Manual para la Herramienta de Precios Chedraui v1.0: Alimentación, Uso y Extracción de Datos

BlackTrust

January 13, 2021

Contents

1 Descripción de la Herramienta

La Herramienta de Precios Chedraui busca precios en tiendas de la competencia con el fin de igualar precios de productos en estas tiendas. En general, la herramienta busca el precio más barato posible, o en su defecto el precio de mercado (moda nacional).

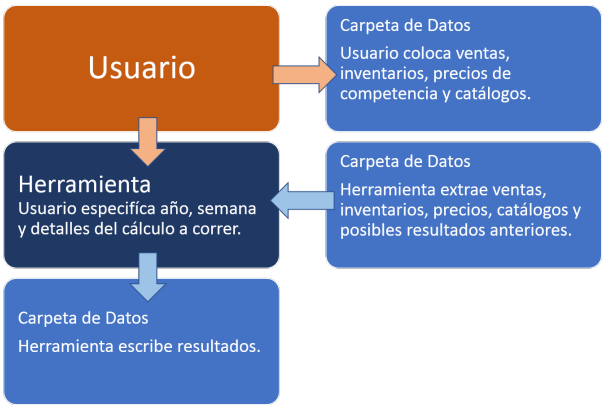
Las tablas de datos son extraidas del mismo equipo donde corre la herramienta, no requiere de conexiones a bases de datos externas. La herramienta es capaz de correr en un cluster, las tablas de datos se extraen del mismo en este caso de un Hadoop Filesystem (HDFS, Filesystem compartido por el cluster).

Los requerimientos físicos y de software se describen en la sección **??**, la preparación de los datos de entrada en la **??**, las funciones de la herramienta y su uso en la **??**, la descripción de los datos de salida en la **??**. Finalmente, el proceso seguido por la herramienta se detalla en la sección **??**, esta misma sección tiene una descripción poco técnica del proceso.

La Herramienta de Precios Chedraui corre sobre la consola de Spark, la cuál a su vez corre sobre la máquina virtual de Java con el lenguaje Scala. Opcionalmente, se puede utilizar el HDFS si se desea correr sobre un cluster. La instalación de las dependencias de la herramienta se cubre en la sección **??**.

En general, las funciones siguen el siguiente proceso:

1. El usuario coloca datos en la máquina de trabajo.
2. El usuario hace una llamada a una función de la herramienta.
3. La herramienta coloca resultados sobre una carpeta.



1.1 Reglas de Negocio y Proceso

A continuación se detallan las reglas de negocio utilizadas por la herramienta. El proceso principal la llevan a cabo cuatro funciones de la herramienta:

1. calcCompetencia Preparación de Precios de Competencia
2. calcDatos Cruce de Precios y Detección de Promociones
3. calcPrecios Sugerencias de Precios
4. calcImpactos Cálculo de Impactos de Cambio de Precio

1.1.1 Preparación de Precios de Competencia

La herramienta extraerá los precios de cada producto de la competencia a nivel UPC/Tienda, luego calculará moda y mediana nacionales en la semana más reciente donde se haya encontrado cada UPC. Adicionalmente, la herramienta calculará la mediana de precio a lo largo de las semanas de un producto a nivel Tienda.

1.1.2 Cruce de Precios y Detección de Promociones

La herramienta extraerá los inventarios y calculará el precio de regulación central:

$$P_{RC} = InvFinVta/InvFinUni \tag{1}$$

A partir de las ventas (o del inventario según especifique el usario) se cruzará la matriz de competencia a nivel Tienda/Depto/SubDepto. La matriz de competencia asocia Tiendas Chedraui a Tiendas de la competencia; cada tienda de la competencia tiene un nivel de prioridad, donde la prioridad máxima prioridad es 1 y la mínima es 3.

Se cruzarán los precios de la competencia. A nivel UPC/Tienda se tomará el precio más reciente de la competencia, a nivel UPC se toma la moda nacional más reciente y a nivel UPC/Tienda se toma la mediana de la Tienda a lo largo

del tiempo. Precios de cada producto serán marcados como promocionales si se cumple alguna de las siguientes condiciones:

$$\begin{aligned} P &< P_{medianaTienda}(100 - porcentaje) \\ P &< P_{modaNacional}(100 - 60) \end{aligned} \tag{2}$$

El porcentaje estará especificado por un catálogo cat_promos. Un precio será marcado como remate si la condición anterior se cumple con porcentaje=60%.

