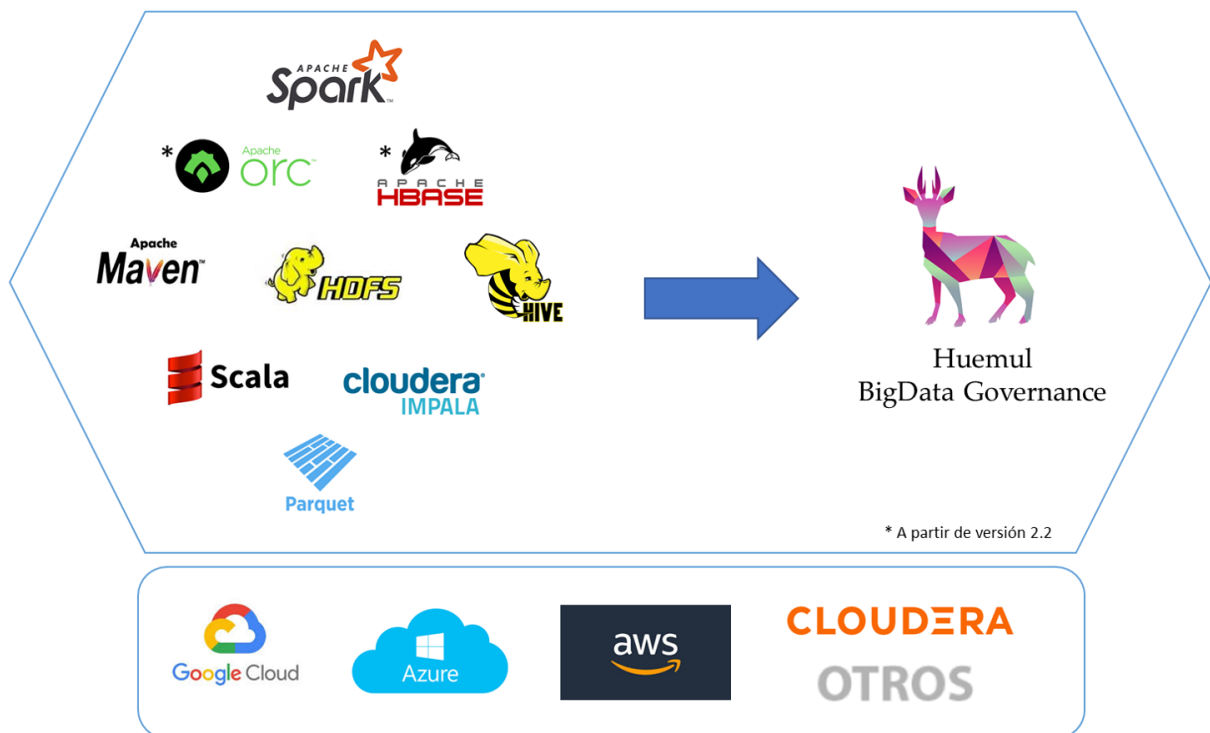


Guía para la ejecución de Huemul con datos del Sistema Financiero Chileno en Cluster HortonWorks

Desarrolla las ingestas de datos con estándares productivos, acelerando la entrega de datos en la **PRIMERA MILLA**



WebSite

www.HuemulSolutions.com

GitHub

<https://github.com/HuemulSolutions/Bigdata-Governance>

Maven <https://mvnrepository.com/artifact/com.huemulsolutions.bigdata/huemul-bigdatagovernance>

Introducción

Los datos del sistema financiero permiten visualizar el estado de resultados y balance de los chilenos a cierta fecha. Este ejemplo permite cargar estos archivos al cluster BigData, estructurarlos y dejarlos disponibles en tablas sobre Hive.

Para ejecutar este ejemplo, se han creado tres ficheros que contienen todo lo necesario, se explican a continuación:

- **demo-sbif.zip:** contiene los datos del proyecto, y archivo ejemplo de configuración para connectionString del modelo de control. Los datos que contiene son los siguientes:
 - NEGOCIO_201806: Datos de negocios identificados para esta demo
 - PRODUCTO_201806: Datos de productos permitidos para esta demo
 - PLANCUENTA_GESTION_001: Asignación de productos y negocios a las cuentas contables.
 - 201806: Carpeta que contiene los datos del sistema financiero para junio del 2018
 - 201807: Carpeta que contiene los datos del sistema financiero para julio del 2018
- **huemul_install_sbif.sh:** Shell para crear directorios en HDFS y copiar archivo RAW.
- **huemul_drivers.zip:** Drivers utilizados por Huemul, contiene lo siguiente:
 - huemul-bigdatagovernance-2.6.1.jar: Framework Huemul BigData
 - huemul-sql-decode-1.0.jar: Componente de Huemul para obtener traza
 - postgresql-9.4.1212.jar: Driver para conexión con modelo de control implementado en PostgreSQL
 - ojdbc7.jar: Driver para conexión con modelo de control implementado en Oracle
 - mysql-connector-java-5.1.48.jar: Driver para conexión con modelo de control implementado en MySQL
 - mssql-jdbc-7.2.1.jre8.jar: Driver para conexión con modelo de control implementado en SQL Server.

Los ficheros pueden ser descargados desde el siguiente link:

www.huemulsolutions.com/ejemplos/sbif_26/demo-sbif.zip

www.huemulsolutions.com/ejemplos/sbif_26/huemul_install_sbif.sh

www.huemulsolutions.com/ejemplos/sbif_26/huemul-drivers.zip

Para efectos de este ejemplo, los códigos se han ejecutado sobre un ambiente Hortonworks versión 2.6.5 en modo Sandbox, por tanto, los tiempos de ejecución no son representativos.

A continuación, se explican los pasos necesarios para subir los datos, ejecutar los procesos y visualizar los resultados.

