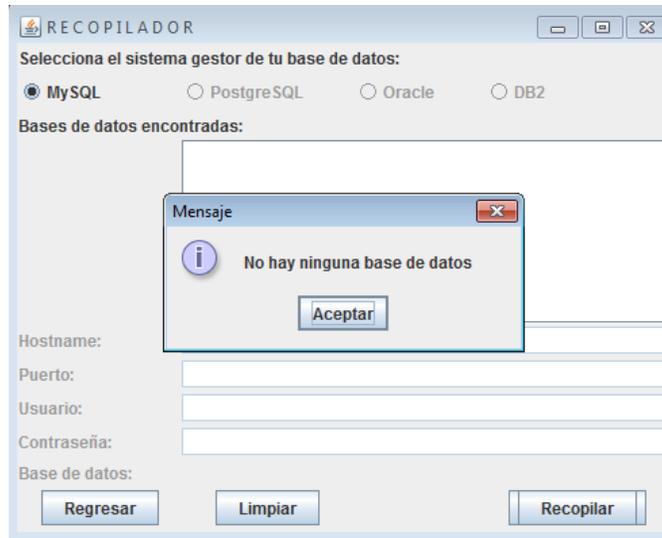


Figura 48

- Presionamos el botón “Limpiar” y nos aparece un mensaje que nos indica que todos los campos marcados con un asterisco (\*) deberán de ser llenados de manera obligatoria para poder recopilar la información que deseamos.

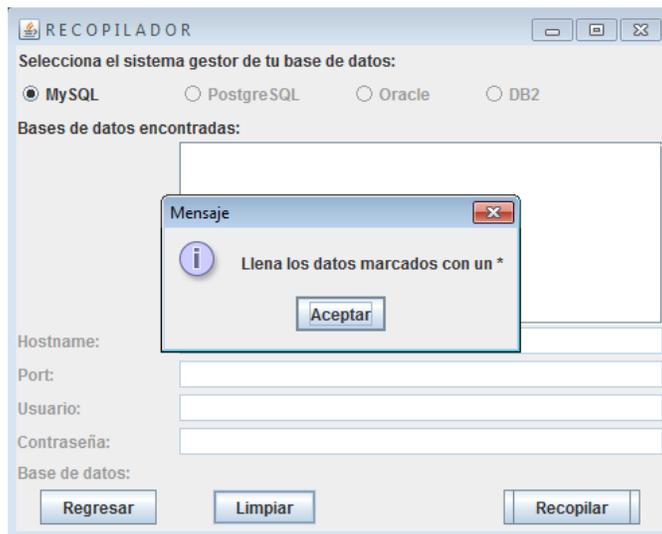


Figura 49

- Llenamos los campos marcados con asterisco.

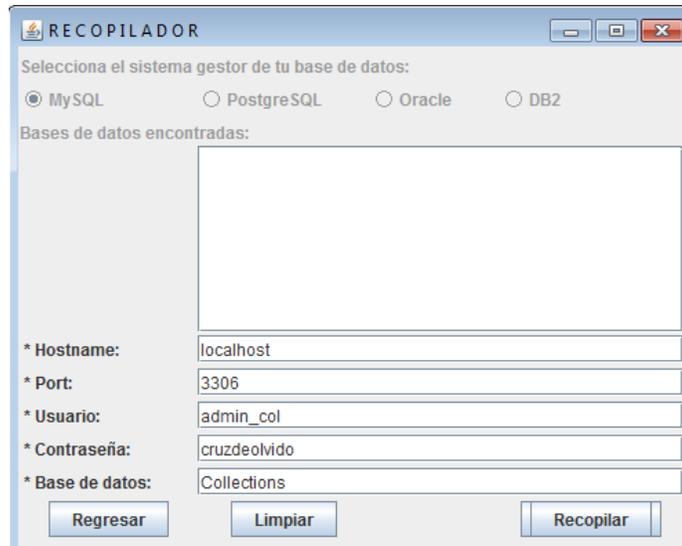


Figura 50

- Presionamos el botón “Recopilar” pero en caso de que haya habido un error en la captura de datos, la aplicación no nos dejará continuar hasta que éste se corrija, aunque ésta aplicación no te indicará el nombre del campo en el que se cometió el error, eso queda como trabajo para el usuario.

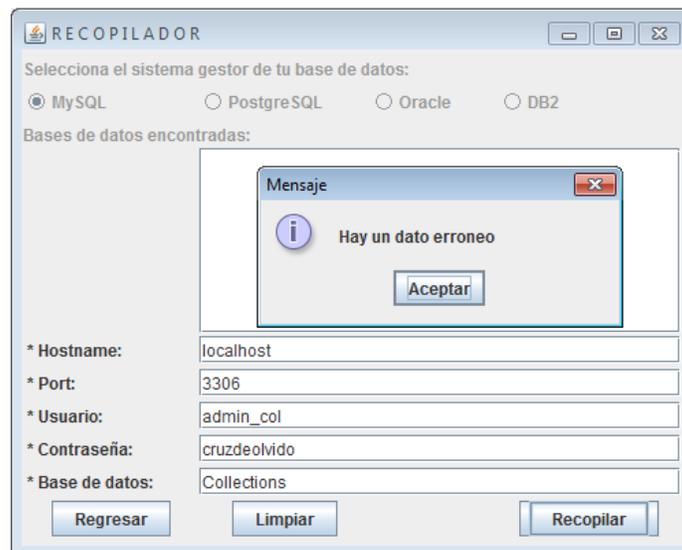


Figura 51

- Tendrá que checar los campos y corregir el dato.

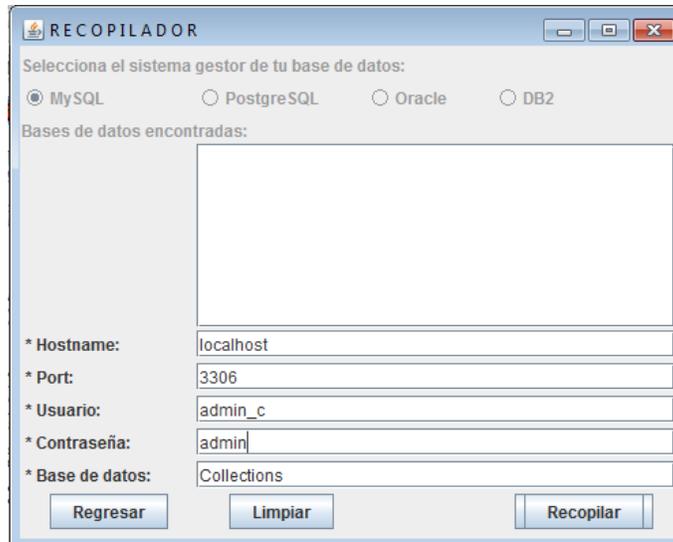


Figura 52

- Una vez que se haya corregido el error, nuevamente presionamos el botón “Recopilar” y si todos los datos están capturados de manera correcta, aparecerá un mensaje con la leyenda “Archivo PDF generado con éxito!!”. Damos clic en “Aceptar” y automáticamente el archivo se guardará en directorio de trabajo actual del usuario.

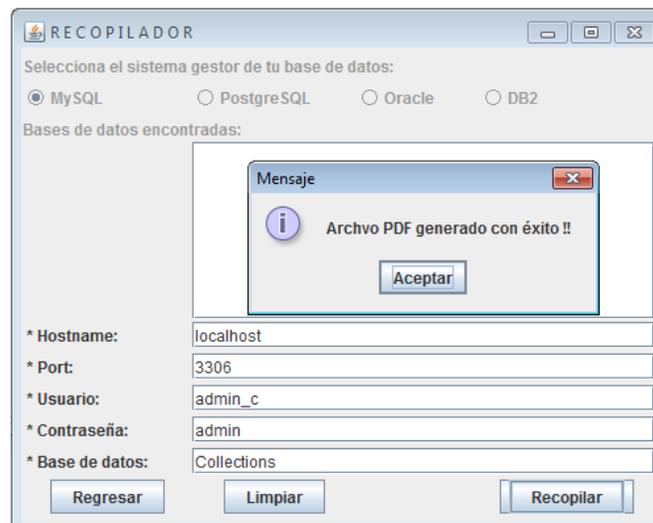


Figura 53

- Nos dirigimos al directorio de trabajo actual del usuario y observamos que ahí se encuentra el archivo generado.

