

ANEXO 5 – INSTALACIÓN PENTAHO PDI – MIMIC-III

Detallamos a continuación el proceso de instalación de Pentaho PDI, la carga del catálogo y la instalación de la BBDD MIMIC-III.

Contenido

A5-1. INSTALACIÓN PDI PENTAHO.....	3
A5-2. INSTALACIÓN MIMIC-III	10

A5-1. INSTALACIÓN PDI PENTAHO

El primer paso es descargar el módulo PDI de la web de Pentaho <http://community.pentaho.com/>, utilizamos la versión community, la disponible a fecha mayo 2017 es la 7.0 y se encuentra en el apartado *Downloads* → *Data integration*.

Es necesario disponer de java instalado, 1.6 o superior, versión de 64bits, antes de proseguir la instalación. La instalación se ha realizado en un equipo linux, en un equipo windows se realiza de forma similar.

Guardamos el fichero **pdi-ce-7.0.0.0-25.zip** en el directorio de instalación y lo descomprimos.

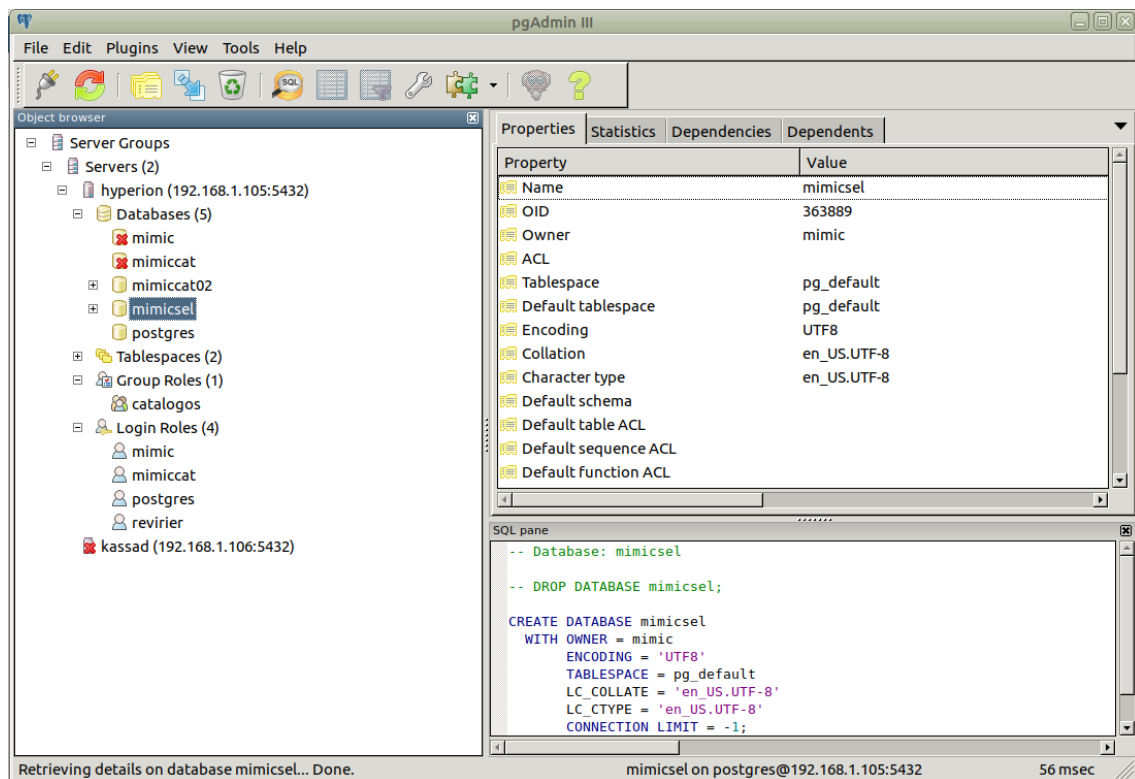
```
revirier@hyperion:/opt/PDI$ sudo mv ~/Downloads/pdi-ce-7.0.0.0-25.zip .
revirier@hyperion:/opt/PDI$ sudo unzip pdi-ce-7.0.0.0-25.zip .
revirier@hyperion:/opt/PDI$ ls -l data-integration
revirier@hyperion:/opt/PDI$ ls -l data-integration/
total 9528
-rw-r--r-- 1 root root 551 nov 5 2016 Carte.bat
-rwxr-xr-x 1 root root 405 nov 5 2016 carte.sh
drwxr-xr-x 2 root root 4096 nov 5 2016 classes
drwxr-xr-x 3 root root 4096 nov 5 2016 Data Integration.app
drwxr-xr-x 2 root root 4096 nov 5 2016 Data Service JDBC Driver
drwxr-xr-x 3 root root 4096 nov 5 2016 docs
-rw-r--r-- 1 root root 138 nov 5 2016 Encr.bat
-rwxr-xr-x 1 root root 140 nov 5 2016 encr.sh
-rw-r--r-- 1 root root 131 nov 5 2016 Import.bat
-rw-r--r-- 1 root root 2354 nov 5 2016 import-rules.xml
-rwxr-xr-x 1 root root 275 nov 5 2016 import.sh
-rw-r--r-- 1 root root 184 nov 5 2016 Kitchen.bat
-rwxr-xr-x 1 root root 354 nov 5 2016 kitchen.sh
drwxr-xr-x 2 root root 4096 nov 5 2016 launcher
drwxr-xr-x 2 root root 20480 nov 5 2016 lib
drwxr-xr-x 6 root root 4096 nov 5 2016 libswt
-rw-r--r-- 1 root root 13366 nov 5 2016 LICENSE.txt
-rw-r--r-- 1 root root 172 nov 5 2016 Pan.bat
-rwxr-xr-x 1 root root 320 nov 5 2016 pan.sh
-rw-r--r-- 1 root root 9181335 nov 5 2016
PentahoDataIntegration_OSS_Licenses.html
drwxr-xr-x 27 root root 4096 nov 5 2016 plugins
-rw-r--r-- 1 root root 213 nov 5 2016 purge-utility.bat
-rwxr-xr-x 1 root root 347 nov 5 2016 purge-utility.sh
drwxr-xr-x 2 root root 4096 nov 5 2016 pwd
-rw-r--r-- 1 root root 1312 nov 5 2016 README.txt
-rw-r--r-- 1 root root 522 nov 5 2016 runSamples.bat
-rwxr-xr-x 1 root root 305 nov 5 2016 runSamples.sh
drwxr-xr-x 5 root root 4096 nov 5 2016 samples
-rw-r--r-- 1 root root 4098 nov 5 2016 set-pentaho-env.bat
-rwxr-xr-x 1 root root 3710 nov 5 2016 set-pentaho-env.sh
drwxr-xr-x 2 root root 4096 nov 5 2016 simple-jndi
-rw-r--r-- 1 root root 3857 nov 5 2016 Spoon.bat
-rwxr-xr-x 1 root root 220 nov 5 2016 spoon.command
-rw-r--r-- 1 root root 98 nov 5 2016 SpoonConsole.bat
-rw-r--r-- 1 root root 1271 nov 5 2016 SpoonDebug.bat
-rwxr-xr-x 1 root root 1051 nov 5 2016 SpoonDebug.sh
-rw-r--r-- 1 root root 370070 nov 5 2016 spoon.ico
-rw-r--r-- 1 root root 1345 nov 5 2016 spoon.png
```

```
-rwxr-xr-x 1 root root 6307 nov 5 2016 spoon.sh
drwxr-xr-x 5 root root 4096 nov 5 2016 system
drwxr-xr-x 3 root root 4096 nov 5 2016 ui
-rwxr-xr-x 1 root root 1628 nov 5 2016 yarn.sh
```