Contexto

Este ejemplo permite ver las distintas opciones de ingestas de datos, y la forma en que Huemul permite la implementación de forma muy sencilla:

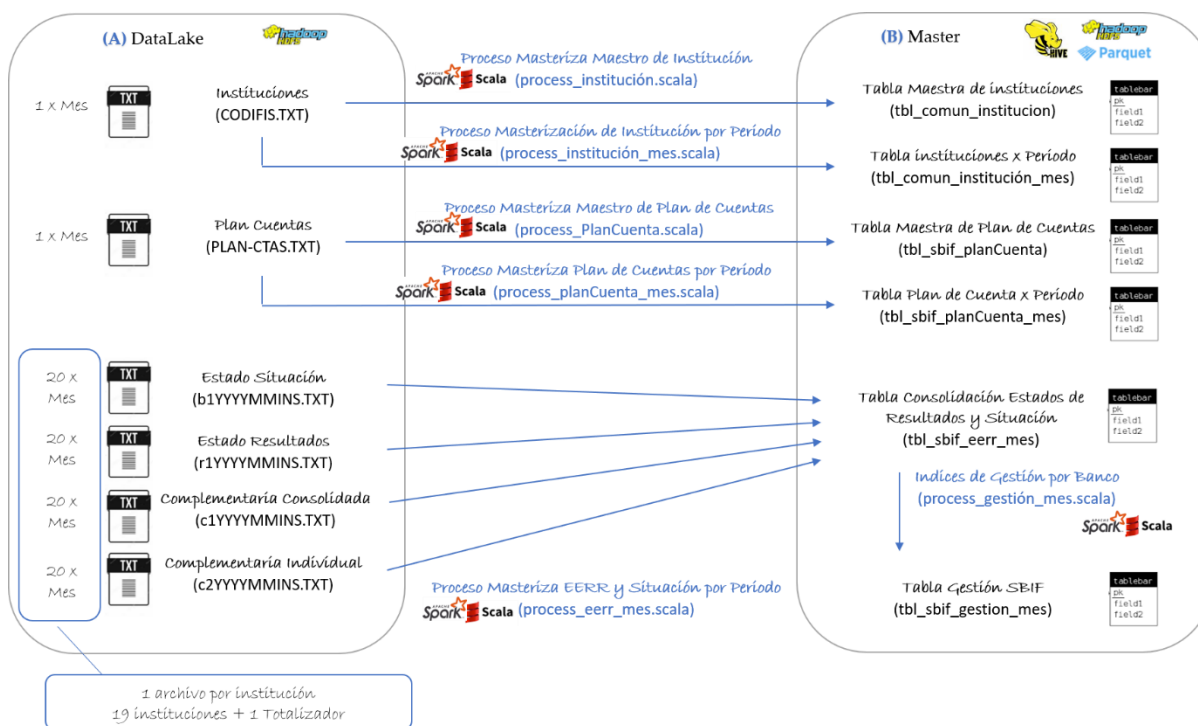


figura 1.1. Estructura de carpetas base en HDFS

Tal como se muestra en la imagen anterior, hay múltiples escenarios para ingestar archivos:

- **una interfaz carga una tabla maestra:** una interfaz (por ejemplo CODIFIS.TXT) se carga sobre una tabla (tbl_comun_institucion). Este ejemplo carga una tabla maestra, es decir, los datos nuevos sobrescriben los datos antiguos.
- **una interfaz carga una tabla transaccional:** a diferencia del ejemplo anterior, la tabla de destino es una tabla transaccional, eso significa que tiene una columna de periodo. Los datos quedan como una foto mes a mes.
- **Muchas interfaces cargan una tabla transaccional:** en este ejemplo se pueden ver como múltiples interfaces se procesan para dejar una sola tabla transaccional. Para ver este ejemplo debes ver el código “process_eerr_mes.scala”
- **Tablas que cargan tablas:** es común que luego de ingestar datos, estos sean procesados para generar nuevos datos. El proceso “process_gestion_mes.scala” permite ver la forma de tomar tablas ya existentes para crear nuevas tablas con nueva información.

Ejecutar Script de ambientación en Clúster

Lo primero que se requiere es subir los archivos de ambientación al clúster, para ello debes seguir los siguientes pasos:

Abrir shell y dar acceso de ejecución sobre el fichero huemul_install_sbif.sh

```
chmod 775 huemul_install_sbif.sh
```

Luego, ejecuta el script:

```
./huemul_install_sbif.sh
```

Este comando tomará un poco de tiempo, debido a que realiza los siguientes pasos:

- creación de estructuras en HDFS
- creación de bases de datos en HIVE
- descomprimir archivos de datos
- copiar archivos de datos en rutas RAW
- descomprimir dependencias requeridas.

Una vez que el script de ambientación ha terminado correctamente, podrás ver las estructuras creadas en el path user/data/production:

/ > user > data > production

Total: 10 files or folders

+ Select All

+ New Folder

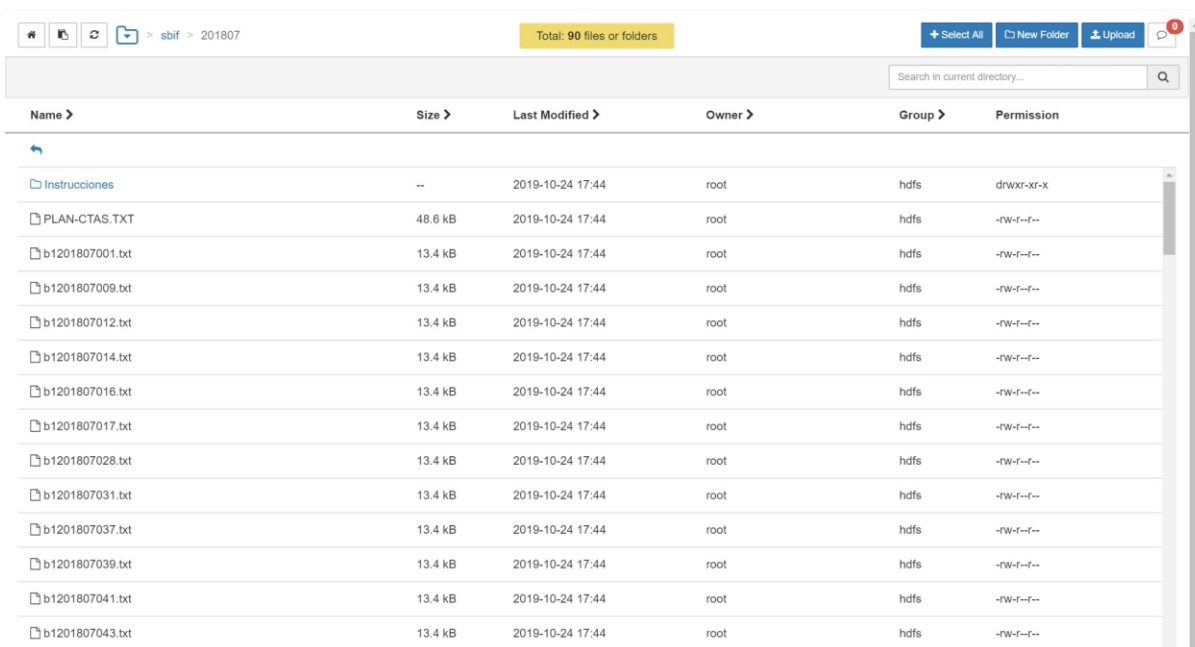
Upload

Search in current directory...