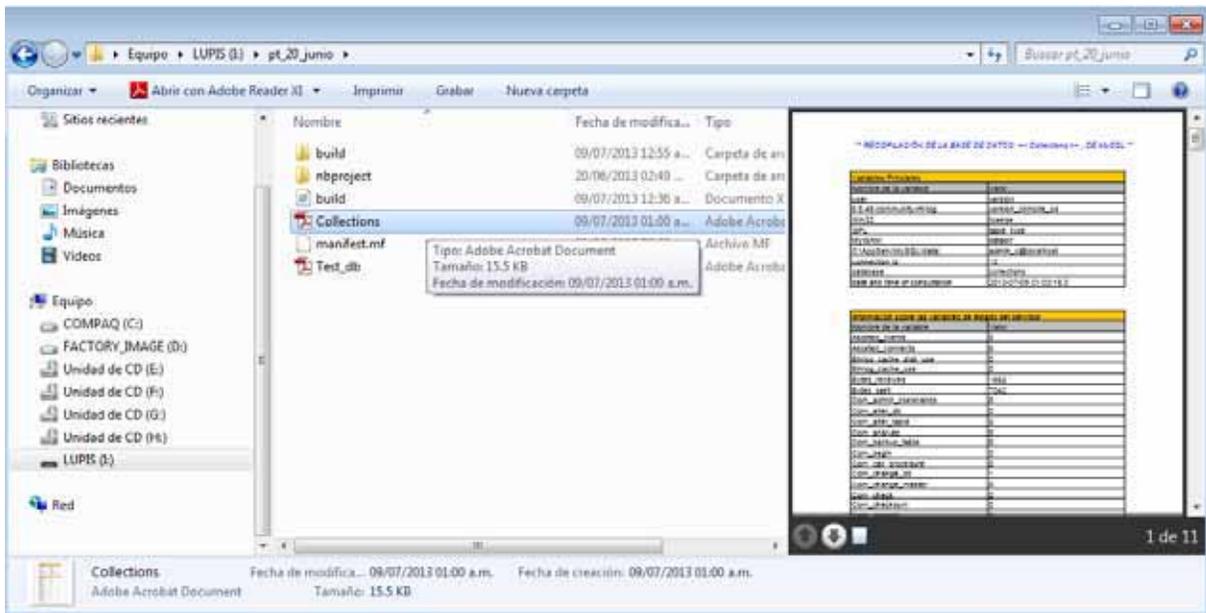


Figura 54

- Abrimos el archivo y checamos la información para los fines que nos convengan.

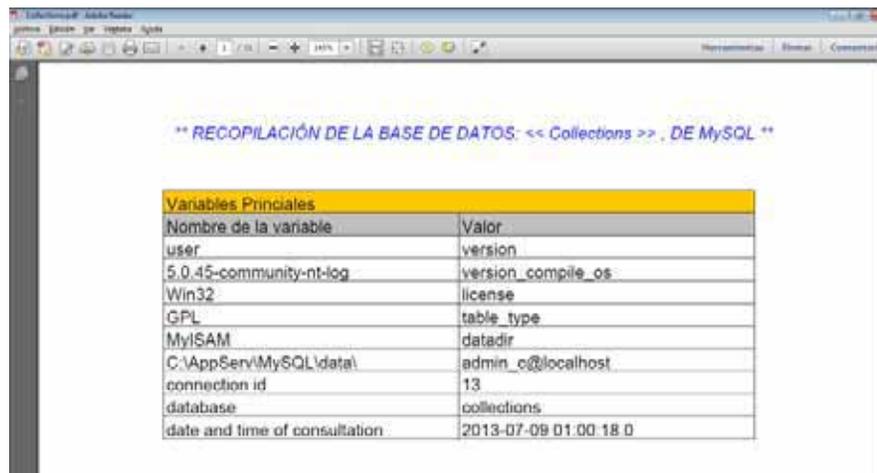


Figura 55

## Apéndice

### Usuarios de las Bases de Datos

El usuario con el que se accede a la información de la base de datos no es el mismo para todos los manejadores, una base de datos puede tener usuarios distintos y con distintos privilegios.

Ilustraremos a continuación los cuatro manejadores con sus respectivos nombres de usuarios para que quede más claro este punto.

- **MySQL v5.2**

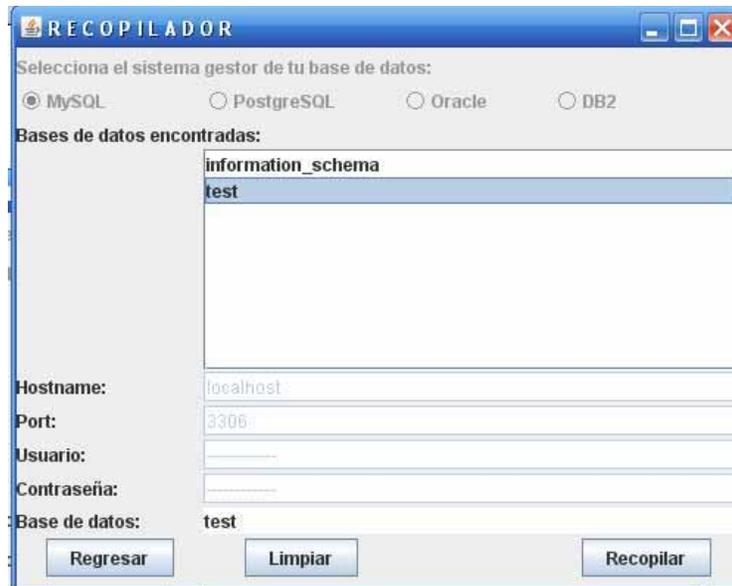


Figura 56

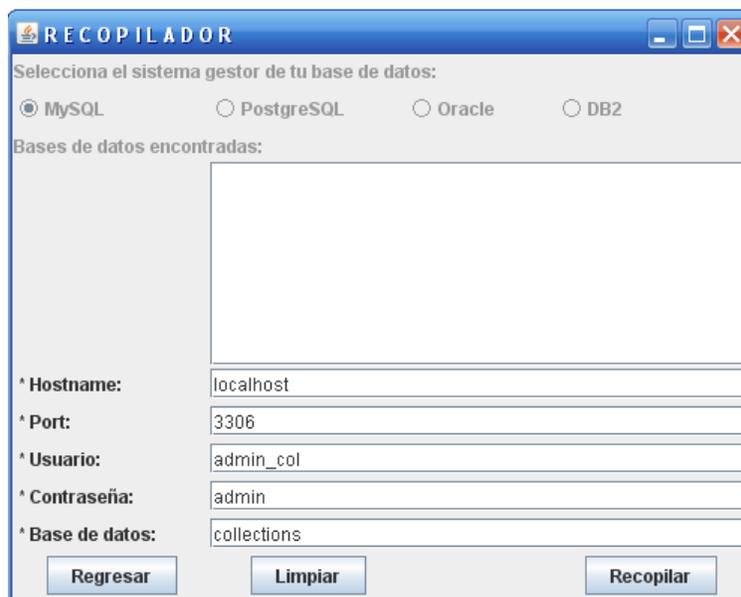


Figura 57

- **PostgreSQL 9.2**

The screenshot shows a window titled "RECOPILADOR" with a blue header. Below the header, the text "Selecciona el sistema gestor de tu base de datos:" is followed by four radio buttons: "MySQL", "PostgreSQL" (which is selected), "Oracle", and "DB2". Below this is a large empty rectangular area. Underneath are five input fields: "Hostname:" with "localhost", "Port:" with "5432", "Usuario:" with "postgres", "Contraseña:" with "dragoon7J70", and "Base de datos:" with "collections". At the bottom are three buttons: "Regresar", "Limpiar", and "Recopilar".

Figura 58

- **DB2 express-c v10.1**

The screenshot shows the same "RECOPILADOR" window, but now "DB2" is selected with the radio button. The "Base de datos:" field now contains "collect". The other fields remain the same: "Hostname:" is "localhost", "Port:" is "50000", "Usuario:" is "admin123", and "Contraseña:" is "dragoon7J70". The buttons "Regresar", "Limpiar", and "Recopilar" are still present at the bottom.

Figura 59

The screenshot shows a window titled "RECOPILADOR" with a subtitle "Selecciona el sistema gestor de tu base de datos:". There are four radio buttons for database systems: MySQL, PostgreSQL, Oracle, and DB2. The DB2 radio button is selected. Below the radio buttons is a large empty rectangular area. Underneath this area are five text input fields: "Hostname" with "localhost", "Port" with "50000", "Usuario" with "admin123", "Contraseña" with "dragoon7J70", and "Base de datos" with "sample". At the bottom of the window are three buttons: "Regresar", "Limpiar", and "Recopilar".

Figura 60

- Oracle database 11g express edition release v11.2.0.2.0

The screenshot shows the same "RECOPILADOR" window, but now the "Oracle" radio button is selected. The "Base de datos" field is empty. The other fields remain the same: "Hostname" is "localhost", "Port" is "1521", "Usuario" is "system", and "Contraseña" is "dragoon7J70". The "Regresar", "Limpiar", and "Recopilar" buttons are still present at the bottom.

Figura 61



Figura 62

En la siguiente imagen podemos observar todos los Archivos con extensión *.pdf*.