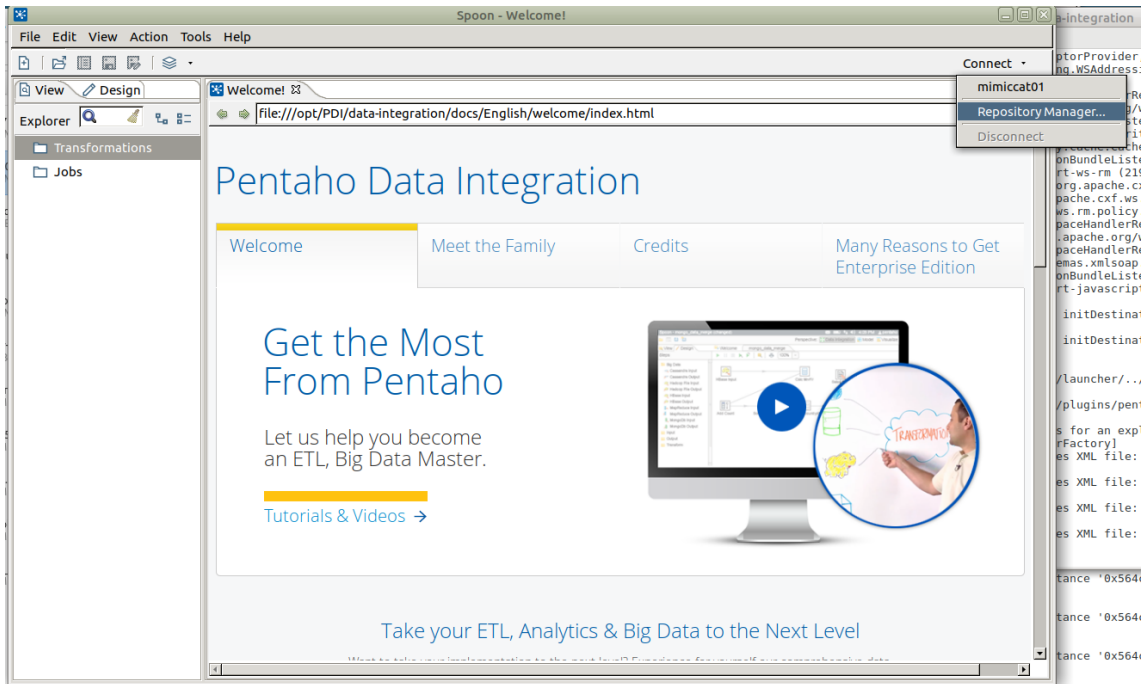
Antes de ejecutarlo, modificamos la memoria disponible editando el fichero *spoon.sh* y ampliando la memoria en la sección de `JAVA_OPTIONS`:

```
if [ -z "$PENTAHO_DI_JAVA_OPTIONS" ]; then
    PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx4096m -XX:MaxPermSize=256m"
fi
```

