

Guía Técnica SIU-Wichi 6.7.0

Fecha actualización: 08/06/2020

Contenido

[1. Introducción](#)

- [1.1. Organización del documento](#)
- [1.2. Información y contacto](#)
- [1.3. Requisitos previos](#)
 - [1.3.1. Paquete SIU-Wichi](#)
 - [1.3.2. Requerimientos de software](#)
 - [1.3.3. Requerimientos de Hardware](#)

[2. Instalación de componentes Pentaho](#)

- [2.1. Introducción](#)
- [2.2. BI-Server](#)
 - [2.2.1. Descargar el BI Server de Pentaho](#)
 - [2.2.2. Descomprimir el archivo](#)
 - [2.2.3. Configurar Java](#)
 - [2.2.3.1. Verificar que Java esté instalado](#)
 - [2.2.3.2. Configurar variables de entorno](#)
 - [2.2.3.3. Parámetros Java](#)
 - [2.2.4. Instalar las bases de datos PostgreSQL](#)
 - [2.2.4.1. Instalación del driver PostgreSQL](#)
 - [2.2.4.2. Scripts SQL para PostgreSQL](#)
 - [2.2.4.3. Ejecutar los scripts SQL](#)
 - [2.2.4.4. Configuración de la seguridad JDBC](#)
 - [2.2.4.5. Configuración de Hibernate y Quartz](#)
 - [2.2.5. Configurar la Base SampleData](#)
 - [2.2.6. Conectividad](#)
 - [2.2.7. Cambiar el propietario](#)
 - [2.2.8. Configuración de ambiente](#)
 - [2.2.8.1. Redirigir inicio](#)
 - [2.2.9. Probar la instalación](#)
 - [2.2.10. Aspectos de seguridad](#)
- [2.3. Componentes visuales](#)
 - [2.3.1. Instalación de componentes](#)
 - [2.3.2. Instalación de saiku](#)
 - [2.3.3. Configuración de saiku](#)
 - [2.3.4. Instalación de Spanish Language Pack](#)
- [2.4. Pentaho Data Integration](#)

[3. Instalación de SIU-Wichi 6.7.0](#)

- [3.1. Creación de la base de datos SIU-Wichi](#)
 - [3.1.1. Creación de la base en Postgres](#)
 - [3.1.2. Creación de la estructura mediante ETL](#)
- [3.2. Configuración de las conexiones JNDI](#)
- [3.3. Registro de esquemas](#)

[3.3.1. Registro de esquemas](#)

[3.3.2. Actualización de esquema \(SIU-Wichi.xml\)](#)

[3.3.3. Mapeo de esquemas para reportes](#)

[3.4. Carpeta SIU-Wichi](#)

[3.5. Instalación de utilidades para administrar el servidor](#)

[3.5.1. Log de accesos](#)

[3.5.2. Perfiles de usuario](#)

[4. Migración de SIU-Wichi versión 6.7.0](#)

[4.1. Introducción](#)

[4.2. Actualizar versión](#)

[4.3. Actualizar solución SIU-Wichi](#)

[4.4. Actualizar esquema \(SIU-Wichi.xml\)](#)

[4.5. Reemplazar ETLs de carga](#)

[4.6. Actualizar datos de Portal de Transparencia](#)

[5. Carga y actualización de datos](#)

[5.1. Consideraciones generales](#)

[5.1.1. Encoding](#)

[5.1.2. Actualizaciones controladas](#)

[5.1.2.1. Pilagá, Mapuche, Rhun y Guaraní 3](#)

[5.1.2.2. Araucano, Diaguita, Kolla, Querandies, Sanavirón/Quilmes y Moodle](#)

[5.1.2.3. Guaraní 2](#)

[5.1.3. Caches de datos](#)

[5.1.3.1. Cache de Mondrian](#)

[5.1.3.2. Cache de Saiku](#)

[5.1.3.3. Cache de CDA](#)

[5.1.4. Nivel de log](#)

[5.2 Configuración de instalaciones](#)

[5.2.1. Introducción](#)

[5.2.1.1. Número de instalación](#)

[5.2.1.2. Cambios en la instalación](#)

[5.2.2. Tablero de Instalaciones](#)

[5.2.3. Tablas involucradas](#)

[5.3. Datos de SIU-Guaraní 2](#)

[5.4. Datos de SIU-Guaraní 3](#)

[5.5. Datos de SIU-Mapuche](#)

[5.6. Datos de SIU-Pilagá](#)

[5.7. Datos de SIU-RHUN](#)

[5.8. Datos de SIU-Araucano](#)

[5.9. Datos de SIU-Diaguita](#)

[5.10. Datos de SIU-Kolla](#)

[5.11. Datos de SIU-Querandies](#)

[5.12. Datos de SIU-Sanavirón/Quilmes](#)

[5.13. Datos de Moodle](#)

[6. Administración del servidor](#)

[6.1. Gestión de la seguridad](#)

[6.1.1. Introducción](#)

[6.1.2. Configuración del Pentaho](#)

[6.1.2.1. Habilitar la seguridad sobre archivos](#)

[6.1.2.2. Desactivar "Login as an Evaluator"](#)

[6.1.3. Gestión de Usuarios en Pentaho](#)

[6.1.4. Habilitación de Cubos](#)

[6.1.4.1. Habilitación de accesos por Unidad Académica](#)

[6.1.4.1.1. Ejemplo de roles y acceso segmentando la información](#)

[6.1.4.2. Datos de configuración](#)

[6.1.4.2.1 - Actualiza datos de Institución](#)

[6.1.4.2.2 - Actualiza Unidades de Análisis](#)

[6.1.4.2.3 - Asignación de Unidades de Análisis](#)

[6.1.4.2.4 - Actualización de Datos](#)

[6.1.4.2.5 Actualización de Roles](#)

[6.1.5. Habilitación de carpetas](#)

[6.2. Configurar certificado SSL](#)

[7. Aspectos avanzados de Pentaho](#)

[7.1. BI-Server](#)

[7.1.1. Desactivar JPivot](#)

[7.1.2. Arranque automático](#)

[7.1.2.1 Arranque automático Systemd](#)

[7.1.3. Servidor SMTP](#)

[7.1.4. Desactivar la base HSQL en el BI Server](#)

[7.1.5. Cambiar imagen de pantalla de Login](#)

[7.1.6. Ocultar carpetas](#)

[7.1.7. Cambiar paleta de colores de saiku](#)

[7.1.8. Iniciar Pentaho en idioma español](#)

[7.2. Tomcat](#)

[7.2.1. Puertos de escucha](#)

[7.2.2. Cambiar URL de acceso](#)

[7.2.3. Duración de la sesión](#)

[7.2.4. Otros puntos de configuración](#)

[7.2.4.1. Ubicación de pentaho-solutions](#)

[7.2.4.2. TrustedIpAddrs](#)

[8. Aspectos avanzados de SIU-Wichi](#)

[8.1. Creación parcial de la base de datos](#)

[8.1.1. Creación de esquemas por sistema fuente](#)

[8.2. Carga de datos](#)

[8.2.1. Separación del proceso de extracción e importación](#)

[8.2.2. Tabla de importaciones](#)

[8.2.3. Procesos de actualización de unidad genérica](#)

[8.2.4. Script para facilitar carga de TXT \[Aporte UNGS\]](#)

[8.2.5. Script para refrescar la cache de mondrian, saiku y CDA cache \[Aporte UNER\]](#)

[8.3. Mantenimiento de la base de datos](#)

[8.3.1. Limpieza de datos innecesarios](#)

[9. Portal de transparencia](#)

[9.1. Portal de transparencia - consideraciones técnicas](#)

[9.2. Portal de transparencia - instalación](#)

[9.2.1. Instalación de Base de Datos](#)

[9.2.2. Instalación del Tablero](#)

[9.2.3. Personalización del Tablero](#)

[9.2.3.1. Personalización del Tablero - Ocultar pestaña](#)

[9.2.3.2. Personalización del Tablero - Cambiar el orden de las pestañas](#)

[9.2.3.3. Personalización del Tablero - Adecuación estética](#)

[9.3. Portal de transparencia - carga de datos](#)

[10. Solución de errores](#)

[10.1. Tablero del Portal Gerencial arroja No Data](#)

1. Introducción

Este documento tiene como objetivo guiar al usuario en la instalación y configuración de un servidor de Pentaho, en especial el **BI Server 6.1**, en la distribución **Ubuntu o Debian de Linux**, la instalación y puesta en funcionamiento del Sistema de Información Gerencial **SIU-Wichi 6.7.0**.

Esta guía está orientada a lectores con perfil técnico, con conocimientos de administración de servidores Linux, SQL y administración de bases de datos PostgreSQL.

1.1. Organización del documento

Esta guía está organizada en 9 capítulos, de la siguiente manera:

El presente capítulo 1, expone información para contactar a SIU y su comunidad de Universidades, donde bajar la versión del sistema, requerimientos de software y hardware necesarios para instalar.

El capítulo 2 explica cómo instalar Pentaho BI-Server, la plataforma sobre la que se instalará SIU-Wichi, como se explica en el capítulo 3. Ambos capítulos son de lectura secuencial.

El capítulo 4 explica cómo migrar datos para los casos en que sea necesario. Al comienzo del mismo se explica en qué casos puede ser saltado.

En el capítulo 5 se explica cómo cargar y actualizar datos. Comienza con consideraciones comunes a todos los sistemas fuente.

En el capítulo 6 se explica la administración del servidor, incluyendo la configuración de la seguridad mediante perfiles de acceso, a nivel de carpetas y de cubos. Se presenta para ser leído luego de la carga de datos dado que puede ser conveniente, aunque es posible abordarlo luego del capítulo 3.

Los capítulos 7 y 8 son de referencia, orientados a situaciones particulares, el 7 en lo referente a Pentaho y el 8 a SIU-Wichi.

El último capítulo plantea errores comunes de instalación.

1.2. Información y contacto

Se puede acceder a este documento y más información sobre SIU-Wichi en la wiki del proyecto SIU-Wichi <http://documentacion.siu.edu.ar/wiki/SIU-Wichi> . Las versiones futuras del documento serán publicadas también en la wiki.

Para consultas se encuentra disponible el foro <http://comunidad.siu.edu.ar/foro/index.php?board=49.0>.

Para contactar al equipo de trabajo SIU-Wichi: wichi@siu.edu.ar

Para contactar a usuarios de todas las universidades: wichi.usuarios@siu.edu.ar

1.3. Requisitos previos

1.3.1. Paquete SIU-Wichi

El paquete SIU-Wichi se puede descargar desde <http://comunidad.siu.edu.ar/> y contiene las siguientes carpetas:

- etl: creación de la base de datos, carga, migraciones, administración de servidor.
- SIU-Wichi.zip: solución a instalar dentro de Pentaho BI Server
- utilidades: esquemas, scripts y archivos necesarios.

1.3.2. Requerimientos de software

Los requerimientos mínimos que recomendamos, según la pruebas realizadas, son los siguientes:

- S.O: Linux - Ubuntu 12.04 o superior / Debian 6.0 o superior
- Pentaho: Bi-Server 6.1 (Edición comunidad)
- B.D: PostgreSQL 9.3 o superior
- Java: JRE versión 1.7

La instalación de Pentaho BI Server se explicará detalladamente más adelante.

1.3.3. Requerimientos de Hardware

Los siguientes son los requerimientos de hardware de referencia, a consensuar con el área pertinente de la Institución

Característica	Servidor mínimo recomendado para realizar una instalación de prueba	Servidor recomendado para una instalación de Producción
Servidores	Unico servidor para instalar base de datos y aplicación Pentaho	• Un servidor de base de datos • Un servidor para la aplicación Pentaho
RAM	4 Gb	8 Gb cada servidor
Disco	40 Gb	• Base de datos: 80 Gb • Pentaho: 30 Gb
Núcleos	2	4 núcleos cada servidor

2. Instalación de componentes Pentaho

2.1. Introducción

En este capítulo se explican las tareas requeridas para instalar y poner en funcionamiento los componentes de Pentaho requeridos por SIU-Wichi. Se instalará Pentaho BI-Server 6.1 Community Edition con sus componentes gráficas, las bases de datos PostgreSQL, y Pentaho Data Integration. Al final de este capítulo deberá tener instalado y funcionando Pentaho BI-Server 6.1, pudiendo utilizar la aplicación *Steel Wheels* que viene preinstalada.

Pentaho BI-Server es la plataforma sobre la que se instala SIU-Wichi ([3.Instalación de SIU-Wichi 6.7.0](#)).

[2.2. BI-Server](#) se detalla la instalación de BI-Server con base de datos PostgreSQL.

[2.3. Componentes visuales](#) se explica cómo instalar los componentes visuales requeridos por SIU-Wichi.

[2.4. Instalación de Pentaho Data Integration](#) se explica como instalar Pentaho PDI, necesaria para varias de las tareas explicadas en la guía.

2.2. BI-Server

2.2.1. Descargar el BI Server de Pentaho

Se debe descargar del sitio Sourceforge.com el BI-Server de Pentaho mediante el link <https://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/6.1/biserver-ce-6.1.0.1-196.zip/download>

El archivo descargado debería tener alrededor de 945 MB. Si la descarga no funciona correctamente, intente seleccionar un mirror distinto al que se asigna por defecto.



No iniciar Pentaho hasta no haber completado todos los puntos de configuración de 2.2. BI-Server

2.2.2. Descomprimir el archivo

Crear el directorio `/usr/local/pentaho` si no existe y descomprimir allí el archivo descargado, utilizando los siguientes comandos en la terminal.

```
$ sudo mkdir /usr/local/pentaho #ignorar posible error de que ya existe
$ cd /usr/local/pentaho
$ sudo unzip <path biserver.zip>/biserver-ce-6.1.0.1-196.zip
```

Como resultado, deberán verse los siguientes directorios:

```

/usr
  /local/
    |-- pentaho
    |   |-- biserver-ce
    |   |   |...

```

2.2.3. Configurar Java

2.2.3.1. Verificar que Java esté instalado

Antes de poder iniciar el servidor de Pentaho hay que verificar que la **JVM** (Java Virtual Machine) esté instalada y que las variables de entorno estén configuradas correctamente.

Para verificar que la JVM esté instalada ejecutar desde la consola el comando **java -version**. Si la JVM está instalada aparecerá algo como lo siguiente:

```

java version "1.7.0_80"
Java(TM) SE Runtime Environment (build 1.7.0_80-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.80-b11, mixed mode)

```

En caso que la JVM no esté instalada se puede instalar descargandola del siguiente link → [Java 1.7](#)
 Luego instale java de la siguiente manera:

Paso 1: Crear el directorio `/usr/local/java` y copiar el archivo descargado:

```

sudo mkdir -p /usr/local/java
sudo cp -r jdk-7u80-linux-x64.tar.gz /usr/local/java/

```

Paso 2: ir a la carpeta creada `/usr/local/java`:

```

cd /usr/local/java

```

Paso 3: Extraer el archivó:

```

sudo tar xvzf jdk-7u80-linux-x64.tar.gz

```

Paso 4: Chequear que el archivo se extrajo correctamente:

```

ls -a jdk1.7.0_80

```

Paso 5: Abrir el archivo `/etc/profile` con sudo:

```

sudo nano /etc/profile

```

Paso 6: ir al final del archivo y agregar las siguientes líneas:

```

JAVA_HOME=/usr/local/java/jdk1.7.0_80
JRE_HOME=/usr/local/java/jdk1.7.0_80
PATH=$PATH:$JRE_HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export JRE_HOME export PATH

```

Paso 7: Actualizar alternativas:

```

sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/java/jdk1.7.0_80/bin/java" 1

```

```

sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/local/java/jdk1.7.0_80/bin/javac" 1
sudo update-alternatives --install "/usr/bin/javaws" "javaws"
"/usr/local/java/jdk1.7.0_80/bin/javaws" 1
sudo update-alternatives --set java /usr/local/java/jdk1.7.0_80/bin/java
sudo update-alternatives --set javac /usr/local/java/jdk1.7.0_80/bin/javac
sudo update-alternatives --set javaws /usr/local/java/jdk1.7.0_80/bin/javaws

```

Paso 8: Recargar profile:

```
source /etc/profile
```

Paso 9: Verificar instalación:

```
java -version
```

Debería recibir un mensaje que muestra

```
java version "1.7.0_80"
```

```
Java(TM) SE Runtime Environment (build 1.7.0_80-b15)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 24.80-b11, mixed mode)
```



Es requisito tener instalado Java: JRE versión 1.7.*

2.2.3.2. Configurar variables de entorno

Ir al directorio de instalación de Java, por defecto es `/usr/lib/jvm` y verificar el nombre de la carpeta de la versión instalada (por defecto `"/usr/lib/jvm/java-1.7.0-openjdk-amd64"`), el cual se utilizará en la variable de entorno `JAVA_HOME`.

Setear la variable de entorno de java editando el archivo `environment` ubicado en el directorio `/etc/` agregando la siguiente línea si no existe:

```
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64
```

Si es necesario modifique de acuerdo a la configuración del java de la máquina.

La nueva variable no estará disponible hasta no iniciar una nueva sesión del shell (cerrar la terminal y volver a abrirla).

2.2.3.3. Parámetros Java

Paso 1: Incrementar la memoria

Para incrementar la cantidad de memoria asignada a la JVM del BI-Server hay que editar el archivo `start-pentaho.sh` ubicado en `<path_pentaho>/biserver-ce`. Se debe modificar el valor del parámetro `-Xmx` como se muestra en el ejemplo siguiente, con un valor adecuado para el servidor.

```

CATALINA_OPTS="-Xms2048m -Xmx6144m -XX:MaxPermSize=256m
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000
-Dfile.encoding=utf8 -DDI_HOME=\"$DI_HOME\""

```

Paso 2: Modo headless

Además, algunas versiones de Java requieren activar explícitamente el modo headless. En la misma línea de código del paso 1 verificar la opción.

```
CATALINA_OPTS="-Xms2048m -Xmx6144m -XX:MaxPermSize=256m  
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000  
-Dfile.encoding=utf8 -DDI_HOME=\"$DI_HOME\" -Djava.awt.headless=true"
```

2.2.3.4. Paquetes complementarios

Según la versión de java puede ser necesario instalar el paquete ttf-dejavu del sistema operativo.

```
$ sudo apt-get install ttf-dejavu
```

2.2.4. Instalar las bases de datos PostgreSQL

El BI-Server usa tres bases de datos internas además de la base de datos siu_wichi. Por defecto las bases internas vienen configuradas en un motor de base de datos Hypersonic que las levanta en memoria. Las bases internas son:

1. **hibernate**, la cual contiene los datos que se relacionan con el registro de auditoría. SIU-Wichi requiere que hibernate esté en PostgreSQL.
2. **quartz**, utilizada para programación de procesos en el tiempo.
3. **jackrabbit**, es el repositorio de soluciones, contiene ejemplos, seguridad de datos y todo lo creado por el usuario.

Además viene configurada en Hypersonic una base de datos con datos de una aplicación de ejemplo.

A continuación describimos los pasos para configurar *hibernate* como base de datos externa. Esto es requerido por SIU-Wichi. Además decidimos configurar la base de datos *quartz* en PostgreSQL y dejar la base de datos de la aplicación de ejemplo en Hypersonic, aunque sería posible hacerlo de otra manera. Para una primera instalación de prueba recomendamos configurar las bases de datos utilizando los nombres de usuario y contraseñas propuestos como ejemplo.

Los scripts utilizados más adelante están basados en los scripts para postgresql que trae el BI-Server (/pentaho/biserver-ce/data/postgresql) con algunas modificaciones.

2.2.4.1. Instalación del driver PostgreSQL

De acuerdo a la versión de Postgres utilizada, puede ser necesario descargar el driver JDBC correspondiente. El BI-Server 6.1 trae el driver para PostgreSQL 9.3 por defecto (postgresql-9.3-1102-jdbc4.).

Puede descargar los drivers del sitio [PostgreSQL JDBC Driver downloads](#) de acuerdo a la versión de PostgreSQL y de Java instaladas. Después de descargar el archivo **jar** correspondiente, moverlo dentro del directorio `/biserver-ce/tomcat/lib/`.

2.2.4.2. Scripts SQL para PostgreSQL

Los scripts para la creación de estas bases se encuentran en `utilidades/instalacion/bases_postgres/` dentro del paquete SIU-Wichi

Hay 3 scripts:

- *create_jcr_postgresql.sql*: Crea la base Jackrabbit.
- *create_quartz_postgresql.sql*: Crea la base Quartz.
- *create_repository_postgresql.sql*: Crea la base Hibernate.

En cada uno de los scripts se crea un usuario con privilegios para la base que se está creando. Ese usuario se crea con contraseña = "password", si desea puede cambiar esta contraseña por cuestiones de seguridad.

A continuación se muestra cómo ejecutar estos scripts.

2.2.4.3. Ejecutar los scripts SQL

Antes de empezar, cambiar al directorio donde se encuentran los scripts. Luego, ejecutar en el orden en que aparecen, los comandos que se muestran, reemplazando los valores en negrita:

```
$ psql -h <ip_server_postgresql> -p <puerto> -U postgres -f create_jcr_postgresql.sql
Password for user postgres: [ingresar la contraseña de postgres]
...
$ psql -h <ip_server_postgresql> -p <puerto> -U postgres -f
create_quartz_postgresql.sql
Password for user postgres: [ingresar la contraseña de postgres]
...
Password for user pentaho_user: [ingresar la palabra "password" o contraseña creada]
...
$ psql -h <ip_server_postgresql> -p <puerto> -U postgres -f
create_repository_postgresql.sql
Password for user postgres: [ingresar la contraseña de postgres]
...
Password for user hibuser: [ingresar la palabra "password" o contraseña creada]
...
```

Ahora abrir la consola de postgres y ejecutar "\l" para ver si se crearon correctamente las bases *jackrabbit*, *hibernate* y *quartz*. (También se puede ver desde el PgAdmin).

```
$ psql -h <ip_server_postgresql> -p <puerto> -U postgres
Password for user postgres: [ingresar la contraseña de postgres]
postgres=# \l
```

Solo como referencia, a continuación se muestra el listado de tablas y bases que deberían haberse creado luego de ejecutar los scripts.

- jackrabbit*
- quartz
 - qrtz5_blob_triggers
 - qrtz5_calendars
 - qrtz5_cron_triggers
 - qrtz5_fired_triggers
 - qrtz5_job_details
 - qrtz5_job_listeners
 - qrtz5_locks
 - qrtz5_paused_trigger_grps
 - qrtz5_scheduler_state

- `qrtz5_simple_triggers`
 - `qrtz5_trigger_listeners`
 - `qrtz5_triggers`
 - `"QRTZ"`
- `hibernate*`
 - `pro_audit`

* Se crearán nuevas tablas dentro de la base después de que se inicie por primera vez el BI-Server.

2.2.4.4. Configuración de la seguridad JDBC

Esta sección describe cómo configurar la seguridad JDBC en el BI-Server de Pentaho para utilizar el servidor PostgreSQL, esto significa que ahora el BI-Server apuntará a las bases `jackrabbit` e `hibernate` en el servidor PostgreSQL en vez de la base HSQL que viene por defecto.

NOTA

*Si ya se cuenta con un usuario al cual se le quiere dar permisos de acceso a la base `hibernate` o realizó un cambio de contraseña en la creación del usuario **hibuser**, se deben modificar todas las ocurrencias de **hibuser** / **password** en esta sección. Esto mismo se aplica a **pentaho_user** / **password** usado para conectarse a la base Quartz y para conectarse a la base `jackrabbit` **jcr_user** / **password**.*

`applicationContext-spring-security-hibernate.properties`

El archivo se encuentra en el directorio `/biserver-ce/pentaho-solutions/system/`.

Una vez abierto el archivo buscar el siguiente código:

```
jdbc.driver=org.hsqldb.jdbcDriver
jdbc.url=jdbc:hsqldb:hsql://localhost:9001/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.HSQLDialect
```

Modificar las secciones remarcadas de manera que queden similar a esto:

```
jdbc.driver=org.postgresql.Driver
jdbc.url=jdbc:postgresql://localhost:5432/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

`hibernate-settings.xml`

El archivo se encuentra en el directorio `/biserver-ce/pentaho-solutions/system/hibernate/`.

Una vez abierto el archivo buscar el siguiente código:

```
<config-file>system/hibernate/hsqldb.hibernate.cfg.xml</config-file>
```

Modificar las secciones remarcadas de manera que queden similar a esto:

```
<config-file>system/hibernate/postgresql.hibernate.cfg.xml</config-file>
```

postgresql.hibernate.cfg.xml

El archivo se encuentra en el directorio `/biserver-ce/pentaho-solutions/system/hibernate/`.

```
<property name="connection.url">jdbc:postgresql://localhost:5432/hibernate</property>
<property name="dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property name="connection.username">hibuser</property>
<property name="connection.password">password</property>
```

Modificar según corresponda.

repository.xml

El archivo se encuentra en el directorio `/biserver-ce/pentaho-solutions/system/jackrabbit/`. El mismo tiene incluido el código de configuración para PostgreSQL como comentarios. Solo hay que comentar el código original y descomentar el correspondiente para PostgreSQL como se muestra a continuación:

- Activar el código para PostgreSQL en la parte **FileSystem**

```
<FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="schema" value="postgresql"/>
  <param name="schemaObjectPrefix" value="fs_repos_"/>
</FileSystem>
```

- Comentar el código original:

```
<FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
  <param name="path" value="{rep.home}/repository"/>
</FileSystem>
```

- Activar el código para PostgreSQL en la parte **DataStore**

```
<DataStore class="org.apache.jackrabbit.core.data.db.DbDataStore">
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="databaseType" value="postgresql"/>
  <param name="minRecordLength" value="1024"/>
  <param name="maxConnections" value="3"/>
  <param name="copyWhenReading" value="true"/>
```

```

    <param name="tablePrefix" value=""/>
    <param name="schemaObjectPrefix" value="ds_repos_" />
  </DataStore>

```

- Comentar el código original

```

<DataStore class="org.apache.jackrabbit.core.data.FileDataStore"/>

```

- Activar el código para PostgreSQL en la parte de **WorkSpace**

```

<FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="schema" value="postgresql"/>
  <param name="schemaObjectPrefix" value="fs_ws_" />
</FileSystem>

```

- Comentar el código original

```

<FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
  <param name="path" value="{wsp.home}" />
</FileSystem>

```

- Activar el código para PostgreSQL en la parte **PersistenceManager** (primera parte).

```

<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.PostgreSQLPersistenceManager">
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="schema" value="postgresql"/>
  <param name="schemaObjectPrefix" value="{wsp.name}_pm_ws_" />
</PersistenceManager>

```

- Comentar el código original

```

<PersistenceManager
class="org.apache.jackrabbit.core.persistence.pool.H2PersistenceManager">
  <param name="url" value="jdbc:h2:{wsp.home}/db"/>
  <param name="schemaObjectPrefix" value="{wsp.name}_" />
</PersistenceManager>

```

- Activar el código para PostgreSQL en la parte de **Versiones**.

```
<FileSystem class="org.apache.jackrabbit.core.fs.db.DbFileSystem">
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="schema" value="postgresql"/>
  <param name="schemaObjectPrefix" value="fs_ver_" />
</FileSystem>
```

- Comentar el código original

```
<FileSystem class="org.apache.jackrabbit.core.fs.local.LocalFileSystem">
  <param name="path" value="{rep.home}/version" />
</FileSystem>
```

- Activar el código para PostgreSQL en la parte de **PersistenceManager** (segunda parte).

```
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.PostgreSQLPersistenceManager">
  <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
  <param name="driver" value="org.postgresql.Driver"/>
  <param name="user" value="jcr_user"/>
  <param name="password" value="password"/>
  <param name="schema" value="postgresql"/>
  <param name="schemaObjectPrefix" value="pm_ver_" />
</PersistenceManager>
```

- Comentar comentar el código original

```
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.pool.H2PersistenceManager">
  <param name="url" value="jdbc:h2:{rep.home}/version/db"/>
  <param name="schemaObjectPrefix" value="version_" />
</PersistenceManager>
```

2.2.4.5. Configuración de Hibernate y Quartz

Hibernate y Quartz necesitan específicamente utilizar las bases `hibernate` y `quartz` que se crearon en el servidor PostgreSQL. Para esto, hay que hacer algunas modificaciones al archivo `context.xml` ubicado en el directorio `/biserver-ce/tomcat/webapps/pentaho/META-INF/`.

NOTA

*Si ya se cuenta con un usuario al cual se le quiere dar permisos de acceso a la base hibernate o realizó un cambio de contraseña en la creación del usuario **hibuser**, se deben modificar todas las ocurrencias de **hibuser** / **password** en esta sección. Esto mismo se aplica a **pentaho_user** / **password** usado para conectarse a la base Quartz.*

context.xml

Las porciones del archivo a modificar son las resaltadas a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="hibuser" password="password"
    driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:hsqldb://localhost/hibernate"
    validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES" />
  <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="pentaho_user" password="password"
    driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:hsqldb://localhost/quartz"
    validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES"/>
</Context>
```

Modificar las partes en **negrita** de manera que queden similar a esto:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="hibuser" password="password"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://localhost:5432/hibernate"
    validationQuery="select 1" />
  <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="pentaho_user" password="password"
    driverClassName="org.postgresql.Driver" url="jdbc:postgresql://localhost:5432/quartz"
    validationQuery="select 1"/>
</Context>
```

pentaho.xml

El directorio `biserver-ce/tomcat/conf/Catalina/localhost` se crea automáticamente una vez que se inicia el BI-Server de Pentaho por primera vez y ahí aparece el archivo `pentaho.xml`, que debemos editar. Por lo tanto, los pasos serían los siguientes:

1. Verificar si existe el directorio `biserver-ce/tomcat/conf/Catalina/localhost`. Si no existe, (no es un problema), saltar el paso 2.
2. En caso de que exista, editar el archivo `pentaho.xml` en dicho directorio, haciendo que quede igual al archivo `context.xml` que se configuró anteriormente.

2.2.5. Configurar la Base SampleData

En [2.2.4. Instalar la base de datos PostgreSQL](#) no se explica como crear la base SampleData (datos de la aplicación de ejemplo) en PostgreSQL ya que es opcional. Si aún así usted instaló SampleData

en PostgreSQL, deberá configurar la conexión correspondiente en la opción de administración de Pentaho, de manera similar a como se configuran las otras conexiones en [3.2. Configuración de las conexiones JNDI](#).

2.2.6. Conectividad

Para acceder al BI-Server desde otras máquinas, deberá configurar la URL del BI-Server

En `biserver-ce/pentaho-solutions/system/server.properties` buscar la siguiente línea:

```
fully-qualified-server-url=http://localhost:8080/pentaho/
```

Y sustituir "localhost:8080" por **IP:puerto** o **nombre:puerto** con los que se va a acceder al servidor.

2.2.7. Cambiar el propietario

Los archivos y directorios de la instalación de pentaho (/usr/local/pentaho/ hacia adentro), tienen como propietario a un usuario de linux llamado *pentaho*. Hay que crearlo si es necesario. En adelante se debe tener cuidado de no cambiar al propietario al editar o manipular los archivos.

Cambiar el propietario de los archivos descomprimidos:

```
$ sudo chown --recursive pentaho:pentaho /usr/local/pentaho
```

2.2.8. Configuración de ambiente

2.2.8.1. Redirigir inicio

Abrir el archivo `plugin.xml` ubicado en `biserver-ce/pentaho-solutions/system/default-plugin/`, dentro del archivo buscar la configuración de `browser.perspective`.