Name >	Size >	Last Modified >	Owner >	Group >	Permission
<div></div>					
<div>analytics</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x
<div>backup</div>	--	2019-10-24 17:40	root	hdfs	drwxr-xr-x
<div>dim</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x
<div>dqerror</div>	--	2019-10-24 17:40	root	hdfs	drwxr-xr-x
<div>master</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x
<div>mdm_oldvalue</div>	--	2019-10-24 17:40	root	hdfs	drwxr-xr-x
<div>raw</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x
<div>reporting</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x
<div>sandbox</div>	--	2019-10-24 17:40	root	hdfs	drwxr-xr-x
<div>temp</div>	--	2019-10-24 17:39	root	hdfs	drwxr-xr-x

figura 2.1. Estructura de carpetas base en HDFS

En el path /user/data/production/raw/sbif/201807



Name >	Size >	Last Modified >	Owner >	Group >	Permission
Instrucciones	--	2019-10-24 17:44	root	hdfs	drwxr-xr-x
PLAN-CTAS.TXT	48.6 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807001.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807009.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807012.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807014.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807016.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807017.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807028.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807031.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807037.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807039.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807041.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--
b1201807043.txt	13.4 kB	2019-10-24 17:44	root	hdfs	-rw-r--r--

figura 2.2. ficheros del sistema financiero, con fecha julio del 2018.

Finalmente, debes incluir los archivos JAR compilados en los proyectos demo:

- **demo_settings-2.6.1.jar:** Contiene los path y connectionString globales para todos los proyectos.
 - Este proyecto usa los siguientes archivos paramétricos:
 - **prod-demo-setting-control-connection:** Contiene el connectionString para el modelo de control
 - **prod-demo-setting-impala-connection:** Contiene el connectionString para conectar con Impala.
- **demo_catalogo-2.6.1.jar:** Carga los datos de productos y negocios. Estas interfaces fueron creadas manualmente como ejemplo para esta demo.
- **demo_sbif-2.6.1.jar:** Carga los datos descargados desde el sitio web de la CMF.

Puedes descargar estos binarios desde los siguientes links:

www.huemulsolutions.com/ejemplos/sbif_261/demo_settings-2.6.1.jar
www.huemulsolutions.com/ejemplos/sbif_261/demo_catalogo-2.6.1.jar
www.huemulsolutions.com/ejemplos/sbif_261/demo_sbif-2.6.1.jar

También puedes descargar el código fuente desde GitHub y compilar el código tú mismo:

[git@github.com:HuemulSolutions/demo_settings.git](https://github.com/HuemulSolutions/demo_settings.git)
[git@github.com:HuemulSolutions/demo_catalogo.git](https://github.com/HuemulSolutions/demo_catalogo.git)
[git@github.com:HuemulSolutions/demo_sbif.git](https://github.com/HuemulSolutions/demo_sbif.git)

Ejecutar Script de Configuración de Modelo de Control.

Huemul requiere de un modelo de control, el cual registra toda la actividad de Huemul, tanto a nivel de procesos (ejecuciones, estado, etc), como también de los metadatos de las interfaces leídas y las tablas creadas.

El modelo de control puede ser implementado en los siguientes motores de base de datos:

- PostgreSQL
- MySQL
- SQL Server
- Oracle

Para efectos de esta guía, utilizaremos una base de datos PostgreSQL versión 10.0 que será creada sobre Azure.



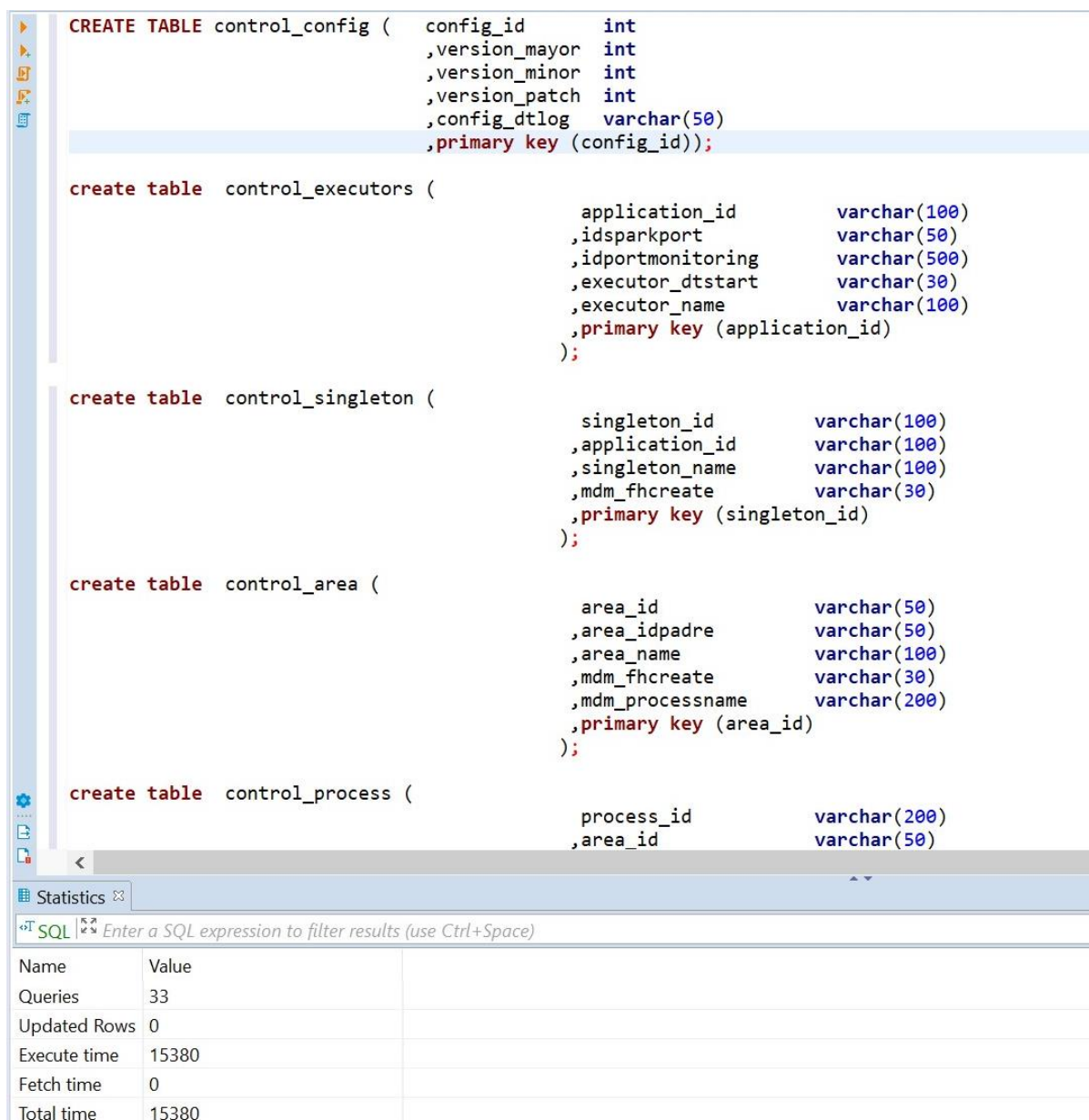
figura 4.1. Creación de Servidor PostgreSQL en Azure.