1.1.3 Sugerencias de Precios

Se obtiene el grupo de cada UPC de acuerdo al catálogo cat_grupos, posteriormente se calcula la moda nacional del grupo (el mínimo de las modas nacionales de cada grupo) y el precio de grupo a nivel tienda (el mínimo de los precios de los productos del grupo a nivel tienda).

Por cada producto se calcula el precio sugerido de acuerdo a las siguientes reglas aplicadas en orden:

1. El precio sugerido inicial por UPC es el precio de la competencia encontrado en las Tiendas de la competencia especificados en la matriz de competencia.
2. Si no se ha encontrado precio o esta marcado como promocional/remate será sustituido por la moda nacional.
3. Si el precio sugerido es mayor al precio de grupo o no se ha encontrado un precio de competencia se sustituye el precio de grupo a nivel tienda.
4. Si el precio sugerido no ha sido encontrado se utiliza la moda de grupo.
5. Se asocia a cada artículo una excepción si al menos a un artículo de su grupo se le da una excepción en el catálogo de excepciones.

Con estas reglas los únicos productos que no presentan sugerencias de precios son aquellos que no se encuentran en las tablas de precios de la competencia.

Finalmente se pueden calcular los impactos que la regulación de precios tiene sobre los precios Chedraui, esta última funcionalidad la lleva a cabo la función calcImpactos.

2 Requerimientos e Instalación

En esta sección se cubren los requerimientos de la herramienta de precios, tanto como software y hardware. Adicionalmente, se cubren los pasos de instalación de la paquetería y librerías requeridas para correr la herramienta.

En resumen, la herramienta de precios es un paquete escrito sobre Apache Spark, el cual a su vez corre sobre la máquina virtual de Java (JVM) y tiene una interfaz nativa en Scala.

Las versiones del software bajo las que la herramienta ha sido escrita y probada son: La herramienta ha sido corrida sobre distintas subversiones de

Software	Versión	Notas
Java	JDK Oracle 8	
Scala	2.11.8	No usar Dotty
Apache Spark	2.2.0	Versiones Spark 2.2.x son compatibles
Hadoop	2.7.3	Opcional para HDFS

Scala 2.12 y sobre OpenJDK 1.8 en lugar de Java 8. Sin embargo, se recomiendan las versiones mencionadas en la lista anterior.

Debido a que se usa Spark y la herramienta trabaja sobre Spark, es posible correr la herramienta sobre un cluster en lugar de una sola máquina. En el caso de correr la herramienta sobre un cluster, se debe utilizar el Hadoop Filesystem (HDFS) para poder distribuir los archivos en el mismo.

2.1 Sistema Operativo y Hardware

Debido a que Spark corre sobre la máquina virtual de Java, en principio es posible correr la herramienta en cualquier sistema operativo que soporte Java. Se recomienda una distribución Linux de 64 bits, la herramienta ha sido escrita y probada en Ubuntu 16.04, pero una distribución basada en Debian (Ubuntu) o Red Hat (Fedora/CentOS) también es recomendada. Cualquier distribución con paquetería Debian o RPM deberían de poder instalar los requerimientos de la herramienta y correrla.

En cuanto a hardware, la variable más importante es la memoria RAM. Los archivos de ventas (1.2 GB) y los archivos de precios de la competencia (800 MB por semana / 5 GB por seis semanas) deben de caber en RAM. Spark es capaz de hacer cómputos sobre disco duro en lugar de RAM, pero la velocidad de procesamiento sufrirá debido a la baja velocidad de acceso a datos en disco duro comparado con RAM; un SSD alivia unpoco este problema pero es mucho mejor dejar esos cálculos en RAM. Tomando en cuenta 6 semanas de datos de precios de la competencia y el archivo de ventas se deben de tener 6 GB de RAM para Spark. Apache recomienda que Spark no supere el 75% de RAM de la máquina, esto da un resultado final de 8 GB de RAM. En la práctica, esta cantidad de RAM es suficiente pero muy restrictiva, se recomiendan más de 12 GB de RAM para persistir más datos en memoria (mediante driver-memory de Spark) y ampliar el heap space de Java.