```
<perspective id="browser.perspective" title="${browse}" layout-priority="-1"
resourcebundle="content/default-plugin/resources/messages/messages"
content-url="mantle/browser">
```

Modificar de acuerdo a lo que se encuentra en negrita:

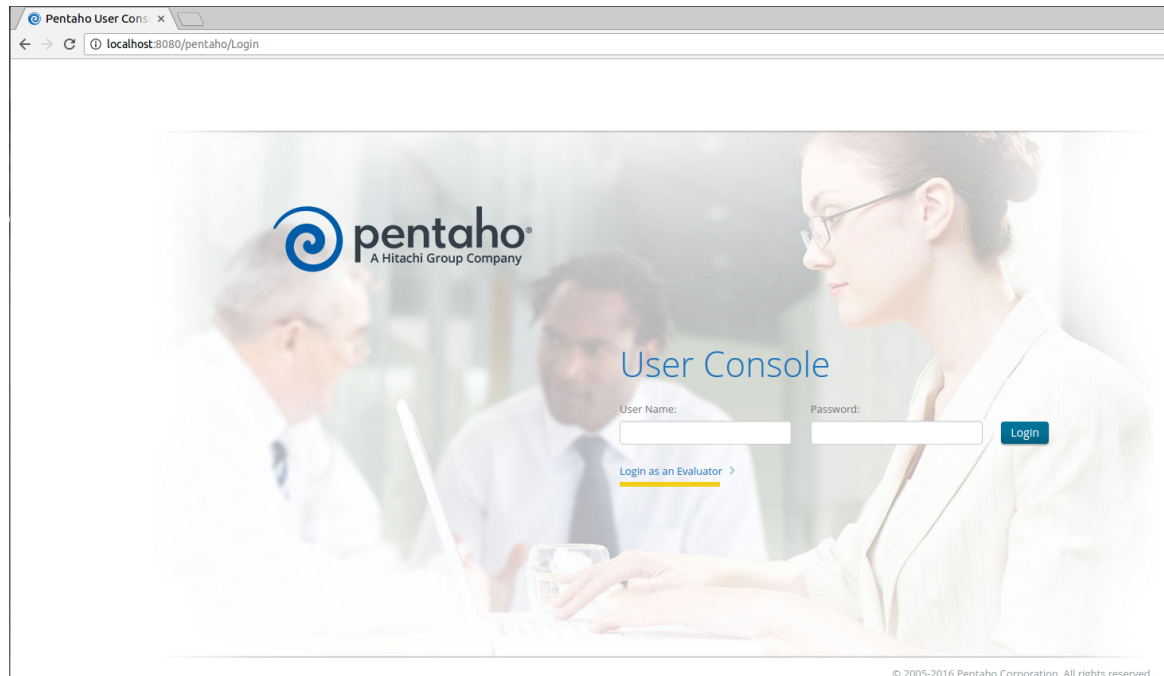
```
<perspective id="browser.perspective" title="${browse}" layout-priority="-3"
resourcebundle="content/default-plugin/resources/messages/messages"
content-url="mantle/browser">
```

2.2.9. Probar la instalación

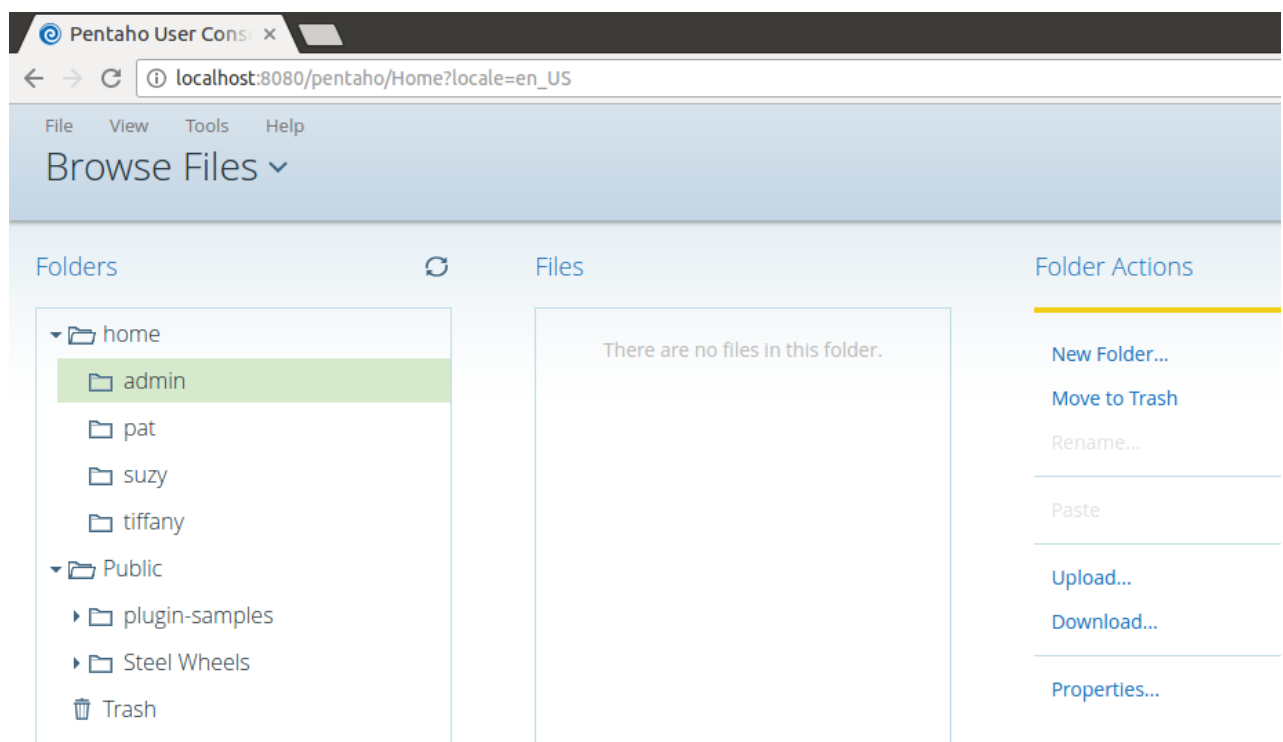
El servidor de Pentaho es una aplicación web que corre en el servidor Apache-Tomcat. Para iniciar el servidor Apache-Tomcat hay que ejecutar el script `start-pentaho.sh` ubicado en el directorio `/biserver-ce/`, como en las siguientes líneas luego de reemplazar el valor en negrita.

```
$ cd <path_biserver>
$ sudo -u pentaho ./start-pentaho.sh
```

Luego se puede acceder mediante el navegador en la url: <http://localhost:8080/pentaho> o <http://<host biserver>:8080/pentaho>. Si el servidor de Pentaho se inició correctamente debería ver la siguiente pantalla de bienvenida:



Luego probar de loguearse con el usuario administrador por defecto (usuario: **admin**, contraseña: **password**), y si funciona correctamente deberá ingresar a la consola de usuario como se muestra en la imagen siguiente:



2.2.10. Aspectos de seguridad

Si siguió los pasos anteriores, el BI-Server debería poder usarse. Sin embargo, para un entorno de producción, recomendamos cambiar todas las contraseñas por defecto y atender a la seguridad de las distintas bases de datos. Además recomendamos considerar los puntos mencionados en el link <https://help.pentaho.com/Documentation/6.1/0H0/060/010> de la wiki de Pentaho.

La documentación completa de la versión 6.1 se accede mediante el link <https://help.pentaho.com/Documentation/6.1>

2.3. Componentes visuales

Para poder utilizar las vistas de Saiku (nuevas o predefinidas), los tableros y los reportes, es necesario instalar los componentes visuales. Estos componentes los instala un usuario administrador desde la Consola de Usuario mediante Marketplace. los componentes necesarios son los siguientes:

- Community Dashboards Framework (CDF)
- Community Data Access (CDA)
- Community Dashboard Editor (CDE)
- Saiku Analytics - versión 3.8.8(PENTAHO6)
- Spanish Language Pack Installer

2.3.1. Instalación de componentes

Algunos de los componentes ya vienen instalados por defecto. CDF, CDA, CDE.

Instalar Spanish Language Pack Installer y Saiku Analytics.

Importante: Se debe instalar versión **3.8.8(PENTAHO6)** que es la Community.

2.3.2. Instalación de saiku

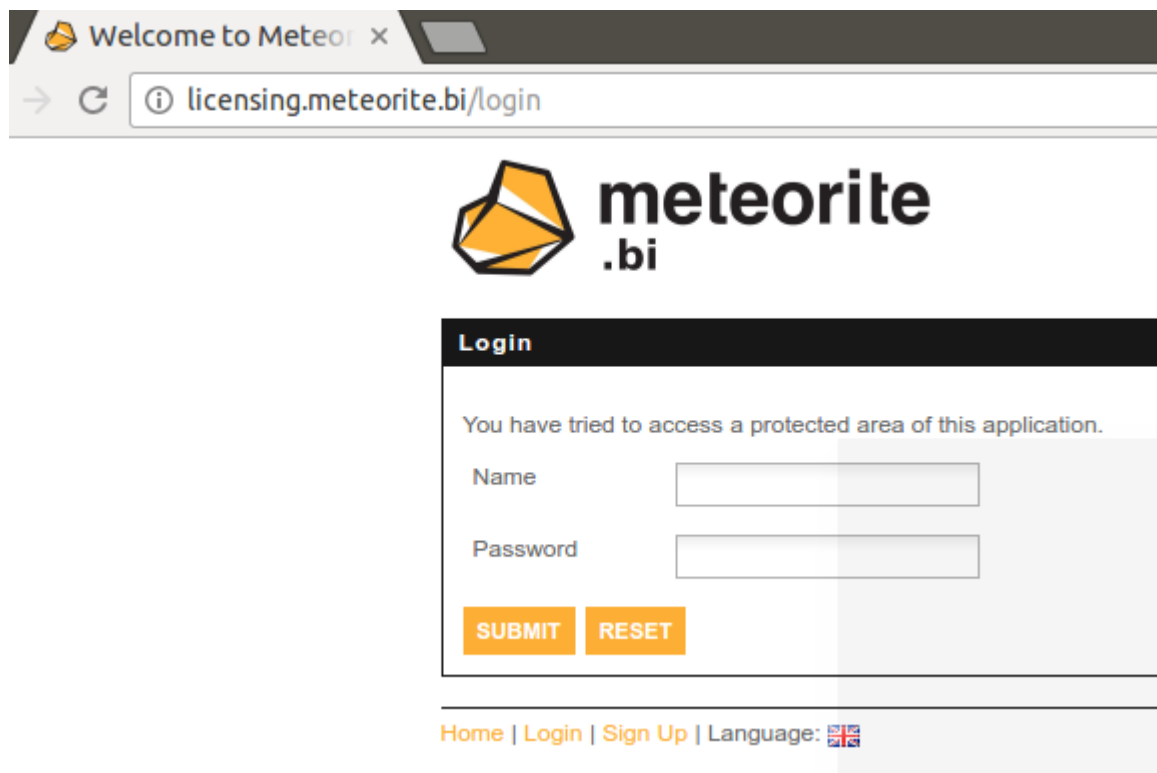
Deberá descargar el componente del siguiente link → [Saiku 3.8.8](#)

Paso 1: Descomprimir el zip en la carpeta `usr/local/pentaho/biserver-ce/pentaho-solutions/system`

Paso 2: Todas las versiones de saiku requieren una licencia. En nuestro caso necesitaremos descargar **Community Edition Free license**. Podemos descargarla en forma gratuita desde <http://licensing.meteorite.bi/>

Paso 3: Acceder a <http://licensing.meteorite.bi/>

Paso 4: hacer clic en *Sign Up*



Welcome to Meteor x

→ ↻ ⓘ licensing.meteorite.bi/login

 **meteorite**.bi

Login

You have tried to access a protected area of this application.

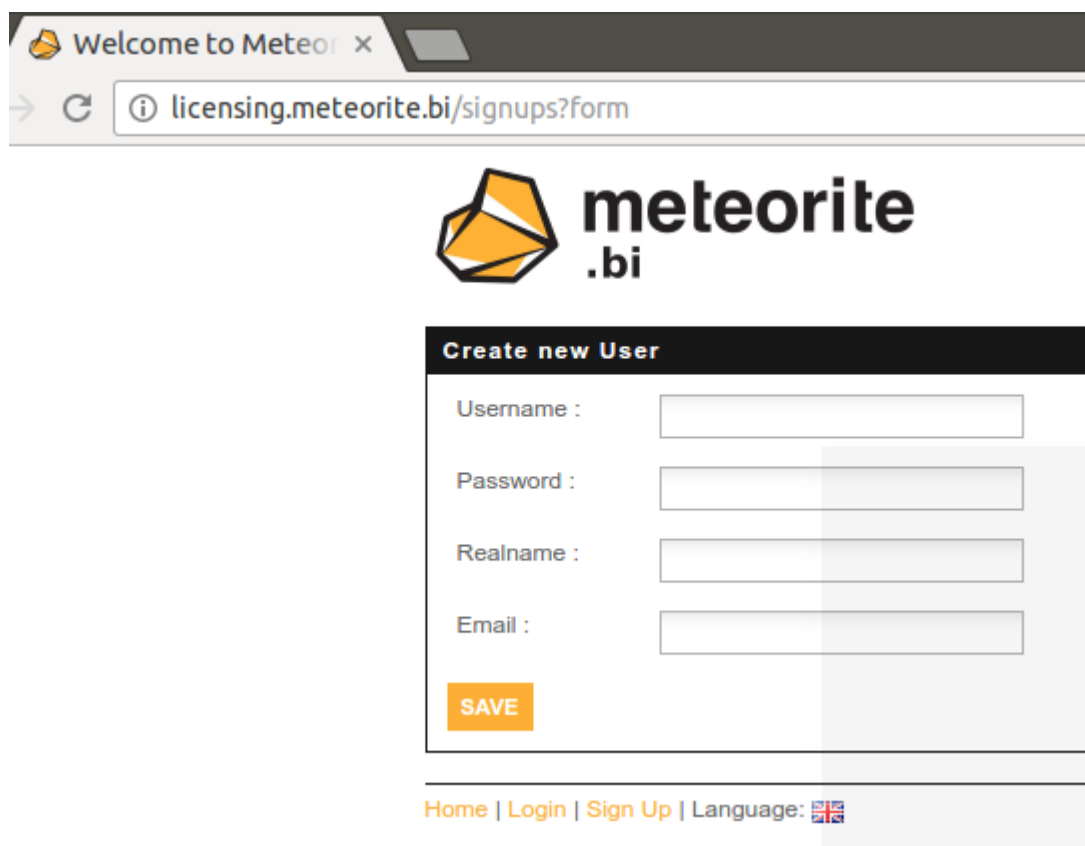
Name

Password

SUBMIT **RESET**

[Home](#) | [Login](#) | [Sign Up](#) | Language: 

Paso 5: Crear un usuario completando los campos requeridos



Welcome to Meteor x

→ ↻ ⓘ licensing.meteorite.bi/signup?form

 **meteorite**.bi

Create new User

Username :

Password :

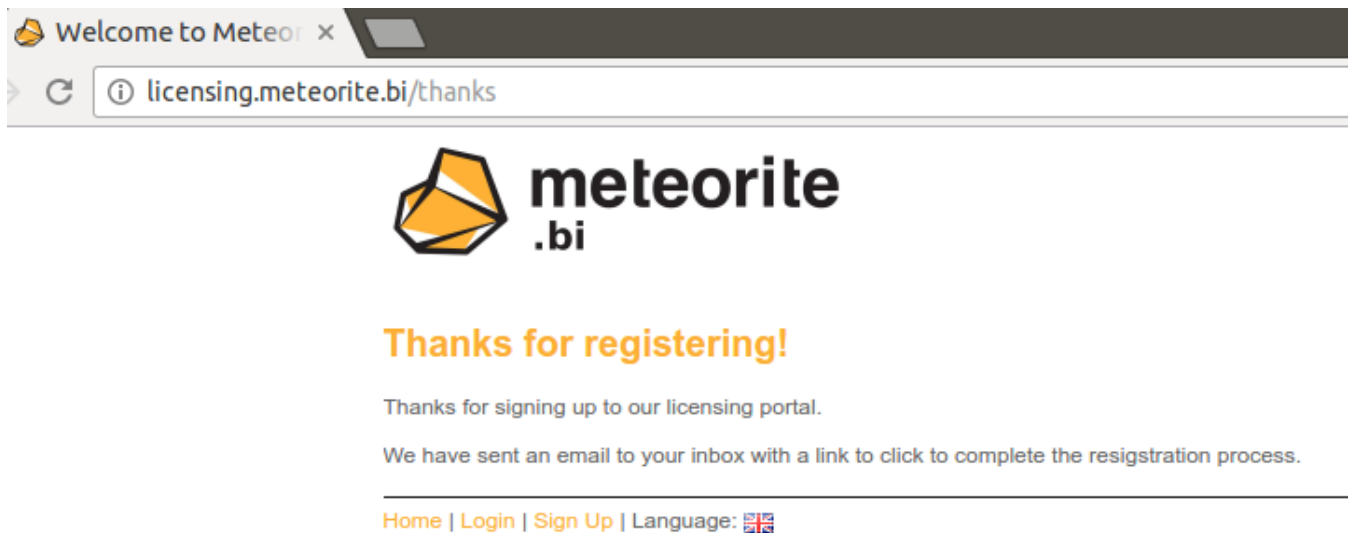
Realname :

Email :

SAVE

[Home](#) | [Login](#) | [Sign Up](#) | Language: 

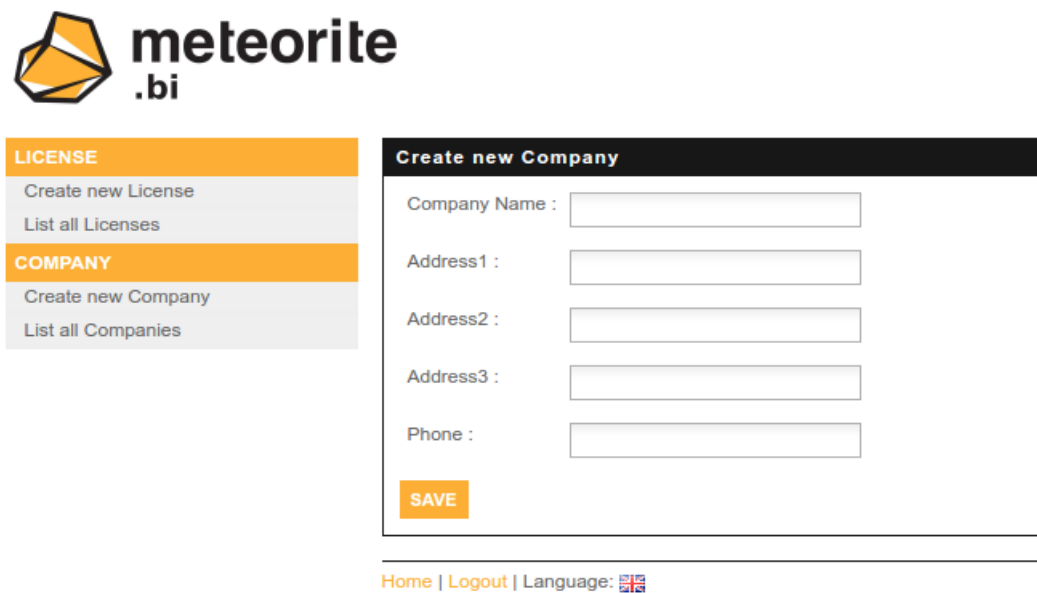
Una vez completos, clic en Save y verá lo siguiente



Paso 6: Debe ingresar al mail que utilizó para crear el usuario. En el mismo deberá tener un correo de *info@meteorite.bi* con asunto *Saiku Analytics - Thanks for registering!*. Para completar el registro, deberá ingresar al link enviado en dicho correo.

Paso 7: El link seleccionado lo enviará nuevamente a la página de Login de *meteorite.bi*. Ingresar con el usuario y contraseña generados anteriormente.

Paso 8: Crear una Compañía haciendo clic en *Create new Company*



Los campos a completar son:

Company Name: nombre de la institución/universidad

Address1: dirección de la institución/universidad

Address2: dirección de la institución/universidad (opcional)

Address3: dirección de la institución/universidad (opcional)

Phone: teléfono de la institución/universidad (opcional)

Una vez completos, clic en **Save**.

Paso 9: Crear una licencia haciendo clic en *Create new License*



LICENSE
Create new License
List all Licenses

COMPANY
Create new Company
List all Companies

Create new License
Hostname :
On a Unix server type the 'hostname' command, on Windows server run 'ipconfig /all'
Max Users :
Only requires setting for a full license.
License Type :
Username :
Company :
SAVE

[Home](#) | [Logout](#) | Language:

Los campos a completar son:

Hostname: nombre del server donde se instaló pentaho 6 (*)

Max Users: cantidad máxima de usuarios que utilizaran saiku

License Type: tipo de licencia requerida. En nuestro caso *COMMUNITY_EDITION*.

Username: nombre de usuario generado

Company: compañía creada en el Paso 10.

(*) para obtener el nombre del servidor, abrir una terminal y escribir el comando *hostname*.

Una vez completos, clic en **Save** y podrá observar algo similar a lo que se muestra en la siguiente imagen:


meteorite
.bi

LICENSE
 Create new License
 List all Licenses
COMPANY
 Create new Company
 List all Companies

Show License
 Hostname :
 Max Users : 100
 Expiry Date :
 License Type : COMMUNITY_EDITION
 Uuid : c22f5f91-5310-4230-9b16-ca3ed9b4b333
 Company : Phoenix S2S, CABA
DOWNLOAD LICENSE

[Home](#) | [Logout](#) | Language:

Paso 10: Descargar la licencia haciendo clic en *Download License*. Descargara un archivo con el nombre *license_hostname.lic*.

Paso 11: Renombrar el archivo descargado a *license.lic*

Paso 12: Copiar el archivo en `/biserver-ce/pentaho-solutions/system/saiku`

Paso 13: Reiniciar Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`

Paso 14: desde la Consola de Usuario, verificar que saiku esté instalado correctamente ingresando a File → New → Saiku Analytics

2.3.3. Configuración de saiku

Paso 1: Detener Pentaho utilizando `/biserver-ce/stop-pentaho.sh`

Paso 2: Editar el archivo `importExport.xml` ubicado en `biserver-ce/pentaho-solutions/system/`

Buscar la siguiente porcion de codigo:

```
<bean id="IRepositoryContentConverterHandler"
class="org.pentaho.platform.plugin.services.importer.DefaultRepositoryContentConverterHandle
r" scope="singleton">
  <constructor-arg>
    <util:map id="convertersMap">
      <entry key="mondrian.xml" value-ref="streamConverter"/>
      <entry key="jpeg" value-ref="streamConverter"/>
    </util:map>
  </constructor-arg>
</bean>
```

Agregar la configuración de saiku de la siguiente manera:

```

<bean id="IRepositoryContentConverterHandler"
class="org.pentaho.platform.plugin.services.importer.DefaultRepositoryContentConverterHandle
r" scope="singleton">
    <constructor-arg>
        <util:map id="convertersMap">
            <entry key="mondrian.xml" value-ref="streamConverter"/>
            <entry key="jpeg" value-ref="streamConverter"/>
            <entry key="saiku" value-ref="streamConverter"/>
        </util:map>
    </constructor-arg>
</bean>

```

Buscar la siguiente porcion de codigo:

```

<bean id="DefaultExportHandler"
class="org.pentaho.platform.plugin.services.importexport.DefaultExportHandler">
    <property name="repository" ref="unifiedRepository" />
    <property name="localeExportList">
        <list>
            <value>.xanalyzer</value>
            <value>.prpti</value>
            <value>.prpt</value>
            <value>.xaction</value>
            <value>.xdash</value>
            <value>.url</value>
            <value>.xanalyzer</value>
            <value>.xjpivot</value>
            <value>.xcdf</value>
        </list>
    </property>
</bean>

```

Agregar la configuración de saiku de la siguiente manera:

```

<bean id="DefaultExportHandler"
class="org.pentaho.platform.plugin.services.importexport.DefaultExportHandler">
    <property name="repository" ref="unifiedRepository" />
    <property name="localeExportList">
        <list>
            <value>.xanalyzer</value>
            <value>.prpti</value>
            <value>.prpt</value>
            <value>.xaction</value>
            <value>.xdash</value>
            <value>.url</value>
            <value>.xanalyzer</value>
            <value>.xjpivot</value>
            <value>.xcdf</value>
            <value>.saiku</value>
        </list>
    </property>
</bean>

```

Buscar la siguiente porcion de codigo:

```

<bean
class="org.pentaho.platform.plugin.services.importer.LocaleImporter">
  <constructor-arg>
    <bean factory-bean="MimeTypeListFactory" factory-method="createMimeTypeList" >
      <constructor-arg
value="org.pentaho.platform.plugin.services.importer.LocaleImporter"/>
    </bean>
  </constructor-arg>
  <constructor-arg>
    <list>
      <value>xaction</value>
      <value>url</value>
      <value>xdash</value>
      <value>prpt</value>
      <value>prpti</value>
      <value>xanalyzer</value>
      <value>wcdf</value>
      <value>xcdf</value>
    </list>
  </constructor-arg>
  <property name="repository" ref="unifiedRepository" />
  <property name="defaultAclHandler" ref="defaultAclHandler" />
</bean>

```

Agregar la configuración de saiku de la siguiente manera:

```

<bean
class="org.pentaho.platform.plugin.services.importer.LocaleImporter">
  <constructor-arg>
    <bean factory-bean="MimeTypeListFactory" factory-method="createMimeTypeList" >
      <constructor-arg
value="org.pentaho.platform.plugin.services.importer.LocaleImporter"/>
    </bean>
  </constructor-arg>
  <constructor-arg>
    <list>
      <value>xaction</value>
      <value>url</value>
      <value>xdash</value>
      <value>prpt</value>
      <value>prpti</value>
      <value>xanalyzer</value>
      <value>wcdf</value>
      <value>xcdf</value>
      <value>saiku</value>
    </list>
  </constructor-arg>
  <property name="repository" ref="unifiedRepository" />
  <property name="defaultAclHandler" ref="defaultAclHandler" />
</bean>

```

Paso 3: Editar el archivo *ImportHandlerMimeTypeDefinitions.xml* ubicado en *biserver-ce/pentaho-solutions/system/*

Buscar la siguiente porción de código:

```
<MimeTypeDefinition mimeType="text/pdf" hidden="false">
  <extension>pdf</extension>
</MimeTypeDefinition>
<MimeTypeDefinition mimeType="text/xml">
  <extension>cda</extension>
....
```

Agregar la configuración de saiku de la siguiente manera:

```
<MimeTypeDefinition mimeType="text/pdf" hidden="false">
  <extension>pdf</extension>
</MimeTypeDefinition>
<MimeTypeDefinition mimeType="application/json">
  <extension>saiku</extension>
</MimeTypeDefinition>
<MimeTypeDefinition mimeType="text/xml">
  <extension>cda</extension>
....
```

Paso 4: Editar el archivo `saiku.min.js` ubicado en `/biserver-ce/pentaho-solutions/system/saiku/ui/`

Buscar la siguiente porción de código, la encontrará dos veces:

```
$(this.el).append("\x3cd\x3e\x3cd\x3e id\x3d'uphead' class\x3d'upgradeheader'\x3eYou are
using Saiku Community Edition, please consider upgrading to \x3ca target\x3d'_blank'
href\x3d'http://meteorite.bi'\x3eSaiku Enterprise\x3c/a\x3e, or entering a \x3ca
href\x3d'http://meteorite.bi/products/saiku/sponsorship'\x3esponsorship agreement with
us\x3c/a\x3e to support development. \x3ca
href\x3d'http://meteorite.bi/products/saiku/community'\x3eOr contribute by joining our
community and helping other users!\x3c/a\x3e\x3c/div\x3e\x3c/div\x3e") }
```

Eliminar, en ambos lugares, lo que está en negrita de modo que quede como sigue:

```
$(this.el).append("") }
```

Buscar la siguiente porción de código, la encontrará cinco veces:

```
add(new SplashScreen)
```

Reemplazar, en los cinco lugares encontrados por:

```
add(new Workspace)
```

Paso 5: copiar el archivo `SaikuWidgetComponent.js` ubicado en la carpeta `utilidades/instalacion/saiku` del paquete SIU-Wichi en el directorio `/biserver-ce/pentaho-solutions/system/saiku/components/saikuWidget/`.

Paso 6: editar el archivo `SaikuEmbed.js` ubicado en `/biserver-ce/pentaho-solutions/system/saiku/ui/js/saiku/embed`

Buscar la siguiente porción de código:

```
var _settings = {  
    server: '/saiku',  
    path: '/rest/saiku/embed',  
    user: 'admin',  
    password: 'admin',  
    blockUI: false  
};
```

Reemplazar los valores en negrita de forma que quede como sigue:

```
var _settings = {  
    server: null,  
    path: null,  
    user: null,  
    password: null,  
    blockUI: false  
};
```

Paso 7: Iniciar Pentaho utilizando `/biserver-ce/start-pentaho.sh`

Paso 8: puede verificar ingresando a la Consola de usuario, opción File → New → Saiku Analytics.

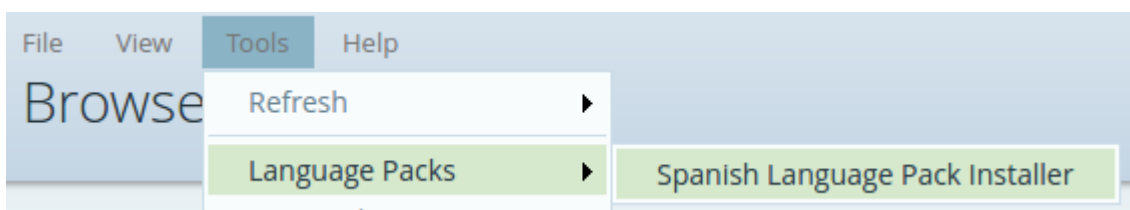
2.3.4. Instalación de Spanish Language Pack

Deberá descargar el componente del siguiente link → [Spanish Language Pack](#)

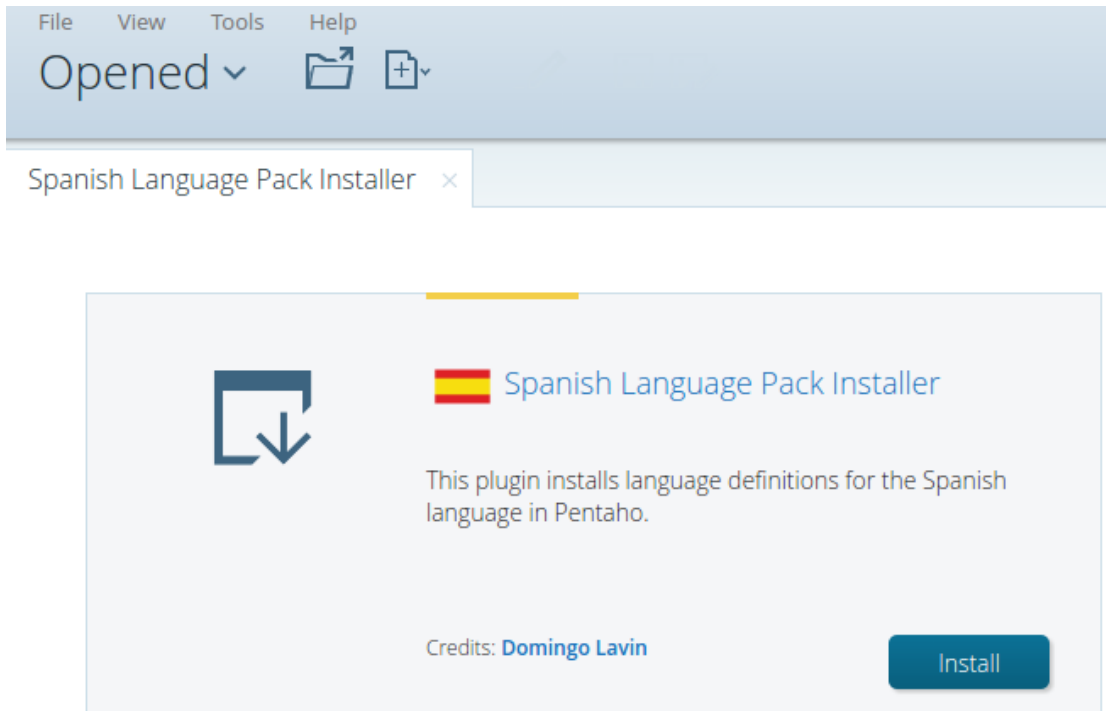
Paso 1: Descomprimir el zip en la carpeta `usr/local/pentaho/biserver-ce/pentaho-solutions/system`

Paso 2: reiniciar el Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`

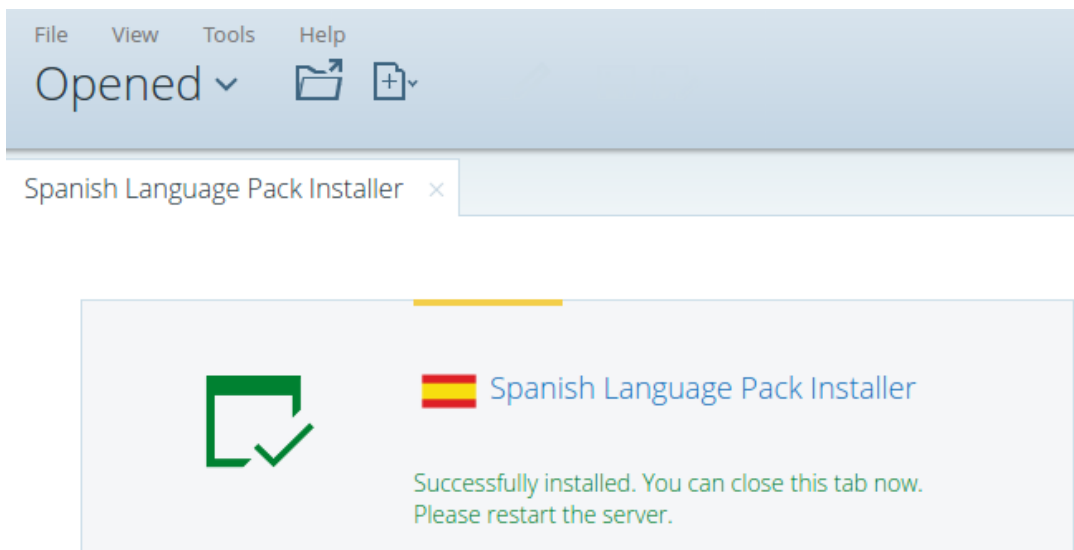
Paso 3: ir a Tools → Language Packs → Spanish Language Pack Installer



Paso 4: hacer clic en Install



Una vez instalado le mostrará la siguiente imagen:



Paso 5: reiniciar el Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`

Paso 6: desde la Consola de Usuario, ir a View → Languages → Spanish

2.4. Pentaho Data Integration

Pentaho Data Integration (PDI), más conocido como kettle, es una herramienta necesaria para ejecutar procesos de extracción, transformación y carga (ETL), necesarios en varias de las tareas explicadas en esta guía.

Paso 1: Descarga.

Descargar Pentaho Data Integration 6.1 del link

<https://sourceforge.net/projects/pentaho/files/Data%20Integration/6.1/pdi-ce-6.1.0.1-196.zip/download>

Paso 2: Descompresión.

Descomprimir el archivo en alguna carpeta, por ejemplo en /usr/local/pentaho/data-integration, reemplazando **RUTA_DESCARGA** en las siguientes líneas y ejecutando.