Una vez creado el servidor de base de datos PostgreSQL, creamos una base de datos llamada "huemul_control". (NOTA: revisa que la IP del clúster pueda acceder al servidor de BBDD).

Luego, debes crear las tablas del modelo de control, para ello debes descargar el script con la versión de Huemul que estés usando (en nuestro caso la versión 2.6.1), el link es el siguiente:

https://github.com/HuemulSolutions/huemul-bigdatagovernance/blob/master/src/main/resources/Instalacion/huemul_bdg_2.6.1_minor.sql

En nuestro caso utilizamos dBeaver para trabajar con la base de datos y ejecutar el script.



```

CREATE TABLE control_config (
    config_id          int
    ,version_mayor     int
    ,version_minor     int
    ,version_patch     int
    ,config_dtlog      varchar(50)
    ,primary key (config_id));

create table control_executors (
    application_id      varchar(100)
    ,idsparkport        varchar(50)
    ,idportmonitoring   varchar(500)
    ,executor_dtstart   varchar(30)
    ,executor_name      varchar(100)
    ,primary key (application_id)
);

create table control_singleton (
    singleton_id        varchar(100)
    ,application_id     varchar(100)
    ,singleton_name     varchar(100)
    ,mdm_fhcreate       varchar(30)
    ,primary key (singleton_id)
);

create table control_area (
    area_id             varchar(50)
    ,area_idpadre       varchar(50)
    ,area_name          varchar(100)
    ,mdm_fhcreate       varchar(30)
    ,mdm_processname    varchar(200)
    ,primary key (area_id)
);

create table control_process (
    process_id          varchar(200)
    ,area_id            varchar(50)

```

Name	Value
Queries	33
Updated Rows	0
Execute time	15380
Fetch time	0
Total time	15380

figura 4.2. Script SQL ejecutado correctamente

Finalmente, debes editar el archivo “prod-demo-setting-control-connection.set” para configurar el connectionString correspondiente a la conexión, este archivo permite que Huemul se comuniquen con la base de datos PostgreSQL.

```
jdbc:postgresql://{{000.000.000.000}}:5432/{{database_name}}?user={{user_name}}&password={{password}}&currentSchema=public
```

figura 4.3. Ejemplo connectionString para modelo de control.

Ejecución del código

Ahora que ya están ambientados los componentes, podemos ejecutar los códigos en spark usando huemul:

Se deben ejecutar en el siguiente orden:

- **com.yourcompany.catalogo.process_negocio**: carga los datos de negocios.
- **com.yourcompany.catalogo.process_producto**: carga los datos de productos.
- **com.yourcompany.sbif.process_mensual**: carga los siguientes datos:
 - **process_institucion**: carga datos de instituciones en tabla maestra. Esta ingesta permite ver el tratamiento de datos semi-estructurados.
 - **process_institucion_mes**: carga datos de instituciones con foto mensual
 - **process_planCuenta**: carga datos del plan de cuentas en tabla maestra
 - **process_planCuenta_mes**: carga datos del plan de cuentas con foto mensual
 - **process_eerr_mes**: carga los datos de estados de resultados y balances de todos los bancos en forma mensual

Ejecución process_negocio

Se ejecuta con el siguiente comando:

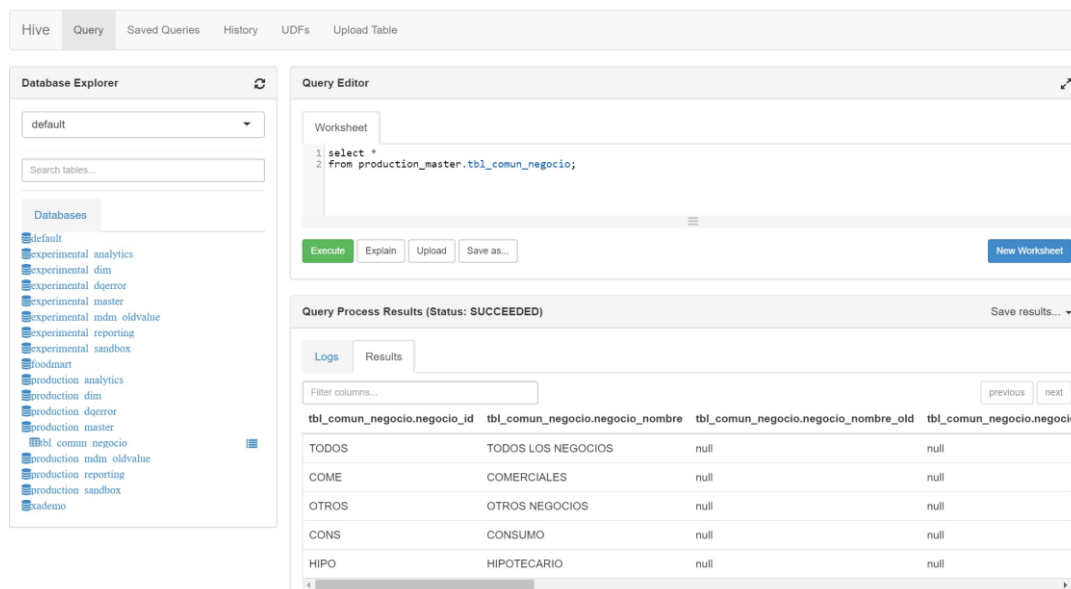
```
spark-submit --master local --jars huemul-bigdatagovernance-2.6.1.jar,huemul-sql-decode-1.0.jar,demo_settings-2.6.1.jar,demo_catalogo-2.6.1.jar,postgresql-9.4.1212.jar --class com.yourcompany.catalogo.process_negocio demo_sbif-2.6.1.jar environment=production,ano=2018,mes=6
```