Los datos que requiere la herramienta son tablas que pueden pesar hasta 2 GB cada una. En la práctica una corrida de una semana puede ocupar en total 20 GB de capacidad (este número depende más de los datos de precios, ventas y artículos que del funcionamiento interno de la herramienta).

El CPU es un tanto más flexible, velocidades de 2.0 GHz son suficientes, el número de núcleos normalmente domina el tiempo de cómputo hasta 8 núcleos (4 núcleos alcanzan un nivel de paralelización suficiente).

	Mínimo	Recomendado
RAM	12 GB	16 GB
Disco	20 GB libres	100 GB
Reloj CPU	2.0 GHz	3.0 GHz
Núcleos CPU	4 núcleos/8 Hilos	8 núcleos/16 Hilos

2.2 Java

2.2.1 Oracle Java

Para las distribuciones de Ubuntu 16.10, 16.04, 15.10, 14.04 y 12.04, así como Linux Mint 18, 17.x y 13 se puede usar el PPA de webupd8team:

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Se tendrán que aceptar los términos y condiciones de Oracle para finalizar la instalación.

Adicionalmente, oracle proporciona el .rpm de JDK 8 en su sitio web:
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Después de bajar el .rpm apropiado para su arquitectura, este se puede instalar con yum:

```
$ sudo yum localinstall nombre_del_rmp.rpm
```

o con zipper según sea el caso:

```
$ zypper install nombre_del_rpm.rpm
```

El comando java -version se puede utilizar para comprobar la versión que se ha instalado.

2.2.2 OpenJDK

Se puede instalar la versión de código abierto de Java para distribuciones basadas en Debian:

```
$ sudo apt-get install default-jre
$ sudo apt-get install default-jdk
```

aquellas con paquetería yum:

```
$ sudo yum install java-1.8.0-openjdk-devel
$ sudo yum install java-1.8.0-openjdk
```

y finalmente aquellas con paquetería zypper:

```
$ zypper in java-1_8_0-openjdk
```

El comando java -version se puede utilizar para comprobar la versión que se ha instalado.

2.3 Scala

Ya que se utilizará Spark, se deben instalar los binarios de Scala (no desde sbt o IntelliJ), los binarios se pueden descargar de la siguiente liga:
https://downloads.lightbend.com/scala/2.11.8/scala-2.11.8.tgz
En caso de que esta liga haya expirado, se pueden encontrar instrucciones para la descarga de dichos binarios aquí:
https://www.scala-lang.org/download/2.11.8.html

Una vez descargados los binarios, se descomprimen y se coloca la carpeta scala-2.11.8 sobre /usr/local/ bajo el nombre de scala/. Esta es la costumbre para sistemas Linux, pero es posible colocársele en otro lado sin ningún problema, se supondrá en este documento que Scala está colocado en /usr/local/scala.

Se recomienda declarar la variable de entorno $SCALA_HOME y expandir el $PATH para los usuarios del sistema de la siguiente manera:

```
$ export SCALA_HOME=/usr/local/scala
$ export PATH=$PATH:$SCALA_HOME/bin
```

Declarese según sean los protocolos de su área de TI (.bashrc, /etc/environment, etc). Se puede comprobar la instalación correcta de Scala al correr simplemente el comando scala en terminal, en caso de no tener el $SCALA_HOME/bin sobre el $PATH tendra que usar la ruta completa a $SCALA_HOME/bin/scala.

2.4 Spark

Spark se debe instalar desde sus binarios, los cuales se pueden descargar aquí:
http://spark.apache.org/downloads.html
Seleccione Spark 2.2.0 construido para Hadoop 2.7.0 en adelante.

Una vez descargados los binarios, se descomprimen y se coloca la carpeta spark-2.2.0-bin-hadoop2.7 sobre /usr/local/ bajo el nombre de spark/. Se supondrá en este documento que Spark está colocado en /usr/local/spark. Se debe remombrar /usr/local/spark/conf/slaves.template a /usr/local/spark/conf/slaves, este es el archivo que le comunica a Spark que máquinas tiene disponibles para trabajar en modo cluster; en caso de no tener un cluster la unica máquina disponible es localhost. La configuración de Spark en cluster se cubrirá en la sección **??**.