```
$ cd /usr/local/pentaho
$ sudo unzip RUTA_DESCARGA/pdi-ce-6.1.0.1-196.zip
```

Paso 3: copiar el jar de postgres descargado en el punto [2.2.4.1. Instalación del driver PostgreSQL](#) y pegarlo dentro de **data-integration/lib****Paso 4:** Memoria de Spoon y Kitchen.

Es conveniente aumentar el límite de memoria de la herramienta Kitchen y Spoon, lo cual es necesario para los procesos de carga de datos. Editar el archivo **spoon.sh**.

Buscar el texto

```
PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m"
```

Paso 5: Permisos.

Corregir los permisos si es necesario, reemplazando **USUARIO** en la siguiente línea y ejecutando

```
$ sudo chown --recursive USUARIO data-integration
```

Luego de estos pasos, la herramienta quedaría instalada. Se puede probar ejecutando los scripts **spoon.sh** (desde el directorio en que se encuentra), **kitchen.sh** o **carte.sh**.

3. Instalación de SIU-Wichi 6.7.0

Antes de continuar con la instalación del sistema SIU-Wichi 6.7.0, hay que asegurarse de tener instalado y configurado lo siguiente:

- [BI-Server 6.1 de Pentaho](#)
- Servidor PostgreSQL 9.x

3.1. Creación de la base de datos SIU-Wichi

La creación de la base de datos `siu_wichi` consiste en la creación de la base de datos en PostgreSQL y la creación de la estructura (DDL). Para poder crear la estructura, será necesario tener instalada la herramienta Pentaho Data Integration (PDI).

3.1.1. Creación de la base en Postgres

El primer paso es crear manualmente la base de datos vacía en PostgreSQL, ya sea desde PgAdmin o desde la línea de comandos.

Es importante establecer la codificación de la base de datos en UTF8.

Debido a la encriptación de las claves se debe instalar, si es necesario, el paquete adicional *contrib* correspondiente a la versión de postgresql instalada.

```
$ apt-get install postgresql-contrib-9.4
```

Reemplazar el valor en **negrita** por el correspondiente a su versión de motor de base de datos.

3.1.2. Creación de la estructura mediante ETL

La creación de la estructura de la base de datos completa se realiza con un solo proceso ETL. Al igual que todos los procesos de ETL se puede ejecutar desde la interfaz gráfica *spoon* o desde la línea de comando. En esta guía mostraremos los pasos de ejecución de procesos de ETL mediante las líneas de comandos.

El proceso `principal_crear_siu_wichi` creará la estructura al ejecutar las siguientes líneas luego de reemplazar los valores en **negrita**.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
-file="<path_etl>/creacion_base_datos/principal_crear_siu_wichi.kjb" \
-param="base_clave=postgres" \
-param="base_nombre=siu_wichi" \
-param="base_host=localhost" \
-param="base_puerto=5432" \
-param="base_usuario=postgres"
```

3.2. Configuración de las conexiones JNDI

Consiste en configurar las conexiones a las bases de datos `siu_wichi` e `hibernate` para poder ser accedidas desde SIU-Wichi mediante JNDI.

Paso 1: detener biserver utilizando `./stop-pentaho.sh`

Paso 2: editar el archivo `web.xml` ubicado en `/biserver-ce/tomcat/webapps/pentaho/WEB-INF`. Buscar el texto

```
<!-- insert additional resource-refs -->
```

e insertar la siguiente porción de código:

```
<resource-ref>
  <description>DBWichi</description>
  <res-ref-name>jdbc/DBWichi</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>

<resource-ref>
  <description>DBHibernate</description>
  <res-ref-name>jdbc/DBHibernate</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

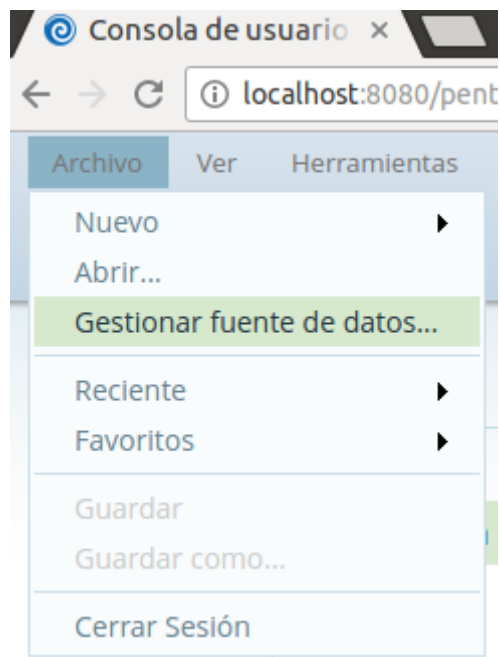
Paso 3: editar el archivo `context.xml` ubicado en `/pentaho/biserver-ce/tomcat/webapps/pentaho/META-INF` e insertar la siguiente porción de código, reemplazando los valores en negrita:

```
<Resource name="jdbc/DBWichi" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
maxIdle="5"
  maxWait="10000" username="postgres" password="postgres"
  driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/siu_wichi"
  validationQuery="select 1" />
  <Resource name="jdbc/DBHibernate" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
maxIdle="5"
  maxWait="10000" username="hibuser" password="password"
  driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/hibernate"
  validationQuery="select 1" />
```

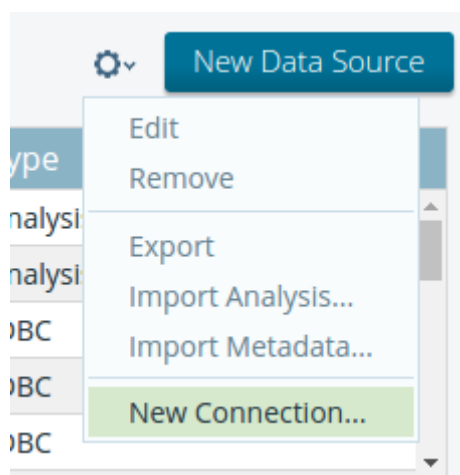
Paso 4: Iniciar Pentaho utilizando `/biserver-ce/start-pentaho.sh`

Paso 5: agregar las conexiones JNDI desde la Consola de Usuario.

- Ir a Archivo → Gestionar fuente de datos... Se abre un pop-up de Data Sources.



- hacer clic en  y New Connection...



- Configurar la conexión DBWichi como se muestra en la imagen.
 - Connection Name: DBWichi
 - Database Type: PostgreSQL
 - Access: JNDI
 - JNDI Name: DBWichi

Database Connection

General
Advanced
Options
Pooling

Connection Name:
DBWichi

Database Type:
Hadoop Hive
Hypersonic
Impala
Cloudera Impala
MS SQL Server
MonetDB
MySQL
Pentaho Data Services
PostgreSQL

Settings
JNDI Name:
DBWichi

Access:
Native (JDBC)
ODBC
JNDI

Adding Databases

Test

OK Cancel

- Probar con el botón Test y aceptar.
- Agregar una nueva conexión, hacer clic en  y New Connection...
- Configurar la conexión DBHibernate como se muestra en la imagen.
 - Connection Name: DBHibernate
 - Database Type: PostgreSQL
 - Access: JNDI
 - JNDI Name: DBHibernate

Database Connection

General
Advanced
Options
Pooling

Connection Name:
DBHibernate

Database Type:
Hadoop Hive
Hypersonic
Impala
Cloudera Impala
MS SQL Server
MonetDB
MySQL
Pentaho Data Services
PostgreSQL

Settings
JNDI Name:
DBHibernate

Access:
Native (JDBC)
ODBC
JNDI

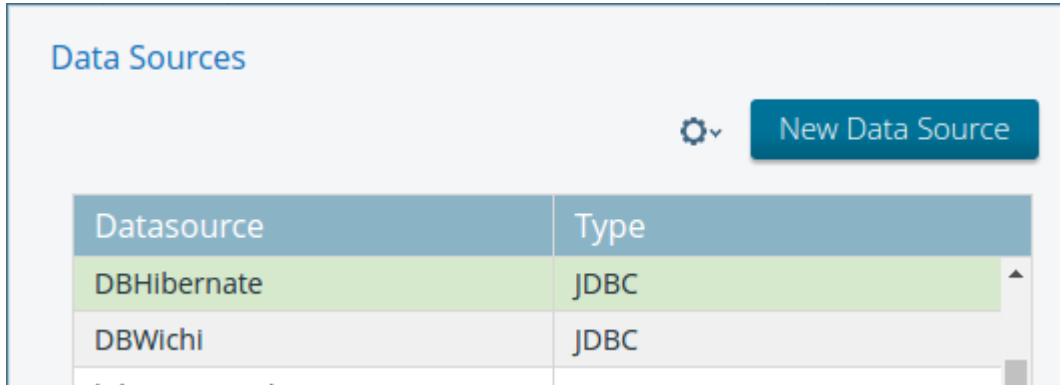
Adding Databases

Test

OK Cancel

- Probar con el botón Test y aceptar.

Las conexiones JNDI se verán como sigue:



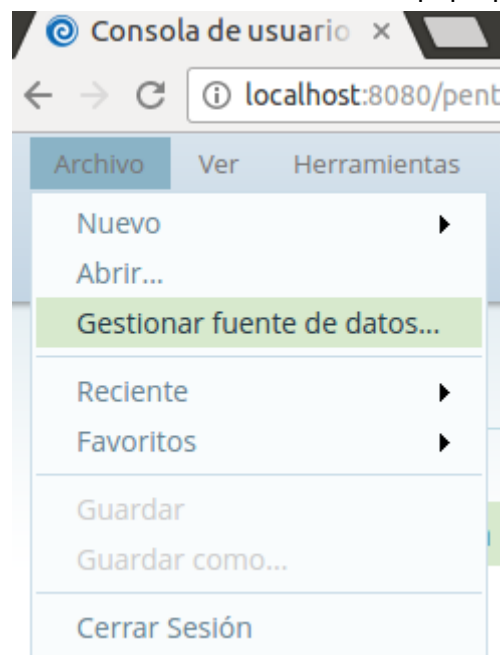
Datasource	Type
DBHibernate	JDBC
DBWichi	JDBC

3.3. Registro de esquemas

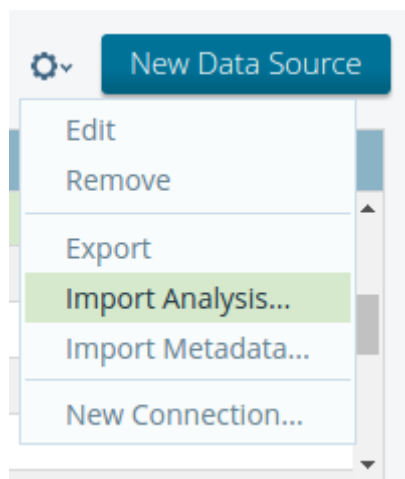
Para que los cubos de análisis estén disponibles y accesibles, hay que registrarlos desde la Consola de Usuario. Debemos registrar los cubos correspondientes a SIU-Wichi y el cubo de Log de Accesos. Luego hay que habilitar el mapeo de mondrian para que los cubos puedan ser utilizados por los reportes del Portal Gerencial.

3.3.1. Registro de esquemas

Paso 1: Ir a Archivo → Gestionar fuente de datos... Se abre un pop-up de Data Sources.



Paso 2: hacer clic en  e Import Analysis...



Se abre el siguiente pop-up

A screenshot of the 'Import Analysis' pop-up window. It has a title bar 'Import Analysis'. Inside, there is a 'Mondrian File:' label followed by a text input field and a blue button with three dots. Below this are two radio buttons: 'Select from available data sources.' (selected) and 'Manually enter data source parameter values.'. Under the selected radio button is a 'Data Source:' label followed by a dropdown menu showing 'AgileBI'. At the bottom right are two buttons: 'Import' (disabled) and 'Close'.

Paso 3: seleccionar el Data Source *DBWichi*.

A screenshot of the 'Import Analysis' pop-up window, similar to the previous one, but the dropdown menu under 'Data Source:' now shows 'DBWichi' instead of 'AgileBI'. The 'Import' button remains disabled.

Paso 4: hacer clic en  y seleccionar el archivo SIU-Wichi.xml ubicado en el directorio */utilidades/esquemas* del paquete SIU-Wichi.

Import Analysis

Mondrian File:

C:\fakepath\SIU-Wichi.xml

☒ Select from available data sources.
☐ Manually enter data source parameter values.

Data Source:

DBWichi

Import Close



Paso 5: importar haciendo clic en

Paso 6: repetir los pasos 2, 3, 4 y 5, seleccionando en el paso 4 el archivo SIU-LogAccesos.xml ubicado en el directorio /utilidades/esquemas del paquete SIU-Wichi.

El registro de cubo se verá como sigue:

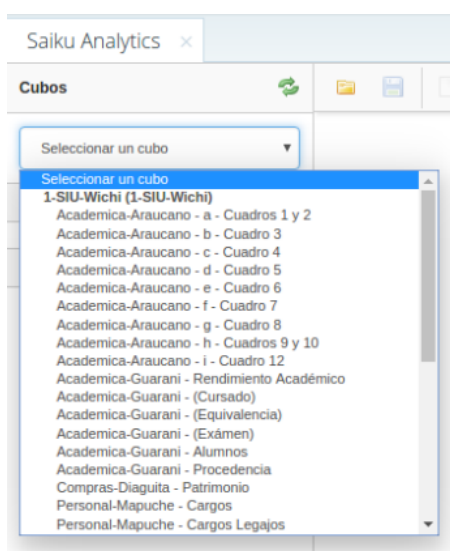
Data Sources

New Data Source

Datasource	Type
1-SIU-Wichi	Analysis
2-Administrar-Servidor	Analysis

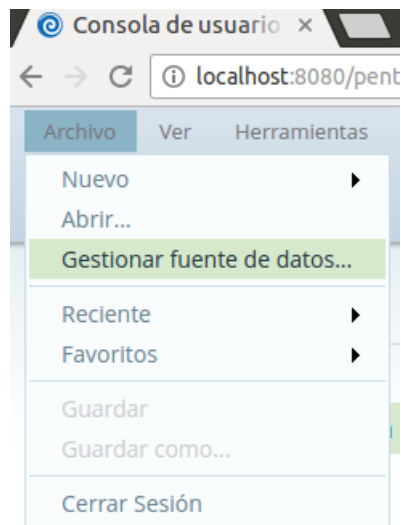
Paso 7: Ir a Herramientas → Actualizar → Caché del esquema mondrian.

Paso 8: Se puede verificar los cubos disponibles accediendo Archivo → Nuevo → Saiku Analytics.

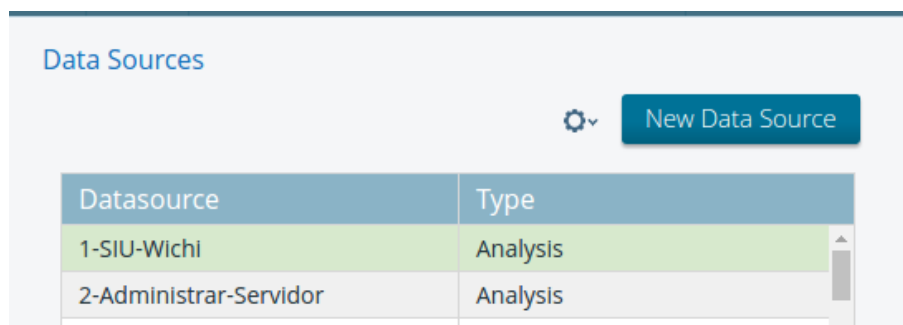


3.3.2. Actualización de esquema (SIU-Wichi.xml)

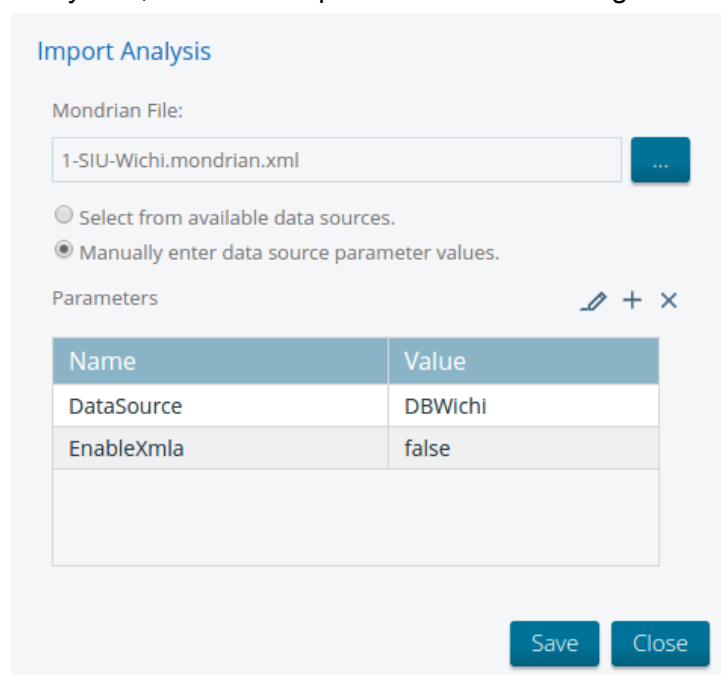
Paso 1: Ir a Archivo → Gestionar fuente de datos... Se abre un pop-up de Data Sources.



Paso 2: hacer clic en el cubo que queremos actualizar, por ejemplo 1-SIU-Wichi.



Paso 3: hacer clic en  y Edit, veremos lo que se muestra en la siguiente imagen

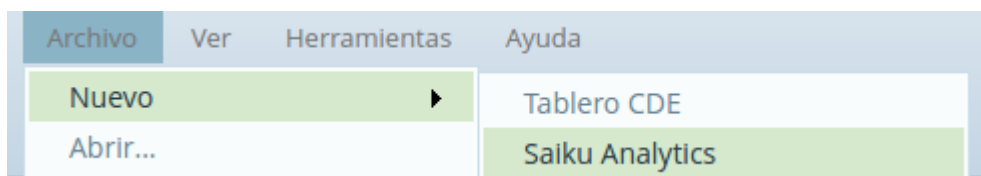


Paso 4: hacer clic en  y seleccionar el archivo SIU-Wichi.xml modificado.

Paso 5: guardamos el cambio haciendo clic en .

Paso 6: ir a Herramientas → Actualizar → Caché del esquema mondrian.

Paso 7: ir a Archivo → Nuevo → Saiku Analytics



Paso 8: refrescar los cubos haciendo clic en .



3.3.3. Mapeo de esquemas para reportes

Paso 1: Abrir el archivo `classic-engine.properties` ubicado en `biserver-ce/tomcat/webapps/pentaho/WEB-INF/classes` y añadir las siguientes líneas de código:

```
# Agregado para el mapping de cubos con reportes
mondrian.spi.CatalogLocator=org.pentaho.reporting.engine.classic.extensions.datasources.mondrian.LegacyCatalogLocator
```

Paso 2: Crear el archivo `mondrian-schema-mapping.properties` dentro de `biserver-ce/tomcat/webapps/pentaho/WEB-INF/classes/` y añadir el mapping para el catálogo mondrian previsto.

```
# -----
# Agregar el mapeo de las conexiones con los esquemas
# -----
SIU-Wichi.xml=mondrian:/1-SIU-Wichi
SIU-LogAccesos.xml=mondrian:/2-Administrar-Servidor
```

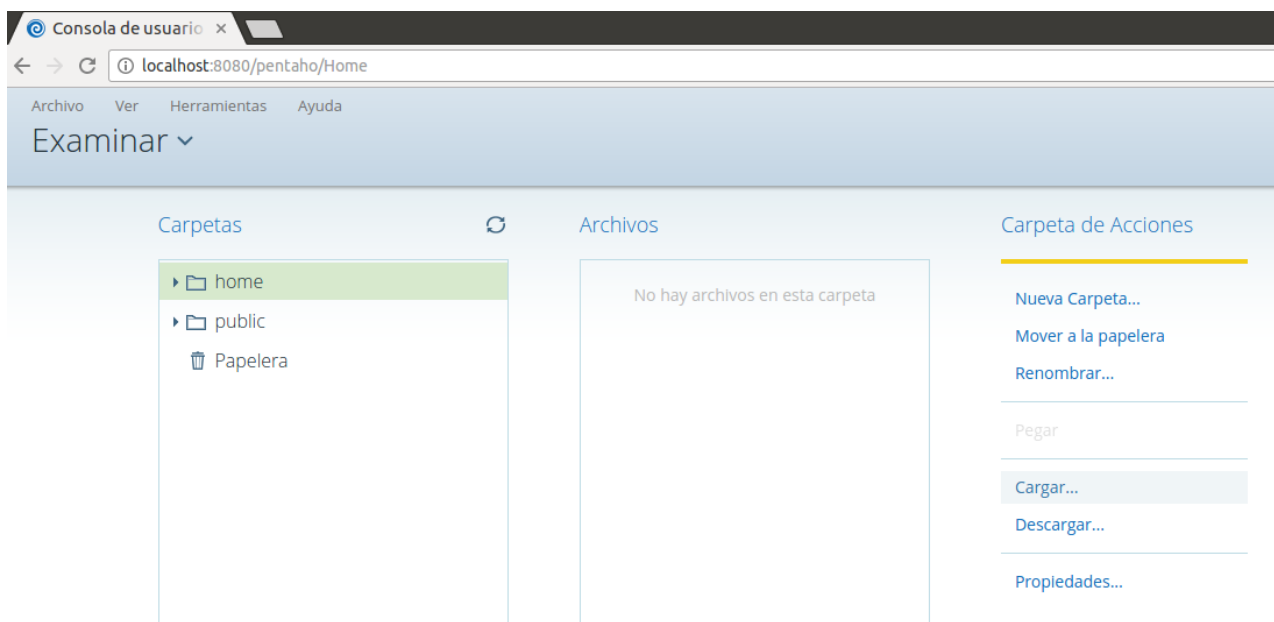
La nueva sintaxis para catálogos mondrian debe atenerse a la regla **mondrian:/**

Paso 3: Reiniciar Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`

3.4. Carpeta SIU-Wichi

Para que la solución SIU-Wichi esté accesible como carpeta, debemos subirla desde la Consola de Usuario.

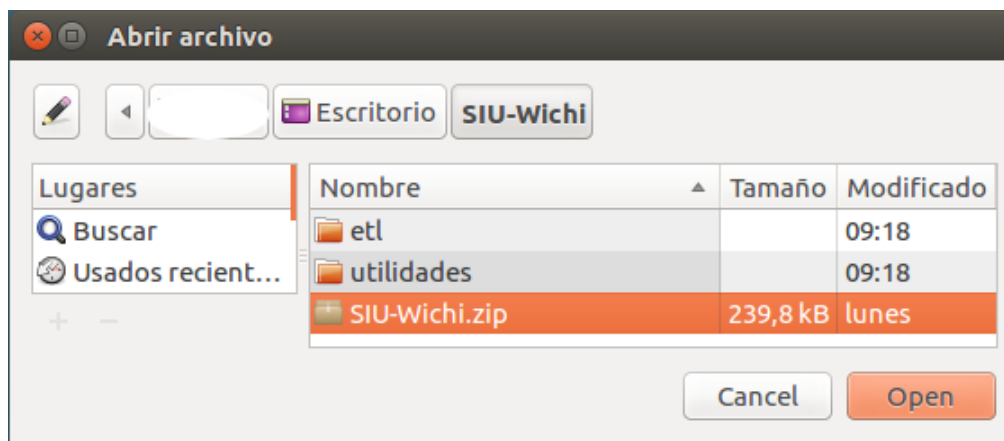
Paso 1: Posicionarse en la carpeta home.



Paso 2: A la derecha podemos visualizar las acciones que podemos realizar sobre una carpeta. Hacemos clic en la acción Cargar...

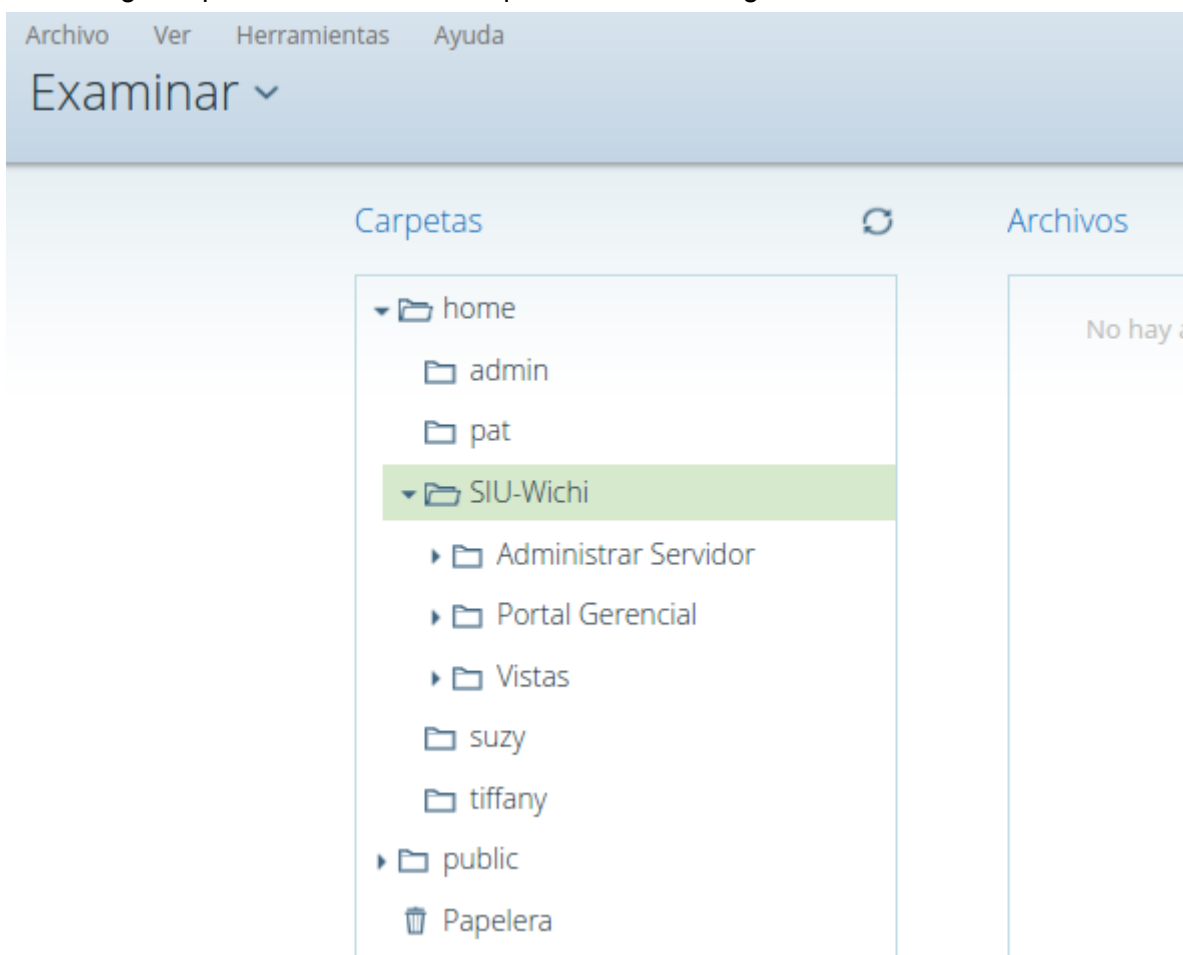


Paso 3: hacer clic en  y seleccionar el archivo SIU-Wichi.zip disponible en el paquete SIU-Wichi.



Paso 4: en Opciones avanzadas existen configuraciones pero dejaremos las que están por defecto. Luego [Aprobar](#), la carga puede tardar unos minutos.

Una vez cargada, podremos observar lo que muestra la imagen:



3.5. Instalación de utilidades para administrar el servidor

A continuación se explica la configuración que permite el funcionamiento de las utilidades de la carpeta *Administrar Servidor*.

3.5.1. Log de accesos

Se puede configurar el BI-Server para monitorear desde la consola de usuario el acceso a las vistas, tableros y reportes guardados. Se configura para que guarde un log en la base hibernate, a partir del cual se vuelcan datos a la base siu_wichi. Luego se podrá acceder desde el cubo **Log Accesos** y los reportes a los datos de log generados.

Un requisito para esto es haber seguido todos los pasos de [2. Instalación de componentes Pentaho](#) sin haber omitido la configuración de la base PostgreSQL. Los pasos necesarios para configurar el monitoreo del log son los siguientes.

Paso 1: Editar el archivo `pentahoObjects.spring.xml` ubicado en `pentaho-solutions/system/`.
Buscar la siguiente porción de código:

```
<bean id="IAuditEntry"  
class="org.pentaho.platform.engine.services.audit.AuditFileEntry" scope="singleton" />
```

y reemplazar por:

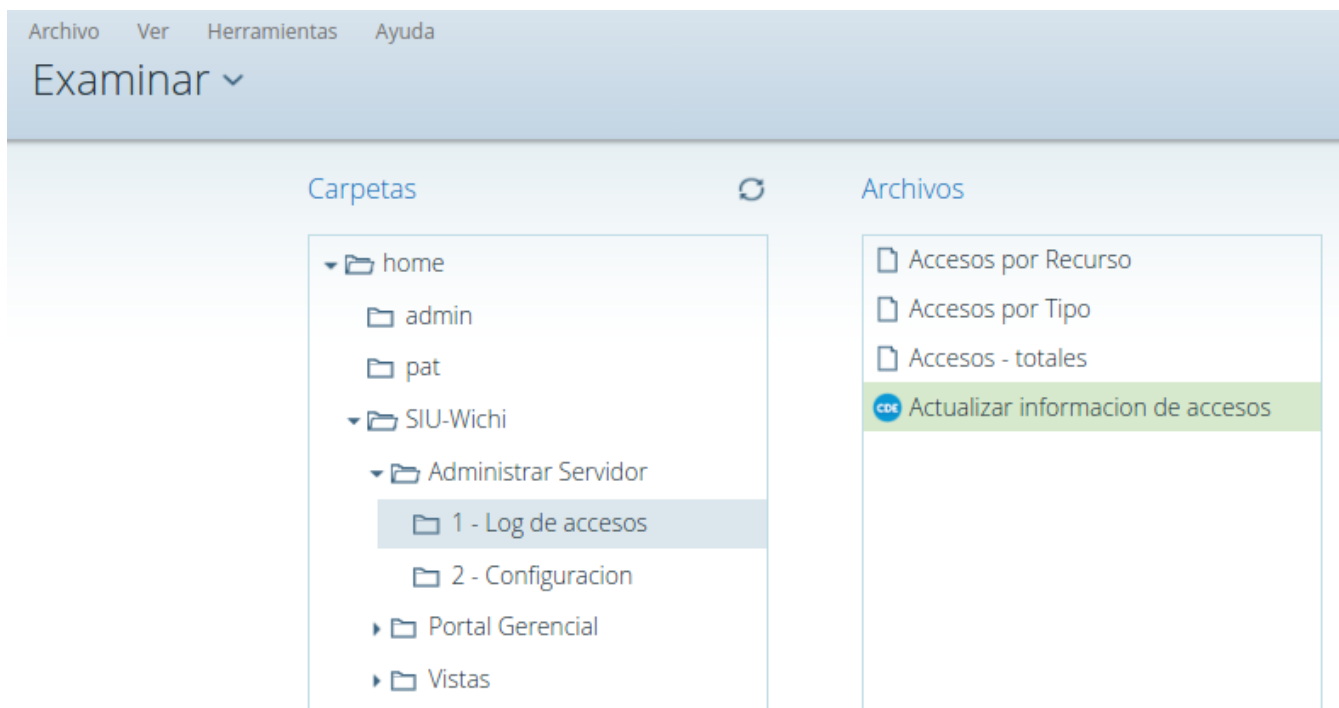
```
<bean id="IAuditEntry"  
class="org.pentaho.platform.engine.services.audit.AuditSQLEntry" scope="singleton" />
```

Paso 2: Reiniciar Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`

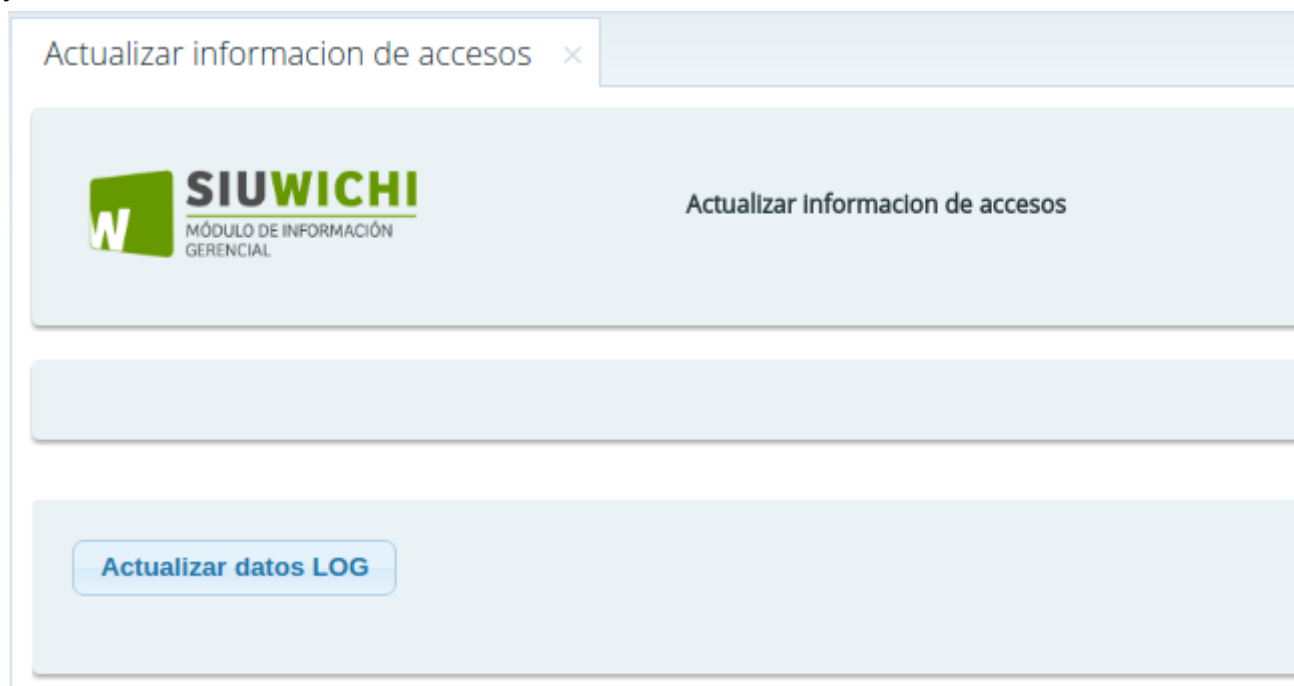
Paso 3: Cargar DW.

Para que los datos sean actualizados en los Reportes y en el Cubo “Log Accesos”, es necesario volcar los datos del log de acceso desde la base hibernate a la base siu_wichi. Esto se logra ejecutando el CDE “**Actualizar información de accesos**” que se encuentra en la carpeta “*Administrar Servidor → Log de accesos*” en la **Consola de Usuario de Pentaho** y luego apretando el botón “**Actualizar datos LOG**”.

Este último paso debe ser ejecutado cada vez que se desee actualizar los datos del log de accesos.



Ejecutar el botón → Actualizar datos LOG



3.5.2. Perfiles de usuario

La gestión de la seguridad mediante perfiles de usuario se explica detalladamente en [6.1. Gestión de la seguridad](#). Involucra tareas de configuración de pentaho, de configuración mediante consola de usuario y mediante consola de administración.

Dependiendo de su caso, quizás prefiera avanzar con la migración y carga de datos para ver al sistema funcionando, y luego ocuparse de la gestión de la seguridad. La instalación por defecto no restringe el acceso.

4. Migración de SIU-Wichi versión 6.7.0

4.1. Introducción

Tenga o no instalada una versión anterior de SIU-Wichi, siempre es posible instalar la versión 6.7.0 de cero, ignorando el resto de este capítulo.

Si usted tiene instalada una versión de SIU-Wichi anterior a la 6.6.1, es posible hacer migraciones sucesivas siguiendo la documentación de las versiones correspondientes. Esto es posible realizarlo aplicando sólo el paso de migración de cada versión.

Este capítulo detalla los pasos de migración desde SIU-Wichi 6.6.1 a 6.7.0. Esta migración requiere:

- Aplicar cambios en la base de datos
- Actualizar solución SIU-Wichi
- Actualizar esquema (SIU-Wichi.xml)
- Reemplazar ETLs de carga
- Cargar los datos del Portal de Transparencia

Luego de la migración podrá continuar cargando actualizaciones de datos con los procesos de ETL de la versión 6.7.0.

4.2. Actualizar versión

La migración de la base de datos siu_wichi se realiza con el siguiente procedimiento:

Paso 1: primer script de migración

Ejecutar el script `1.sql`, que se encuentra en el paquete del SIU-Wichi 6.7.0 en el directorio `/etl/migraciones_base_datos/661_670`.

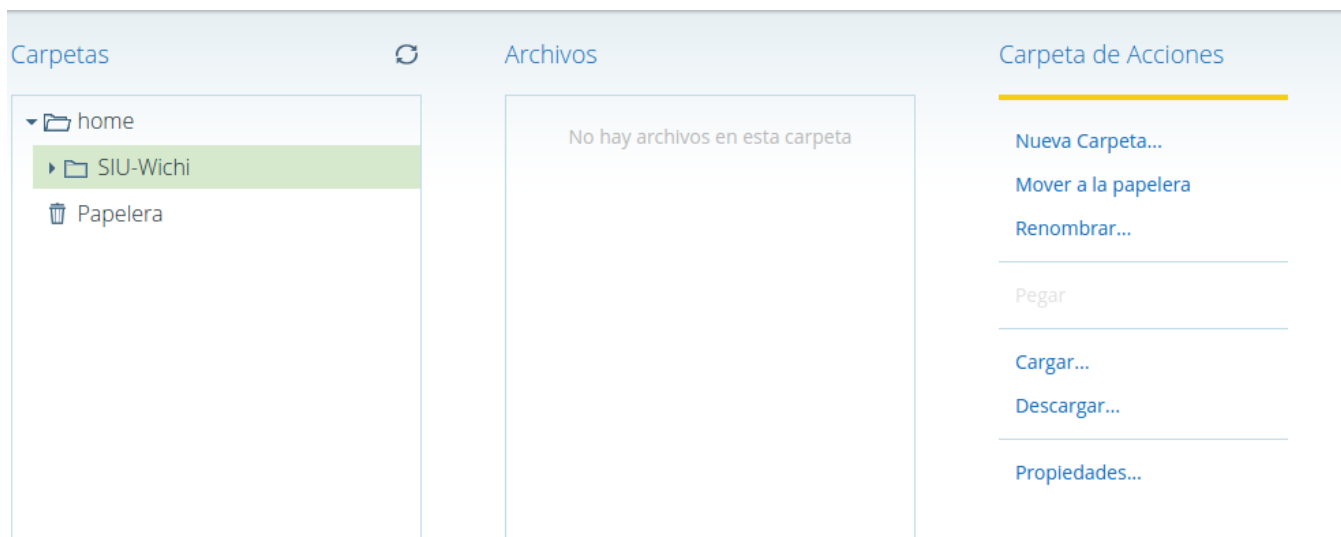
Paso 2: segundo script de migración

Ejecutar el script `2.sql`, que se encuentra en el mismo directorio.

4.3. Actualizar solución SIU-Wichi

Para actualizar la solución SIU-Wichi, debemos realizar los siguientes pasos:

Paso 1: Posicionarse en la carpeta SIU-Wichi.



Paso 2: A la derecha podemos visualizar las acciones que podemos realizar sobre una carpeta. Hacemos clic en la acción Mover a la papelera

Paso 3: realizar la sección [3.4. Carpeta SIU-Wichi](#) de la guía. Este punto explica como subir por medio de la opción “Cargar” el archivo SIU-Wichi.zip ubicado en la carpeta SIU-Wichi 6.7.0 de la versión.

4.4. Actualizar esquema (SIU-Wichi.xml)

Para actualizar el esquema SIU-Wichi, debemos realizar los pasos:

Paso 1 → [6.1.4.2.5 Actualización de Roles](#)

Paso 2 → [3.3.2. Actualización de esquema \(SIU-Wichi.xml\)](#) en ese orden.

4.5. Reemplazar ETLs de carga

Reemplazar los ETLs de carga de cada módulo por los ETLs de la última versión que se encuentran en la carpeta → /SIU-Wichi_6.7.0/etl/

4.6. Actualizar datos de Portal de Transparencia

Es necesario volver a importar los datos del portal de transparencia para poder visualizar los distintos Niveles de Título (Pregrado, Grado y Posgrado) en los tableros Académicos. Solución de tickets:

#858	Portal de Transparencia: agregar cantidad de tipo Titulo posgrado a los tableros de alumnos e ingresantes
------	---

#859	Portal de Transparencia: tableros académicos, agregar filtro Nivel (Pregrado, Grado, Posgrado, Todos)
------	---

Ver realizar actualización de datos ver: [9.3. Portal de transparencia - carga de datos](#)

5. Carga y actualización de datos

En este capítulo se explica cómo cargar datos en SIU-Wichi 6.7.0 para cada uno de los sistemas fuente. Recomendamos leer las siguientes consideraciones generales antes de pasar a las secciones específicas por sistema.

5.1. Consideraciones generales

La siguiente tabla muestra un resumen de las características de los procesos de carga de cada sistema fuente.

Sistema	Encoding	Actualización	Obtención de datos
Guarani 2	ISO-8859	controlada	Txt, múltiples unidades
Guarani 3	UTF8	controlada	Extracción directa
Pampa/Mapuche	(a confirmar)	controlada	Txt
Pilagá	-	controlada	Extracción directa
Rhun	UTF8	controlada	Txt
Araucano	UTF8	completa	Txt
Diaguita	-	completa	Extracción directa
Kolla	UTF8	completa	Extracción directa
Querandies	-	completa	Directa desde SPU
Sanavirón/Quilmes	-	completa	Extracción directa
Moodle	UTF8	completa	Extracción directa

5.1.1. Encoding

Hay diversos factores que determinan cómo se codifican los caracteres en los archivos extraídos. En la tabla anterior se muestran los valores esperados. Los procesos de carga de todos los sistemas trabajan con archivos en UTF8, por lo tanto puede ser necesario convertirlos.

Para determinar la codificación de los archivos puede utilizarse el comando *file* de linux. Por ejemplo:

```
$ file directorio_txt/*
```

Muestra la codificación para cada archivo. Cuando muestra “ASCII text” se puede interpretar como que el archivo no contiene caracteres latinos. Lo mismo cuando muestra “Fortran program”. Los casos de ISO-8859 (también llamado Latin1) deben ser convertidos a UTF8. Para eso puede utilizarse el script *convEncoding.sh* que se encuentra en el directorio */utilidades/general* del paquete SIU-Wichi.

5.1.2. Actualizaciones controladas

5.1.2.1. Pilagá, Mapuche, Rhun y Guaraní 3

Para la carga de datos de los sistemas *Pilagá*, *Mapuche* y *Rhun*, SIU-Wichi cuenta con un mecanismo de control y verificación de manera de asegurar que no se carguen datos incorrectamente y que correspondan al período que efectivamente queremos cargar. Además se permite actualizar/rectificar los datos correspondientes a un período en particular de una manera sencilla y segura.

Para esto se utilizan dos parámetros en el JOB principal que son **período** y **actualización**. El parámetro *período* corresponde al período de los datos que se está importando y su formato varía con cada sistema (ej: para Mapuche período=AAAAMM).

Con el mecanismo de actualizaciones controladas, el usuario puede ingresar en el parámetro **actualización**:

- **0** si el período a cargar es un período nuevo (no cargado previamente)
- **1** si los datos a cargar reemplazan los datos de un período ya cargado, en cuyo caso el proceso eliminará solo los datos correspondientes antes de realizar la carga.

Para la carga de datos de Guaraní 3 se utiliza solo el parámetro **período**, actualizando los datos para ese período o importándolos en caso de que no existan.

5.1.2.2. Araucano, Diaguita, Kolla, Querandíes, Sanavirón/Quilmes y Moodle

Para la carga de datos de los sistemas *Araucano*, *Diaguita*, *Kolla*, *Querandíes* y *Sanavirón/Quilmes* el proceso de ETL borra y vuelve a cargar el lote completo de datos que se descarga del sistema fuente. Por esto es que para este caso no se utilizan los parámetros *período* e *incremental*.

5.1.2.3. Guaraní 2

La carga de datos extraídos del sistema SIU-Guaraní 2 tiene algunas particularidades que difieren al resto de los procesos de carga. El proceso detecta automáticamente qué unidades académicas se están cargando. Los datos existentes de las unidades académicas correspondientes son eliminados y reemplazados. Por eso los parámetros explicados en 5.1.2.1. Pilagá, Mapuche, Rhun y Guaraní 3 no aplica a los datos Guaraní 2.

5.1.3. Caches de datos

Al finalizar cualquier proceso de carga, se deberá refrescar la cache de Mondrian, la de Saiku y la de CDA, para que los cambios tengan efecto.

5.1.3.1. Cache de Mondrian

La cache de Mondrian se refresca mediante la Consola de Usuario (*Herramientas -> Actualizar ->*

Cache del esquema Mondrian). Una vez realizado este paso podrá ver los nuevos datos.

5.1.3.2. Cache de Saiku

La cache de Saiku se refresca desde el modo de edición de una vista. Se puede lograr creando una nueva vista con la opción Archivo / Nuevo / Saiku Analytics, luego haciendo clic en , y cerrando la vista.

5.1.3.3. Cache de CDA

La cache de CDA se refresca mediante la Consola de Usuario (*Herramientas -> Actualizar -> CDA Cache*). Una vez realizado este paso podrá ver los nuevos datos.

5.1.4. Nivel de log

En las secciones siguientes se mostrarán líneas de comando de cómo usar *kitchen* para invocar los procesos de carga. Además de los parámetros indicados, considere los siguientes parámetros generales, especialmente para detectar y comunicar problemas:

-log=archivo: envía la salida al archivo especificado.

-level=Debug: establece el nivel de detalle del log en Debug.

5.2 Configuración de instalaciones

5.2.1. Introducción

Le llamamos instalación a cada instancia o entorno operativo de cada uno de los sistemas del SIU de los que SIU-Wichi puede incorporar datos. Por ejemplo, el listado de instalaciones para una institución (universidad) puede ser la siguiente:

- SIU-Mapuche
- SIU-Pilagá
- SIU-Guaraní, Fac. Económicas
- SIU-Guaraní, Fac. Exactas-Grado
- SIU-Guaraní, Fac. Exactas-Postgrado
- SIU-Araucano
- SIU-Querandies (SPU)
- SIU-Sanavirón/Quilmes
- Moodle

Esta concepción permite a SIU-Wichi manejar datos de distintas instalaciones ordenadamente.

Una ventaja adicional es que los procesos de carga de datos que obtienen la información conectándose directamente a la base fuente no requieran los parámetros de conexión cada vez que se ejecutan. Por otro lado se requiere mantenimiento de la configuración de instalaciones.

5.2.1.1. Número de instalación

Los procesos de carga de datos solicitan entre otros parámetros el número de instalación. El número de instalación puede ser consultado en el tablero de instalaciones (explicado más adelante en este capítulo).

Es un número compuesto con el formato *id-código*. Ambas partes se definen al momento de crear la instalación, pero el tablero solo permite al usuario modificar el código, de manera que pueda elegir un nombre fácil de recordar. El id asegura que los números de instalación no se repitan.

5.2.1.2. Cambios en la instalación

La información de las instalaciones debe ser mantenida. Ante algunos cambios en los servidores de los sistemas fuente puede darse el caso en que deba decidir si crear una instalación nueva o modificar los parámetros de una instalación existente. La decisión debe tomarse teniendo en cuenta el uso de los datos ya cargados a SIU-Wichi.

Es *seguro* crear una instalación nueva en los casos en que los datos ya cargados de la instalación se eliminarán manualmente o seguirán existiendo dentro de SIU-Wichi en forma independiente de los datos que tendrá la nueva instalación.

Sin embargo ante cambios puramente técnicos de servidor, puerto, etc. o ante migraciones de versión del sistema fuente, los datos posteriores deberían reemplazar a los datos anteriores, es decir, no existirán de manera independiente de aquellos, y por lo tanto, se tratará de la misma instalación. En estos casos no se debe crear una instalación nueva sino modificar la existente.

5.2.2. Tablero de Instalaciones

Dicho tablero brinda la posibilidad de crear y modificar instalaciones, se llama “0 - Configurar Instalaciones” y se encuentra en la carpeta “2 – Configuración” dentro de “Administrar Servidor”.

Para crear una nueva instalación basta con presionar el botón “Nueva Instalación” completar los datos correspondientes y luego presionar el botón “Guardar”.

Para completar los datos se deberá elegir un sistema fuente y dependiendo del sistema, se mostrarán los campos a completar. Esto se debe a que ciertos sistemas (Pilagá, Diaguíta, Kolla, Guaraní 3 y Moodle), necesitan los parámetros de conexión a la base de datos del sistema fuente para hacer la extracción de datos. En el caso de Moodle, además, se requieren otros tres parámetros que detallan más la conexión. Por otro lado el Querandíes requiere información que identifique a la Universidad en la base de datos de la Secretaría de Políticas Universitarias. A continuación se adjuntan imágenes que reflejan la diferencia en los datos a completar:

0 - Configurar Instalaciones x

Fuente	Descripción	Numero Instalación	Conexión
SIU-Araucano	SIU-Araucano	8-261	No requiere parámetros
SIU-Diaguita	SIU-Diaguita	4-866	192.168.132.228:5432/diaguita Usuario: diaguita_penta
SIU-Pilaga	uytrr	19-66666	localhost:5432/basepilaga Usuario: postgres
VALOR ERRONEO	VALOR ERRONEO	-1-VALOR ERRONEO	No requiere parámetros
SIU-Guarani	Instalacion de todas las facultades	12-001	No requiere parámetros
SIU-Guarani	SIU-Guarani - Facultad De Ciencias Veterinarias (unidad_id=99)	10-776	No requiere parámetros
SIU-Mapuche	SIU-Mapuche	5-917	No requiere parámetros
SIU-Guarani3	g3unm	14-g3unm	192.168.132.113:5432/mm_unm_g3 Usuario: postgres
RHUN-SPU	RHUN-SPU	9-246	No requiere parámetros
SIU-Diaguita	Nueva conexion de diaguita	20-3131	10.12.13.14:5432/basediaguita Usuario: sabrina

Showing 1 to 10 of 18 entries

Previous 1 2 Next

Nueva Instalación

Luego de presionar “Nueva Instalación”:

RHUN-SPU	RHUN-SPU	9-246	No re
SIU-Diaguita	Nueva conexion de diaguita	20-3131	10.12

Showing 1 to 10 of 18 entries

Cancelar

Sistema Fuente: 1-SIU-Guarani ▼

Cod de Instalación:

Descripción:

Guardar

RHUN-SPU	RHUN-SPU	9-246	No r
SIU-Diaguia	Nueva conexion de diaguia	20-3131	10.1

Showing 1 to 10 of 18 entries

Cancelar

Sistema Fuente: 3-SIU-Pilaga ▼

Cod de Instalación:

Descripción:

Host:

Puerto:

Base de Datos:

Nombre de Usuario:

Clave:

Guardar

SIU-Kolla	kolla4	13-55	192.1 Usuar
-----------	--------	-------	----------------

Showing 1 to 10 of 17 entries

Cancelar

Sistema Fuente: 5-SIU-Querandies ▼

Cod de Instalación:

Descripción:

Codigo SAF:

Nombre de Usuario:

Clave:

Guardar

Cancelar

Sistema Fuente: 13-Moodle ▼

Cod de Instalación:

Descripción:

Host:

Puerto:

Base de Datos:

Nombre de Usuario:

Clave:

Motor(Postgres o MySQL):

Nro Plataforma en G3:

Instalacion activa (S o N):

Guardar

Para realizar una modificación es suficiente con clickear sobre la fila de la instalación que se quiere modificar y automáticamente se desplegarán los campos que se puedan modificar de la misma. Como se dijo anteriormente, los campos modificables dependen del sistema fuente de la instalación. A continuación se adjuntan dos imágenes mostrando la diferencia de los campos modificables:

SIU-Guarani3	g3unm	14-g3unm	192.168.132.113:5432/mm_unm_g3 Usuario: postgres
RHUN-SPU	RHUN-SPU	9-246	No requiere parámetros
SIU-Diaguita	Nueva conexion de diaguita	20-3131	10.12.13.14:5432/basediaguita Usuario: sabrina

Showing 1 to 10 of 18 entries Previous **1** 2 N

Nueva Instalación

Cancelar

Cod de Instalación:

Host:

Puerto:

Base de Datos:

Nombre de Usuario:

Clave:

Modificar

SIU-Guarani3	g3unm	14-g3unm	192.168.132.113:5432/mm_unm_g3 Usuario: postgres
RHUN-SPU	RHUN-SPU	9-246	No requiere parámetros
SIU-Diaguita	Nueva conexion de diaguita	20-3131	10.12.13.14:5432/basediaguita Usuario: sabrina

Showing 1 to 10 of 18 entries Previous **1** 2

Nueva Instalación

Cancelar

Cod de Instalación:

Modificar

Como se puede ver en las imágenes, para el sistema fuente RHUN-SPU en la columna Conexión figura que no requiere parámetros, por lo tanto solo se puede modificar el código de instalación que concatenado con el id de instalación forman el número de instalación utilizado para los procesos de carga.

También se puede dar el caso en el que se haya creado una instalación que requiera parámetros de conexión (Pilagá, Diaguita, Kolla, Guarani3 , Sanavirón/Quilmes o Moodle) y alguno de ellos no fue completado. En ese caso, en la columna Conexión en vez de mostrar los datos, se indica que los parámetros están incompletos.

Show 10 entries Search: 			
Fuente	Descripción	Numero Instalación	Conexión
SIU-Araucano	SIU-Araucano	8-261	No requiere parámetros
SIU-Diaguita	SIU-Diaguita	4-866	192.168.132.228:5432/diaguita Usuario: diaguita_penta
SIU-Diaguita	Nueva conexion de diaguita	20-3131	PARAMETROS INCOMPLETOS

En la imagen se puede apreciar que para la nueva conexión Diaguita al menos un parámetro está incompleto. Como se explicó anteriormente, clickeando sobre dicha fila se mostrarán los campos para completar los parámetros faltantes.

Nueva Instalación

Cancelar

Cod de Instalación:

Host:

Puerto:

Base de Datos:

Nombre de Usuario:

Clave:

3131

10.12.13.14

5432

basediaguita

.....

Modificar

Una vez completado todos los parámetros, en la columna Conexión se mostraran de la siguiente manera: Host:Puerto/BaseDeDatos Usuario. Por cuestiones de seguridad la clave no se muestra en la tabla y se muestra encriptada en los campos para realizar las modificaciones.

Cómo se adelantó, para el caso de Querandies, además del código de instalación se debe ingresar el [código SAF](#) de la Institución, **un usuario con acceso de consulta al sistema Querandies de la SPU para esa institución y su clave**. Normalmente estos datos los posee el personal que realiza la carga de datos de Infraestructura de la Universidad en el sistema SIU-Querandies. Tenga en cuenta también que el [código SAF](#) debe estar cargado en la tabla d_institución (mediante el tablero indicado en [6.1.4.2.1 - Actualiza datos de Institución](#))

Showing 1 to 4 of 4 entries

Previous 1 Next

Nueva Instalación

Cancelar

Cod de Instalación:

Código SAF:

Nombre de Usuario:

Clave:

2

805

consulta

.....

Modificar

Para el caso de Moodle, los parámetros extra son los últimos tres:

Sistema Fuente:

Cod de Instalación:

Descripción:

Host:

Puerto:

Base de Datos:

Nombre de Usuario:

Clave:

Motor(Postgres o MySQL):

Nro Plataforma en G3:

Instalacion activa (S o N):

13-Moodle

Guardar

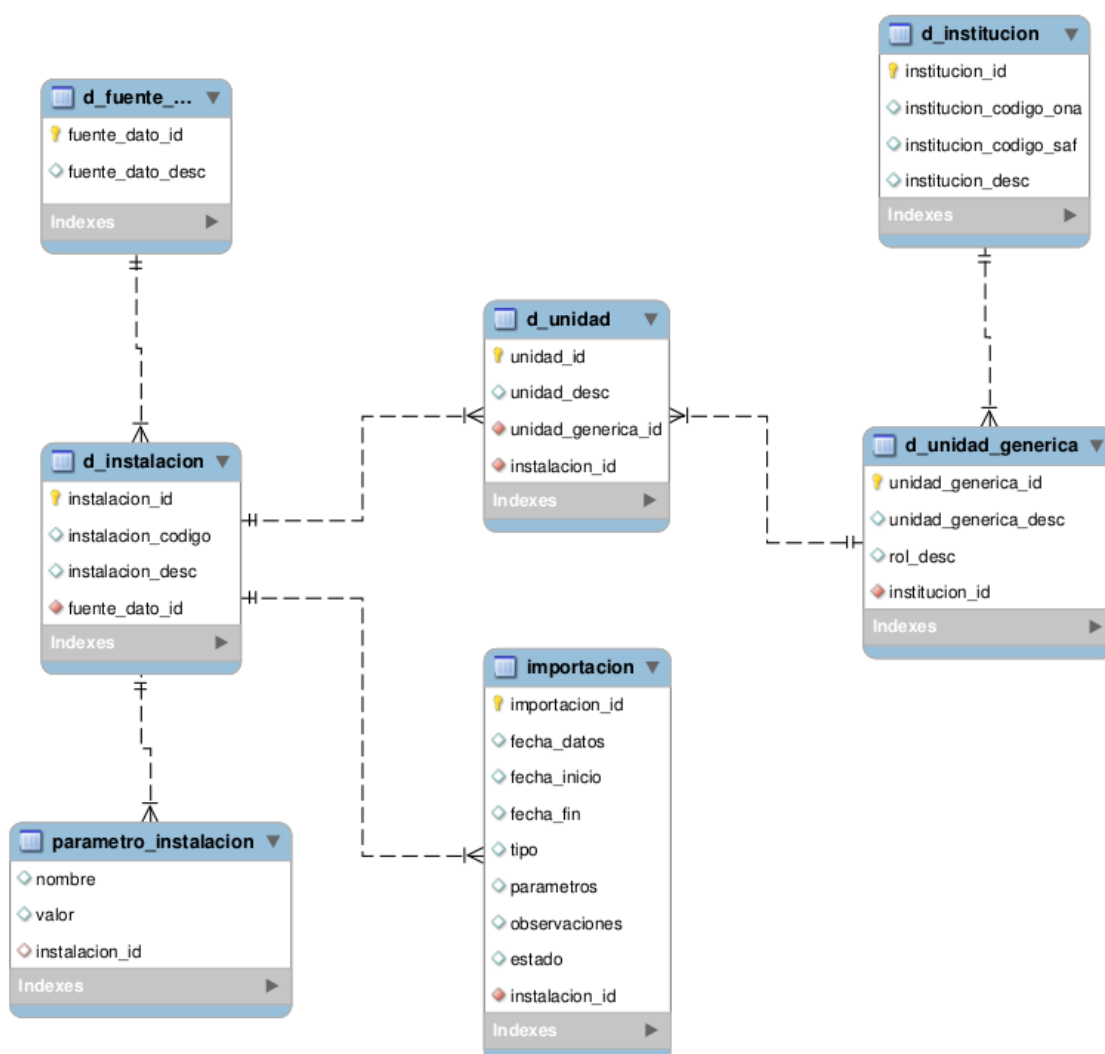
donde en Motor, se debe indicar si el motor de la base de datos moodle es Postgres o MySQL. En Nro Plataforma se debe indicar a qué número de plataforma de Guaraní 3 corresponde la base de

datos moodle que estamos configurando. Esto se puede ver en la tabla **negocio.int_pv_plataformas** de la base de Guaraní 3. En el caso de no tener vinculado Moodle con Guaraní 3, poner Nro Plataforma=1 y luego a medida que se agregan instalaciones de moodle, incrementar ese número (2,3 ...,etc).

En Instalacion activa se debe poner S, si quiero cargar esa instalación de moodle, o N para los casos en que por el momento no me interese cargarla.

5.2.3. Tablas involucradas

El siguiente diagrama muestra las tablas involucradas.



5.3. Datos de SIU-Guaraní 2

A partir de la versión SIU-Wichi 5.1.0, el ETL de carga, permite cargar datos de una, dos, o N unidades académicas al mismo tiempo y el proceso de carga de los diferentes cubos se realiza mediante la ejecución de un solo JOB de kettle. Sin embargo **los datos provenientes de instalaciones distintas deberán cargarse por separado** (más información en [5.2. Configuración de instalaciones](#)).

Para esto se necesita contar con los lotes de txt de de cada unidad académica y de cada uno de los cubos, Rendimiento Académico, Alumnos y Procedencia. En caso de que la cantidad de unidades académicas sea grande, podrá optarse por cargarlas en subconjuntos, a fin de acelerar el proceso.

Importante: La extracción de los datos desde el sistema SIU-Guaraní se deberá hacer con toda la historia completa, o sea que a la hora de generar los datos poner como Año Académico el año de inicio de actividades, en el caso de los cubos de Procedencia y Alumnos, y para el de Rendimiento Académico se deberá ingresar el “Año Académico Desde” y “Año Académico Hasta” de manera de cubrir todo el rango de años disponibles.

Los pasos para cargar datos de SIU-Guaraní, son los siguientes.

Paso 1: En el módulo /submódulo/ Interfaz/ Guarani - Araucano:

Orden 2: Actualizar Alumnos y Egresados

Este paso es requisito previo para generar datos para el cubo 05 - Alumnos - Arau

Paso 2: Generar los txt desde el sistema SIU-Guarani por el módulo Interfaz/ Guarani Datawarehouse, opciones:

- Genera datos temporales DW
- Genera archivos cubo 02v2 - Rend. Acad.
- Genera archivos cubo 03 - Procedencia
- Genera archivos cubo 05 - Alumnos - Arau

Paso 3: Estructura de directorios

Copiar los archivos TXT extraídos de SIU-Guaraní correspondientes a el o los cubos, en los directorios correspondientes. Para evitar trabajo innecesario del servidor, elimine de la estructura los datos de importaciones anteriores. No incluya datos provenientes de instalaciones distintas (más información en [5.2. Configuración de instalaciones](#)).