El resultado de esta ejecución es el siguiente:

```
19/10/24 20:59:02 INFO huemulsolutions: processName: com.yourcompany.catalogo.datalake.raw_negocio_mes, ProcessExec_Id: 2019102420590200243004,
05832040288
19/10/24 20:59:02 INFO huemulsolutions: Step: Start
19/10/24 20:59:02 INFO huemulsolutions: Step: Abre archivo RDD y devuelve esquemas para transformar a DF
19/10/24 20:59:02 INFO huemulsolutions: Reading File: /user/data/production/raw/catalogo/NEGOCIO_201806.txt
19/10/24 20:59:02 INFO huemulsolutions: 2 first line example of file: /user/data/production/raw/catalogo/NEGOCIO_201806.txt
19/10/24 20:59:06 INFO huemulsolutions: CONS:CONSUMO
19/10/24 20:59:06 INFO huemulsolutions: HIPO:HIPOTECARIO
19/10/24 20:59:06 INFO huemulsolutions: N° Columns in RowFile: 2
19/10/24 20:59:06 INFO huemulsolutions: Step: Aplicando Filtro
19/10/24 20:59:06 INFO huemulsolutions: Step: Transformando datos a dataframe
19/10/24 20:59:07 INFO huemulsolutions: Step: Valida que cantidad de registros esté entre 10 y 100
19/10/24 20:59:14 INFO huemulsolutions: FINISH processName: com.yourcompany.catalogo.datalake.raw_negocio_mes, ProcessExec_Id: 201910242059020
024300405832040288, Time Elapsed: 00:00:12
19/10/24 20:59:14 INFO huemulsolutions: HuemulControl: table instance: tbl_comun_negocio
19/10/24 20:59:14 INFO huemulsolutions: Step: Generar Logica de Negocio
19/10/24 20:59:15 INFO huemulsolutions: Step: QUITAR!!! Generar Estadísticas de las columnas SOLO PARA PRIMERA EJECUCION
19/10/24 21:00:10 INFO huemulsolutions: Step: Asocio columnas de la tabla con nombres de campos de SQL
19/10/24 21:00:10 INFO huemulsolutions: Step: Ejecuta Proceso
19/10/24 21:00:10 INFO huemulsolutions: processName: com.huemulsolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 201910242100100066501755,
011140288
19/10/24 21:00:10 INFO huemulsolutions: Step: Start
19/10/24 21:00:10 INFO huemulsolutions: Step: Ref & Master: Select distinct
19/10/24 21:00:10 INFO huemulsolutions: Step: Ref & Master: Select Old Table
19/10/24 21:00:10 INFO huemulsolutions: create empty dataframe because production_master.tbl_comun_negocio does not exist
19/10/24 21:00:10 INFO huemulsolutions: Step: Ref & Master: Full Join
19/10/24 21:00:11 INFO huemulsolutions: Step: Ref & Master: Update & Insert Logic
19/10/24 21:00:12 INFO huemulsolutions: Step: Ref & Master: Hash Code
19/10/24 21:00:13 INFO huemulsolutions: Step: Ref & Master: Statistics
19/10/24 21:00:31 INFO huemulsolutions: Step: Ref & Master: Validating Insert & Update
19/10/24 21:00:31 INFO huemulsolutions: Step: Ref & Master: Final Table
19/10/24 21:00:32 INFO huemulsolutions: Step: Ref & Master: MDM Old Value Full Trace
19/10/24 21:00:32 INFO huemulsolutions: Step: Start DataQuality
19/10/24 21:00:53 INFO huemulsolutions: Step: Start ForeignKey
19/10/24 21:00:53 INFO huemulsolutions: Step: Validating errors
19/10/24 21:00:53 INFO huemulsolutions: Step: Compare Schema Final DF
19/10/24 21:00:54 INFO huemulsolutions: Step: Start Save
19/10/24 21:00:54 INFO huemulsolutions: Step: Save: Drop ActionType column
19/10/24 21:00:54 INFO huemulsolutions: Step: Save: Set num FileParts
19/10/24 21:00:54 INFO huemulsolutions: Step: Save: Overwrite Master & Ref Data
19/10/24 21:01:06 INFO huemulsolutions: Step: Save: Drop Hive table Def
19/10/24 21:01:07 INFO huemulsolutions: Step: Save: Create Table in Hive Metadata
19/10/24 21:01:08 INFO huemulsolutions: Step: Register Master Information
19/10/24 21:01:08 INFO huemulsolutions: FINISH processName: com.huemulsolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 20191024210010006
6501755011140288, Time Elapsed: 00:00:57
19/10/24 21:01:08 INFO huemulsolutions: FINISH ALL OK
19/10/24 21:01:08 INFO huemulsolutions: FINISH processName: com.yourcompany.catalogo.process_negocio, ProcessExec_Id: 201910242059020011500156,
600340288, Time Elapsed: 00:02:06
[root@sandbox-hdp ~]#
```

figura 3.1. Ejecución

La siguiente imagen muestra la tabla cargada con los datos:



Query Editor

```
1 select *
2 from production_master.tbl_comun_negocio;
```

Query Process Results (Status: SUCCEEDED)

Logs Results

Filter columns...