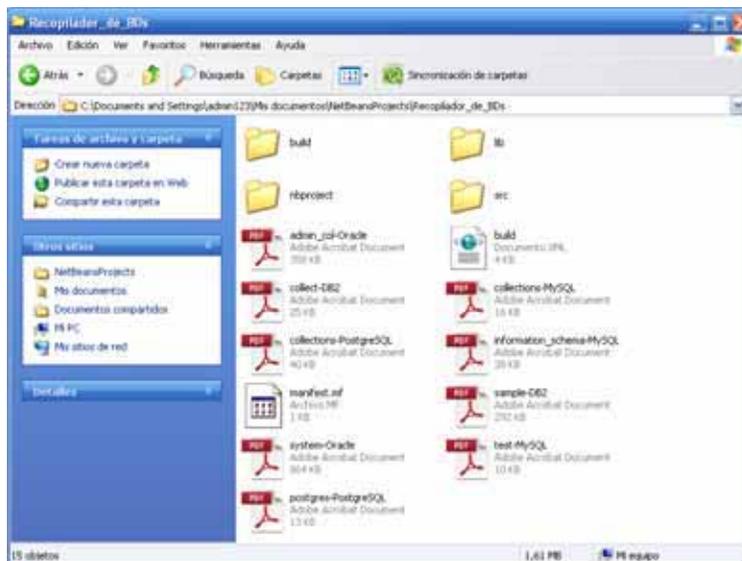


Figura 63

## Base de Datos utilizada para las pruebas

Para realizar las pruebas de los 4 manejadores, utilizamos una base de datos llamada “collections”, para cada uno de ellos se tuvieron que generar los “scripts”, éstos nos sirven para exportar la base de datos en un archivo con extensión *.sql*, además necesita dos parámetros de entrada, el nombre de usuario y la contraseña para poder acceder a la base de datos.

Para realizar las pruebas de los 4 manejadores, utilizamos una base de datos llamada “collections”. Sabemos que existen muchas formas de crear bases de datos en los manejadores pero una de las mejores formas es utilizando scripts, es decir, un texto con instrucciones. En caso de que se quieran guardar estos scripts, los deberían de guardar con la extensión *sql*.

A continuación se muestran los “scripts” para cada uno de los manejadores:

### Script de la base de datos collections de MySQL

```
-- phpMyAdmin SQL Dump
-- version 2.10.3
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 11-12-2011 a las 17:48:01
-- Versión del servidor: 5.0.51
-- Versión de PHP: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Base de datos: `collections`
--
-----

--
-- Estructura de tabla para la tabla `catt_tipo_recurso`
--

CREATE TABLE `catt_tipo_recurso` (
  `id_tipo_recurso` varchar(3) NOT NULL,
  `descripcion_tipo_recurso` varchar(20) NOT NULL,
  PRIMARY KEY (`id_tipo_recurso`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `catt_tipo_recurso`
--

INSERT INTO `catt_tipo_recurso` VALUES ('rev', 'revista');
INSERT INTO `catt_tipo_recurso` VALUES ('lib', 'libro');
INSERT INTO `catt_tipo_recurso` VALUES ('enc', 'enciclopedia');
```

```

INSERT INTO `catt_tipo_recurso` VALUES ('dic', 'diccionario');
INSERT INTO `catt_tipo_recurso` VALUES ('cd', 'cdaudio');
INSERT INTO `catt_tipo_recurso` VALUES ('dvd', 'pelicula');

-----

--
-- Estructura de tabla para la tabla `contacto`
--

CREATE TABLE `contacto` (
  `id_contacto` int(11) NOT NULL auto_increment,
  `nombre_contacto` varchar(50) NOT NULL,
  `mail_contacto` varchar(50) default NULL,
  PRIMARY KEY (`id_contacto`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=9 ;

--
-- Volcar la base de datos para la tabla `contacto`
--

INSERT INTO `contacto` VALUES (1, 'Ricardo Rodriguez Nogal', 'rrn1979@gmail.com');
INSERT INTO `contacto` VALUES (5, 'Pedro Martinez Garcia', 'pmg@mail.com');
INSERT INTO `contacto` VALUES (3, 'Carlos Magaña Salazar', 'cms@hotmail.com');
INSERT INTO `contacto` VALUES (4, 'Chokis Peralta Ruiz', 'cpr@hotmail.com');
INSERT INTO `contacto` VALUES (7, 'Ayumu Kurosawa', 'ayumuk@mail.com');
INSERT INTO `contacto` VALUES (8, 'Veronica Bonilla Acevedo', 'vba@mail.com');

-----

--
-- Estructura de tabla para la tabla `prestamo`
--

CREATE TABLE `prestamo` (
  `id_prestamo` int(11) NOT NULL auto_increment,
  `id_recurso` varchar(20) NOT NULL,
  `id_contacto` int(11) NOT NULL,
  `inicio_prestamo` date default NULL,
  `fin_prestamo` date default NULL,
  PRIMARY KEY (`id_prestamo`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

--
-- Volcar la base de datos para la tabla `prestamo`
--

INSERT INTO `prestamo` VALUES (2, '789682', 1, '2011-12-11', '2011-12-30');
INSERT INTO `prestamo` VALUES (3, '341298', 1, '2011-12-11', '2011-12-21');
INSERT INTO `prestamo` VALUES (4, '111111', 8, '2011-12-11', '2011-12-31');

-----

--
-- Estructura de tabla para la tabla `recurso`
--

CREATE TABLE `recurso` (
  `id_recurso` varchar(20) NOT NULL,
  `id_tipo_recurso` varchar(3) NOT NULL,
  `titulo_recurso` varchar(100) NOT NULL,
  `clasificacion_recurso` varchar(50) default NULL,

```

```

`edicion_recurso` varchar(3) default NULL,
`editorial_recurso` varchar(20) default NULL,
`autor_recurso` varchar(100) default NULL,
`genero_recurso` varchar(100) default NULL,
`prestado_recurso` tinyint(1) NOT NULL default '0',
PRIMARY KEY (`id_recurso`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `recurso`
--

INSERT INTO `recurso` VALUES ('123456', 'lib', 'Los elementos de la Noche', 'Poesia', '4', 'era', 'Jose Emilio Pacheco',
NULL, 0);
INSERT INTO `recurso` VALUES ('789682', 'lib', 'El Principio del Placer', 'Cuento', '4', 'Serie del volador', 'Jose Emilio
Pacheco', NULL, 1);
INSERT INTO `recurso` VALUES ('509761', 'rev', 'Dibujarte', 'Dibujo', '7', NULL, NULL, NULL, 0);
INSERT INTO `recurso` VALUES ('341298', 'enc', 'Diccionario Larousse', 'Ciencia', '12', NULL, NULL, NULL, 1);
INSERT INTO `recurso` VALUES ('782687', 'rev', 'Arqueologia Mexicana', 'Arqueologia', '12', NULL, NULL, NULL,
0);

```

## Script de la base de datos collections de PosrtgreSQL

```

CREATE TABLE catt_tipo_recurso (
  id_tipo_recurso varchar(3) NOT NULL,
  descripcion_tipo_recurso varchar(20) NOT NULL,
  PRIMARY KEY (id_tipo_recurso)
);

INSERT INTO catt_tipo_recurso VALUES ('rev', 'revista');
INSERT INTO catt_tipo_recurso VALUES ('lib', 'libro');
INSERT INTO catt_tipo_recurso VALUES ('enc', 'enciclopedia');
INSERT INTO catt_tipo_recurso VALUES ('dic', 'diccionario');
INSERT INTO catt_tipo_recurso VALUES ('cd', 'cdaudio');
INSERT INTO catt_tipo_recurso VALUES ('dvd', 'pelicula');

-----

--
-- Estructura de tabla para la tabla contacto
--

CREATE TABLE contacto (
  id_contacto serial NOT NULL,
  nombre_contacto varchar(50) NOT NULL,
  mail_contacto varchar(50) default NULL,
  PRIMARY KEY (id_contacto)
);

--
-- Volcar la base de datos para la tabla contacto
--

INSERT INTO contacto VALUES (1, 'Ricardo Rodriguez Nogal', 'rrn1979@gmail.com');
INSERT INTO contacto VALUES (5, 'Pedro Martinez Garcia', 'pmg@mail.com');
INSERT INTO contacto VALUES (3, 'Carlos Magaña Salazar', 'cms@hotmail.com');
INSERT INTO contacto VALUES (4, 'Chokis Peralta Ruiz', 'cpr@hotmail.com');
INSERT INTO contacto VALUES (7, 'Ayumu Kurosawa', 'ayumuk@mail.com');