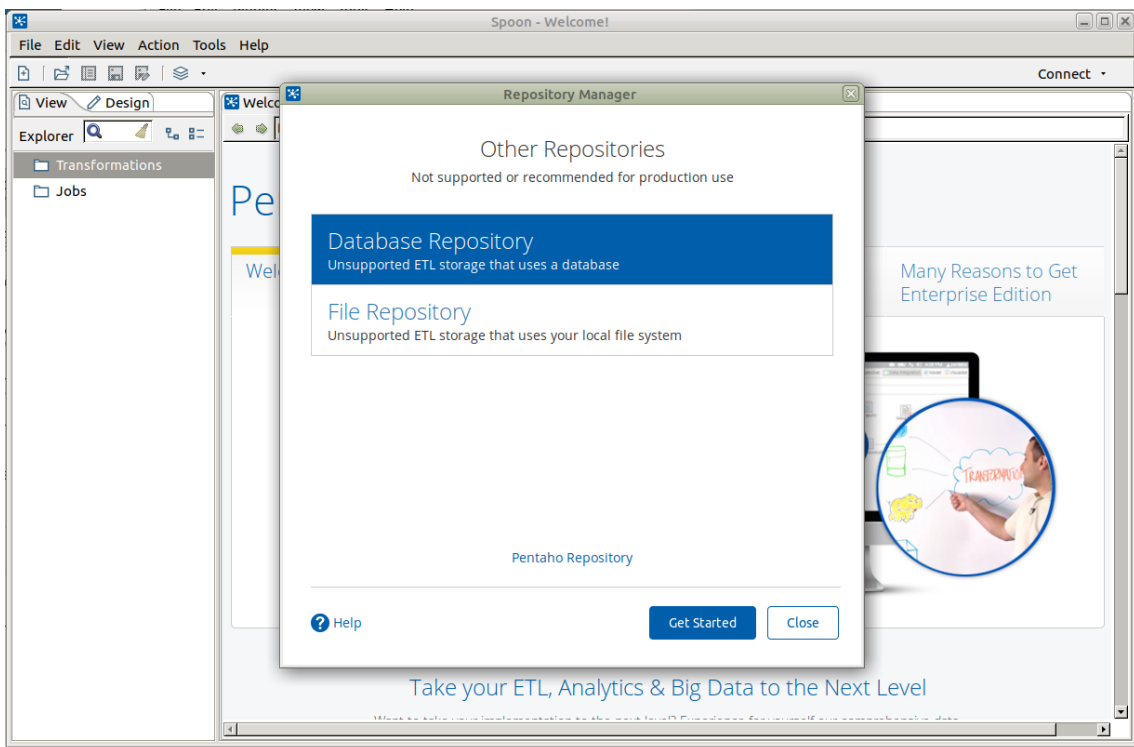
Y creamos las BBDD necesarias para el repositorio y las tablas intermedias, en este caso las creamos en PostgreSQL, pero se pueden crear en cualquiera de los gestores de BBDD soportados por Pentaho.