tbl_comun_negocio.negocio_id	tbl_comun_negocio.negocio_nombre	tbl_comun_negocio.negocio_nombre_old	tbl_comun_negocio.negocio_nombre
TODOS	TODOS LOS NEGOCIOS	null	null
COME	COMERCIALES	null	null
OTROS	OTROS NEGOCIOS	null	null
CONS	CONSUMO	null	null
HIPO	HIPOTECARIO	null	null

figura 3.2. Resultados de la ejecución del process_negocio

Ejecución process_producto

Se ejecuta con el siguiente comando:

```
spark-submit --master local --jars huemul-bigdatagovernance-2.6.1.jar,huemul-sql-decode-1.0.jar,demo_settings-2.6.1.jar,demo_catalogo-2.6.1.jar,postgresql-9.4.1212.jar --class com.yourcompany.catalogo.process_producto demo_sbif-2.6.1.jar environment=production,ano=2018,mes=6
```

El resultado de esta ejecución es el siguiente:

```

19/10/24 21:20:13 INFO huemulolutions: Step: Abre DataLake
19/10/24 21:20:13 INFO huemulolutions: processName: com.yourcompany.catalogo.dataLake.raw_producto_mes, ProcessExec_Id: 201910242120130092900
486580645509
19/10/24 21:20:13 INFO huemulolutions: Step: Start
19/10/24 21:20:13 INFO huemulolutions: Step: Abre archivo RDD y devuelve esquemas para transformar a DF
19/10/24 21:20:13 INFO huemulolutions: Reading File: /user/data/production/raw/catalogo/PRODUCTO_201806.txt
19/10/24 21:20:14 INFO huemulolutions: 2 first line example of file: /user/data/production/raw/catalogo/PRODUCTO_201806.txt
19/10/24 21:20:19 INFO huemulolutions: CUOTA PRESTAMOS EN CUOTAS
19/10/24 21:20:19 INFO huemulolutions: CTACTE CUENTA CORRIENTE
19/10/24 21:20:19 INFO huemulolutions: N° Columns in RowFile: 2
19/10/24 21:20:19 INFO huemulolutions: Step: Aplicando Filtro
19/10/24 21:20:19 INFO huemulolutions: Step: Transformando datos a dataframe
19/10/24 21:20:19 INFO huemulolutions: Step: Valida que cantidad de registros esté entre 2 y 100
19/10/24 21:20:31 INFO huemulolutions: FINISH processName: com.yourcompany.catalogo.dataLake.raw_producto_mes, ProcessExec_Id: 20191024212013
0092900486580645509, Time Elapsed: 00:00:17
19/10/24 21:20:31 INFO huemulolutions: HueMulControl: table instance: tbl_comun_producto
19/10/24 21:20:31 INFO huemulolutions: Step: Generar Logica de Negocio
19/10/24 21:20:32 INFO huemulolutions: Step: Asocia columnas de la tabla con nombres de campos de SQL
19/10/24 21:20:32 INFO huemulolutions: Step: Ejecuta Proceso
19/10/24 21:20:32 INFO huemulolutions: processName: com.huemulolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 201910242120320033301624
471145509
19/10/24 21:20:32 INFO huemulolutions: Step: Start
19/10/24 21:20:32 INFO huemulolutions: Step: Ref & Master: Select distinct
19/10/24 21:20:32 INFO huemulolutions: Step: Ref & Master: Select Old Table
19/10/24 21:20:32 INFO huemulolutions: create empty dataframe because production_master.tbl_comun_producto does not exist
19/10/24 21:20:32 INFO huemulolutions: Step: Ref & Master: Full Join
19/10/24 21:20:33 INFO huemulolutions: Step: Ref & Master: Update & Insert Logic
19/10/24 21:20:33 INFO huemulolutions: Step: Ref & Master: Hash Code
19/10/24 21:20:34 INFO huemulolutions: Step: Ref & Master: Statistics
19/10/24 21:21:11 INFO huemulolutions: Step: Ref & Master: Validating Insert & Update
19/10/24 21:21:11 INFO huemulolutions: Step: Ref & Master: Final Table
19/10/24 21:21:11 INFO huemulolutions: Step: Ref & Master: MDM Old Value Full Trace
19/10/24 21:21:11 INFO huemulolutions: Step: Start DataQuality
19/10/24 21:21:44 INFO huemulolutions: Step: Start ForeignKey
19/10/24 21:21:44 INFO huemulolutions: Step: Validating errors
19/10/24 21:21:44 INFO huemulolutions: Step: Compare Schema Final DF
19/10/24 21:21:44 INFO huemulolutions: Step: Start Save
19/10/24 21:21:44 INFO huemulolutions: Step: Save: Drop ActionType column
19/10/24 21:21:44 INFO huemulolutions: Step: Save: Set num FileParts
19/10/24 21:21:44 INFO huemulolutions: Step: Save: Overwrite Master & Ref Data
19/10/24 21:22:03 INFO huemulolutions: Step: Save: Drop Hive table Def
19/10/24 21:22:04 INFO huemulolutions: Step: Save: Create Table in Hive Metadata
19/10/24 21:22:05 INFO huemulolutions: Step: Register Master Information
19/10/24 21:22:05 INFO huemulolutions: FINISH processName: com.huemulolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 20191024212032003
3301624471145509, Time Elapsed: 00:01:33
19/10/24 21:22:05 INFO huemulolutions: FINISH ALL OK
19/10/24 21:22:05 INFO huemulolutions: FINISH processName: com.yourcompany.catalogo.process_producto, ProcessExec_Id: 20191024212013008770016
0202145509, Time Elapsed: 00:01:51
[root@sandbox-hdp ~]#

```

figura 3.3. Ejecución de process_producto

La siguiente imagen muestra la tabla cargada con los datos:

DFs Upload Table

Query Editor

Worksheet

```

1 select *
2 from production_master.tbl_comun_producto;

```

Execute

Explain

Upload

Save as...

New Worksheet

Query Process Results (Status: SUCCEEDED)

Save results...