```

```
INSERT INTO contacto VALUES (8, 'Veronica Bonilla Acevedo', 'vba@mail.com');
```

```
CREATE TABLE prestamo (  
  id_prestamo serial NOT NULL,  
  id_recurso varchar(20) NOT NULL,  
  id_contacto numeric(11) NOT NULL,  
  inicio_prestamo date default NULL,  
  fin_prestamo date default NULL,  
  PRIMARY KEY (id_prestamo)  
);
```

```
INSERT INTO prestamo VALUES (2, '789682', 1, '2011-12-11', '2011-12-30');  
INSERT INTO prestamo VALUES (3, '341298', 1, '2011-12-11', '2011-12-21');  
INSERT INTO prestamo VALUES (4, '111111', 8, '2011-12-11', '2011-12-31');
```

```
CREATE TABLE recurso (  
  id_recurso varchar(20) NOT NULL,  
  id_tipo_recurso varchar(3) NOT NULL,  
  titulo_recurso varchar(100) NOT NULL,  
  clasificacion_recurso varchar(50) default NULL,  
  edicion_recurso varchar(3) default NULL,  
  editorial_recurso varchar(20) default NULL,  
  autor_recurso varchar(100) default NULL,  
  genero_recurso varchar(100) default NULL,  
  prestado_recurso smallint NOT NULL default '0',  
  PRIMARY KEY (id_recurso)  
);
```

```
INSERT INTO recurso VALUES ('123456', 'lib', 'Los elementos de la Noche', 'Poesia', '4', 'era', 'Jose Emilio Pacheco',  
NULL, 0);  
INSERT INTO recurso VALUES ('789682', 'lib', 'El Principio del Placer', 'Cuento', '4', 'Serie del volador', 'Jose Emilio  
Pacheco', NULL, 1);  
INSERT INTO recurso VALUES ('509761', 'rev', 'Dibujarte', 'Dibujo', '7', NULL, NULL, NULL, 0);  
INSERT INTO recurso VALUES ('341298', 'enc', 'Diccionario Larousse', 'Ciencia', '12', NULL, NULL, NULL, 1);  
INSERT INTO recurso VALUES ('782687', 'rev', 'Arqueologia Mexicana', 'Arqueologia', '12', NULL, NULL, NULL, 0);
```

## Script de la base de datos collections de DB2

```
-- Base de datos: `collections`  
--  
-----  
--  
-- Estructura de tabla para la tabla `catt_tipo_recurso`  
--  
create database COLLECT;  
connect to COLLECT;  
  
CREATE TABLE catt_tipo_recurso (  
  id_tipo_recurso varchar(3) NOT NULL,  
  descripcion_tipo_recurso varchar(20) NOT NULL,  
  PRIMARY KEY (id_tipo_recurso)  
);  
  
--  
-- Volcar la base de datos para la tabla `catt_tipo_recurso`
```

```

--
INSERT INTO catt_tipo_recurso VALUES ('rev', 'revista');
INSERT INTO catt_tipo_recurso VALUES ('lib', 'libro');
INSERT INTO catt_tipo_recurso VALUES ('enc', 'enciclopedia');
INSERT INTO catt_tipo_recurso VALUES ('dic', 'diccionario');
INSERT INTO catt_tipo_recurso VALUES ('cd', 'cdaudio');
INSERT INTO catt_tipo_recurso VALUES ('dvd', 'pelicula');

-----

--
-- Estructura de tabla para la tabla `contacto`
--

CREATE TABLE contacto (
  id_contacto INTEGER NOT NULL,
  nombre_contacto varchar(50) NOT NULL,
  mail_contacto varchar(50) default NULL,
  PRIMARY KEY (id_contacto)
);

create sequence contacto_seq
start with 1
increment by 3;

--
-- Volcar la base de datos para la tabla `contacto`
--

INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Ricardo Rodriguez Nogal', 'rrn1979@gmail.com');
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Pedro Martinez Garcia', 'pmg@mail.com');
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Carlos Magaña Salazar', 'cms@hotmail.com');
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Chokis Peralta Ruiz', 'cpr@hotmail.com');
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Ayumu Kurosawa', 'ayumuk@mail.com');
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Veronica Bonilla Acevedo', 'vba@mail.com');

-----

--
-- Estructura de tabla para la tabla `prestamo`
--

CREATE TABLE prestamo (
  id_prestamo integer NOT NULL,
  id_recurso varchar(20) NOT NULL,
  id_contacto integer NOT NULL,
  inicio_prestamo date default NULL,
  fin_prestamo date default NULL,
  PRIMARY KEY (id_prestamo)
);

create sequence prestamo_seq
start with 2
increment by 1;

--
-- Volcar la base de datos para la tabla `prestamo`
--

INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '789682', 1, '2011-12-11', '2011-12-30');
INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '341298', 1, '2011-12-11', '2011-12-21');

```

```
INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '111111', 8, '2011-12-11', '2011-12-31');
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `recurso`  
--
```

```
CREATE TABLE recurso (  
  id_recurso varchar(20) NOT NULL,  
  d_tipo_recurso varchar(3) NOT NULL,  
  titulo_recurso varchar(100) NOT NULL,  
  clasificacion_recurso varchar(50) default NULL,  
  edicion_recurso varchar(3) default NULL,  
  editorial_recurso varchar(20) default NULL,  
  autor_recurso varchar(100) default NULL,  
  genero_recurso varchar(100) default NULL,  
  prestado_recurso SMALLINT default '0' NOT NULL ,  
  PRIMARY KEY (id_recurso)  
);
```

```
--  
-- Volcar la base de datos para la tabla `recurso`  
--
```

```
INSERT INTO recurso VALUES ('123456', 'lib', 'Los elementos de la Noche', 'Poesia', '4', 'era', 'Jose Emilio Pacheco',  
NULL, 0);  
INSERT INTO recurso VALUES ('789682', 'lib', 'El Principio del Placer', 'Cuento', '4', 'Serie del volador', 'Jose Emilio  
Pacheco', NULL, 1);  
INSERT INTO recurso VALUES ('509761', 'rev', 'Dibujarte', 'Dibujo', '7', NULL, NULL, NULL, 0);  
INSERT INTO recurso VALUES ('341298', 'enc', 'Diccionario Larousse', 'Ciencia', '12', NULL, NULL, NULL, 1);  
INSERT INTO recurso VALUES ('782687', 'rev', 'Arqueologia Mexicana', 'Arqueologia', '12', NULL, NULL, NULL, 0);
```

## Script de la base de datos collections de Oracle

```
--  
-- Base de datos: `collections`  
--
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `catt_tipo_recurso`  
--
```

```
CREATE TABLE catt_tipo_recurso (  
  id_tipo_recurso varchar(3) NOT NULL,  
  descripcion_tipo_recurso varchar(20) NOT NULL,  
  PRIMARY KEY (id_tipo_recurso)  
);
```

```
--  
-- Volcar la base de datos para la tabla `catt_tipo_recurso`  
--
```

```
INSERT INTO catt_tipo_recurso VALUES ('rev', 'revista');  
INSERT INTO catt_tipo_recurso VALUES ('lib', 'libro');  
INSERT INTO catt_tipo_recurso VALUES ('enc', 'enciclopedia');  
INSERT INTO catt_tipo_recurso VALUES ('dic', 'diccionario');  
INSERT INTO catt_tipo_recurso VALUES ('cd', 'cdaudio');
```

```
INSERT INTO catt_tipo_recurso VALUES ('dvd', 'pelicula');
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `contacto`  
--
```

```
CREATE TABLE contacto (  
  id_contacto NUMBER,  
  nombre_contacto varchar(50) NOT NULL,  
  mail_contacto varchar(50) default NULL,  
  PRIMARY KEY (id_contacto)  
);
```

```
create sequence contacto_seq  
start with 1  
increment by 3;
```

```
--  
-- Volcar la base de datos para la tabla `contacto`  
--
```

```
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Ricardo Rodriguez Nogal', 'rrn1979@gmail.com');  
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Pedro Martinez Garcia', 'pmg@mail.com');  
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Carlos Magaña Salazar', 'cms@hotmail.com');  
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Chokis Peralta Ruiz', 'cpr@hotmail.com');  
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Ayumu Kurosawa', 'ayumuk@mail.com');  
INSERT INTO contacto VALUES (contacto_seq.NextVal, 'Veronica Bonilla Acevedo', 'vba@mail.com');
```

```
-----
```

```
--  
-- Estructura de tabla para la tabla `prestamo`  
--
```

```
CREATE TABLE prestamo (  
  id_prestamo NUMBER,  
  id_recurso varchar(20) NOT NULL,  
  id_contacto NUMBER(11) NOT NULL,  
  inicio_prestamo date default NULL,  
  fin_prestamo date default NULL,  
  PRIMARY KEY (id_prestamo)  
);
```

```
create sequence prestamo_seq  
start with 2  
increment by 1;
```

```
--  
-- Volcar la base de datos para la tabla `prestamo`  
--
```

```
-- Formato para date: "DD-MON-YY"
```

```
INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '789682', 1, sysdate, sysdate);  
INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '341298', 1, '11-12-13', '21-12-12');  
INSERT INTO prestamo VALUES (prestamo_seq.NextVal, '111111', 8, '11-12-13', '31-12-13');
```

```
-----
```

```
--
```

```

-- Estructura de tabla para la tabla `recurso`
--
CREATE TABLE recurso (
  id_recurso varchar(20) NOT NULL,
  id_tipo_recurso varchar(3) NOT NULL,
  titulo_recurso varchar(100) NOT NULL,
  clasificacion_recurso varchar(50) default NULL,
  edicion_recurso varchar(3) default NULL,
  editorial_recurso varchar(20) default NULL,
  autor_recurso varchar(100) default NULL,
  genero_recurso varchar(100) default NULL,
  prestado_recurso NUMBER(1) default '0' NOT NULL,
  PRIMARY KEY (id_recurso)
);

--
-- Volcar la base de datos para la tabla `recurso`
--

INSERT INTO recurso VALUES ('123456', 'lib', 'Los elementos de la Noche', 'Poesia', '4', 'era', 'Jose Emilio Pacheco',
NULL, 0);
INSERT INTO recurso VALUES ('789682', 'lib', 'El Principio del Placer', 'Cuento', '4', 'Serie del volador', 'Jose Emilio
Pacheco', NULL, 1);
INSERT INTO recurso VALUES ('509761', 'rev', 'Dibujarte', 'Dibujo', '7', NULL, NULL, NULL, 0);
INSERT INTO recurso VALUES ('341298', 'enc', 'Diccionario Larousse', 'Ciencia', '12', NULL, NULL, NULL, 1);
INSERT INTO recurso VALUES ('782687', 'rev', 'Arqueologia Mexicana', 'Arqueologia', '12', NULL, NULL, NULL, 0);