```

directorio_txt/
  unidad_academica_1/
    rendimiento/ → txt's cubo rendimiento académico
    procedencia/ → txt's cubo procedencia
    alumnos/ → txt's cubo alumnos
  ...
  unidad_academica_n/
    rendimiento/
    procedencia/
    alumnos/
  
```

Ejemplo:

```

facultad_ingenieria/
  rendimiento/
  procedencia/
  alumnos/
facultad_arquitectura/
  
```

```
rendimiento/
procedencia/
alumnos/
```

Paso 4: Codificación.

Si el sistema SIU-Guaraní genera los TXT con codificación distinta a UTF-8, hay que convertirlos antes de la importación. Por ejemplo, si SIU-Guaraní corre sobre un servidor Windows, probablemente genere los archivos con codificación ISO-8859-1 (ver [5.1.1. Encoding](#)). Si es necesario convertir los archivos TXT de ISO-8859-1 a UTF-8, hacer lo siguiente:

- a- Crear un directorio vacío para cada cubo, donde se copiarán los TXT convertidos a UTF-8.
- b- Para cada cubo, ejecutar el script que se encuentra en el directorio **utilidades** del paquete SIU_Wichi:

```
$ <path_utilidades>/general/convEncoding.sh <directorio_iso_8859_1>
<directorio_destino>
```

El script mostrará la codificación en que se encuentran los archivos y pedirá confirmación.

Paso 5: ETL de carga.

Ejecutar el job de carga unificado para los datos de Guaraní. Ejecutar el siguiente comando luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
-file="<path_etl>/carga_de_datos/guarani/principal_carga_guarani.kjb" \
-param="base_usuario=postgres" \
-param="base_clave=xxxxxxx" \
-param="base_nombre=siu_wichi" \
-param="base_host=localhost" \
-param="base_puerto=5432" \
-param="rutatxt=/home/wichi/guarani" \
-param="instalacion=1-583"
```

El parámetro *instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id` “-” `instalacion_codigo`.

Puede utilizar el script de carga [8.2.4. Script para facilitar carga de TXT \[Aporte UNGS\]](#)

5.4. Datos de SIU-Guaraní 3

A partir de la versión SIU-Wichi 5.6.0, el ETL de carga, permite importar los datos de Guaraní 3 (version 3.11.0 o posterior). A partir de la versión 6.3.0 los datos se cargaran en un esquema distinto al de los datos de Guaraní 2. Por lo que se recomienda borrar los datos existentes de las instalaciones provenientes de G3 del esquema guaraní, utilizando el script `3.sql` que se encuentra en el paquete del

SIU-Wichi 6.3.0 en el directorio `/etl/migraciones_base_datos/620_630` y volver a importarlos utilizando los etl del paquete SIU-Wichi 6.7.0. En caso de **datos provenientes de instalaciones distintas deberán cargarse por separado** (más información en [5.2. Configuración de instalaciones](#)).

El proceso de carga se encarga de conectarse a la base de datos de Guaraní, realizar la extracción y luego la importación de datos a Wichi. Para el caso que se necesite realizar el proceso por separado, en la siguiente sección se explica cómo hacerlo ([8.2.1. Separación del proceso de extracción e importación](#)).

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso.

Parámetro	Descripción
<code>base_wichi_host</code>	IP o nombre del host donde se encuentra la base siu_wichi
<code>base_wichi_puerto</code>	puerto al en el que escucha el servidor de postgres de wichi
<code>base_wichi_nombre</code>	nombre de la base de datos de wichi
<code>base_wichi_usuario</code>	usuario que tiene acceso al servidor de postgres de wichi
<code>base_wichi_clave</code>	clave del usuario de la base siu_wichi
<code>carga_comentario</code>	(Opcional)
<code>carga_directorioTemporal</code>	directorio temporal vacío para armado del paquete
<code>carga_instalacion</code>	Número de la instalación. Para que el sistema extraiga datos de distintas bases de datos Guaraní, identifica la instalación.
<code>carga_moodle</code>	SI para el caso que tenga una base Moodle configurada o NO.
<code>carga_periodo</code>	Periodo que se quiere cargar(AAAA), o 0 para cargar todos los periodos.
<code>paquete_directorio</code>	(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de "-"
<code>paquete_encryptacion</code>	(Opcional) tipo de encriptación: publica ninguna
<code>paquete_passPgp</code>	(Opcional) (Solo si se usa encriptación)
<code>paquete_rutaGpg</code>	(Solo si se usa encriptación) ruta al ejecutable de gpg
<code>paquete_userIdServidorCentral</code>	(Solo si se usa encriptación) user ID de gpg del servidor central Wichi
<code>paquete_wildcardPgp</code>	(Solo si se usa encriptación) Generalmente se puede dejar vacío
<code>performance_commit_size</code>	(Opcional) cada cuantos insert se hace commit

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro `carga_directorioTemporal`.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL `carga_de_datos/guarani3/principal_carga_guarani.kjb`, como se muestra a

continuación luego de reemplazar los valores en negrita.

```
$ $ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/guarani3/principal_carga_guarani.kjb" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_puerto=5432" \
  -param="base_wichi_usuario=postgres" \
  -param="carga_comentario=ejemplo" \
  -param="carga_moodle=NO" \
  -param="carga_directorioTemporal=/home/wichi/guarani3" \
  -param="carga_instalacion=5-631" \
  -param="carga_periodo=0"
```

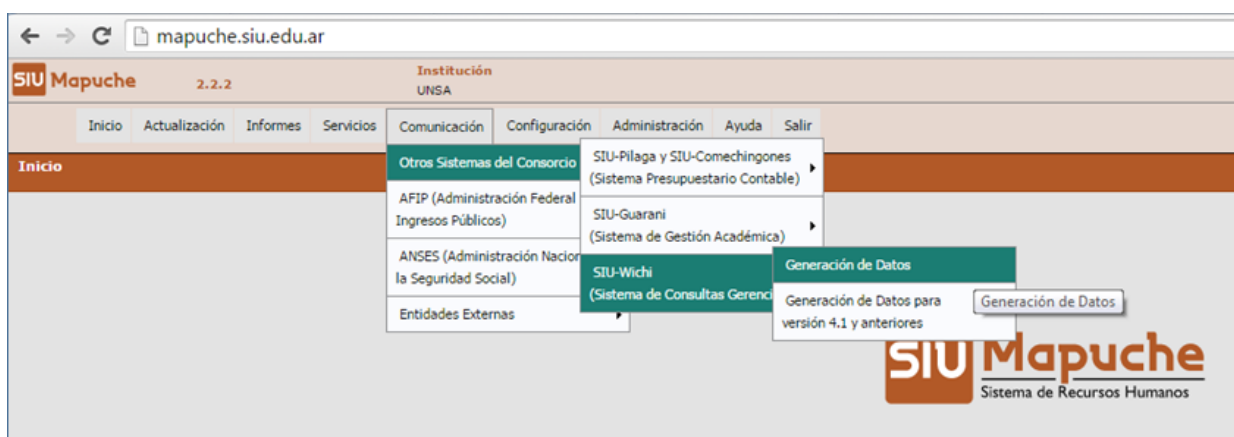
El parámetro *carga_instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla *d_instalaciones* concatenando los campos *instalacion_id* “-” *instalacion_codigo*.

5.5. Datos de SIU-Mapuche

Los pasos para cargar datos de SIU-Pampa/Mapuche, son los siguientes.

Paso 1: Generar los txt desde el sistema SIU-Mapuche, menú Comunicación \ Otros sistemas del Consorcio SIU \ SIU-Wichi \ Generación de Datos

Importante: esto se debe hacer cada mes antes del cierre mensual.



Paso 2: Copia:

Copiar los archivos TXT extraídos de SIU-Mapuche un directorio temporal.

Paso 3: Codificación.

Si el sistema SIU-Mapuche genera los TXT con codificación distinta a UTF-8, hay que convertirlos antes de la importación. Por ejemplo, si SIU-Mapuche corre sobre un servidor Windows, probablemente genere los archivos con codificación ISO-8859-1 (también conocida como LATIN-1). Pero los factores que determinan la codificación de los TXT son diversos.

Si es necesario convertir los archivos TXT de ISO-8859-1 a UTF-8, hacer lo siguiente:

- a- Crear un directorio vacío, donde se copiarán los TXT convertidos a UTF-8.
- b- Ejecutar el script que se encuentra en el directorio **utilidades** del paquete SIU_Wichi_6.7.0.zip:

```
$ <path_utilidades>/general/convEncoding.sh <directorio_iso_8859_1>
<directorio_destino>
```

El script mostrará la codificación en que se encuentran los archivos y pedirá confirmación.

Paso 4: ETL de carga.

Ejecutar el job de carga de datos de Mapuche con el siguiente comando luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
-file="<path_etl>/carga_de_datos/mapuche/importacion_mapuche.kjb" \
-param="base_usuario=postgres" \
-param="base_clave=xxxxxxx" \
-param="base_nombre=siu_wichi" \
-param="base_host=localhost" \
-param="base_puerto=5432" \
-param="rutatxt=/home/wichi/mapuche/txt_utf8" \
-param="actualizacion=0" \
-param="periodo=201204" \
-param="instalacion=2-375" \
-param="observaciones=ejemplo"
```

El **período** se ingresa con el formato AAAAMM (ej: 201204) y corresponde al período de los datos que se está importando.

Con el mecanismo de actualizaciones controladas, el usuario puede ingresar en el parámetro **actualización**:

- **0** si el período a cargar es un período nuevo (no cargado previamente)
- **1** si los datos a cargar reemplazan los datos de un período ya cargado, en cuyo caso el proceso eliminará solo los datos correspondientes antes de realizar la carga.

El parámetro *instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id` “-” `instalacion_codigo`. Ver [5.2. Configuración de instalaciones](#)

Puede utilizar el script de carga [8.2.4. Script para facilitar carga de TXT \[Aporte UNGS\]](#)

5.6. Datos de SIU-Pilagá

Se implementó un proceso que se conecta a la base de datos de SIU-Pilagá y realiza la extracción de datos y posterior importación a SIU-Wichi.

Es importante aclarar que para cargar servicios públicos de los años 2014, 2015, 2016 y 2017 se debe actualizar la base pilaga a la última versión, esto se hace instalando la versión de histórico de ejercicio → http://documentacion.siu.edu.ar/wiki/SIU-Pilaga/version3.3.4/historicos_ejercicios.

Funciona correctamente hasta el año 2014, no es posible cargar datos de servicios públicos anteriores a ese año.

Puede darse el caso de que algunas universidades necesiten dividir el proceso (en [8.2.1. Separación del proceso de extracción e importación](#) se explica como hacerlo), pero estimamos que no es el caso general.

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso unificado.

Parámetro	Descripción
<i>actualizacion</i>	• 0 si el año (ejercicio) a cargar es nuevo (no estaba cargado previamente) • 1 si los datos a cargar reemplazan los datos de un año (ejercicio) ya cargado, en cuyo caso el proceso eliminará solo los datos correspondientes antes de realizar la carga. (ver 5.1.2. Actualizaciones controladas)
<i>base_clave</i>	Clave del usuario de la base siu_wichi
<i>base_host</i>	IP o nombre del host de la base siu_wichi
<i>base_nombre</i>	Nombre de la base de datos de siu_wichi
<i>base_puerto</i>	Puerto del servidor de la base siu_wichi
<i>base_usuario</i>	Usuario de la base siu_wichi
<i>comentario</i>	(Opcional) Comentario a cargar en la tabla de importaciones
<i>commit_size</i>	(Opcional) Cada cuantos insert se hace commit en la base
<i>directorioDestino</i>	Directorio donde se va a crear el paquete de datos. Luego de ejecutar el proceso, el paquete y/o el directorio se podrá eliminar.
<i>directorioTemporal</i>	Directorio temporal vacío para armado del paquete. Luego de ejecutar el proceso, el directorio se podrá eliminar.
<i>directorioTxt</i>	Directorio vacío, donde se van a crear archivos txt extraídos de Pilagá. Luego de ejecutar el proceso, el directorio se podrá eliminar.
<i>encriptacion</i>	(Opcional) Tipo de encriptación: <i>publica</i> <i>ninguna</i> . Para el caso general no es pertinente la encriptación, por lo tanto el valor por defecto es <i>ninguna</i> .
<i>instalacion</i>	Número de la instalación. Para que el sistema extraiga datos de distintas bases de datos Pilagá, identifica la instalación.
<i>passPgp</i>	(Opcional, solo para encriptación pública)

<i>periodo</i>	Período (año) al que corresponden los datos importados ej: 2012.
<i>pilaga_base_host</i>	IP o nombre del host donde se encuentra la base de datos de Pilagá
<i>pilaga_base_nombre</i>	Nombre de la base de datos de Pilaga
<i>pilaga_base_pass</i>	Contraseña de la base de datos de Pilaga
<i>pilaga_base_puerto</i>	Puerto de la base de datos de Pilaga
<i>pilaga_base_usuario</i>	Usuario de la base de datos de Pilaga
<i>rutaGpg</i>	(Opcional, solo para encriptación pública) Ruta al ejecutable de GPG
<i>userIdServidorCentral</i>	(Opcional, solo para encriptación pública) User ID de GPG del servidor central Wichi
<i>wildcardPgp</i>	(Opcional, solo para encriptación pública) Generalmente no es necesario

Paso 1: Crear directorios.

En caso de que no los tenga, crear dos directorios vacíos de uso temporal, para usar en los parámetros `directorioTemporal` y `directorioTxt`.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL `carga_de_datos/pilaga/principal_carga_pilaga.kjb`, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
-file="<path_etl>/carga_de_datos/pilaga/principal_carga_pilaga.kjb" \
-param="periodo=AAAA" \
-param="actualizacion=0" \
-param="base_usuario=postgres" \
-param="base_clave=xxxxxxx" \
-param="base_host=localhost" \
-param="base_nombre=siu_wichi" \
-param="base_puerto=5432" \
-param="comentario=ejemplo" \
-param="directorioDestino=/home/wichi/pilaga" \
-param="directorioTemporal=/home/wichi/pilaga/temp" \
-param="directorioTxt=/home/wichi/pilaga/txt" \
-param="instalacion=3-829" \
```

El parámetro *instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id` “-” `instalacion_codigo`. Ver [5.2. Configuración de instalaciones](#)

Se sugiere utilizar un único número de instalación, modificando el nombre de la base de datos de cada ejercicio (año), el host, puerto, usuario y clave según sea necesario.

Por ejemplo para la carga de la base pilaga_2019

Cod de Instalación:	952
Host:	localhost
Puerto:	5432
Base de Datos:	pilaga_2019
Nombre de Usuario:	coco
Clave:	*****

Luego utilizamos el mismo número de instalación 952, apuntando a la base pilaga_2020

Cod de Instalación:	952
Host:	localhost
Puerto:	5432
Base de Datos:	pilaga_2020
Nombre de Usuario:	coco
Clave:	*****

Paso 3: Borrar directorios.

Opcionalmente borrar los directorios creados en el paso 1.

5.7. Datos de SIU-RHUN

SIU-Wichi permite actualizaciones controladas de los datos de RHUN. El *período* se ingresa con el formato AAAAMM (ej: 201502) y corresponde al período de los datos que se está importando, mientras que **actualización** puede tomar los valores 0 = “No es una actualización” y 1 = “Sí es una actualización”.

Paso 1: Descargar los archivos txt desde <http://spupedidos.siu.edu.ar/>

Menú → Importación

Opción → Estado de importación

Estado del Proceso de Importación

Institución: Universidad Nacional d [dropdown]
 Conjunto de Información: Información de Personal [dropdown]
 Año Período: 2015 [dropdown]
 Período: -- Todos -- [dropdown]
 Estado: Importado OK [dropdown]
 Desde Fecha: [calendar]
 Hasta Fecha: [calendar]
 Tipo: -- SELECCIONE -- [dropdown]

[Filtrar] [Limpiar]

Az

Cod RRHH Importación	Institución	Conjunto de Información	Período	Tipo	Fecha y Hora de Envío	Estado	Observaciones					
16468	Universidad Nacional	Información de Personal	Enero 2015	Original	05/02/2015 11:49:40	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
16844	Nacional	Información de Personal	Febrero 2015	Original	10/03/2015 9:17:28	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
17257	Nacional	Información de Personal	Marzo 2015	Original	06/04/2015 13:31:03	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
17781	Nacional	Información de Personal	Abril 2015	Original	30/04/2015 10:19:54	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
17984	Nacional	Información de Personal	Mayo 2015	Original	01/06/2015 14:06:25	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
18280	Nacional	Información de Personal	Junio 2015	Original	02/07/2015 9:15:16	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
18506	Nacional	Información de Personal	Julio 2015	Original	05/08/2015 13:43:49	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
18663	Nacional	Información de Personal	Agosto 2015	Original	02/09/2015 12:57:51	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
18803	Nacional	Información de Personal	Septiembre 2015	Original	02/10/2015 10:05:53	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
18973	Nacional	Información de Personal	Octubre 2015	Original	04/11/2015 12:15:01	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
19091	Nacional	Información de Personal	Noviembre 2015	Original	02/12/2015 12:47:07	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]
19232	Nacional	Información de Personal	Diciembre 2015	Original	29/12/2015 11:33:09	Importado OK		[icon]	[icon]	[icon]	[icon]	[icon]

Paso 2: Copia

Copiar los archivos txt descargados de Rhun a un directorio accesible.

Paso 3: ETL de carga.

Ejecutar el job de carga de datos de Rhun con el siguiente comando luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/rhun/importacion_rhun.kjb" \
  -param="base_usuario=postgres" \
  -param="base_clave=xxxxxxx" \
  -param="base_nombre=siu_wichi" \
  -param="base_host=localhost" \
  -param="base_puerto=5432" \
  -param="rutatxt=/home/wichi/rhun" \
  -param="instalacion=4-261" \
  -param="observaciones=ejemplo" \
  -param="periodo=AAAAMM" \
  -param="actualizacion=0"
```

Con el mecanismo de actualizaciones controladas, el usuario puede ingresar en el parámetro **actualizacion**:

- **0** si el período a cargar es un período nuevo (no cargado previamente)
- **1** si los datos a cargar reemplazan los datos de un período ya cargado, en cuyo caso el proceso eliminará solo los datos correspondientes antes de realizar la carga.

El parámetro *instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla d_instalaciones concatenando los campos instalacion_id “-” instalacion_codigo. Ver [5.2. Configuración de instalaciones](#)

Puede utilizar el script de carga [8.2.4. Script para facilitar carga de TXT \[Aporte UNGS\]](#)

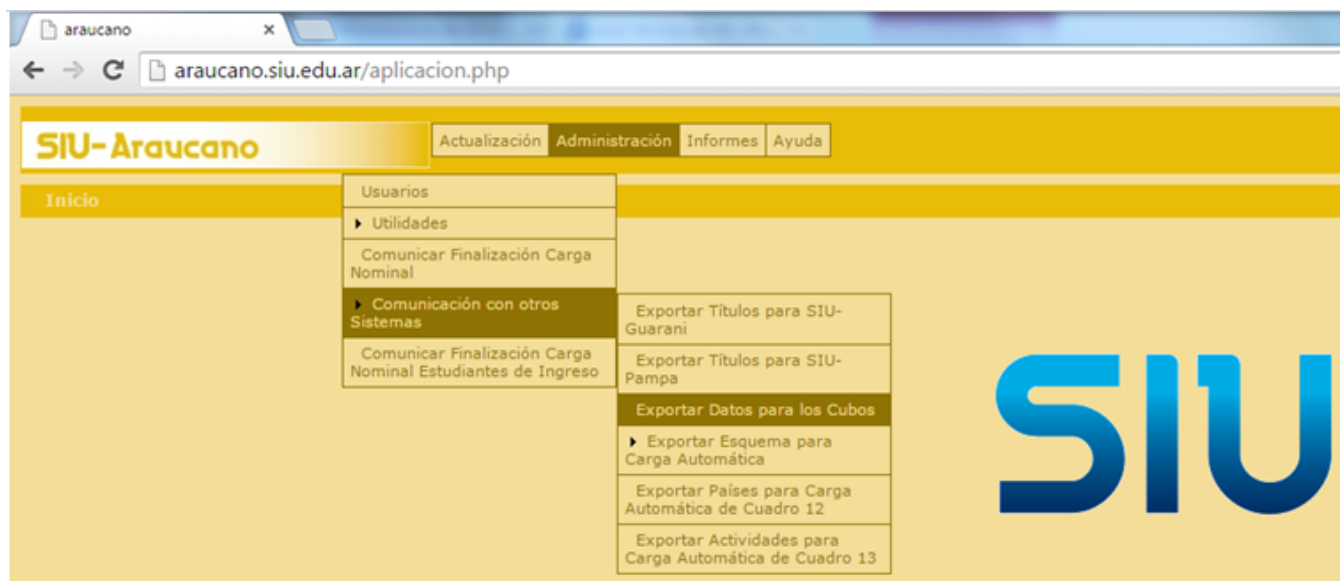
5.8. Datos de SIU-Araucano

Los pasos para cargar datos de SIU-Araucano, son los siguientes.

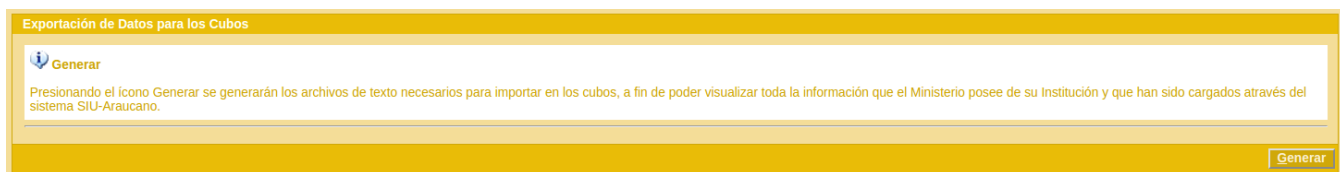
Paso 1: Acceder a <http://araucano.siu.edu.ar/> con usuario y clave correspondiente a la Universidad.

Paso 2: Generar archivos txt desde el sistema Araucano

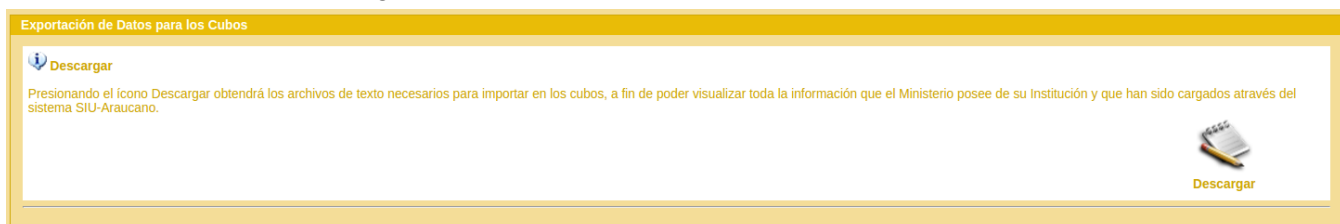
Generar datos desde Araucano, en menú Administración \ Comunicación con otros sistemas \ opción Exportar datos para cubos (son los mismos txt para SIU-Wichi y O3)



Seleccionar botón → Generar:



Seleccionar botón → Descargar:



Paso 3: Copia.

Copiar los archivos TXT descargados de SIU-Araucano correspondientes, en directorios accesibles.

Paso 4: ETL de carga.

Ejecutar el job de carga de datos de Araucano con el siguiente comando luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/araucano/importacion_araucano.kjb" \
  -param="base_usuario=postgres" \
  -param="base_clave=xxxxxxx" \
  -param="base_nombre=siu_wichi" \
  -param="base_host=localhost" \
  -param="base_puerto=5432" \
  -param="rutatxt=/home/wichi/araucano" \
  -param="instalacion=6-731"
```

El parámetro *instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id` “-” `instalacion_codigo`. Ver [5.2. Configuración de instalaciones](#)

5.9. Datos de SIU-Diaguíta

Se implementó un proceso que se conecta a la base de datos de SIU-Diaguíta y realiza la extracción de datos y posterior importación a SIU-Wichi.

Puede darse el caso de que algunas universidades necesitan dividir el proceso (en [8.2.1. Separación del proceso de extracción e importación](#) se explica como hacerlo), pero estimamos que no es el caso general.

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso.