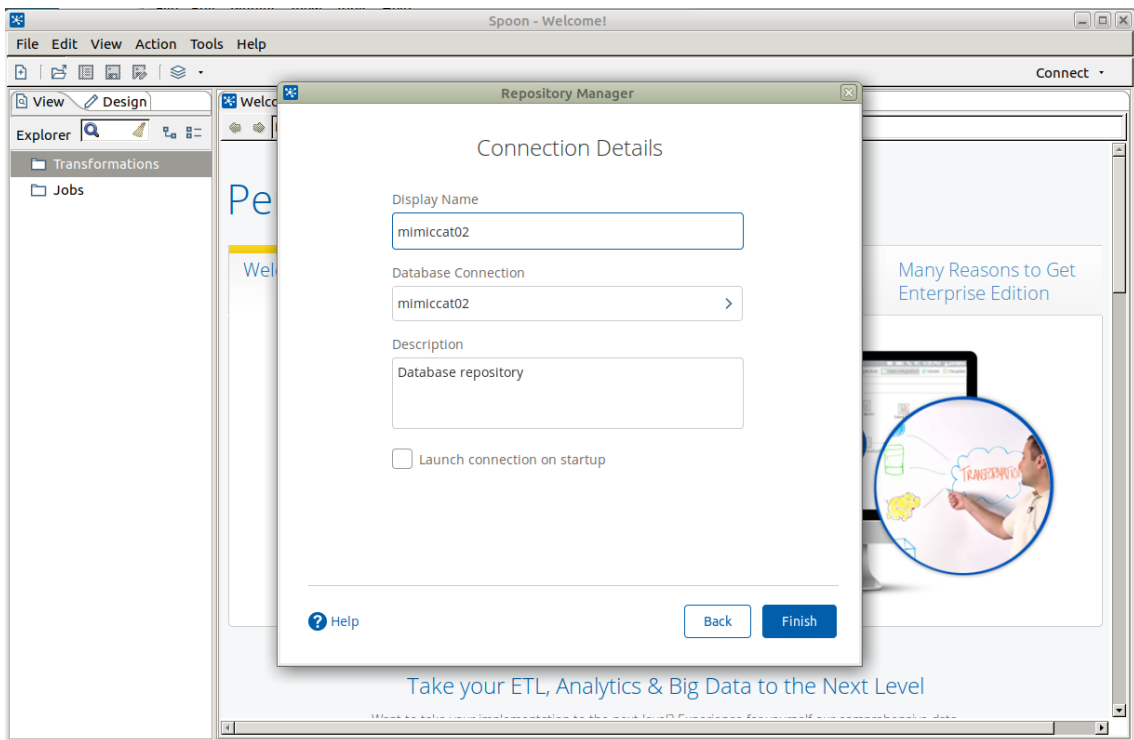
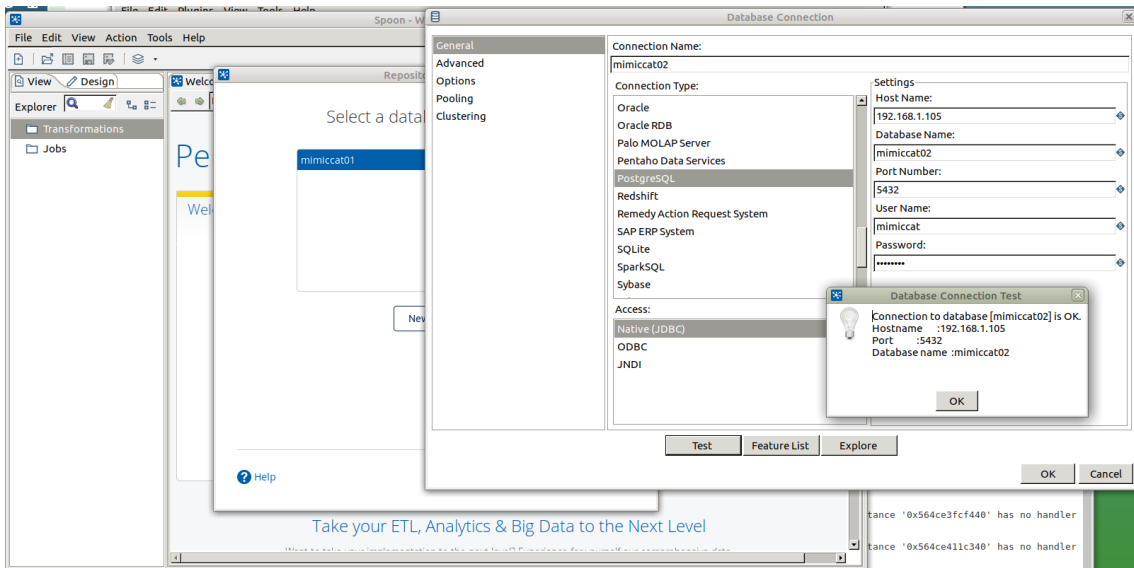
Una vez disponibles las BBDD necesarias: *mimicat02* (catálogo), *mimic-iii* y *mimicsel* ejecutamos el script *spoon.sh*. En la pantalla principal de Pentaho se debe seleccionar la opción de **connect** → **repository manager**