```

## Estructura de la base de datos de prueba “collections”

A continuación se muestra la Estructura de la base de datos llamada collections, se observa que tiene cuatro tablas: catt\_tipo\_recurso, contacto, préstamo y recurso. Esta estructura la obtuvimos solamente utilizando PhpMyAdmin pero no debe variar demasiado en comparación con la estructura de otro manejador.



Además se muestra la información detallada de cada una de las tablas.

🏠 Servidor: localhost ▶ 📁 Base de datos: collections

### catt\_tipo\_recurso

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_tipo_recurso	varchar(3)	No		
descripcion_tipo_recurso	varchar(20)	No		

### contacto

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_contacto	int(11)	No		
nombre_contacto	varchar(50)	No		
mail_contacto	varchar(50)	Si	NULL	

📄 Últimas noticias

### prestamo

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_prestamo	int(11)	No		
id_recurso	varchar(20)	No		
id_contacto	int(11)	No		
inicio_prestamo	date	Si	NULL	
fin_prestamo	date	Si	NULL	

### recurso

Campo	Tipo	Nulo	Predeterminado	Comentarios
id_recurso	varchar(20)	No		
id_tipo_recurso	varchar(3)	No		
titulo_recurso	varchar(100)	No		
clasificacion_recurso	varchar(50)	Si	NULL	
edicion_recurso	varchar(3)	Si	NULL	
editorial_recurso	varchar(20)	Si	NULL	
autor_recurso	varchar(100)	Si	NULL	
genero_recurso	varchar(100)	Si	NULL	
prestado_recurso	tinyint(1)	No	0	

## Salida de una corrida de la aplicación

En el siguiente apartado mostraremos el contenido del un archivo *.pdf* que fue generado, para ejemplificar utilizaremos el manejador MySQL.

En las celdas iluminadas de color amarillo, se especifica a grandes rasgos el contenido de las tablas. El contenido de las tablas está organizado de la siguiente manera:

- Variables principales
- Información sobre las variables de estado del servidor
- Tablas existentes
- Información general acerca de cada tabla
- Información de las columnas de cada tabla
- Información del estado de integridad de cada tabla

- Información de índice de cada tabla

(El contenido del archivo generado puede variar, ya que dependerá de los privilegios que el usuario de la base de datos tenga)

**\*\* RECOPIACIÓN DE LA BASE DE DATOS: << collections >> , DE MySQL \*\***

Propiedades del sistema	
Nombre de la variable	Valor
OS name	Windows XP
OS arch	x86
OS version	5.1
User's account name	admin123
User's home directory	C:\Documents and Settings\admin123
User's current working directory	C:\Documents and Settings\admin123\Mis documentos\NetBeansProjects\Recopilador de BDs

Variables Principales	
Nombre de la variable	Valor
version	5.0.51b-community-nt-log
version compile os	Win32
license	GPL
table type	MyISAM
datadir	C:\AppServ\MySQL\data\
user	admin_col@localhost
connection id	20
database	collections
date and time of consultation	2013-07-17 08:00:14.0

Información sobre las variables de estado del servidor	
Nombre de la variable	Valor
Aborted clients	0
Aborted connects	1
Binlog cache disk use	0
Binlog cache use	0
Bytes received	1662
Bytes sent	7041
Com admin commands	0
Com alter db	0
Com alter table	0
Com analyze	0
Com backup table	0
Com begin	0
Com call procedure	0
Com change db	1
Com change master	0
Com check	0
Com checksum	0
Com commit	0
Com create db	0
Com create function	0
Com create index	0
Com create table	0
Com create user	0
Com dealloc sql	0
Com delete	0
Com delete multi	0
Com do	0
Com drop db	0
Com drop function	0

Com drop index	0
Com drop table	0
Com drop user	0
Com execute sql	0
Com flush	0
Com grant	0
Com ha close	0
Com ha open	0
Com ha read	0
Com help	0
Com insert	0
Com insert select	0
Com kill	0
Com load	0
Com load master data	0
Com load master table	0
Com lock tables	0
Com optimize	0
Com preload keys	0
Com prepare sql	0
Com purge	0
Com purge before date	0
Com rename table	0
Com repair	0
Com replace	0
Com replace select	0
Com reset	0
Com restore table	0
Com revoke	0
Com revoke all	0
Com rollback	0
Com savepoint	0
Com select	6
Com set option	4
Com show binlog events	0
Com show binlogs	0
Com show charsets	0
Com show collations	1
Com show column types	0
Com show create db	0
Com show create table	0
Com show databases	0
Com show errors	0
Com show fields	0
Com show grants	0
Com show innodb status	0
Com show keys	0
Com show logs	0

Com show master status	0
Com show ndb status	0
Com show new master	0
Com show open tables	0
Com show privileges	0
Com show processlist	0
Com show slave hosts	0
Com show slave status	0
Com show status	1
Com show storage engines	0
Com show tables	0
Com show triggers	0
Com show variables	8
Com show warnings	1
Com slave start	0
Com slave stop	0
Com stmt close	0
Com stmt execute	0
Com stmt fetch	0
Com stmt prepare	0
Com stmt reset	0
Com stmt send long data	0
Com truncate	0
Com unlock tables	0
Com update	0
Com update multi	0
Com xa commit	0
Com xa end	0
Com xa prepare	0
Com xa recover	0
Com xa rollback	0
Com xa start	0
Compression	OFF
Connections	4
Created tmp disk tables	0
Created tmp files	0
Created tmp tables	8
Delayed errors	0
Delayed insert threads	0
Delayed writes	0
Flush commands	1
Handler commit	0
Handler delete	0
Handler discover	0
Handler prepare	0
Handler read first	0
Handler read key	0
Handler read next	0

Handler read prev	0
Handler read rnd	0
Handler read rnd next	364
Handler rollback	0
Handler savepoint	0
Handler savepoint rollback	0
Handler update	0
Handler write	489
InnoDB buffer pool pages data	0
InnoDB buffer pool pages dirty	0
InnoDB buffer pool pages flushed	0
InnoDB buffer pool pages free	0
InnoDB buffer pool pages latched	0
InnoDB buffer pool pages misc	0
InnoDB buffer pool pages total	0
InnoDB buffer pool read ahead rnd	0
InnoDB buffer pool read ahead seq	0
InnoDB buffer pool read requests	0
InnoDB buffer pool reads	0
InnoDB buffer pool wait free	0
InnoDB buffer pool write requests	0
InnoDB data fsyncs	0
InnoDB data pending fsyncs	0
InnoDB data pending reads	0
InnoDB data pending writes	0
InnoDB data read	0
InnoDB data reads	0
InnoDB data writes	0
InnoDB data written	0
InnoDB dblwr pages written	0
InnoDB dblwr writes	0
InnoDB log waits	0
InnoDB log write requests	0
InnoDB log writes	0
InnoDB os log fsyncs	0
InnoDB os log pending fsyncs	0
InnoDB os log pending writes	0
InnoDB os log written	0
InnoDB page size	0
InnoDB pages created	0
InnoDB pages read	0
InnoDB pages written	0
InnoDB row lock current waits	0
InnoDB row lock time	0
InnoDB row lock time avg	0
InnoDB row lock time max	0
InnoDB row lock waits	0
InnoDB rows deleted	0