Logs

Results

Filter columns...

previous next

tbl_comun_producto.producto_id	tbl_comun_producto.producto_nombre	tbl_comun_producto.producto_nombre_old	tbl_comun_producto.
TODOS	TODOS LOS PRODUCTOS	null	2019-10-24 21:21:45.9
DEPOSITO	DEPOSITOS A PLAZO	null	2019-10-24 21:21:45.9
LINCRED	LINEA DE CREDITO	null	2019-10-24 21:21:45.9
OTROS	OTROS PRODUCTOS	null	2019-10-24 21:21:45.9
CTAAHORRO	CUENTAS DE AHORRO	null	2019-10-24 21:21:45.9
TDC	TARJETA DE CREDITO	null	2019-10-24 21:21:45.9
CTACTE	CUENTA CORRIENTE	null	2019-10-24 21:21:45.9

figura 3.4. Resultados de la ejecución del process_producto

Ejecución process_mensual

Se ejecuta con el siguiente comando:

```
spark-submit --master local --jars huemul-bigdatagovernance-2.6.1.jar,huemul-sql-decode-1.0.jar,demo_settings-2.6.1.jar,demo_catalogo-2.6.1.jar,postgresql-9.4.1212.jar --class com.yourcompany.sbif.process_mensual demo_sbif-2.6.1.jar environment=production,ano=2018,mes=6
```

El resultado de esta ejecución es el siguiente:

```
20
19/10/24 21:40:56 INFO huemulsolutions: Step: Start
19/10/24 21:40:56 INFO huemulsolutions: Step: Abre archivo RDD y devuelve esquemas para transformar a DF
19/10/24 21:40:56 INFO huemulsolutions: Reading File: /user/data/production/raw/sbif/201806/c2201806999.txt
19/10/24 21:40:56 INFO huemulsolutions: 2 first line example of file: /user/data/production/raw/sbif/201806/c2201806999.txt
19/10/24 21:40:56 INFO huemulsolutions: 999 SISTEMA FINANCIERO
19/10/24 21:40:56 INFO huemulsolutions: 9910000 2963801313659.00
19/10/24 21:40:56 INFO huemulsolutions: N° Columns in RowFile: 2
19/10/24 21:40:56 INFO huemulsolutions: Step: Aplicando Filtro y leyendo formato
19/10/24 21:40:56 INFO huemulsolutions: Step: Transformando datos a dataframe
19/10/24 21:40:57 INFO huemulsolutions: Step: Valida que cantidad de registros esté entre 1 y 1000
19/10/24 21:40:57 INFO huemulsolutions: Step: Valida que código dentro del archivo sea igual al parámetro recibido
19/10/24 21:40:57 INFO huemulsolutions: FINISH processName: com.yourcompany.sbif.dataLake.raw_C2_mes, ProcessExec_Id: 201910242140560073183632
379443126, Time Elapsed: 00:00:00
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_sbif_eerr_mes
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_comun_institucion_mes
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_comun_institucion
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_sbif_planCuenta_mes
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_sbif_planCuenta
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_comun_negocio
19/10/24 21:40:57 INFO huemulsolutions: HuemulControl: table instance: tbl_comun_producto
19/10/24 21:40:57 INFO huemulsolutions: Step: Generar Logica de Negocio
19/10/24 21:41:00 INFO huemulsolutions: Step: Asocia columnas de la tabla con nombres de campos de SQL
19/10/24 21:41:00 INFO huemulsolutions: Step: Ejecuta Proceso
19/10/24 21:41:00 INFO huemulsolutions: processName: com.huemulsolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 201910242141000033285353
558843126
19/10/24 21:41:00 INFO huemulsolutions: Step: Start
19/10/24 21:41:00 INFO huemulsolutions: Step: Transaction: Validating fields
19/10/24 21:41:00 INFO huemulsolutions: Step: Transaction: Create Hash Field
19/10/24 21:41:00 INFO huemulsolutions: Step: Transaction: Get Statistics info
19/10/24 21:41:16 INFO huemulsolutions: Step: Start DataQuality
19/10/24 21:42:04 INFO huemulsolutions: Step: Start ForeignKey
19/10/24 21:44:03 INFO huemulsolutions: Step: Validating errors
19/10/24 21:44:03 INFO huemulsolutions: Step: Compare Schema Final DF
19/10/24 21:44:03 INFO huemulsolutions: Step: Start Save
19/10/24 21:44:03 INFO huemulsolutions: Step: Save: Validating N° partitions
19/10/24 21:44:26 INFO huemulsolutions: Step: Save: Drop old partition
19/10/24 21:44:26 INFO huemulsolutions: Step: Save: Set num FileParts
19/10/24 21:44:26 INFO huemulsolutions: Step: Save: OverWrite partition with new data
19/10/24 21:44:55 INFO huemulsolutions: Step: Save: Drop Hive table Def
19/10/24 21:44:58 INFO huemulsolutions: Step: Save: Create Table in Hive Metadata
19/10/24 21:44:59 INFO huemulsolutions: Step: Save: Repair Hive Metadata
19/10/24 21:45:02 INFO huemulsolutions: Step: Register Master Information
19/10/24 21:45:02 INFO huemulsolutions: FINISH processName: com.huemulsolutions.bigdata.tables.huemul_Table, ProcessExec_Id: 20191024214100003
3285353558843126, Time Elapsed: 00:04:02
19/10/24 21:45:02 INFO huemulsolutions: FINISH ALL OK
19/10/24 21:45:02 INFO huemulsolutions: FINISH processName: com.yourcompany.sbif.process_eerr_mes, ProcessExec_Id: 201910242139270091900187823
643126, Time Elapsed: 00:05:34
[root@sandbox-hdp ~]#
```

figura 3.5. Ejecución de process_mensual

La siguiente imagen muestra la tabla **tbl_comun_institucion** con los datos cargados:

Worksheet			
<pre>1 select * 2 from production_master.tbl_comun_institucion;</pre>			
<div>Execute Explain Upload Save as...</div> <div>New Worksheet</div>			
Query Process Results (Status: SUCCEEDED)			
Save results...			
Logs Results			
Filter columns...			
tbl_comun_institucion.ins_id	tbl_comun_institucion.ins_nombre	tbl_comun_institucion.mdm_fhnew	tbl_comun_institucion.mdm_processn
009	BANCO INTERNACIONAL	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
053	BANCO RIPLEY	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
028	BANCO BICE	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
012	BANCO DEL ESTADO DE CHILE	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
037	BANCO SANTANDER-CHILE	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
060	CHINA CONSTRUCTION BANK (5)	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci
031	HSBC BANK (CHILE)	2019-10-24 21:28:18.893	com_yourcompany_sbif_process_instituci

figura 3.6. tabla `tbl_comun_institucion`

La siguiente imagen muestra la tabla `tbl_comun_institucion_mes` con los datos cargados:

Worksheet

```
1 select periodo_mes, *
2 from production_master.tbl_comun_institucion_mes;
```

Execute

Explain

Upload

Save as...