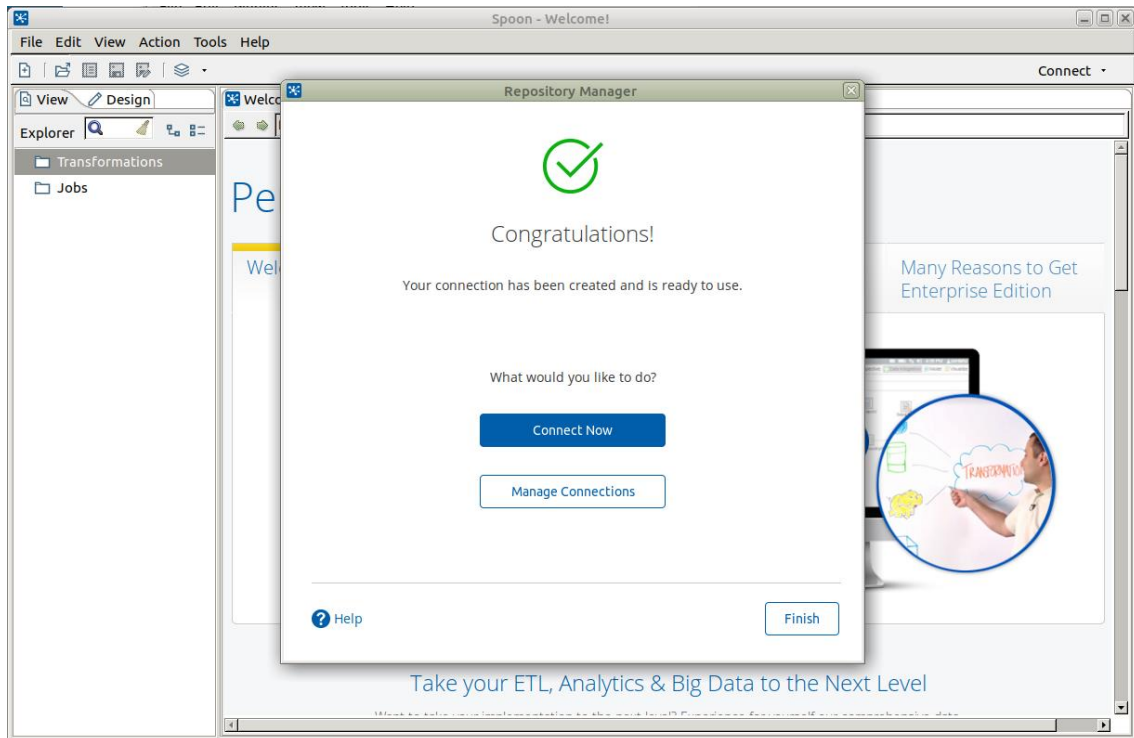
Y elegir la opción **other repositories** → **Database Repository**