Parámetro	Descripción
<code>base_wichi_host</code>	IP o nombre del host donde se encuentra la base <code>siu_wichi</code>
<code>base_wichi_puerto</code>	puerto al en el que escucha el servidor de postgres de wichi
<code>base_wichi_nombre</code>	nombre de la base de datos de wichi
<code>base_wichi_usuario</code>	usuario que tiene acceso al servidor de postgres de wichi
<code>base_wichi_clave</code>	clave del usuario de la base <code>siu_wichi</code>
<code>carga_actualizacion</code>	Deshabilitado en esta version (se ignora). Los valores posibles son 0/1/auto .
<code>carga_comentario</code>	(Opcional)
<code>carga_directorioTemporal</code>	directorio temporal vacío para armado del paquete

<i>carga_instalacion</i>	<i>Número de la instalación. Para que el sistema extraiga datos de distintas bases de datos Pilagá, identifica la instalación.</i>
<i>carga_periodo</i>	<i>Deshabilitado en esta version (se ignora).</i>
<i>paquete_directorio</i>	<i>(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de "-"</i>
<i>paquete_encryptacion</i>	<i>(Opcional) tipo de encriptación: publica ninguna</i>
<i>paquete_passPgp</i>	<i>(Opcional) (Solo si se usa encriptación)</i>
<i>paquete_rutaGpg</i>	<i>(Solo si se usa encriptación) ruta al ejecutable de gpg</i>
<i>paquete_userIdServidorCentral</i>	<i>(Solo si se usa encriptación) user ID de gpg del servidor central Wichi</i>
<i>paquete_wildcardPgp</i>	<i>(Solo si se usa encriptación) Generalmente se puede dejar vacío</i>
<i>performance_commit_size</i>	<i>(Opcional) cada cuantos insert se hace commit</i>

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro `carga_directorioTemporal`.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL `carga_de_datos/diaguaita/principal_carga_diaguaita.kjb`, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/diaguaita/principal_carga_diaguaita.kjb" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_puerto=5432" \
  -param="base_wichi_usuario=postgres" \
  -param="carga_comentario=ejemplo" \
  -param="carga_directorioTemporal=/home/wichi/diaguaita" \
  -param="carga_instalacion=5-631"
```

El parámetro `carga_instalacion` se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id` “-” `instalacion_codigo`. Ver [5.2. Configuración de instalaciones](#)

5.10. Datos de SIU-Kolla

Se implementó un proceso que se conecta a la base de datos de SIU-Kolla y realiza la extracción de datos y posterior importación a SIU-Wichi.

Puede darse el caso de que algunas universidades necesiten dividir el proceso (en [8.2.1. Separación del proceso de extracción e importación](#) se explica como hacerlo), pero estimamos que no es el caso general.

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso.

Parámetro	Descripción
<i>base_wichi_host</i>	<i>IP o nombre del host donde se encuentra la base siu_wichi</i>
<i>base_wichi_puerto</i>	<i>puerto al en el que escucha el servidor de postgres de wichi</i>
<i>base_wichi_nombre</i>	<i>nombre de la base de datos de wichi</i>
<i>base_wichi_usuario</i>	<i>usuario que tiene acceso al servidor de postgres de wichi</i>
<i>base_wichi_clave</i>	<i>clave del usuario de la base siu_wichi</i>
<i>carga_comentario</i>	<i>(Opcional)</i>
<i>carga_directorioTemporal</i>	<i>directorio temporal vacío para armado del paquete</i>
<i>carga_instalacion</i>	<i>id de la instalacion "-" código de la instalación</i>
<i>carga_periodo</i>	<i>Año al que corresponden los datos importados ej:2019. Cero significa todos.</i>
<i>paquete_directorio</i>	<i>(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de "-"</i>
<i>paquete_encryptacion</i>	<i>(Opcional) tipo de encriptación: publica ninguna</i>
<i>paquete_passPgp</i>	<i>(Opcional) (Solo si se usa encriptación)</i>
<i>paquete_rutaGpg</i>	<i>(Solo si se usa encriptación) ruta al ejecutable de gpg</i>
<i>paquete_userIdServidorCentral</i>	<i>(Solo si se usa encriptación) user ID de gpg del servidor central Wichi</i>
<i>paquete_wildcardPgp</i>	<i>(Solo si se usa encriptación) Generalmente se puede dejar vacío</i>
<i>performance_commit_size</i>	<i>(Opcional) cada cuantos insert se hace commit</i>

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro *carga_directorioTemporal*.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL *carga_de_datos/kolla/carga_kolla.kjb*, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/kolla/carga_kolla.kjb" \
  -param="base_wichi_usuario=postgres" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_puerto=5432" \
  -param="carga_directorioTemporal=/home/wichi/kolla" \
  -param="carga_instalacion=9-632" \
  -param="carga_periodo=0"
```

El parámetro *carga_instalacion* se obtiene del campo "Número de Instalación" del tablero

“Instalaciones” o se puede obtener de la tabla d_instalaciones concatenando los campos instalacion_id “-” instalacion_codigo. Configuran una instalación distinta para cada base de datos Kolla. Ver [5.2. Configuración de instalaciones](#)

Paso 3: Cargar el esquema en Pentaho

A diferencia de todas las otras fuentes de datos el SIU-Kolla tiene la característica que dependiendo de las preguntas que se hagan en las distintas encuestas son las dimensiones que figuran para clasificar los encuestados. Esto significa que para cada importación el proceso genera un archivo llamado EsquemaKolla_<numero de instalacion>-<codigo de instalacion>.xml en la carpeta /carga_de_datos/kolla/. Este archivo debe ser registrado como se indica en [3.3.1. Registro de esquemas](#)

5.11. Datos de SIU-Querandíes

Se implementó un proceso que se conecta a la base de datos de SIU-Querandíes alojada en servidores de la Secretaría de Políticas Universitarias y realiza:

1. La conexión a esa base.
2. El filtrado de los datos por [código SAF](#) (para identificar a la Institución).
3. La validación de permiso de acceso en base al usuario y clave ingresados.
4. La importación completa en la base de SIU-Wichi.

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso. Tenga en cuenta también que el [código SAF](#) debe estar cargado en la tabla d_institución (mediante el tablero indicado en [6.1.4.2.1 - Actualiza datos de Institución](#))

Parámetro	Descripción
base_wichi_host	IP o nombre del host donde se encuentra la base siu_wichi
base_wichi_puerto	puerto al en el que escucha el servidor de postgres de wichi
base_wichi_nombre	nombre de la base de datos de wichi
base_wichi_usuario	usuario que tiene acceso al servidor de postgres de wichi
base_wichi_clave	clave del usuario de la base siu_wichi
carga_comentario	(Opcional)
carga_directorioTemporal	directorio temporal vacío para armado del paquete
carga_instalacion	id de la instalacion “-” codigo de la instalación
paquete_directorio	(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de “-”
paquete_rutaGpg	ruta al ejecutable de gpg
performance_commit_size	(Opcional) cada cuantos insert se hace commit

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro carga_directorioTemporal.

Paso 2: Cargar clave de encriptado.

El archivo con todos los datos de Infraestructura de su Institución debe pasar desde el servidor de la SPU hasta el de la Institución, por esa razón el mismo viaja encriptado y por lo tanto hay que importar la clave utilizada a los fines de poder desencriptarlo.

Para importar la clave debe ejecutar en la interfaz de comandos:

`gpg --import QuerandiesWichi-priv.key` (el archivo QuerandiesWichi-priv.key está en la distribución, en la carpeta `etl/carga_de_datos/querandies`).

Si por alguna razón cambia el servidor desde el que ejecuta el Paso 3 deberá antes realizar este Paso 2 en el mismo para importar la clave de autenticación.

En algunos casos es posible que el proceso de importación le solicite la frase de autenticación, si es así la misma es "ParaElWichi620".

Paso 3: Ejecutar el paso de ETL.

Ejecutar el paso de ETL `carga_de_datos/querandies/principal_carga_querandies.kjb`, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/querandies/principal_carga_querandies.kjb" \
  -param="base_wichi_usuario=postgres" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_puerto=5432" \
  -param="carga_directorioTemporal=/home/wichi/querandies" \
  -param="carga_instalacion=x-yyyy"
```

El parámetro `carga_instalacion` se obtiene del campo "Número de Instalación" del tablero ubicado en carpeta Administrar Servidor / 2 - Configuración / "0 - Configurar Instalaciones" o se puede obtener de la tabla `d_instalaciones` concatenando los campos `instalacion_id (x)`-`instalacion_codigo (yyyy)`. Ver [5.2. Configuración de instalaciones](#)

5.12. Datos de SIU-Sanavirón/Quilmes

A partir de la versión SIU-Wichi 6.4.0, el ETL de carga, permite importar los datos de Sanavirón/Quilmes (versión 1.0.0).

El proceso de carga se encarga de conectarse a la base de datos de Sanavirón, realizar la extracción y luego la importación de datos a Wichi. Para el caso que se necesite realizar el proceso por separado, en la siguiente sección se explica cómo hacerlo ([8.2.1. Separación del proceso de extracción e importación](#)).

Antes de enumerar los pasos a seguir, se muestra una tabla de referencia de los parámetros que admite el proceso.

Parámetro	Descripción
<code>base_wichi_host</code>	IP o nombre del host donde se encuentra la base <code>siu_wichi</code>
<code>base_wichi_puerto</code>	puerto al en el que escucha el servidor de <code>postgres</code> de <code>wichi</code>

<i>base_wichi_nombre</i>	<i>nombre de la base de datos de wichi</i>
<i>base_wichi_usuario</i>	<i>usuario que tiene acceso al servidor de postgres de wichi</i>
<i>base_wichi_clave</i>	<i>clave del usuario de la base siu_wichi</i>
<i>carga_comentario</i>	<i>(Opcional)</i>
<i>carga_directorioTemporal</i>	<i>directorio temporal vacío para armado del paquete</i>
<i>carga_instalacion</i>	<i>Número de la instalación. Para que el sistema extraiga datos de la base de datos Sanavirón, identifica la instalación.</i>
<i>paquete_directorio</i>	<i>(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de "-"</i>
<i>paquete_encryptacion</i>	<i>(Opcional) tipo de encriptación: publica ninguna</i>
<i>paquete_passPgp</i>	<i>(Opcional) (Solo si se usa encriptación)</i>
<i>paquete_rutaGpg</i>	<i>(Solo si se usa encriptación) ruta al ejecutable de gpg</i>
<i>paquete_userIdServidorCentral</i>	<i>(Solo si se usa encriptación) user ID de gpg del servidor central Wichi</i>
<i>paquete_wildcardPgp</i>	<i>(Solo si se usa encriptación) Generalmente se puede dejar vacío</i>
<i>performance_commit_size</i>	<i>(Opcional) cada cuantos insert se hace commit</i>

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro *carga_directorioTemporal*.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL *carga_de_datos/sq/principal_carga_sq.kjb*, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ $ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/sq/principal_carga_sq.kjb" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_puerto=5432" \
  -param="base_wichi_usuario=postgres" \
  -param="carga_comentario=ejemplo" \
  -param="carga_directorioTemporal=/home/wichi/sq" \
  -param="carga_instalacion=5-631" \
```

El parámetro *carga_instalacion* se obtiene del campo “Número de Instalación” del tablero “Instalaciones” o se puede obtener de la tabla *d_instalaciones* concatenando los campos *instalacion_id* “-” *instalacion_codigo*.

Ver [5.2. Configuración de instalaciones](#)

5.13. Datos de Moodle

A partir de la versión SIU-Wichi 6.7.0, el ETL de carga, permite importar los datos de Moodle de versiones 3.4 o 3.5 (las versiones mas nuevas no han sido probadas).

Es necesario descargar el driver correspondiente para poder conectarse a la base de moodle (jar de Postgresql o MySQL) y luego moverlo dentro del directorio de Pentaho Data Integration (PDI):
`/PDI/lib/`.

Existen dos opciones de carga:

1. Moodle integrado con Guaraní 3
2. Solo Moodle

En ambos casos el proceso de carga se conecta a la/s base/s de datos de Moodle que hayan sido configuradas, realizar la extracción y luego la importación de datos a Wichi.

1. **Carga de Moodle integrado con Guaraní 3:** para este caso es necesario indicar a qué plataforma de Guaraní corresponde/n la/s base/s moodle a cargar (esto se puede ver en la tabla **negocio.int_pv_plataformas** de la base de Guaraní) y luego al correr la carga de datos de Guaraní 3, setear el parámetro **carga_moodle = SI**
2. **Carga solo Moodle:** para este caso, se deben configurar los parámetros que muestra la siguiente tabla

Parámetro	Descripción
<i>base_wichi_host</i>	<i>IP o nombre del host donde se encuentra la base siu_wichi</i>
<i>base_wichi_puerto</i>	<i>puerto al en el que escucha el servidor de postgres de wichi</i>
<i>base_wichi_nombre</i>	<i>nombre de la base de datos de wichi</i>
<i>base_wichi_usuario</i>	<i>usuario que tiene acceso al servidor de postgres de wichi</i>
<i>base_wichi_clave</i>	<i>clave del usuario de la base siu_wichi</i>
<i>carga_directorioTemporalMoodle</i>	<i>directorio temporal vacío para armado del paquete</i>
<i>paquete_directorio</i>	<i>(Opcional) si se especifica, al terminar deja una copia del paquete de datos wichi. Debe ser distinto de "-"</i>
<i>paquete_encryption</i>	<i>(Opcional) tipo de encriptación: publica ninguna</i>
<i>performance_commit_size</i>	<i>(Opcional) cada cuantos insert se hace commit</i>

Paso 1: Crear directorio temporal.

Asegurarse de tener disponible un directorio temporal vacío para usar con el parámetro **carga_directorioTemporalMoodle**.

Paso 2: Ejecutar el paso de ETL.

Ejecutar el paso de ETL `carga_de_datos/moodle/principal_carga_moodle.kjb`, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ $ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/moodle/principal_carga_moodle.kjb" \
  -param="base_wichi_clave=xxxxxxx" \
  -param="base_wichi_host=localhost" \
  -param="base_wichi_nombre=siu_wichi" \
  -param="base_wichi_puerto=5432" \
  -param="base_wichi_usuario=postgres" \
  -param="carga_directorioTemporalMoodle=/home/wichi/moodle" \
```

6. Administración del servidor

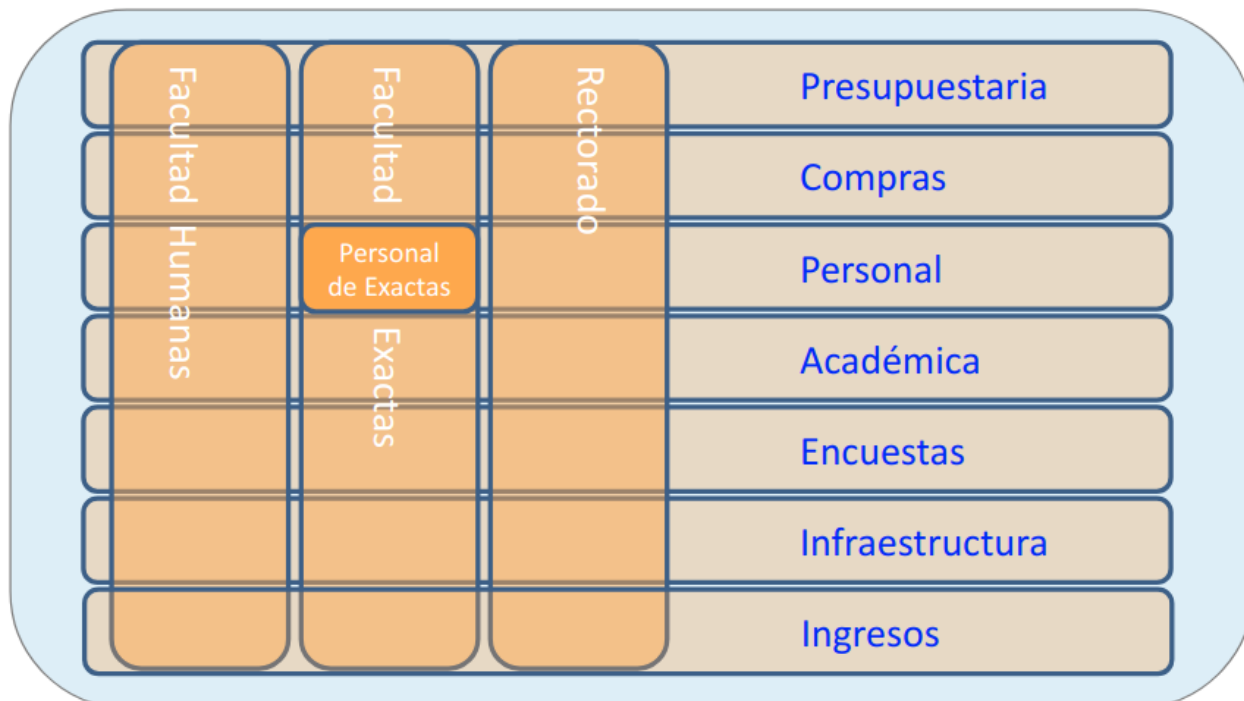
6.1. Gestión de la seguridad

6.1.1. Introducción

El SIU-Wichi se plantea como una base para el análisis de datos Institucional, de esta forma la información existente puede ser clasificada, por ejemplo, de la siguiente forma:

Roles y acceso segmentando la información

Base de datos Universitaria (SIU-Wichi)



Vemos aquí cortes horizontales que tienen que ver con el origen de la información:

Presupuestaria, correspondiente con el sistema fuente Pilagá.

Personal, correspondiente con los sistemas fuente Mapuche y RHUN.

Académica, correspondiente con los sistemas fuente Araucano y Guaraní.

Compras, correspondiente con el sistema fuente Diaguita.

Encuestas, correspondiente con el sistema fuente Kolla.

Infraestructura, correspondiente con el sistema fuente Querandíes.

Ingresos, corresponde con el sistema fuente Sanavirón/Quilmes.

Y así se seguirán sumando otras fuentes de gestión de la información con la entrega de futuras versiones de SIU-Wichi).

Por otro lado están graficados cortes verticales que indican la Unidad Presupuestaria o Académica a la que le corresponde la información, en este caso puede ser alguna Facultad, Rectorado, algún Colegio Universitario, Instituto de Investigación, Sede, etc. A estas Unidades les llamaremos de ahora en más **“Unidades de Análisis”**.

El recuadro contiene la información global institucional que es posible analizar con SIU-Wichi, permitiendo a la institución determinar quienes son los usuarios que accederán a dicha información, como por ejemplo Rector, el Consejo Superior y otras autoridades.

En el medio, se puede ver una zona de color naranja intenso que señala la porción de datos que puede acceder por ejemplo Recursos Humanos de una Facultad determinada(Exactas).

El objetivo de este documento es explicar cómo se hace para configurar distintos tipos de usuarios para que puedan acceder a la porción de datos que les corresponde.

6.1.2. Configuración del Pentaho

Es necesario verificar los siguientes parámetros para asegurarnos que el Pentaho está correctamente configurado y permitirá la gestión de usuarios, roles y permisos en forma segura:

6.1.2.1. Habilitar la seguridad sobre archivos

Se debe configurar los tipos de archivo sobre los cuales Pentaho administra la seguridad. Los dashboards (.wcdf, .cdfde) y las vistas de Saiku (.saiku) por defecto no permiten restricción de seguridad, por lo que cualquier usuario puede editarlas.

Para definir las extensiones de los componentes sobre los cuales se quiere configurar la seguridad se debe editar el archivo `/biserver-ce/pentaho-solutions/system/pentaho.xml` y buscar la siguiente porción de código:

```
</cache-provider>
<!-- Insert additional pentaho-system -->
<audit>
```

Insertar luego de `</cache-provider>` y antes de `<audit>`, la siguiente porción de código:

```
<acl-publisher>
  <default-acls>
    <acl-entry role="Administrator" acl="FULL_CONTROL" />
    <acl-entry role="Authenticated" acl="EXECUTE" />
  </default-acls>
  <overrides>
    <file path="/pentaho-solutions/admin">
      <acl-entry role="Administrator" acl="FULL_CONTROL" />
    </file>
  </overrides>
</acl-publisher>
<acl-voter>
  <admin-role>Administrator</admin-role>
</acl-voter>
<acl-files>xaction,url,prpt,prpti,xdash,xcdf,wcdf,saiku,cdfde</acl-files>
<anonymous-authentication>
  <anonymous-user>anonymousUser</anonymous-user>
  <anonymous-role>Anonymous</anonymous-role>
</anonymous-authentication>
```

De esta manera se puede definir para los tableros y vistas de Saiku, dentro de la opción de Administración, los permisos para cada usuario o rol.

Para ver los cambios reiniciar el BI-Server.

6.1.2.2. Desactivar “Login as an Evaluator”

Para desactivar el Login de los usuarios de prueba se debe editar el archivo `/biserver-ce/pentaho-solutions/system/pentaho.xml` y buscar la siguiente porción de código:

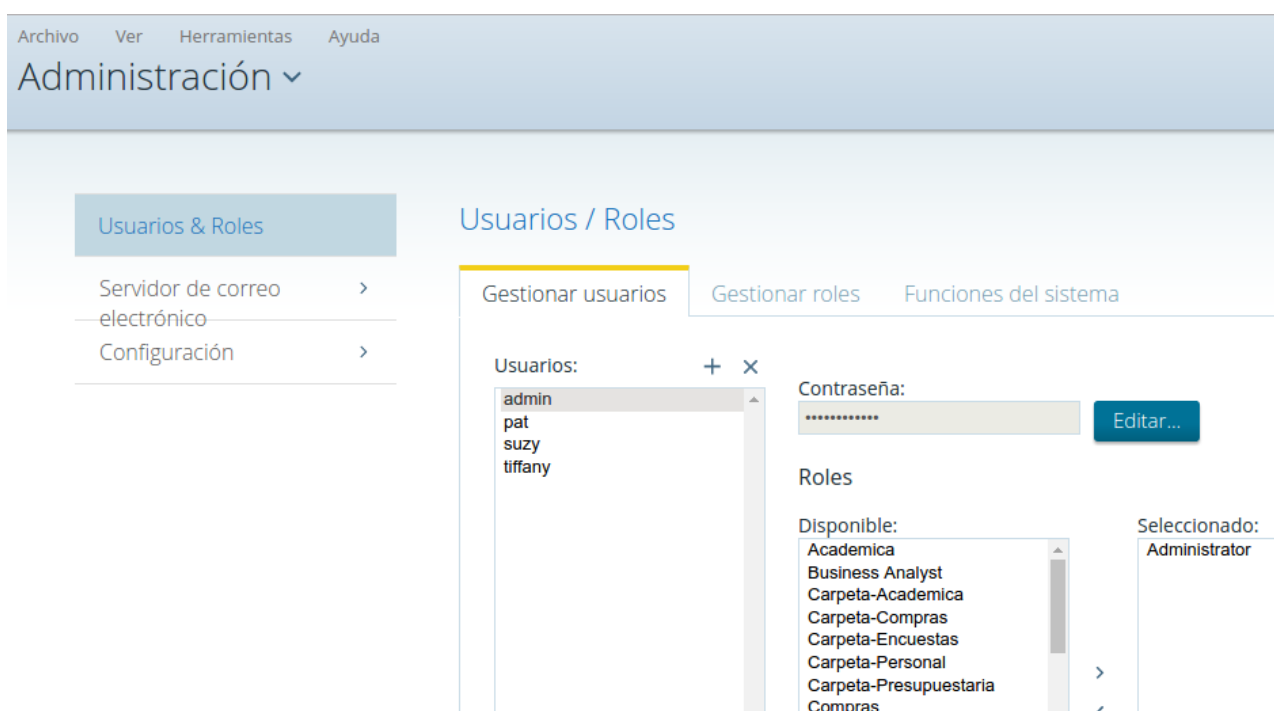
```
<login-show-users-list>true</login-show-users-list>
<!--
  If true, show hint about sample users. Ultimately, should replace
login-show-users-list.
-->
<login-show-sample-users-hint>true</login-show-sample-users-hint>
```

Modificar el texto en negrita de modo que quede como sigue:

```
<login-show-users-list>false</login-show-users-list>
<!--
  If true, show hint about sample users. Ultimately, should replace
login-show-users-list.
-->
<login-show-sample-users-hint>false</login-show-sample-users-hint>
```

6.1.3. Gestión de Usuarios en Pentaho

La gestión de usuarios en Pentaho se realiza mediante la opción Administración. Para acceder a ella en la pantalla inicial hacer clic en Examinar y de la lista desplegable elegir la opción Administración.



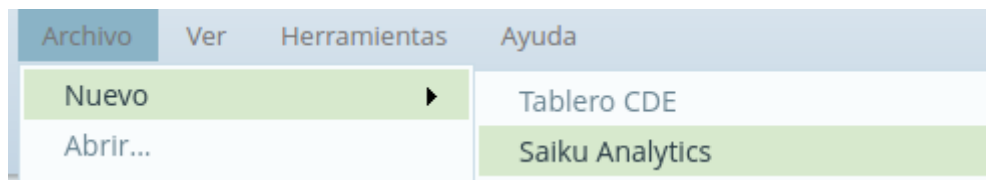
En este panel se pueden ver tres solapas que nos permiten ver los usuarios y los roles, agregarlos, borrarlos y relacionarlos.

El SIU-Wichi posee un componente que genera automáticamente el conjunto de Roles necesarios para particionar la base de datos y brindarle permiso a cada tipo de usuario, por esta razón se recomienda realizar todo el procedimiento indicado en el punto siguiente antes de acceder a esta

opción de administración a fin de poder visualizar los Roles creados.

6.1.4. Habilitación de Cubos

Una vez personalizado el acceso a los usuarios por carpetas es probable que se pueda obtener acceso indeseado a información creando nuevas vistas. Esto se realiza mediante la opción *Archivo* → *Nuevo* → *Saiku Analytics*



Para que el acceso a toda la información esté también limitado debe existir el perfil de acceso a los roles en el esquema pentaho (SIU-Wichi.xml)

El esquema "SIU-Wichi.xml" puede ser configurado de tal forma que los usuarios de facultades accedan sólo a la información que les corresponde. Esta configuración será explicada en las próximas páginas.

6.1.4.1. Habilitación de accesos por Unidad Académica

Este apartado explica cómo hacer para poder filtrar la información y permitir su análisis en forma vertical (según el gráfico mostrado en [6.1.1. Introducción](#)). Es decir cómo habilitar a los usuarios de una Dependencia para que sólo vean información de la misma.

Esta configuración requiere la operación de personal de soporte Informático, es decir el administrador técnico del sitio tomando la definición que brinden los usuarios de la gestión. A continuación se explica la mecánica utilizada y la forma de configuración y mantenimiento.

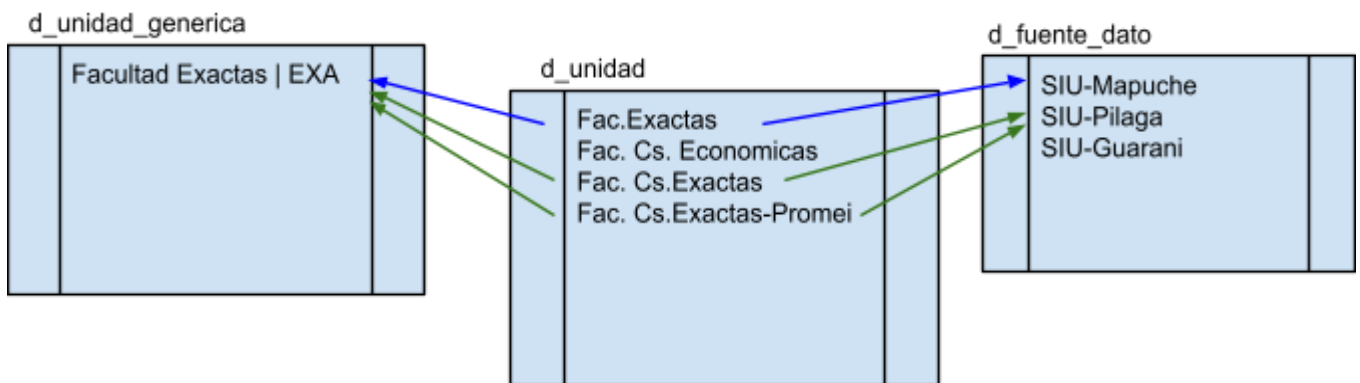
Las tareas a realizar son:

- Unificar las distintas formas de identificar a las dependencias por parte de los sistemas. Esto significa que lo primero a hacer es definir todas las divisiones verticales en las que nos va a interesar particionar la información, les daremos el nombre de **Unidades de Análisis**.
- A cada Unidad de Análisis le definiremos un nombre de Rol para luego asignárselo a los usuarios creados.
- Definir en el esquema de análisis de datos de pentaho la asignación de cada rol con un valor de dimensión específico.

Para gestionar estos perfiles de datos se han creado un conjunto de tablas en la base de datos siu_wichi, dentro del esquema public.

Ejemplo:

Se muestra a continuación un caso en el que la Facultad Exactas está identificada de una forma en Mapuche (Fac. Exactas) y de dos formas distintas en Pilaga (Fac. Cs.Exactas y Fac. Cs.Exactas-Promei) y cómo se modela esta realidad en las tablas del esquema anterior. En el mismo ejemplo se puede ver que la Unidad de Análisis "Facultad Exactas" tiene definido el rol EXA que luego deberá ser asignado a todos los usuarios que se deseen crear con estas características de acceso.

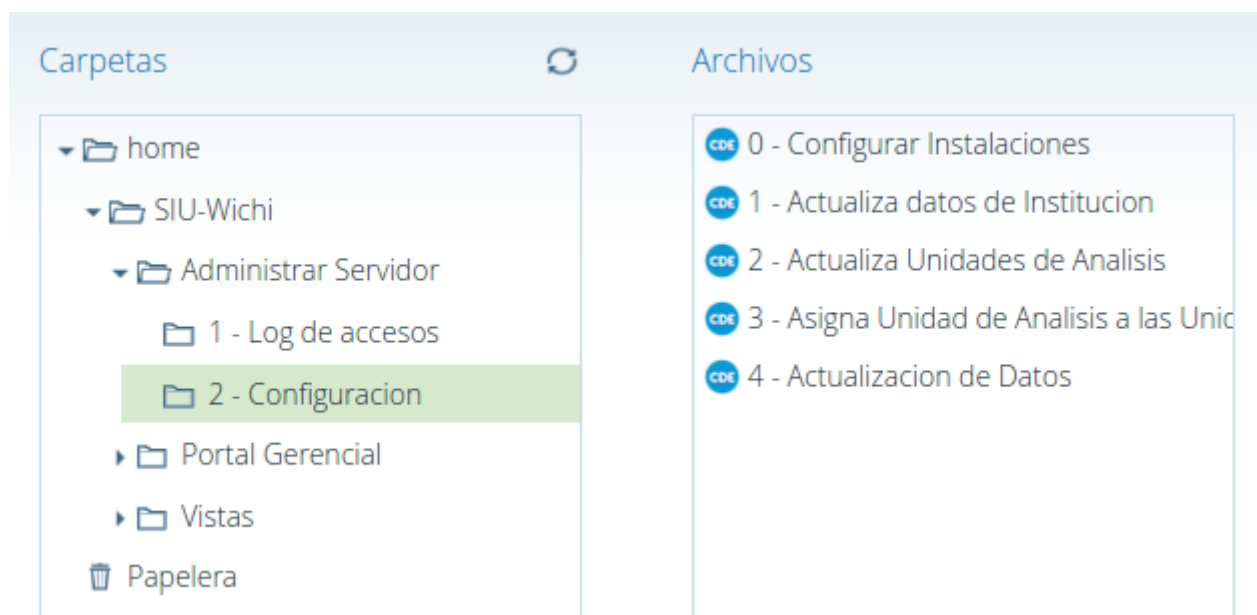


6.1.4.1.1. Ejemplo de roles y acceso segmentando la información

Puede ver un ejemplo detallado en <https://goo.gl/JTqSSf>

6.1.4.2. Datos de configuración

Para configurar el acceso particionado de la información se han definido un conjunto de opciones accesibles desde la carpeta **Administrar Servidor** → **2 - Configuración**.



6.1.4.2.1 - Actualiza datos de Institución

1 - Actualiza datos de Institucion x

 **SIUWICHI**
MÓDULO DE INFORMACIÓN
GERENCIAL

1 - Actualiza datos de Institucion

Actualización de datos de la Institución
(esta información está en la tabla `public.d_institucion`).

Código SAF:

800

Código ONA:

40

Nombre:

Universidad Nacional de Prueba

Sigla:

UNP

Actualizar datos Institución

[Ver códigos SAF y ONA](#)

6.1.4.2.2 - Actualiza Unidades de Análisis

Con esta opción se cargan las Unidades de Análisis en que se desea filtrar la información (habitualmente principales dependencias de la universidad, por ejemplo facultades y rectorado).

Lista de Unidades de Análisis

(esta información está en la tabla `public.d_unidad_generica`).

Show entries Search:

Código	Descripción	Rol de usuario
11	Agronomía	FacAgro
5	Arte	FacArte
102	Campus	Rec016
10	Derecho	FacDer
7	Económicas	FacEcon
6	Exactas	FacExa
8	Humanas	FacHum
1	Ingeniería	FacIng
9	Instituto	UEQQN
4	Medicina	ESSalud

Showing 1 to 10 of 14 entries

Datos de la Unidad de Análisis

(Tanto la descripción de la Unidad como del Rol deben ser únicos)

Descripción:

Rol de Usuario:

[Alta de Unidad de Análisis](#)

6.1.4.2.3 - Asignación de Unidades de Análisis

Esta opción permite asignar las Facultades utilizadas por los sistemas SIU a las Unidades de Análisis definidas en Wichi.

Asignaciones de Unidades de Análisis

(esta información está en la tabla `public.d_unidad`).

Show entries Search:

Código	Unidad de Análisis	Instalación	Unidad usada en Sistema Fuente
915	Agronomía	Instalacion de todas las facultades	Facultad De Agronomía De Azul
913	Económicas	Instalacion de todas las facultades	Facultad De Ciencias Económicas
916	Exactas	Instalacion de todas las facultades	Facultad De Ciencias Exactas
912	Humanas	Instalacion de todas las facultades	Facultad De Ciencias Humanas
914	Veterinarias	Instalacion de todas las facultades	Facultad De Ciencias Veterinarias
917	Agronomía	kollaunne	Facultad de Arquitectura y Urbanismo
920	Arte	kollaunne	Facultad de Ciencias Agrarias
918	Sociales	kollaunne	Facultad de Humanidades
921	Ingeniería	kollaunne	Facultad de Medicina
919	Derecho	kollaunne	Facultad de Odontología

Showing 1 to 10 of 914 entries

Instalación:

Unidad Académica:

Unidad de Análisis asignada:

[Guarda la asignación](#)

6.1.4.2.4 - Actualización de Datos

Una vez completados estos pasos ya se ha cargado toda la información necesaria para que Pentaho tenga definidos todos los roles necesarios a los fines de brindar acceso parcializado según las necesidades de la Institución. En la opción siguiente (4 - Actualización de Datos) se puede revisar la configuración definida y sobre el final de la misma están los botones para impactar todos los cambios tanto en la base de datos como en el esquema Pentaho para dejar disponibles los roles a los fines de asignárselos a los usuarios que se deseen definir.

En primer término es necesario prestar especial atención a los datos que figuran en la tabla titulada "Lista de Unidades a las que le falta asignación", dicha tabla debería estar vacía, de lo contrario significa que aún hay Facultades/Dependencias a las que no se le ha definido la Unidad de Análisis en la que queda incluida. En este caso estos datos no quedarán asignados a ningún rol visualizable en forma parcial, sólo lo podrán ver los usuarios de la Administración Central que tienen acceso a toda la información.

Ejecutar los procesos para actualizar la base de datos, para ésto existen los botones que se visualizan a continuación y cuya funcionalidad se explica posteriormente:

4 - Actualización de Datos ×

Actualización de Unidades de Análisis en las tablas de datos

Se deben ejecutar los botones "Actualizar" si la tabla contiene registros sin unidad generica (cantidad mayor a cero)

Se deben ejecutar los botones "Reasignar" si se han cambiado asignaciones de Unidades de Análisis y se debe reagrupar todo nuevamente.

Registros de Guaraní sin Unidades Académica Generica asignada	865146	Actualizar G2	Reasignar G2
Registros de Guaraní3 sin Unidades Académica Generica asignada	372779	Actualizar G3	Reasignar G3
Registros de Araucano sin Unidades Académica Generica asignada	7783	Actualizar	Reasignar
Registros de Mapuche sin Unidades Académica Generica asignada	8345	Actualizar	Reasignar

Esta información está organizada por fuente de datos o sistema origen, esto significa que la primer "fila" se refiere a Guaraní, la segunda a Araucano y así sucesivamente.

En cada fila hay en primer término una tabla que indica la cantidad de registros de datos que existen provenientes del sistema correspondiente que NO tienen asignada la Unidad de Análisis. Por lo tanto un valor cero en dicha tabla indica que no es necesario presionar ninguno de los botones que figuran a su derecha.

Si la tabla correspondiente indica un valor mayor a cero producto de haber incorporado datos luego de la última actualización debe presionar el botón “Actualizar” que asignará las Unidades de Análisis faltantes. Si por otro lado ha estado reasignando Unidades de Análisis en forma distinta a una asignación anterior debe presionar el botón “Reasignar” que se encargará de actualizar TODOS los registros de datos según la nueva configuración. Luego de presionar alguno de estos botones la tabla debería indicar un valor cero, de lo contrario aún no se ha distribuido toda la información.

6.1.4.2.5 Actualización de Roles

Cómo último proceso se debe actualizar el esquema SIU-Wichi.xml según los roles y Unidades Académicas Genéricas que han definido. **Siempre utilizar el archivo Siu-Wichi.xml que viene en la versión**, ubicado en el directorio `/utilidades/esquemas` del paquete SIU-Wichi.

Mediante el ETL → `principal_actualizar_roles.kjb`, el mismo por un lado carga en el esquema Pentaho de Wichi todos los roles definidos y a cada uno lo configura filtrando la Unidad de Análisis que le corresponda para todos los cubos existentes en el sistema. Por otro lado crea todos los roles en la base de datos de tal forma que cuando entre a la pantalla de administración de Pentaho los vea disponibles para poder asignarlos a los usuarios que desee crear.

- a) **Ejecutar el ETL** `/administracion_servidor/actualizar_roles/principal_actualizar_roles.kjb`, como se muestra a continuación luego de reemplazar los valores en negrita.

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/administracion_servidor/actualizar_roles/principal_actualizar_
  roles.kjb" \
  -param="base_usuario=postgres" \
  -param="base_clave=xxxxxxx" \
  -param="base_nombre=siu_wichi" \
  -param="base_host=localhost" \
  -param="base_puerto=5432" \
  -param="directorio_xml=/utilidades/esquemas" \
  -param="pentaho_url_puerto=localhost:8080" \
  -param="pentaho_usuario=admin" \
  -param="pentaho_password=password" \
```