New Worksheet

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

Filter columns...

previous

next

periodo_mes	tbl_comun_institucion_mes.ins_id	tbl_comun_institucion_mes.ins_nombre	tbl_comun_institucion_mes.mdm_fhnew	tbl_con
2018-06-01	001	BANCO DE CHILE	2019-10-24 21:30:37.022	com_yo
2018-06-01	009	BANCO INTERNACIONAL	2019-10-24 21:30:37.022	com_yo
2018-06-01	012	BANCO DEL ESTADO DE CHILE	2019-10-24 21:30:37.022	com_yo
2018-06-01	014	SCOTIABANK CHILE (1)	2019-10-24 21:30:37.022	com_yo
2018-06-01	016	BANCO DE CREDITO E INVERSIONES	2019-10-24 21:30:37.022	com_yo
2018-06-01	017	BANCO DO BRASIL S.A.	2019-10-24 21:30:37.022	com_yo
2018-06-01	028	BANCO BICE	2019-10-24 21:30:37.022	com_yo

figura 3.7. tabla `tbl_comun_institucion_mes`

La siguiente imagen muestra la tabla `tbl_sbif_plancuenta` con los datos cargados:

Worksheet

```

1 select *
2 from production_master.tbl_sbif_plancuenta;

```

Execute Explain Upload Save as... New Worksheet

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

Filter columns... previous next

tbl_sbif_plancuenta.plancuenta_id	tbl_sbif_plancuenta.plancuenta_nombre	tbl_sbif_plancuenta.plancuenta_nombre_old	tbl_sbif_plancuenta.plancuenta_nombre
2700200	Provisi?n para dividendos m?nimos	null	null
4250102	Comisiones por operaci?n de tarjetas de d?bito y cajeros	null	null
8145000	Colocaciones de consumo	null	null
4200500	Comisiones por cobranzas, recaudaciones y pagos	null	null
2100300	Otras obligaciones a la vista	null	null
4100430	Intereses de otros cr?ditos para vivienda	null	null
4150301	Intereses por dep?sitos a plazo	null	null
4610500	Impuestos, contribuciones y aportes	null	null
8315000	Boletas de garant?a	null	null
8315100	Boletas de garant?a en moneda chilena.	null	null

figura 3.8. tabla tbl_sbif_plancuenta

La siguiente imagen muestra la tabla **tbl_sbif_plancuenta_mes** con los datos cargados:

Worksheet

```

1 select periodo_mes, *
2 from production_master.tbl_sbif_plancuenta_mes;

```

Execute Explain Upload Save as... New Worksheet

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

Filter columns... previous next

periodo_mes	tbl_sbif_plancuenta_mes.plancuenta_id	tbl_sbif_plancuenta_mes.plancuenta_nombre	tbl_sbif_plancuenta_mes.mdm_fhnew
2018-06-01	1750000	IMPUESTOS DIFERIDOS	2019-10-24 21:38:14.702
2018-06-01	4200300	Comisiones por servicios de tarjetas	2019-10-24 21:38:14.702
2018-06-01	1600200	Otros activos fijos	2019-10-24 21:38:14.702
2018-06-01	4150200	Contratos de retrocompra	2019-10-24 21:38:14.702
2018-06-01	4150572	Reajustes por bonos hipotecarios	2019-10-24 21:38:14.702
2018-06-01	5009000	RESULTADO DE OPERACIONES CONTINUAS	2019-10-24 21:38:14.702
2018-06-01	1130100	Documentos a cargo de otros bancos (canje)	2019-10-24 21:38:14.702
2018-06-01	1302602	Cr?ditos con garant?a CORFO	2019-10-24 21:38:14.702
2018-06-01	1500000	INTANGIBLES	2019-10-24 21:38:14.702
2018-06-01	4200200	Comisiones por avales y cartas de cr?dito	2019-10-24 21:38:14.702
2018-06-01	5010000	UTILIDAD (PERDIDA) CONSOLIDADA DEL EJERCICIO	2019-10-24 21:38:14.702

figura 3.9. tabla tbl_sbif_plancuenta_mes

La siguiente imagen muestra la tabla **tbl_sbif_eerr_mes** con los datos cargados:

Worksheet

```
1 select periodo_mes, *
2 from production_master.tbl_sbif_eerr_mes;
```

Execute Explain Upload Save as... New Worksheet

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

Filter columns... previous next

periodo_mes	tbl_sbif_eerr_mes.ins_id	tbl_sbif_eerr_mes.plancuenta_id	tbl_sbif_eerr_mes.eerr_origen	tbl_sbif_eerr_mes.eerr_monto_clp
2018-06-01	001	1000000	B1	16935765000000
2018-06-01	001	1100000	B1	665536000000
2018-06-01	001	1100100	B1	515823000000
2018-06-01	001	1100200	B1	146589000000
2018-06-01	001	1100300	B1	3124000000
2018-06-01	001	1100400	B1	0
2018-06-01	001	1130000	B1	375604000000
2018-06-01	001	1130100	B1	183003000000
2018-06-01	001	1130200	B1	192601000000
2018-06-01	001	1150000	B1	1236941000000
2018-06-01	001	1150100	B1	977775000000

figura 3.10. tabla tbl_sbif_eerr_mes

Finalmente, se muestra un resumen de los dos períodos cargados en la tabla **tbl_sbif_eerr_mes**:

Worksheet

```
1 select periodo_mes
2     , count(distinct ins_id) as num_instituciones
3     , count(1) as num_filas_por_mes
4 from production_master.tbl_sbif_eerr_mes
5 group by periodo_mes;
```

Execute Explain Upload Save as... New Worksheet

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

Filter columns... previous next

periodo_mes	num_instituciones	num_filas_por_mes
2018-06-01	21	8778
2018-07-01	22	9196

figura 3.11. Resumen de dos periodos cargados en tabla tbl_sbif_eerr_mes