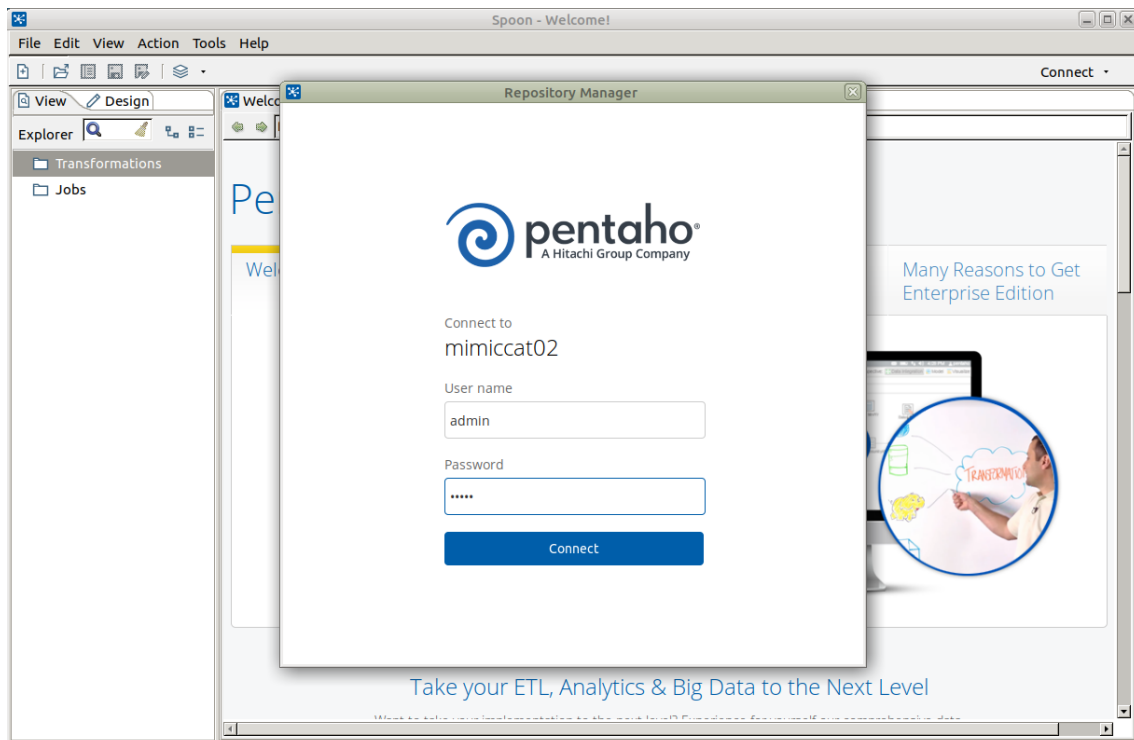
Creamos una conexión de BBDD nueva:



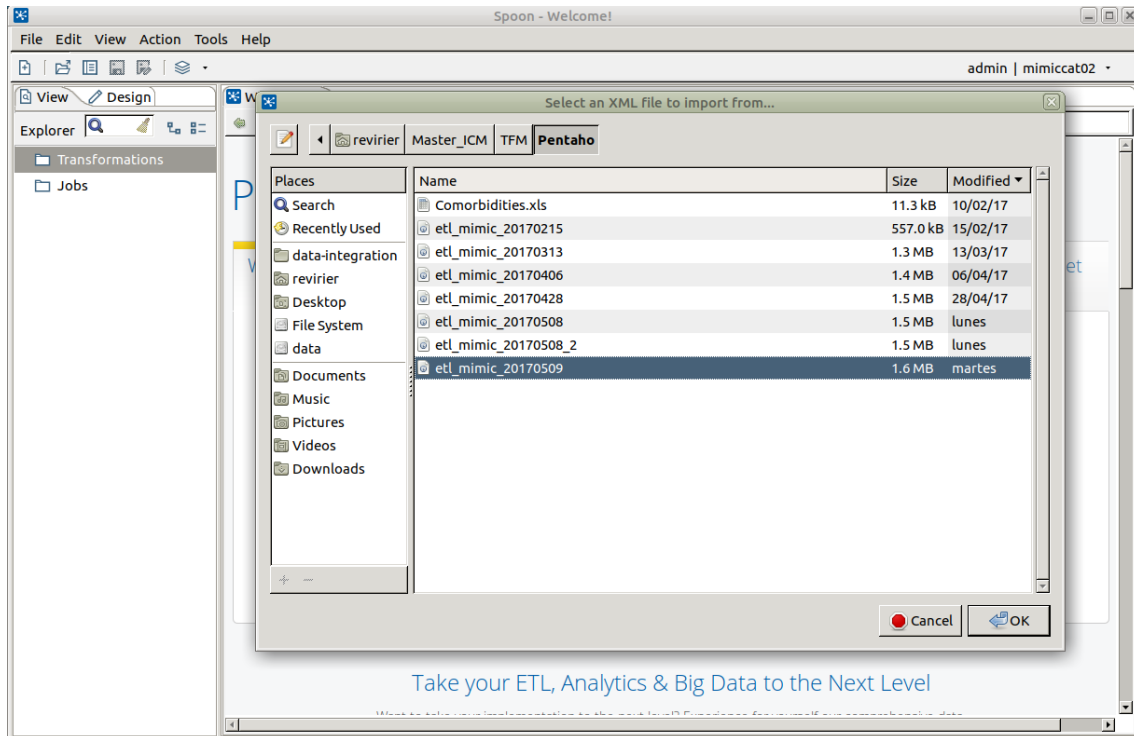
Y nos conectamos a ella:



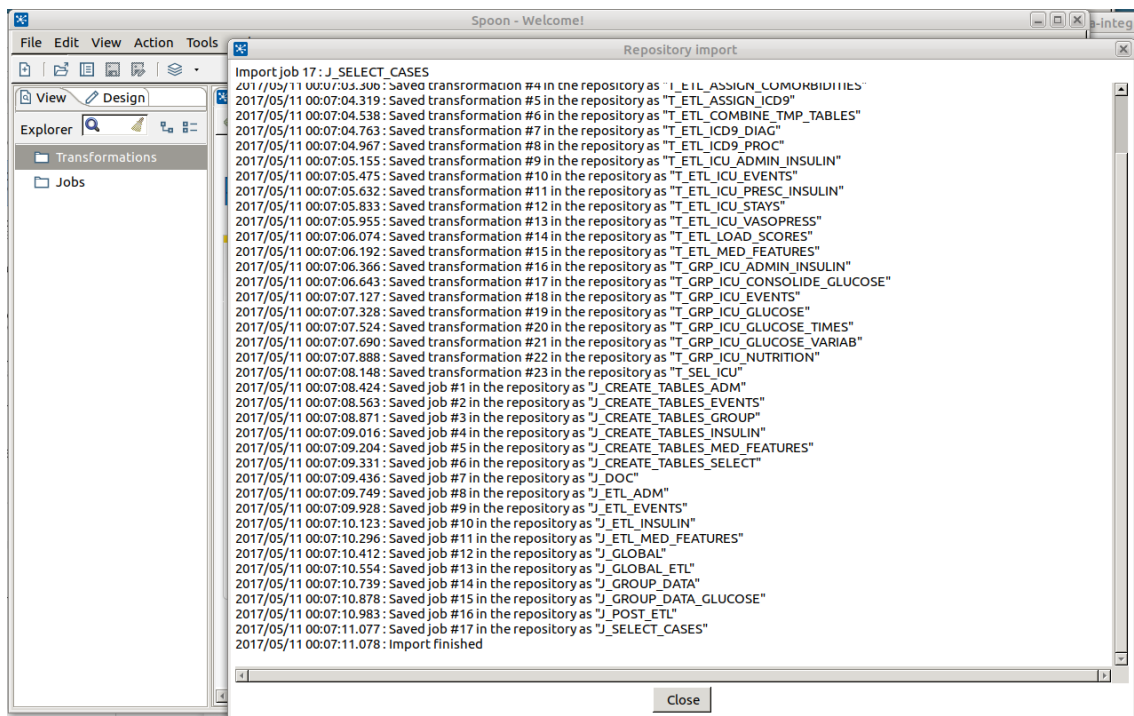
La conexión a los catálogos se realiza con el usuario *admin* / clave *admin* por defecto:



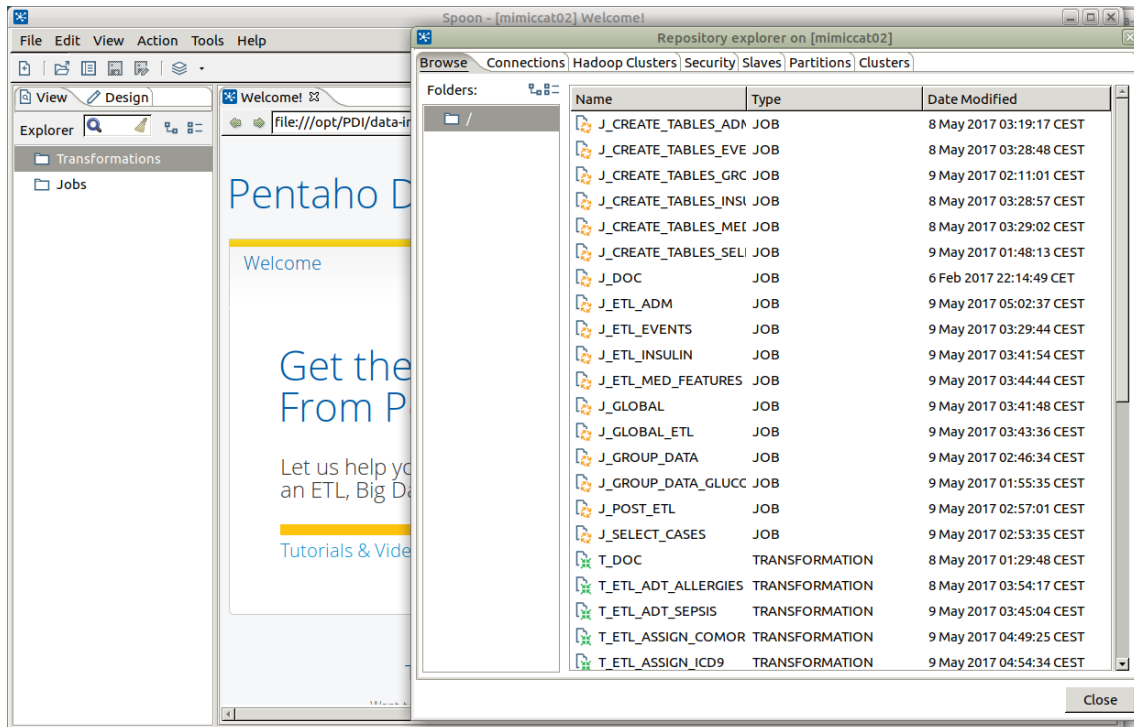
Una vez conectados al repositorio debemos importar el fichero XML que contiene la exportación del catálogo desarrollado en este trabajo, en el menú **Tools** → **Import Repository** elegimos el fichero correspondiente:



Contestamos no a la pregunta de si aplicamos un conjunto de reglas y aceptamos la instalación en el directorio raíz del repositorio:



Una vez terminado explorando el repositorio desde **Tools** → **Repository** → **Explore...**



comprobamos que se han cargado correctamente las transformaciones y trabajos.

Cargando cualquier trabajo, por ejemplo, el inicial J_GLOBAL, podemos ver en el panel de la izquierda las conexiones de datos que es necesario configurar con los datos específicos de cada entorno: **mimic_in** que se corresponde con la BBDD MIMIC-III, **mimic_out** que es la BBDD intermedia *mimicseI* que hemos creado anteriormente y **temp_db** BBDD intermedia en memoria que no es necesario reconfigurar.

A5-2. INSTALACIÓN MIMIC-III

Para la instalación de la BBDD MIMIC-III existen tres alternativas, documentadas en el sitio web de MIMIC (<https://mimic.physionet.org/>) y en MIMIC-CODE (<https://github.com/MIT-LCP/mimic-code>)

- Instalar PostgreSQL, obtener los ficheros CSV con el contenido de las tablas, crear una BBDD vacía y seguir los pasos detallados en <https://mimic.physionet.org/tutorials/install-mimic-locally-ubuntu/> para cargar los ficheros CSV en una nueva base de datos.
- Instalar PostgreSQL (u otro gestor de BBDD soportado), obtener los ficheros CSV con el contenido de las tablas, clonar el repositorio mimic-code y seguir las instrucciones de <https://github.com/MIT-LCP/mimic-code/tree/master/buildmimic/postgres> para ejecutar los scripts de instalación.
- Instalar PostgreSQL y MIMIC-III en una máquina virtual mediante Vagrant siguiendo las instrucciones del repositorio <https://github.com/nsh87/mimic-iii-vm>

Las dos últimas opciones automatizan todo el proceso, la carga de los ficheros CSV en la BBDD PostgreSQL tarda unas 7 u 8 horas dependiendo de las características del equipo y la configuración del gestor de BBDD.

Las vistas materializadas se pueden instalar ejecutándose desde cualquier entorno de consultas SQL conectado a la BBDD MIMIC-III o siguiendo las instrucciones del repositorio <https://github.com/MIT-LCP/mimic-code/blob/master/Makefile.md>