Innodb rows inserted	0
Innodb rows read	0
Innodb rows updated	0
Key blocks not flushed	0
Key blocks unused	8981
Key blocks used	0
Key read requests	0
Key reads	0
Key write requests	0
Key writes	0
Last query cost	0.000000
Max used connections	2
Not flushed delayed rows	0
Open files	1
Open streams	0
Open tables	0
Opened tables	0
Prepared stmt count	0
Qcache free blocks	0
Qcache free memory	0
Qcache hits	0
Qcache inserts	0
Qcache lowmem prunes	0
Qcache not cached	0
Qcache queries in cache	0
Qcache total blocks	0
Questions	29
Rpl status	NULL
Select full join	0
Select full range join	0
Select range	0
Select range check	0
Select scan	8
Slave open temp tables	0
Slave retried transactions	0
Slave running	OFF
Slow launch threads	0
Slow queries	0
Sort merge passes	0
Sort range	0
Sort rows	0
Sort scan	0
Ssl accept renegotiates	0
Ssl accepts	0
Ssl callback cache hits	0
Ssl cipher	
Ssl cipher list	
Ssl client connects	0

Ssl connect renegotiates	0
Ssl ctx verify depth	0
Ssl ctx verify mode	0
Ssl default timeout	0
Ssl finished accepts	0
Ssl finished connects	0
Ssl session cache hits	0
Ssl session cache misses	0
Ssl session cache mode	NONE
Ssl session cache overflows	0
Ssl session cache size	0
Ssl session cache timeouts	0
Ssl sessions reused	0
Ssl used session cache entries	0
Ssl verify depth	0
Ssl verify mode	0
Ssl version	
Table locks immediate	12
Table locks waited	0
Tc log max pages used	0
Tc log page size	0
Tc log page waits	0
Threads cached	0
Threads connected	2
Threads created	2
Threads running	1
Uptime	33796
Uptime since flush status	33796

Tablas existentes
catt tipo recurso
contacto
prestamo
recurso

Información general acerca de cada tabla	
Name	catt tipo recurso
Engine	MyISAM
Version	10
Row format	Dynamic
Rows	6
Avg row length	20
Data length	120
Max data length	281474976710655
Index length	2048

Data free	0
Auto increment	
Create time	2013-06-10 23:03:54.0
Update time	2013-06-10 23:30:06.0
Check time	
Collation	utf8 general ci
Checksum	
Create options	
Comment	
Name	contacto
Engine	MyISAM
Version	10
Row format	Dynamic
Rows	6
Avg row length	46
Data length	280
Max data length	281474976710655
Index length	2048
Data free	0
Auto increment	9
Create time	2013-06-10 23:03:54.0
Update time	2013-06-10 23:30:07.0
Check time	
Collation	utf8 general ci
Checksum	
Create options	
Comment	
Name	prestamo
Engine	MyISAM
Version	10
Row format	Dynamic
Rows	3
Avg row length	28
Data length	84
Max data length	281474976710655
Index length	2048
Data free	0
Auto increment	5
Create time	2013-06-10 23:03:55.0
Update time	2013-06-10 23:30:07.0
Check time	
Collation	utf8 general ci
Checksum	
Create options	
Comment	

Name	recurso
Engine	MyISAM
Version	10
Row format	Dynamic
Rows	5
Avg row length	62
Data length	312
Max data length	281474976710655
Index length	2048
Data free	0
Auto increment	
Create time	2013-06-10 23:03:55.0
Update time	2013-06-10 23:30:07.0
Check time	
Collation	utf8_general_ci
Checksum	
Create options	
Comment	

Información de las columnas de la tabla: catt_tipo_recurso					
Field	Type	Null	Key	Default	Extra
id_tipo_recurso	varchar(3)	NO	PRI		
descripcion_tipo_recurso	varchar(20)	NO			

Información de las columnas de la tabla: contacto					
Field	Type	Null	Key	Default	Extra
id_contacto	int(11)	NO	PRI		auto_increment
nombre_contacto	varchar(50)	NO			
mail_contacto	varchar(50)	YES			

Información de las columnas de la tabla: prestamo					
Field	Type	Null	Key	Default	Extra
id_prestamo	int(11)	NO	PRI		auto_increment
id_recurso	varchar(20)	NO			
id_contacto	int(11)	NO			
inicio_prestamo	date	YES			

fin_prestamo	date	YES			
--------------	------	-----	--	--	--

Información de las columnas de la tabla: recurso					
Field	Type	Null	Key	Default	Extra
id_recurso	varchar(20)	NO	PRI		
id_tipo_recurso	varchar(3)	NO			
titulo_recurso	varchar(100)	NO			
clasificacion_recurso	varchar(50)	YES			
edicion_recurso	varchar(3)	YES			
editorial_recurso	varchar(20)	YES			
autor_recurso	varchar(100)	YES			
genero_recurso	varchar(100)	YES			
prestado_recurso	tinyint(1)	NO		0	

Información del estado de integridad de la tabla: catt_tipo_recurso			
Table	Op	Msg type	Msg text
collections.catt_tipo_recurso	check	status	OK

Información del estado de integridad de la tabla: contacto			
Table	Op	Msg type	Msg text
collections.contacto	check	status	OK

Información del estado de integridad de la tabla: prestamo			
Table	Op	Msg type	Msg text
collections.prestamo	check	status	OK

Información del estado de integridad de la tabla: recurso			
Table	Op	Msg type	Msg text
collections.recurso	check	status	OK

Número de filas de cada tabla

Nombre de la tabla	Número de filas
catt tipo recurso	6
contacto	6
prestamo	3
recurso	5

Información de índice de cada tabla	
Table	catt tipo recurso
Non unique	0
Key name	PRIMARY
Seq in index	1
Column name	id tipo recurso
Collation	A
Cardinality	6
Sub part	
Packed	
Null	
Index type	BTREE
Comment	
Table	contacto
Non unique	0
Key name	PRIMARY
Seq in index	1
Column name	id contacto
Collation	A
Cardinality	6
Sub part	
Packed	
Null	
Index type	BTREE
Comment	
Table	prestamo
Non unique	0
Key name	PRIMARY
Seq in index	1
Column name	id prestamo
Collation	A
Cardinality	3
Sub part	
Packed	
Null	
Index type	BTREE
Comment	

Table	recurso
Non unique	0
Key name	PRIMARY
Seq_in_index	1
Column name	id recurso
Collation	A
Cardinality	5
Sub part	
Packed	
Null	
Index type	BTREE
Comment	

## Código fuente

En esta sección se mostrará parte de código que se utilizó para el desarrollo del proyecto. Por simplicidad sólo se mostrará el código que forma parte del módulo de conexión a los diferentes manejadores y el módulo de recopilación de un solo manejador, el motivo es el siguiente, la funcionalidad y los resultados se obtienen de manera correcta de los cuatro manejadores, la metodología utilizada es la misma, lo único que cambia son los comandos propios de cada manejador.

```
//Clase ConectaMySQL.java
```

```
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ConectaMySQL
{
    Connection conexion = null;

    public Connection getConnection()
    {
        return conexion;
    }

    public boolean Conectar(String host, String puerto, String bd, String usuario, String pswd )
    {
        try
        {
            try
```



```

        {
            return null;
        }
        return resultado;
    }

    public void desconectarBaseDatos()
    {
        try
        {
            conexion.close();
        }
        catch (SQLException ex)
        {
            Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

}

//Clase ConectaPosrtgreSQL.java
package Swing9;

import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ConectaPostgreSQL
{
    Connection conexion = null;

    public Connection getConnection()
    {
        return conexion;
    }

    public boolean Conectar(String host, String puerto, String bd, String usuario, String pswd )
    {
        try
        {
            try
            {
                Class.forName("org.postgresql.Driver").newInstance();
            }
            catch ( InstantiationException | IllegalAccessException ex)
            {
                Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        catch (ClassNotFoundException ex)
        {
            Logger.getLogger(Conecta5.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

    }
    //jdbc:postgresql://localhost/escolares?"
    String cad = "jdbc:postgresql://" + host + "/" + bd + "?";

    if(host==null)
    {
        conexion = DriverManager.getConnection("jdbc:postgresql://localhost/");
    }
    else
    {
        conexion = DriverManager.getConnection (cad ,usuario, pswd);
    }

}
catch(SQLException ex)
{
    return false;
}
return true;
}

public boolean ejecutarSQL(String sql)
{
    try
    {
        Statement sentencia = conexion.createStatement();
        sentencia.executeUpdate(sql);
    }
    catch(SQLException ex)
    {
        return false;
    }
    return true;
}

public ResultSet ejecutarSQLSelect(String sql)
{
    ResultSet resultado;

    try
    {
        Statement sentencia = conexion.createStatement();
        resultado = sentencia.executeQuery(sql);
    }
    catch(SQLException ex)
    {
        return null;
    }
    return resultado;
}

public void desconectarBaseDatos()
{
    try
    {
        conexion.close();
    }
}

```

```

    }
    catch (SQLException ex)
    {
        Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

//Clase ConectaOracle.java

//Clase ConectaDB2.java

//Módulo que genera el documento .pdf