En el parámetro **directorio_xml** deberá ingresar el path del directorio donde se encuentra el archivo *SIU-Wichi.xml*, ubicado en el directorio `/utilidades/esquemas` del paquete SIU-Wichi.

En ese mismo directorio (**directorio_xml**) se generará un nuevo esquema SIU-Wichi.xml, el cual contendrá los roles definidos. Además se genera un backup del esquema SIU-Wichi_fecha.xml por cualquier inconveniente que pueda surgir.

- b) **Subir y actualizar el archivo SIU-Wichi.xml** en Pentaho menú Archivo / opción Gestionar Fuente de datos. Indicado en la sección [3.3.2. Actualización de esquema \(SIU-Wichi.xml\)](#)
- c) Luego de estos pasos ya está listo para **crear los usuarios y asignar roles** correspondientes al tipo de acceso que desea brindarles. Esta operación se realiza por opción Administración del menú de opciones desplegable, veamos a continuación un ejemplo:

Administración ▾

Usuarios & Roles

Servidor de correo electrónico >

Configuración >

Usuarios / Roles

Gestionar usuarios

Gestionar roles

Funciones del sistema

Roles

+ x

FacHum
 FacHum-Academica
 FacHum-Compras
 FacHum-Encuestas
 FacHum-Personal
 FacHum-Presupuestaria
 FacIng
 FacIng-Academica
 FacIng-Compras
 FacIng-Encuestas
 FacIng-Personal
 FacIng-Presupuestaria
 FacSoc
 FacSoc-Academica
 FacSoc-Compras
 FacSoc-Encuestas
 FacSoc-Personal
 FacSoc-Presupuestaria

Permisos de operación

- ☐ Administer Security
- ☒ Read Content
- ☐ Publish Content
- ☐ Manage Data Sources
- ☒ Execute
- ☐ Schedule Content
- ☒ Create Content

Usuarios

Disponible:

admin
 demo
 pat
 sabri
 suzy
 tiffanv

>

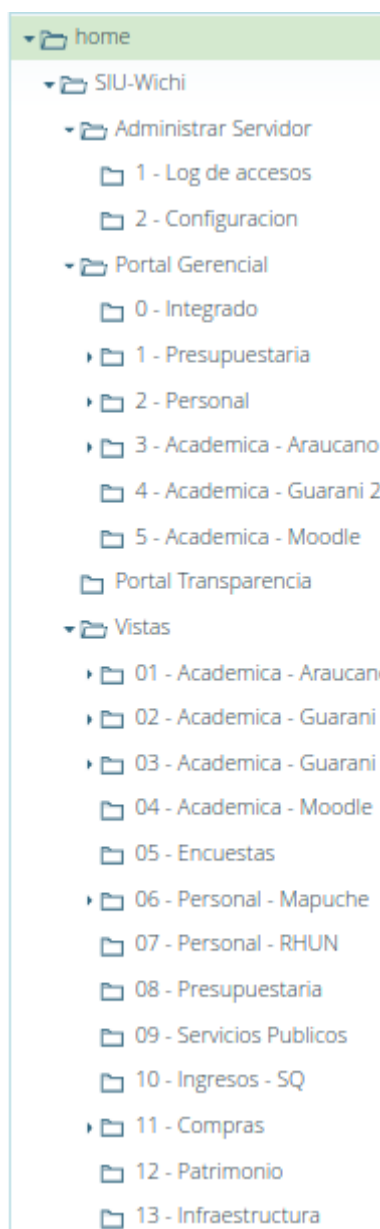
En este caso se pueden ver roles definidos para tres facultades (Facultad de Humanas, Ingeniería y Sociales) y para cada una de ellas existen cinco roles bien diferenciados. Todo usuario al que se le asigne el rol básico (FacHum, FacIng o FacSoc) podrán acceder a TODA la información de la Facultad correspondiente. Por otro lado si se le desea brindar a un usuario acceso a la información de personal y académica de la Facultad de Sociales pero no a información presupuestaria y compras se le deben asignar los roles FacSoc-Academica y FacSoc-Personal.

6.1.5. Habilitación de carpetas

El proceso ejecutado mediante el etl “Actualización de Roles” indicado en el apartado 6.1.4.2.5 adiciona a los roles creados para la partición de la información cinco roles más que pueden ser utilizados para filtrar el acceso a las carpetas disponibles en el SIU-Wichi, estos roles son:

- Carpeta-Presupuestaria
- Carpeta-Personal
- Carpeta-Academica
- Carpeta-Compras
- Carpeta-Encuestas
- Carpeta-Infraestructura
- Carpeta-Ingresos

Cuando se entra al portal se pueden visualizar las siguientes carpetas:



Si desea restringir el acceso a alguna de estas carpetas se recomienda proceder de la siguiente forma. Seleccionar la carpeta correspondiente, elegir Propiedades de las opciones de Acciones en el menú de la derecha de como se muestra a continuación:

Carpetas

SIU-Wichi

Administrar Servidor

Portal Gerencial

Portal Transparencia

Vistas

01 - Academica - Araucano

02 - Academica - Guarani 2

03 - Academica - Guarani 3

04 - Academica - Moodle

05 - Encuestas

06 - Personal - Mapuche

07 - Personal - RHUN

08 - Presupuestaria

09 - Servicios Publicos

10 - Ingresos - SQ

11 - Compras

12 - Patrimonio

13 - Infraestructura

Archivos

Cobrado Anual por Concepto de In

Cobrado Fuente 12 por Unidad Pre

Cobrado por Unidad Presupuestari

Credito por Unidad Presupuestaria

Credito por Unidad Presupuestaria

Credito por Unidad Presupuestaria

Credito por Unidad Presupuestaria

Devengado de Sueldos y Otros gas

Devengado por Inciso y Partida pri

Ejecucion Mensual del Devengado

Ejecucion por Finalidad y Funcion

Ejecucion por Fuente

Ejecucion por Grupo Presupuestari

Ejecucion por Inciso

Ejecucion por Mes

Ejecucion por Mes_g

Ejecucion por Programa

Ejecucion por Unidad Presupuestar

Ejecucion por Unidad Presupuestar

Carpeta de Acciones

Nueva Carpeta...

Mover a la papelera

Renombrar...

Pegar

Cargar...

Descargar...

Propiedades...

Seleccionar la solapa Compartir, destildar el check *Heredar permisos de carpeta*, eliminar el **Authenticated(Rol)** e incorporar el rol correspondiente de los cinco creados (por ejemplo **Carpeta-Presupuestaria**):

1 - Presupuestaria Propiedades

General Compartir

☐ Heredar permisos de carpeta

Usuarios y roles:

Carpeta-Presupuestaria
Administrator

Agregar... Remove

Permisos para Carpeta-Presupuestaria:

☐ Gestionar permisos (Control completo)
☐ Eliminar
☐ Escribir
☒ Leer

Aprobar Cancelar

Luego, para cada usuario que se crea, hay que acceder a la pantalla de Administración → Gestionar usuarios y ponerle el o los roles según el acceso que se le desee habilitar (Carpeta-Presupuestaria, Carpeta-Personal, Carpeta-Academica, Carpeta-Compras, Carpeta-Encuestas, Carpeta-Infraestructura y Carpeta-Ingresos).

Para que todo usuario que sea de una Unidad Académica no pueda ver ninguna de las carpetas (“Administrar Servidor”, “Portal Gerencial” y “Vistas”) se debe acceder también a ellas y deshabilitar el acceso al usuario Authenticated(Rol).

6.2. Configurar certificado SSL



Agradecemos el aporte de este punto a Roberto Carlos Ruiz de la UNSL

Usando Letsencrypt

1- Instalar certbot

Debian 8 - Agregar a sources.list

```
deb http://ftp.debian.org/debian jessie-backports main
```

```
apt-get install certbot -t jessie-backports
```

Debian 9

```
apt-get install certbot
```

2- Generación de Certificado

```
keytool -genkey -alias tomcat -keyalg RSA -keystore /usr/local/pentaho/biserver-ce/tomcat/.keystore  
-keysize 2048
```

#IMPORTANTE! EN CN USAR EL DOMINIO DEL SISTEMA ejemplo: "example.com"

```
keytool -list -keystore /usr/local/pentaho/biserver-ce/tomcat/.keystore -v -storepass "you_password" >  
key.check
```

```
keytool -certreq -alias tomcat -file request.csr -keystore /usr/local/pentaho/biserver-ce/tomcat/.keystore  
-storepass "you_password"
```

```
certbot certonly --csr ./request.csr --standalone
```

```
keytool -import -trustcacerts -alias tomcat -file 0001_chain.pem -keystore  
/usr/local/pentaho/biserver-ce/tomcat/.keystore -storepass "you_password"
```

3- Configuración

Editar /usr/local/pentaho/biserver-ce/tomcat/conf/server.xml

```
<Connector port="8080" enableLookups="false"  
    redirectPort="8443" />  
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
    maxThreads="200" scheme="https" secure="true"  
    clientAuth="false" sslProtocol="TLS"  
    keystoreFile="/usr/local/pentaho/biserver-ce/tomcat/.keystore"  
    KeystorePass="you_password" />
```

Editar /usr/local/pentaho/biserver-ce/tomcat/webapps/pentaho/WEB-INF/web.xml y agregar:

```
<security-constraint>  
    <web-resource-collection>  
        <web-resource-name>Automatic Forward to HTTPS/SSL  
    </web-resource-name>  
        <url-pattern>/*</url-pattern>  
    </web-resource-collection>  
    <user-data-constraint>  
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
    </user-data-constraint>
```

</security-constraint>

Editar /usr/local/pentaho/biserver-ce/pentaho-solutions/system/server.properties

fully-qualified-server-url=https://localhost:8443/pentaho/

Con iptables redireccionar el puerto 80 al 8080

iptables -A PREROUTING -t nat -p tcp --dport 80 -j REDIRECT --to-ports 8080

SIU-Wichi con SSL usando crt y key (COMODO CA)

Asumiendo:

- * you_cert.key: la clave privada de su certificado
- * you_cert.crt - tu certificado
- * domainIntermediate.crt - Validación de la organización intermedia
- * inter.crt: la CA intermedia que firmó su certificado
- * root.crt - la CA raíz que firmó la CA intermedia

cat domainIntermediate.crt inter.crt root.crt > chain.crt

En el caso de COMODO CA, descargarla desde:

<https://support.comodo.com/index.php?/comodo/Knowledgebase/Article/View/979/108/domain-validation-sha-2>

En mi caso puntual

<https://support.comodo.com/index.php?/Knowledgebase/Article/GetAttachment/979/1056459>

Renombrar

mv comodo-rsa-domain-validation-sha-2-w-root.ca-bundle you_chain.crt

openssl pkcs12 -export -chain -inkey you_cert.key -in you_cert.crt -name "wichi" -CAfile you_chain.crt -out wichi.p12

Cuando se le solicite la contraseña de exportación, ingrese algo y no lo deje en blanco.

Ahora usa keytool para verificar:

keytool -list -v -storetype pkcs12 -keystore wichi.p12

Ingrese la contraseña de exportación para la contraseña del almacén de claves.

Configuración

Editar /usr/local/pentaho/biserver-ce/tomcat/conf/server.xml

<Connector port="8080" enableLookups="false"
 redirectPort="8443" />

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

```
maxThreads="200" SSLEnabled="true" scheme="https" secure="true"  
keystoreFile="/usr/local/pentaho/biserver-ce/tomcat/ssl/wichi.p12"  
keystorePass="you_password" clientAuth="false" sslProtocol="TLS"  
keystoreType="PKCS12" />
```

Editar /usr/local/pentaho/biserver-ce/tomcat/webapps/pentaho/WEB-INF/web.xml y agregar:

```
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>Automatic Forward to HTTPS/SSL  
  </web-resource-name>  
    <url-pattern>/*</url-pattern>  
  </web-resource-collection>  
    <user-data-constraint>  
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
    </user-data-constraint>  
</security-constraint>
```

Editar /usr/local/pentaho/biserver-ce/pentaho-solutions/system/server.properties

fully-qualified-server-url=https://localhost:8443/pentaho/

Con iptables redireccionar el puerto 80 al 8080

```
iptables -A PREROUTING -t nat -p tcp --dport 80 -j REDIRECT --to-ports 8080
```

Fuentes:

<https://community.letsencrypt.org/t/configuring-lets-encrypt-with-tomcat-6-x-and-7-x/32416>
<http://tkurek.blogspot.com.ar/2013/07/tomcat-7-http-to-https-redirect.html>

7. Aspectos avanzados de Pentaho

En este capítulo se explican algunos puntos de configuración avanzada de Pentaho BI Server. Aplique solo los puntos que considere necesarios en su instalación en particular.

7.1. BI-Server

7.1.1. Desactivar JPivot

Debido a que utilizamos saiku para analizar las vistas, ya que es mucho más ágil y simple de usar, podemos desactivar el plugin JPivot. Para ello se debe renombrar el archivo *plugin.xml* ubicado en *biserver-ce/pentaho-solutions/system/pentaho-jpivot-plugin/* como *plugin_desac.xml*.

7.1.2. Arranque automático

Antes de continuar, es conveniente verificar la configuración de regionalización del sistema operativo. En Ubuntu, lo podemos hacer con el comando

```
$ locale
```

Observe los valores que se obtienen para las variables **LANG** y **LC_TYPE**. Los valores deberían ser algo como **es_AR.UTF-8** o **es_ES.UTF-8**. Es importante que la codificación esté en UTF-8. En los scripts de inicio que se muestran más abajo, sustituya los valores en las líneas que comienzan con `cmd="sudo -u pentaho...` por los valores de su sistema.

Se puede crear un script de inicio del servidor de Pentaho para que se ejecute cuando arranca el sistema operativo. En este caso, se crea el archivo `pentaho-biserver.sh` que se encuentra en `/etc/init.d`. (Nota: hay que tener permisos de root, y hay que usar sudo para copiar o mover el script en esa ubicación.)

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          pentaho-biserver
# Required-Start:    postgresql $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Pentaho Bi-Server
# Description:       Pentaho Bi-Server
#
### END INIT INFO

# Script para iniciar automaticamente del servidor de Pentaho

cd /usr/local/pentaho/biserver-ce
# set up command for pentaho user, set java environment
cmd="sudo -u pentaho LANG=es_ES.UTF-8 LC_TYPE=es_ES.UTF-8
JAVA_HOME=/usr/lib/jvm/default-java JAVA_OPTS=-Djava.awt.headless=true"
case "$1" in
  start)
    # run the original pentaho start script
    $cmd ./start-pentaho.sh >> /var/log/pentaho-biserver.log 2>&1 &
    ;;
  stop)
    # run the original pentaho stop script
    $cmd ./stop-pentaho.sh >> /var/log/pentaho-biserver.log 2>&1 &
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  *)
    echo "Usage: $0 {start|stop|restart}"
    exit 1
```

```
esac
exit 0
```

Para usar el script manualmente:

```
$ cd /etc/init.d
$ sudo ./pentaho-biserver.sh
Usage: ./pentaho-biserver.sh {start|stop|restart|status}
$ sudo ./pentaho-biserver.sh start
$ sudo ./pentaho-biserver.sh stop
```

En las distribuciones Linux basadas en Debian, incluyendo Ubuntu, se puede utilizar la aplicación **update-rc.d**, mediante la cual se setean los links simbólicos, haciendo que el script se ejecute en el momento que el Servidor arranca, iniciando el servidor de Pentaho (y parando el servidor de pentaho cuando se apaga o reinicia el servidor):

```
$ sudo update-rc.d pentaho-biserver.sh defaults
```

Luego, el siguiente comando debería mostrar los links generados

```
$ ls /etc/rc?.d/*pentaho*
```

7.1.2.1 Arranque automático Systemd



Agradecemos el aporte de este punto a Roberto Carlos Ruiz de la UNSL

Otra opción de arranque automático, sería la siguiente:

Crear el archivo pentaho.service

```
$sudo nano /etc/systemd/system/pentaho.service
```

```
[Unit]
Description=Pentho Bi-server
After=syslog.target
After=network.target
[Service]
Type=forking
User=pentaho
Group=pentaho
Environment=JAVA_HOME
ExecStart=/usr/local/pentaho/biserver-ce/start-pentaho.sh
ExecStartPost=/bin/echo pentaho...end of unitfile
ExecStop=/usr/local/pentaho/biserver-ce/stop-pentaho.sh
TimeoutSec=300
```

```
[Install]
```

WantedBy=graphical.target

A continuación

```
$sudo systemctl enable pentaho.service  
$sudo systemctl start pentaho.service  
$sudo systemctl daemon-reload
```

7.1.3. Servidor SMTP

Para configurar el servidor de Pentaho para usar un servidor SMTP (servidor de correo) para el envío de reportes por email, etc hay que ingresar a la Consola de Usuario. Luego hacer clic en *Examinar* y seleccionar *Administración* del combo desplegable. Una vez allí, seleccionar la opción *Servidor de correo electrónico* como se muestra en la imagen:

The screenshot shows the 'Administración' (Administration) page in the Pentaho User Console. The left sidebar contains a menu with 'Usuarios & Roles', 'Servidor de correo electrónico' (highlighted), and 'Configuración'. The main content area is titled 'Servidor de correo' and contains the 'Configuración del servidor' (Server Configuration) section. This section includes the following fields and options:

- SMTP Host:** A text input field.
- Puerto:** A text input field containing the value '25'.
- Uso de autenticación:** A checked checkbox.
- Nombre de usuario:** A text input field.
- Contraseña:** A text input field.
- Protocolo:** A dropdown menu with 'SMTP' selected.
- Todos los correos electrónicos vendrán desde:** A text input field.
- Email \"From\" Name:** A text input field.
- Inciar uso TLS:** An unchecked checkbox.
- Uso SSL:** An unchecked checkbox.

At the bottom of the configuration section, there is a blue button labeled 'Prueba de configuración de correo electrónico' and a grey button labeled 'Guardar'.

Estos son los parámetros que pueden ser configurados para tener soporte SMTP:

- SMTP Host
 - Dirección del servidor de email SMTP Ej: smtp.gmail.com
- Puerto

- Puerto del servidor SMTP Ej: para GMail es el 587
- Uso de autenticación
 - Tildar el check si el servidor de email requiere la autenticación al enviar.
- Nombre de usuario
 - Nombre de usuario usado para autenticar con el servidor SMTP, Uso de autenticación debe estar tildado.
- Contraseña
 - Contraseña que se usa para autenticar con el servidor SMTP, Uso de autenticación debe estar tildado.
- Protocolo
 - El protocolo de transporte para acceder al servidor de email. Generalmente es smtp Ej: para GMail es smtps
- Todas los correos electrónicos vendrán desde
 - La dirección de origen "from" que envía mails desde el servidor de Pentaho Ej: serverpentaho@gmail.com
- Email From Name
 - El nombre de origen "from" que envía mails desde el servidor de Pentaho Ej: Soporte de Pentaho
- Iniciar uso TLS
 - Si el servidor SMTP server usa autenticación TLS tildar el check. Ej: GMail usa TLS, por lo que debería estar tildado
- Uso SSL
 - Si el servidor SMTP server usa autenticación SSL tildar el check.

7.1.4. Desactivar la base HSQL en el BI Server

En caso de que su configuración no utilice ninguna de las bases de datos Hypersonic, se puede desactivar el arranque del motor. Por defecto con la versión 6.1 la base HSQL se inicia automáticamente.

Buscar los siguientes bloques de código en el archivo `web.xml` del directorio `biserver-ce/tomcat/webapps/pentaho/WEB-INF`:

```
<context-param>
<param-name>hsqldb-databases</param-name>
<param-value>sampledata@../../data/hsqldb/sampledata,hibernate@../../data/hsqldb/hibernate,quartz@../../data/hsqldb/quartz</param-value>
</context-param>

<listener>
<listener-class>org.pentaho.platform.web.http.context.HsqldbStartupListener</listener-class>
</listener>
```

En caso de que desee conservar la base de datos de ejemplo SampleData, deje solo el arranque de la misma, de manera que quede como:

```
<context-param>
<param-name>hsqldb-databases</param-name>
<param-value>sampledata@../../data/hsqldb/sampledata</param-value>
```

```
</context-param>
```

Para desactivar completamente la base HSQL se puede eliminar el siguiente código o simplemente comentarlo de manera que quede como:

```
<!--
<context-param>
<param-name>hsqldb-databases</param-name>
<param-value>sampledata@../../data/hsqldb/sampledata,hibernate@../../data/hsqldb/hibernate,quartz@../../data/hsqldb/quartz</param-value>
</context-param>
-->

<!--
<listener>
<listener-class>org.pentaho.platform.web.http.context.Hsqldb
StartupListener</listener-class>
</listener>
-->
```

7.1.5. Cambiar imagen de pantalla de Login

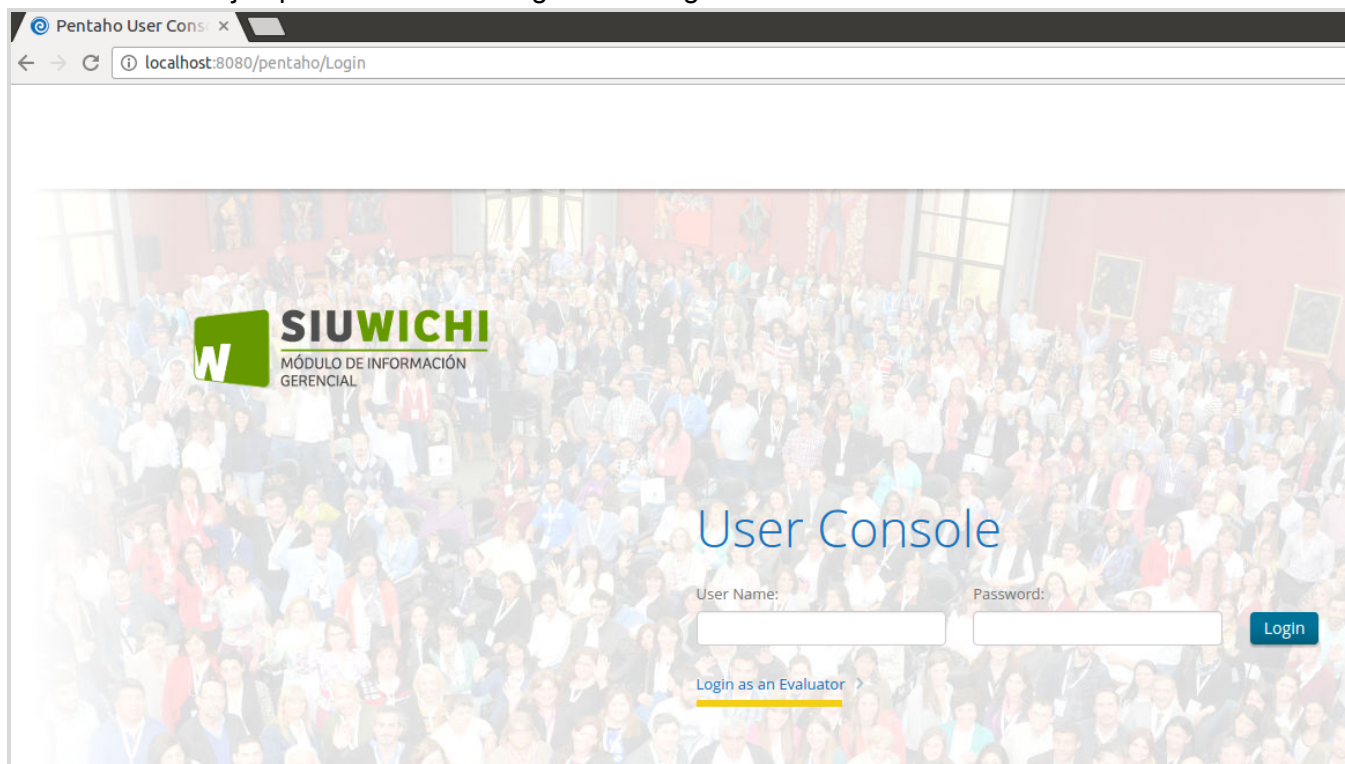
Para modificar la imagen de fondo de la pantalla de Login deberá reemplazar el archivo *login-crystal-bg.jpeg* ubicado en

/biserver-ce/pentaho-solutions/system/common-ui/resources/themes/crystal/images

Para modificar el logo de la pantalla de Login deberá reemplazar el archivo *puc-login-logo.png* ubicado en

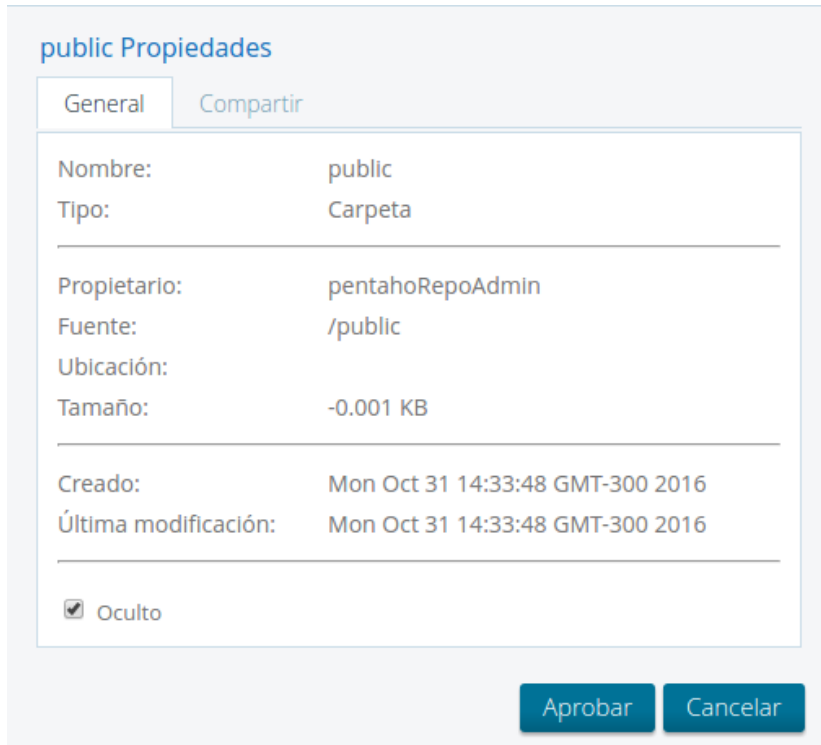
/biserver-ce/pentaho-solutions/system/common-ui/resources/themes/images

Como modo de ejemplo se muestra la siguiente imagen:



7.1.6. Ocultar carpetas

Si desea ocultar una carpeta, como por ejemplo la carpeta **public** que viene por defecto, deberá posicionarse sobre la misma y elegir del panel de *Acciones por Carpeta*, la opción *Propiedades*. Luego tildar el check *Oculto*.



public Propiedades

General Compartir

Nombre: public
 Tipo: Carpeta

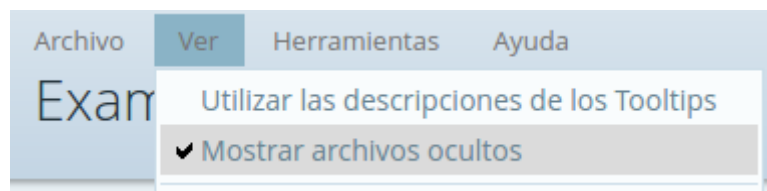
Propietario: pentahoRepoAdmin
 Fuente: /public
 Ubicación:
 Tamaño: -0.001 KB

Creado: Mon Oct 31 14:33:48 GMT-300 2016
 Última modificación: Mon Oct 31 14:33:48 GMT-300 2016

☒ Oculto

Aprobar Cancelar

Luego podrá ver las carpetas o archivos ocultos ingresando en la opción Ver → Mostrar archivos ocultos



7.1.7. Cambiar paleta de colores de saiku

Editar el archivo `biserver-ce/pentaho-solutions/system/saiku/ui/saiku.min.js`

Buscar la siguiente porción de código:

```
a.colors=["#AE1717","#AE5B17","#0E6868"];
```

Reemplazar por la paleta de colores deseada, como modo de ejemplo:

```
a.colors=["#1f77b4", "#aec7e8", "#ff7f0e", "#ffbb78", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5", "#8c564b", "#c49c94", "#e377c2", "#7b6d2", "#7f7f7f", "#c7c7c7", "#bcbd22", "#dbdb8d", "#17becf", "#9edae5"];
```

7.1.8. Iniciar Pentaho en idioma español

Una vez instalado el paquete de lenguaje español como se indica en la sección [2.3.4. Instalación de Spanish Language Pack](#), puede configurar la plataforma pentaho para que siempre inicie en español. Para esto deberá verificar la configuración de regionalización del sistema operativo. En Ubuntu, lo podemos hacer con el comando

```
$ locale
```

Observe los valores que se obtienen para las variables **LANG** y **LC_TYPE**. Los valores deberían ser algo como **es_AR.UTF-8** o **es_ES.UTF-8**.

Luego deberá buscar en el archivo `web.xml` del directorio `biserver-ce/tomcat/webapps/pentaho/WEB-INF` la siguiente línea:

```
<!-- insert additional context-params
```

Deberá insertar luego de esa línea, el siguiente bloque de código:

```
<context-param>
  <param-name>locale-language</param-name>
  <param-value>es</param-value>
</context-param>

<context-param>
  <param-name>locale-country</param-name>
  <param-value>ES</param-value>
</context-param>

<context-param>
  <param-name>encoding</param-name>
  <param-value>UTF-8</param-value>
</context-param>
```

Una vez guardado estos cambios se debe reiniciar Pentaho utilizando `/biserver-ce/stop-pentaho.sh` y luego `/biserver-ce/start-pentaho.sh`.

7.2. Tomcat

7.2.1. Puertos de escucha

Por defecto, el BI-Server viene configurado para escuchar en el **puerto 8080** y en la url <http://localhost:8080>. En caso que quiera o deba modificar estos valores, también se deberá **cambiar la url del servidor** (explicado más abajo).

Acceder al archivo: `/biserver-ce/tomcat/conf/server.xml` y buscar las siguientes líneas:

```
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
```

```
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->

<Connector URIEncoding="UTF-8" port="8080" protocol="HTTP/1.1"
connectionTimeout="20000" redirectPort="8443" />
```

Se deberá cambiar el valor 8080 por otro puerto no usado. Además debería cambiarse el puerto de shutdown.

Finalmente, dado que cambia la URL de acceso, hay que reflejar el cambio en server.xml, como se explica en [7.2.2. Cambiar URL de acceso](#). Una vez guardada la nueva configuración y reiniciado el servidor, se podrá acceder con el nuevo puerto, por ejemplo: <http://localhost:8081/pentaho>.

7.2.2. Cambiar URL de acceso

Si se va a acceder al servidor mediante una URL distinta a localhost:8080, se debe editar el archivo que se indica a continuación.

En `/biserver-ce/pentaho-solutions/system/server.properties` buscar las siguientes líneas:

```
# FullyQualifiedServerUrl is used only in the case of offline content generation
# and whenever something need to talk back to the server
fully-qualified-server-url=http://localhost:8080/pentaho/
```

Y donde aparece `http://localhost/pentaho/`, sustituir "localhost" por la IP:Puerto o la dirección deseada del servidor. Para que los cambios tomen efecto hay que reiniciar el servidor.

7.2.3. Duración de la sesión

Por defecto, la Consola de Usuario de Pentaho viene configurada con un tiempo de timeout para la sesión de 30 minutos.

Modificar el archivo `web.xml` en `/biserver-ce/tomcat/webapps/pentaho/WEB-INF`
Buscar la configuración:

```
<!-- insert additional servlet mappings -->
<session-config>
<session-timeout>120</session-timeout>
</session-config>
```

El número es el tiempo de sesión en minutos.

Modificar y reiniciar el BI-Server.

7.2.4. Otros puntos de configuración

Para configurar el servidor Apache-Tomcat de Pentaho, la mayoría de los cambios se hacen en el archivo `web.xml` que se encuentra en el directorio `biserver-ce/tomcat/webapps/pentaho/WEB-INF/`.

Los siguiente items y otros se pueden configurar en Pentaho:

- Ubicación de pentaho-solutions
- TrustedIpAddr
- URL de acceso
- Deshabilitar el arranque de la base HSQL
- Configuración de idioma y localización (no se explican en este documento)

7.2.4.1. Ubicación de pentaho-solutions

El parámetro solution-path le permite al BI-Server de Pentaho saber ubicar el directorio pentaho-solutions. Por defecto se setea en el directorio `/biserver-ce/pentaho/`.

Si decidió utilizar un servidor Apache-Tomcat existente (o movió el directorio `pentaho-solutions`) necesitará apuntar este a donde se encuentra el directorio `pentaho-solutions`. En este ejemplo el directorio `pentaho-solutions` se encuentra en `/usr/local/pentaho/`, y el fragmento de código del `solution-path` se ve como:

```
<context-param>
  <param-name>solution-path</param-name>
  <param-value>/usr/local/pentaho/pentaho-solutions/</param-value>
</context-param>
```

7.2.4.2. TrustedIpAddr

Si se va a acceder al BI-Server mediante una dirección IP o hostname distinto al que viene por defecto 127.0.0.1, se debe modificar el proxy de confianza para que coincida con la dirección o el hostname para que los plugins de Pentaho funcionen correctamente. Además, si tiene aplicaciones que acceden a los recursos del BI-Server, como APIs REST, debe agregar la dirección IP de esa aplicación para que sus solicitudes sean aceptadas.

Abrir el archivo y encontrar el siguiente código:

```
<filter>
  <filter-name>Proxy Trusting Filter</filter-name>
  <filter-class>org.pentaho.platform.web.http.filters.ProxyTrustingFilter</filter-class>
  <init-param>
    <param-name>TrustedIpAddr</param-name>
    <param-value>127.0.0.1,0\:0\:0\:0\:0\:0\:0\:0\:1(%.+)*$</param-value>
    <description>Comma separated list of IP addresses of a trusted
hosts.</description>
  </init-param>
</filter>
```

Agregar separadas con coma, las ip de confianza.

8. Aspectos avanzados de SIU-Wichi

En este capítulo no se explican procesos generales, sino aquellos a aplicar en los casos en que sea necesario.

8.1. Creación parcial de la base de datos

8.1.1. Creación de esquemas por sistema fuente

Puede darse el caso en que se necesite eliminar los esquemas de un sistema fuente y volver a crearlos, sin volver a crear el resto.

Los procesos para crear los esquemas por sistema son los siguientes.

Sistema Fuente	Proceso
<i>Araucano</i>	<path_etl> /araucano/principal_creacion_araucano.kjb
<i>Guaraní</i>	<path_etl> /guarani/principal_creacion_guarani.kjb
<i>Mapuche</i>	<path_etl> /mapuche/principal_creacion_mapuche.kjb
<i>Pilaga</i>	<path_etl> /pilaga/principal_creacion_pilaga.kjb
<i>Rhun</i>	<path_etl> /rhun/principal_creacion_rhun.kjb
<i>Kolla</i>	<path_etl> /kolla/principal_creacion_kolla.kjb
<i>Diaguíta</i>	<path_etl> /diaguíta/principal_creacion_diaguíta.kjb
<i>Querandies</i>	<path_etl> /querandies/principal_creacion_querandies.kjb
<i>Sanaviron/Quilmes</i>	<path_etl> /sq/principal_creacion_sq.kjb

8.2. Carga de datos

8.2.1. Separación del proceso de extracción e importación

El proceso de carga en Pilagá o Kolla se puede separar en extracción e importación, corriéndolos por separado. Esto implicará un manejo manual o transferencia del paquete de datos, por lo tanto recomendamos considerar la funcionalidad de encriptación y firma.

Los procesos de extracción e importación para Pilaga son respectivamente:

- **<path_etl>**/carga_de_datos/pilaga/extraccion/principal_extraccion_pilaga.kjb
- **<path_etl>**/carga_de_datos/pilaga/importacion/principal_importacion_pilaga.kjb

Los procesos de extracción e importación para Kolla son respectivamente:

- **<path_etl>**/carga_de_datos/kolla/extraccion/1_1_principal_extraccion_kolla.kjb
- **<path_etl>**/carga_de_datos/kolla/importacion/1_2_principal_importacion_kolla.kjb

8.2.2. Tabla de importaciones

La tabla *public.importacion* de la base *siu_wichi* contiene una entrada para cada importación de datos. El campo estado indica si la importación pudo terminar o si fue interrumpida.

8.2.3. Procesos de actualización de unidad genérica

Es posible correr por línea de comando los procesos de actualización de unidad académica genérica. Esto puede ser útil sobre todo cuando la cantidad de datos es muy grande. La siguiente tabla muestra las rutas a los procesos.

Sistema Fuente	Proceso
<i>Araucano</i>	<path_etl> /carga_de_datos/araucano/importacion_araucano_dw/actualizaciones_unidad_generica.kjb
<i>Guaraní</i>	<path_etl> /carga_de_datos/guarani/importacion_guarani_dw/actualizaciones_unidad_generica.kjb
<i>Mapuche</i>	<path_etl> /carga_de_datos/mapuche/importacion_mapuche_dw/actualizaciones_unidad_generica.kjb
<i>Rhun</i>	<path_etl> /carga_de_datos/rhun/importacion_rhun_dw/actualizaciones_unidad_generica.kjb
<i>Pilagá</i>	<path_etl> /carga_de_datos/pilaga/importacion/pilaga_dw/actualizaciones_unidad_generica.kjb
<i>Diaguita</i>	<path_etl> /carga_de_datos/diaguita/importacion/diaguita_dw/actualizaciones_unidad_generica.kjb
<i>Querandies</i>	<path_etl> /carga_de_datos/querandies/importacion/querandies_dw/actualizaciones_unidad_generica.kjb
<i>Sanaviron/Quilmes</i>	<path_etl> /carga_de_datos/sq/importacion/sq_dw/actualizaciones_unidad_generica.kjb

Los parámetros de estos procesos son los siguientes:

Parámetro	Descripción
<i>base_clave</i>	Contraseña de la base de datos
<i>base_host</i>	Host de la base de datos

<i>base_nombre</i>	Nombre de la base de datos
<i>base_puerto</i>	Puerto de la base de datos
<i>base_usuario</i>	Usuario de la base de datos
<i>forzar_uag_ft</i>	0=actualizar, 1=reasignar

8.2.4. Script para facilitar carga de TXT [Aporte UNGS]



Agradecemos el aporte de este punto a Gonzalo Pablo Ribera de UNGS

Script de carga de los sistemas que utilizan TXT como fuente de datos, actualizado a la versión 6.5.0. Los sistemas a cargar serían:

- Guaraní 2
- Mapuche
- Rhun

El script *carga_wichi_txt.sh* se encuentra en la carpeta <Paquete_SIU-Wichi>/utilidades/general. Antes de ejecutarlo hay que editarlo y configurar algunas variables (datos de conexión, path_etl y path_kitchen).

Se puede ejecutar desde cualquier lado sin problemas pasándole sólo dos parámetros : sistema a cargar y la ruta con los txt(ya convertidos a UTF8 en los casos que corresponda), luego el script pedirá el resto de los parámetros obligatorios según el sistema.

8.2.5. Script para refrescar la cache de mondrian, saiku y CDA cache [Aporte UNER]



Agradecemos el aporte de este punto a Diego Sposito de UNER

Este script sirve para refrescar tanto la cache de mondrian(cambios en la base de datos), la de saiku(cambio en el esquema SIU-Wichi.xml) y la cache de los CDA(Community Data Access: usados en tableros de Portal de Transparencia y Gerencial).

El script *clear_cache.sh* se encuentra en la carpeta <Paquete_SIU-Wichi>/utilidades/general. Antes de ejecutarlo hay que editarlo y configurar algunas variables (usuario y clave del administrador, url de pentaho).

8.3. Mantenimiento de la base de datos

8.3.1. Limpieza de datos innecesarios

A partir de SIU-Wichi 5.0.0, los datos guardados en DSA, es decir en los esquemas *_dsa, se pueden eliminar luego de la carga de datos. Esto se debe a que para todos los sistemas fuente se hace una copia de las dimensiones del Datawarehouse a DSA antes de comenzar la carga. Tanto los datos de TMP como las tablas de hecho de DSA, no son reutilizados, y se pisan cada vez que se hace una importación.

Por lo tanto los esquemas DSA y los esquemas TMP pueden ser limpiados manualmente a fin de reducir el volumen de la base de datos. Para esto el siguiente script genera las líneas TRUNCATE TABLE adecuadas:

```
select 'TRUNCATE TABLE ' || table_schema || '.' || table_name || ';' as linea
from information_schema.tables i
where i.table_schema in(
    'araucano_tmp', 'araucano_dsa',
    'guarani_tmp', 'guarani_dsa',
    'mapuche_tmp', 'mapuche_dsa',
    'pilaga_tmp', 'pilaga_dsa',
    'rhun_tmp', 'rhun_dsa')
order by table_schema, table_name
```

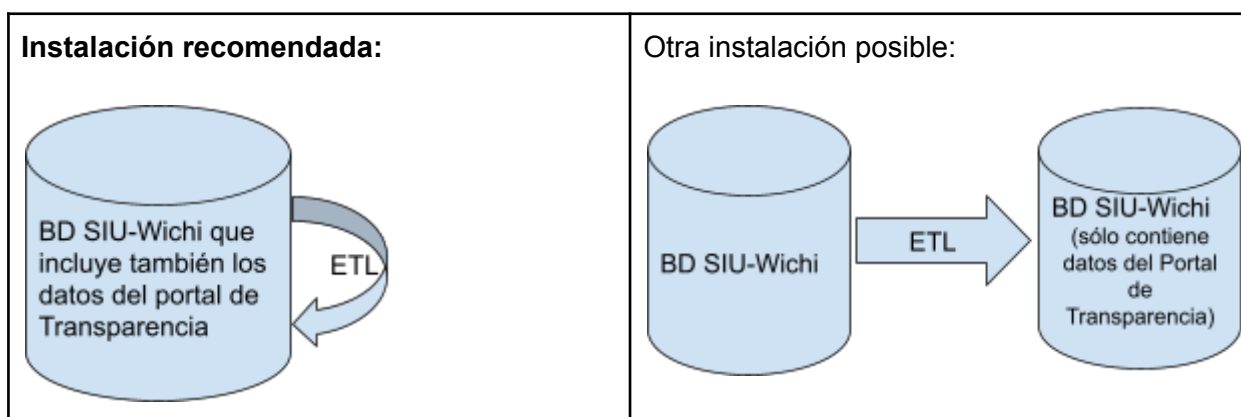
9. Portal de transparencia

A partir de la versión de SIU-Wichi 6.6.0 se incorporó un “Portal de Transparencia” que es un tablero que puede ser configurado como de acceso público (sin formulario de login) y posibilita visualizar información sumariada de distintas fuentes de datos existentes en la base de datos SIU-Wichi, utilizando los esquemas de base de datos portal_transparencia y portal_transparencia_dsa. El objeto de este tablero es que la Universidad muestre información accesible directamente desde algún portal web a la comunidad.

9.1. Portal de transparencia - consideraciones técnicas

Las siguientes son algunas consideraciones técnicas a tener en cuenta para la puesta en producción de este portal:

- Todos los datos de este portal provienen de la base de datos SIU-Wichi y son procesados mediante un ETL que se mostrará a continuación.
- Recomendamos que los datos se alojen directamente en la misma base de datos del SIU-Wichi pero si desea instalar otra base de datos e inclusive otro Pentaho para este portal no existen restricciones, el ETL permite tomar los datos de la base SIU-Wichi y alojarlos en la nueva. Es importante destacar que esta nueva base de datos tendrá **exactamente la misma estructura que la base de datos SIU-Wichi** pero contendrá únicamente los datos de este portal. Ambas bases deberían **mantenerse siempre actualizadas en las mismas versiones**.



9.2. Portal de transparencia - instalación

9.2.1. Instalación de Base de Datos

Para la primer opción (recomendada) asumimos que ya posee una base de SIU-Wichi y por lo tanto en esta versión ya la ha actualizado. Esta base ya posee todas las tablas necesarias para utilizar el portal.

Si por alguna razón desea que la información del portal de transparencia esté alojada en una base de datos distinta puede crear la misma mediante el proceso [3.1. Creación de la base de datos SIU-Wichi](#)

9.2.2. Instalación del Tablero

Dentro de la solución SIU-Wichi se ha incorporado el tablero del Portal de Transparencia, el mismo está alojado en **SIU-Wichi → Portal Transparencia** y se llama **Transparencia**.

Dado que el objeto de este tablero es el acceso público y sin necesidad de tener un usuario de Pentaho se requieren los siguientes pasos para poder incorporarlo en cualquier sitio web que se desee:

1. Habilitar el acceso a una página incorporando usuario y clave mediante get

1. edite el archivo:	biserver-ce/pentaho-solutions/system/security.properties
2. Asigne	requestParameterAuthenticationEnabled=true
3. Baje el servidor Pentaho y súbalo de nuevo.	

2. Crear un usuario, por ejemplo "usuario_transparencia" con clave "clave_transparencia"
3. Acceder al tablero Transparencia mediante la opción "Open in a new window".
4. Copiar el link que figura en la ventana y agregarle al final el siguiente texto:

?userid=usuario_transparencia&password=clave_transparencia

(Con el nombre de usuario y clave que haya generado)

Si además le interesa que el link abra directamente un tablero distinto al primero de presupuesto puede agregar al final de la url el texto &tablero=nn (donde nn es el número de tablero que desea que abra al inicio, es un número entre 1 y 26, 221, 241 y 261).

5. Con todo ese link generado abrir una solapa de incógnito y verificar que esta URL abre el tablero sin necesidad de registrarse y no muestra el entorno de Pentaho. Este link es el que podrá pegar en cualquier sitio web desde el que desea que se tenga acceso.

6. Además para que la página funcione correctamente deberá configurar la fuente de datos que la misma requiere, el nombre de la fuente de datos (Connection Name) debe ser “transparencia” y si ha optado por utilizar la misma base de datos que ya posee de SIU-Wichi lo único que tiene que hacer es crear esta fuente conectada a la conexión JNDI existente llamada DBWichi mediante este [paso](#). Si ha optado por tener dos bases separadas deberá crear la conexión “transparencia” siguiendo el siguiente [paso](#).

Database Connection

General
Advanced
Options
Pooling

Connection Name:
transparencia

Database Type:
Generic database
H2
Hadoop Hive 2
Hadoop Hive
Hypersonic
Impala
Cloudera Impala
MS SQL Server
MonetDB

Adding Databases

Access:
Native (JDBC)
ODBC
JNDI

Settings
JNDI Name:
DBWichi

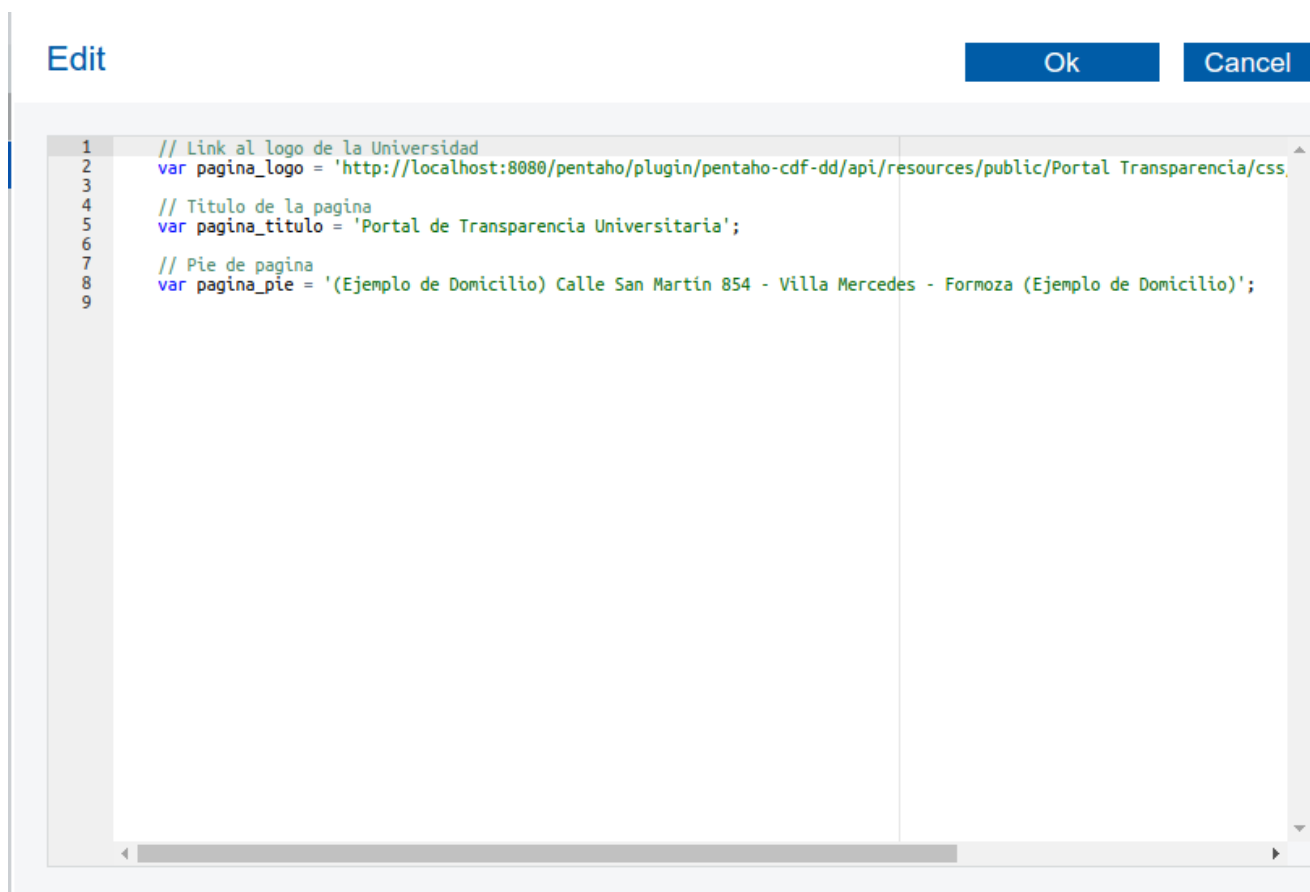
Test

OK Cancel

9.2.3. Personalización del Tablero

Si bien el tablero se entrega con los fuentes y puede ser totalmente personalizado, inicialmente se habilitan tres elementos para la personalización básica y que no requieren intervenir en el código, los pasos para esta personalización son:

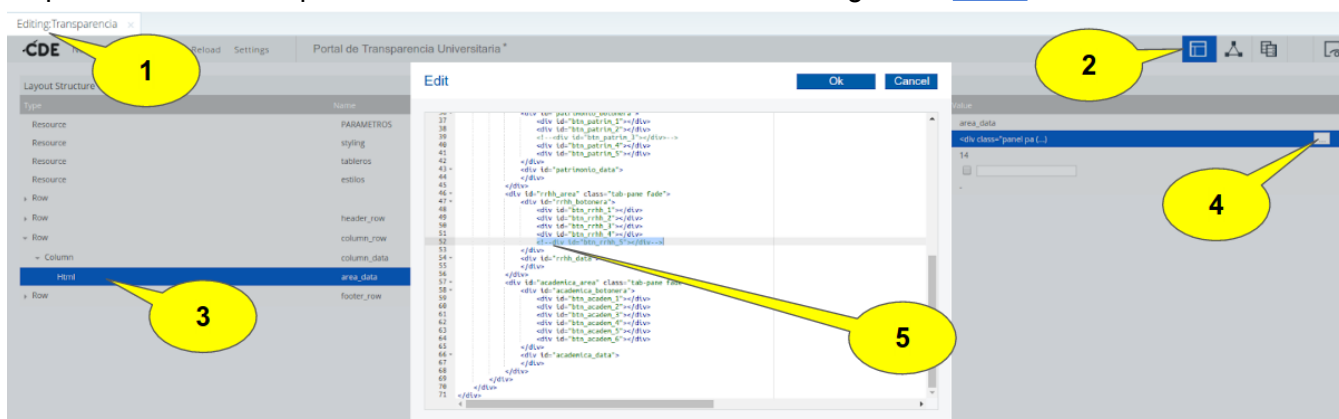
1. Abrir el Tablero con la opción “Edit”.
2. Dentro del “Layout Structure” están los “Resouces” y el primero se llama PARAMETROS, éste debe abrirlo haciendo click sobre él.
3. A la derecha aparecerá una entrada “Resouce code” y más a la derecha un botón con , presione sobre él.
4. Se abrirá el formulario que se muestra a continuación en el que deberá cargar la imagen del título (el nombre y logo de su Universidad), luego el título y finalmente el pie de la página con su dirección postal o información que crea de interés.



5. Luego de incorporar los datos que desea presiona el botón Ok y finalmente presiona el link “Save” de la pantalla principal. Ya debería tener este tablero personalizado.

9.2.3.1. Personalización del Tablero - Ocultar pestaña

Es posible ocultar una pestaña o toda un área del tablero de la siguiente [forma](#):



Los pasos son:

1. Abrir el pentaho con usuario administrador y entrar al portal y **Editarlo**.
2. Presionar el botón de **Layout**.
3. Seleccionar el componente html (**area_data**)
4. Presionar los tres puntos suspensivos a la derecha para acceder al código.
5. Comentar el botón que se desea inhabilitar, en este caso cambiar:

`<div id="btn_rrhh_5"></div>`

por:

`<!--div id="btn_rrhh_5"></div-->`

9.2.3.2. Personalización del Tablero - Cambiar el orden de las pestañas

Al igual que se ve en el punto anterior, accediendo al mismo código, se podrá observar:

```
<li id='li_presupuesto' onclick...../li>
<li id='li_compras' onclick...../li>
<li id='li_patrimonio' onclick...../li>
<li id='li_rrhh' onclick...../li>
<li id='li_academica' onclick...../li>
```

Este orden secuencial marca el orden de las solapas de izquierda a derecha por lo tanto si se desea cambiar el mismo sólo hay que cambiar de ubicación de cada <li.....> en este código.

9.2.3.3. Personalización del Tablero - Adecuación estética

El tablero se entrega con una estética predefinida pero la misma se puede adecuar mucho en base a la configuración existente en el archivo css/estilos.css pudiendo en su Institución modificar la gráfica simplemente reemplazando ese archivo a los fines de que no impacte en el comportamiento del tablero. De esta forma recomendamos re-escribir el archivo estilos.css y guardar una copia para poder recuperar la estética cuando se aplique una nueva versión mediante la recuperación de este archivo con la estética que desean.

9.3. Portal de transparencia - carga de datos

Cada vez que deseamos actualizar los datos del portal de transparencia hay que tener en cuenta que los mismos serán obtenidos de la base SIU-Wichi y por lo tanto ésta deberá estar actualizada previamente a la fecha que se desee, una vez que se tenga esa certeza la forma de incorporar los datos en el portal es mediante la ejecución del siguiente ETL:

```
$ cd <path_data_integration>
$ ./kitchen.sh \
  -file="<path_etl>/carga_de_datos/transparencia/main_carga_datos_portal_transpar
  encia.kjb" \
  -param="wichi_host=localhost" \
  -param="wichi_puerto=5432" \
  -param="wichi_base=siu_wichi" \
  -param="wichi_usuario=postgres" \
  -param="wichi_clave=postgres" \
  -param="transparencia_host=localhost" \
  -param="transparencia_puerto=5432" \
  -param="transparencia_base=siu_wichi" \
  -param="transparencia_usuario=postgres" \
  -param="transparencia_clave=postgres"
```

Todos los textos en negrita deben ser reemplazados por los valores correspondientes a su instalación. Los diez parámetros que posee el ETL corresponden a cinco parámetros para cada conexión a Base de Datos, una conexión es al Wichi que contiene los datos y la otra es a la base que contiene los datos de Transparencia. En este caso las cinco entradas correspondientes a "wichi_..." son iguales a las cinco "transparencia_..." debido a que los datos coexisten en la misma base (recomendado), si

fuesen dos bases distintas habrá dos definiciones diferentes. La fecha de esta última actualización figuraría entre paréntesis a la derecha del pie de página del Portal.

Luego de correr este etl, se debe actualizar CDA cache desde el menú Herramientas → Actualizar → CDA Cache o bien configurar el script clear_cache.sh para que se ejecute automatico([sección 8.2.5](#))

10. Solución de errores

En este capítulo no se explican soluciones a errores comunes en la instalación.

10.1. Tablero del Portal Gerencial arroja No Data

Si al ejecutar un tablero del Portal gerencial arroja “No Data”, se debe posicionar sobre la solapa del tablero, hacer clic derecho y seleccionar la opción Recargar.