```

public Document archivopdf(String titulo)
{
    /*** Creando PDF ***/
    Document documento = new Document();// Se crea el documento

    try
    {
        try // Se asocia el documento al OutputStream
        { // Se crea el OutputStream para el fichero donde queremos dejar el pdf.
            PdfWriter.getInstance(documento, new FileOutputStream(titulo+".pdf"));
        }
        catch (FileNotFoundException ex)
        {
            Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    catch (DocumentException ex)
    {
        Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
    }

    documento.open();

    try
    {
        Paragraph parrafo1 = new Paragraph( "    ** RECOPIACIÓN DE LA BASE DE
DATOS: << "+titulo+" >> , DE MySQL **\n\n\n",
            FontFactory.getFont("arial", // fuente
            12, // tamaño
            Font.ITALIC, // estilo
            BaseColor.BLUE)); // color));
        parrafo1.setAlignment(1);//el 1 es para centrar
        documento.add(parrafo1);//imprime antes q todo, add orden d como imprimir las cosas
    }
    catch (DocumentException ex)
    {
        Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

}
//declaras cada tabla con sus columnas
PdfPTable tabla1 = new PdfPTable(1);//encabezado
PdfPTable tabla2 = new PdfPTable(2);
PdfPTable tabla3 = new PdfPTable(1);
PdfPTable tabla4 = new PdfPTable(2);
PdfPTable tabla5 = new PdfPTable(1);
PdfPTable tabla6 = new PdfPTable(1);//una columna tabla cualquier nombre
PdfPTable tabla7 = new PdfPTable(1);
PdfPTable tabla8 = new PdfPTable(2);//dos columnas
PdfPTable tabla10 = new PdfPTable(6);
PdfPTable tabla11 = new PdfPTable(6);
PdfPTable tablaN = new PdfPTable(1);
PdfPTable tablaM = new PdfPTable(2);
PdfPTable tabla12 = new PdfPTable(1);
PdfPTable tabla13 = new PdfPTable(2);
//declaras una celda con lo q quieres q tenga la celda
PdfPCell cell1 = new PdfPCell(new Paragraph("Variables Principales"));
PdfPCell cell2 = new PdfPCell(new Paragraph("Nombre de la variable"));
PdfPCell cell3 = new PdfPCell(new Paragraph("Valor"));
PdfPCell cell4 = new PdfPCell(new Paragraph("user"));
PdfPCell cell5 = new PdfPCell(new Paragraph("connection id"));
PdfPCell cell6 = new PdfPCell(new Paragraph("database"));
PdfPCell cell7 = new PdfPCell(new Paragraph("date and time of consultation"));
PdfPCell cell8 = new PdfPCell(new Paragraph("Información sobre las variables de estado
del servidor"));
PdfPCell cell9 = new PdfPCell(new Paragraph("Tablas exitentes"));
PdfPCell cell10 = new PdfPCell(new Paragraph("Información general acerca de cada
tabla"));
PdfPCell cell11 = new PdfPCell(new Paragraph(" "));//no escribes nada
PdfPCell cell13 = new PdfPCell(new Paragraph("Field"));
PdfPCell cell14 = new PdfPCell(new Paragraph("Type"));
PdfPCell cell15 = new PdfPCell(new Paragraph("Null"));
PdfPCell cell16 = new PdfPCell(new Paragraph("Key"));
PdfPCell cell17 = new PdfPCell(new Paragraph("Default"));
PdfPCell cell18 = new PdfPCell(new Paragraph("Extra"));
PdfPCell cell19 = new PdfPCell(new Paragraph("Table"));
PdfPCell cell20 = new PdfPCell(new Paragraph("Op"));
PdfPCell cell21 = new PdfPCell(new Paragraph("Msg_type"));
PdfPCell cell22 = new PdfPCell(new Paragraph("Msg_text"));
PdfPCell cell23 = new PdfPCell(new Paragraph("Nombre de la tabla"));
PdfPCell cell24 = new PdfPCell(new Paragraph("Número de filas"));
PdfPCell cellN = new PdfPCell(new Paragraph("Número de filas de cada tabla"));
PdfPCell cell25 = new PdfPCell(new Paragraph("Información de índice de cada tabla"));

cell1.setBackgroundColor(BaseColor.ORANGE);
cell2.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell3.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell8.setBackgroundColor(BaseColor.ORANGE);
cell9.setBackgroundColor(BaseColor.ORANGE);
cell10.setBackgroundColor(BaseColor.ORANGE);
cell11.setBackgroundColor(BaseColor.BLUE);//la coloreas de azul
cell13.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell14.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell15.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell16.setBackgroundColor(BaseColor.LIGHT_GRAY);

```

```

cell17.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell18.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell19.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell20.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell21.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell22.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell23.setBackgroundColor(BaseColor.LIGHT_GRAY);
cell24.setBackgroundColor(BaseColor.LIGHT_GRAY);
cellN.setBackgroundColor(BaseColor.ORANGE);
cell25.setBackgroundColor(BaseColor.ORANGE);
tabla1.addCell(cell1);//a tabla 1 metele lo q tenga la celda 1 q es varialbes principales y el

```

color orange

```

tabla2.addCell(cell2);//
tabla2.addCell(cell3);
tabla2.addCell(cell4);
tabla3.addCell(cell8);
tabla4.addCell(cell2);
tabla4.addCell(cell3);
tabla5.addCell(cell9);
tabla7.addCell(cell10);
tablaN.addCell(cellN);
tablaM.addCell(cell23);
tablaM.addCell(cell24);
tabla12.addCell(cell25);

```

```

for (String elemento: sent.Sentencia_1_mysql())
{
    //String muestraSql = "select user(), connection_id()";
    ResultSet rs1 = miConexion.ejecutarSQLSelect(elemento);

    try
    {
        while (rs1.next())
        {
            tabla2.addCell(rs1.getString(1));
            tabla2.addCell(rs1.getString(2));
        }
    }
    catch (SQLException ex)
    {
        Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

//el número id de conexión al servidor
String su = "select user()";
ResultSet rs2 = miConexion.ejecutarSQLSelect(su);
try
{
    while (rs2.next())
    {
        tabla2.addCell(rs2.getString(1));
    }
}

```

```

}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

tabla2.addCell(cell5);
//el nombre del usuario actual de MySQL
String ci = "select connection_id";
ResultSet rs3 = miConexion.ejecutarSQLSelect(ci);
try
{
    while (rs3.next())
    {
        tabla2.addCell(rs3.getString(1));
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

tabla2.addCell(cell6);
//la base de datos en uso
String db = "select database()";
ResultSet rs4 = miConexion.ejecutarSQLSelect(db);
try
{
    while (rs4.next())
    {
        tabla2.addCell(rs4.getString(1));
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

tabla2.addCell(cell7);
//fecha y hora de la consulta
String fyh = "select now()";
ResultSet rs5 = miConexion.ejecutarSQLSelect(fyh);
try
{
    while (rs5.next())
    {
        tabla2.addCell(rs5.getString(1));
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

//Información sobre las variables de estado del servidor
String ss = "show status;";

```

```

ResultSet rs6 = miConexion.ejecutarSQLSelect(ss);
try
{
    while (rs6.next())
    {
        tabla4.addCell(rs6.getString(1)); //muestra dos resultados en una fila que es
variable_name y Value
        tabla4.addCell(rs6.getString(2));
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

//Tablas
String st = "show tables;";
ResultSet rs7 = miConexion.ejecutarSQLSelect(st);
try
{
    while (rs7.next())
    {
        tabla6.addCell(rs7.getString(1)); //obten esa cadena y 1 solo muestra un resultado mysql
(por fila)
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

//Información acerca de cada tabla
String sts = "show table status;";
ResultSet rs8 = miConexion.ejecutarSQLSelect(sts);
try
{
    while (rs8.next())
    {
        tabla8.addCell("Name"); //izq a der, arriba abajo
        tabla8.addCell(rs8.getString(1)); //numerar para q saque los resultados en cada fila con
su encabezado
        tabla8.addCell("Engine");
        tabla8.addCell(rs8.getString(2));
        tabla8.addCell("Version");
        tabla8.addCell(rs8.getString(3));
        tabla8.addCell("Row_format");
        tabla8.addCell(rs8.getString(4));
        tabla8.addCell("Rows");
        tabla8.addCell(rs8.getString(5));
        tabla8.addCell("Avg_row_length");
        tabla8.addCell(rs8.getString(6));
        tabla8.addCell("Data_length");
        tabla8.addCell(rs8.getString(7));
        tabla8.addCell("Max_data_length");
        tabla8.addCell(rs8.getString(8));
        tabla8.addCell("Index_length");
    }
}

```

```

        tabla8.addCell(rs8.getString(9));
        tabla8.addCell("Data_free");
        tabla8.addCell(rs8.getString(10));
        tabla8.addCell("Auto_increment");
        tabla8.addCell(rs8.getString(11));
        tabla8.addCell("Create_time");
        tabla8.addCell(rs8.getString(12));
        tabla8.addCell("Update_time");
        tabla8.addCell(rs8.getString(13));
        tabla8.addCell("Check_time");
        tabla8.addCell(rs8.getString(14));
        tabla8.addCell("Collation");
        tabla8.addCell(rs8.getString(15));
        tabla8.addCell("Checksum");
        tabla8.addCell(rs8.getString(16));
        tabla8.addCell("Create_options");
        tabla8.addCell(rs8.getString(17));
        tabla8.addCell("Comment");
        tabla8.addCell(rs8.getString(18));
        tabla8.addCell(cell11);//relleno azul
        tabla8.addCell(cell11);//relleno azul porque son dos columnas
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

try
{
    documento.add(tabla1);
    documento.add(tabla2);
    documento.add(new Paragraph("\n\n"));//retornos de carro o linea en blanco
    documento.add(tabla3);
    documento.add(tabla4);
    documento.add(new Paragraph("\n\n"));
    documento.add(tabla5);
    documento.add(tabla6);
    documento.add(new Paragraph("\n\n"));
    documento.add(tabla7);
    documento.add(tabla8);
    documento.add(new Paragraph("\n"));
} catch (DocumentException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

String st2 = "show tables;";
ResultSet rs9 = miConexion.ejecutarSQLSelect(st2);
try
{
    while(rs9.next())
    { //consultar las columnas de cada una de las tablas:
        String sts2 = "describe "+rs9.getString(1)+" ";
        ResultSet rs10 = miConexion.ejecutarSQLSelect(sts2);
        PdfPTable tablaX = new PdfPTable(1);
    }
}

```

```

PdfPTable tablaY = new PdfPTable(6);//tiene 6 columnas
PdfPCell cellX = new PdfPCell(new Paragraph("Información de las columnas de la
tabla: "+rs9.getString(1)));
cellX.setBackgroundColor(BaseColor.ORANGE);
tablaX.addCell(cellX);
tablaY.addCell(cell13);//field
tablaY.addCell(cell14);//type
tablaY.addCell(cell15);//null
tablaY.addCell(cell16);//key
tablaY.addCell(cell17);//default
tablaY.addCell(cell18);//extra
while(rs10.next())
{
    tablaY.addCell(rs10.getString(1));//las llena de izq a der, de arr a abaj pero esta tiene
6 columnas por eso las llena asi
    tablaY.addCell(rs10.getString(2));
    tablaY.addCell(rs10.getString(3));
    tablaY.addCell(rs10.getString(4));
    tablaY.addCell(rs10.getString(5));
    tablaY.addCell(rs10.getString(6));
}
try
{
    documento.add(new Paragraph("\n\n"));
    documento.add(tablaX);
    documento.add(tablaY);
} catch (DocumentException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

String st3 = "show tables;";
ResultSet rs11 = miConexion.ejecutarSQLSelect(st3);
try
{
    while(rs11.next())
    { //consultar el estado de integridad de una tabla
        String ct = "check table "+rs11.getString(1)+" ";
        ResultSet rs12 = miConexion.ejecutarSQLSelect(ct);
        PdfPTable tablaW = new PdfPTable(1);
        PdfPTable tablaZ = new PdfPTable(4);
        PdfPCell cellW = new PdfPCell(new Paragraph("Información del estado de integridad
de la tabla: "+rs11.getString(1)));
        cellW.setBackgroundColor(BaseColor.ORANGE);
        tablaW.addCell(cellW);
        tablaZ.addCell(cell19);
        tablaZ.addCell(cell20);
        tablaZ.addCell(cell21);
        tablaZ.addCell(cell22);
        while(rs12.next())

```

```

        {
            tablaZ.addCell(rs12.getString(1));
            tablaZ.addCell(rs12.getString(2));
            tablaZ.addCell(rs12.getString(3));
            tablaZ.addCell(rs12.getString(4));
        }
    try
    {
        documento.add(new Paragraph("\n\n"));
        documento.add(tablaW);
        documento.add(tablaZ);
    } catch (DocumentException ex)
    {
        Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

String st4 = "show tables;";
ResultSet rs13 = miConexion.ejecutarSQLSelect(st4);
try
{
    while(rs13.next())
    { //número de filas de una tabla
        String sc = "select count(*) from "+rs13.getString(1)+" ";
        ResultSet rs14 = miConexion.ejecutarSQLSelect(sc);
        while(rs14.next())
        {
            tablaM.addCell(rs13.getString(1));
            tablaM.addCell(rs14.getString(1));
        }
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}
try
{
    documento.add(new Paragraph("\n\n"));
    documento.add(tablaN);
    documento.add(tablaM);
} catch (DocumentException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

String st5 = "show tables;";
ResultSet rs15 = miConexion.ejecutarSQLSelect(st5);
try
{
    while(rs15.next())

```

```

    {
        //Información de índice de una tabla
        String si = "show index from "+rs15.getString(1)+";";
        ResultSet rs16 = miConexion.ejecutarSQLSelect(si);

        while (rs16.next())
        {
            tabla13.addCell("Table");
            tabla13.addCell(rs16.getString(1));
            tabla13.addCell("Non_unique");
            tabla13.addCell(rs16.getString(2));
            tabla13.addCell("Key_name");
            tabla13.addCell(rs16.getString(3));
            tabla13.addCell("Seq_in_index");
            tabla13.addCell(rs16.getString(4));
            tabla13.addCell("Column_name");
            tabla13.addCell(rs16.getString(5));
            tabla13.addCell("Collation");
            tabla13.addCell(rs16.getString(6));
            tabla13.addCell("Cardinality");
            tabla13.addCell(rs16.getString(7));
            tabla13.addCell("Sub_part");
            tabla13.addCell(rs16.getString(8));
            tabla13.addCell("Packed");
            tabla13.addCell(rs16.getString(9));
            tabla13.addCell("Null");
            tabla13.addCell(rs16.getString(10));
            tabla13.addCell("Index_type");
            tabla13.addCell(rs16.getString(11));
            tabla13.addCell("Comment");
            tabla13.addCell(rs16.getString(12));
            tabla13.addCell(cell11);
            tabla13.addCell(cell11);
        }
    }
}
catch (SQLException ex)
{
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

try
{
    documento.add(new Paragraph("\n\n"));
    documento.add(tabla12);
    documento.add(tabla13);
} catch (DocumentException ex) {
    Logger.getLogger(Ventana5.class.getName()).log(Level.SEVERE, null, ex);
}

documento.close();
/** Fin de Creando PDF */
return documento;
}

```

## Conclusiones

En este proyecto si se logró desarrollar un programa que permite visualizar un reporte de la información de la configuración y estado de los manejadores de bases de datos relacionales: MySQL, PostgreSQL, DB2 y Oracles de forma fácil.

Se investigaron los parámetros de configuración, se crearon los módulos de conexión, se creó el módulo de recopilación de información del contenido y parámetros de configuración para los diferentes manejadores de bases de datos relacionales.

Además de lo mencionado anteriormente, se creó el módulo de reportes donde se puede visualizar la información recopilada.

La mayor ventaja de este proyecto consiste en que se puede obtener toda la información relevante de una base de datos de forma sencilla y además de una manera fácil de analizar ya que la información se presenta por medio de tablas; las tablas incluyen información tal como Nombre de la variable y valor que se obtiene dependiendo de la variable que se haya utilizado, las tablas están divididas por categorías de acuerdo a la información que se arroja, toda ésta información recopilada en un documento con extensión *.pdf* para mejor visualización.

Este proyecto es muy eficiente ya que no se necesitan muchos recursos para utilizarse.

En un futuro, éste proyecto se podrá extender y fortalecer incorporando otros manejadores de bases de datos, para esto se podría seguir la misma metodología y solo bastaría con identificar los comandos equivalentes a los manejadores que se trabajaron en este proyecto.

## Referencias y Bibliografía

Página web: [1] MySQL. (01, Julio, 2013) *MySQL: Developer Zone* [En línea] Disponible en: <http://dev.mysql.com/>

Página web: [2] PostgreSQL. (02, Julio, 2013) *PostgreSQL: The world's most advanced open source database* [En línea] Disponible en: <http://www.postgresql.org>

Página web: [3] International Business Machines. (02, Julio, 2013) *IBM - DB2 database software* [En línea] Disponible en: <http://www.ibm.com/software/data/db2/>

Página web: [4] ORACLE. (03, Julio, 2012) *Oracle | Hardware and Software, Engineered to Work Together* [En línea] Disponible en: <http://www.oracle.com/index.html>, 2012.

Tesis: [5]Juan Manuel Saldaña Vilches, “Integración de una base de datos a una data warehouse para el análisis y explotación de la información en el área de GTS,” tesis, Instituto Politécnico Nacional, 2010.

Página web: [6]ORACLE. (12, Julio, 2013) *¿Qué es Java y por qué lo necesito?* [En línea] Disponible en: [http://www.java.com/es/download/faq/whatis\\_java.xml](http://www.java.com/es/download/faq/whatis_java.xml)

Página web: [7] ORACLE. (13, Julio, 2013) Java SE Downloads [En línea] Disponible en: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Página web: [8] iText. (13, Julio, 2013) Free / Open Source PDF Library for Java and C# [En línea] Disponible en: <http://itextpdf.com/>