Se recomienda declarar la variable de entorno $SPARK_HOME y expandir el $PATH para los usuarios del sistema de la siguiente manera:

```
$ export SPARK_HOME=/usr/local/scala
$ export PATH=$PATH:$SPARK_HOME/bin
```

Declarese según sean los protocolos de su área de TI (.bashrc, /etc/environment, etc). Se puede comprobar la instalación correcta de Scala al correr simplemente el comando spark-shell en terminal, en caso de no tener el $SPARK_HOME/bin sobre el $PATH tendra que usar la ruta completa a $SPARK_HOME/bin/spark-shell.

2.5 Hadoop

Hadoop se utiliza en el caso de correr la herramienta sobre un cluster, se puede instalar desde sus archivos binarios:
https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3 Los archivos binarios se encuentran en el archivo comprimido hadoop-2.7.3.tar.gz.

Una vez descargados los binarios, se descomprimen y se coloca la carpeta hadoop-2.7.3 sobre /usr/local/ bajo el nombre de hadoop/. Se supondrá en este documento que Hadoop está colocado en /usr/local/hadoop. La configuración de Hadoop se cubrirá en **??**.

2.6 Configuración de Cluster

Esta sección detalla la configuración de un cluster de máquinas que corren la herramienta sobre Spark y Hadoop. Por concretitud, se supondrá que se tienen 4 máquinas distintas bajo la misma red, cada una con IP fija y nombres de máquina Maq1, Maq2, Maq3 y Maq4. Todas las máquinas deben de compartir el mismo nombre de usuario para manejar Spark+Hadoop, tradicionalmente se utiliza el nombre hduser, pero puede ser cualquier nombre. Se supondrá el nombre black para como nombre de usuario de este punto en adelante.

El esquema de trabajo de Spark/Hadoop requiere una máquina maestra (o master), la cual recibe instrucciones del usuario y coordina a las demás para trabajar juntas. A las máquinas de trabajo se les refiere como trabajadoras (workers) o esclavas (slaves); la nomenclatura exacta depende de la fuente. En este documento se utilizará la nomenclatura de maestra/trabajadoras.

La máquina maestra también puede ser esclava y en esta se coleccionan los datos finales no paralelizados, así que normalmente se escoge la máquina con mayor memoria RAM. En este documento se escogerá Maq1 como la maestra.

En todas las máquinas se debe instalar Java, Scala, Spark y Hadoop. En el caso de Scala, Spark y Hadoop, se recomienda automatizar el copiado de cada carpeta de programa a /usr/local después de haber editado sus hojas de configuración de acuerdo a las instrucciones de esta sección.

2.6.1 Configuración SSH y de Hosts

Spark/Hadoop utiliza SSH para manejar las señales entre las máquinas, así que se requiere instalar el cliente y el servidor SSH. Las instrucciones para Debian, yum y zypper son las siguientes:

```
$ sudo apt-get install openssh-server openssh-client
$ sudo yum install openssh-server openssh-clients
$ sudo zypper install openSSH
```

Se recomienda no utilizar el puerto 22 para ssh, en dado caso se debe especificar el puerto por medio de la variable de entorno $SPARK_SSH_OPTS en la máquina maestra, por ejemplo:

```
export SPARK_SSH_OPTS="-p NumeroDePuerto"
```

Adicionalmente, se requiere que las máquinas tengan capacidad de iniciar sesiones SSH entre ellas por medio de llaves SSH. Si las máquinas no tienen llaves SSH estas se pueden generar con el comando ssh-keygen, genere las llaves de acuerdo a los protocolos de su rama de Seguridad de la Información. En este documento se supondrá que los pares de llaves (pública y privada) se generaron en el directorio /home/black/.ssh con los nombres id_rsa y id_rsa.pub (respectivamente).

Se necesitan añadir las IPs de las máquinas al archivo Hosts de cada máquina, por ejemplo el archivo /etc/hosts de Maq2 se puede ver así:

```
127.0.0.1 localhost
127.0.1.1 Maq2
IP1 Red1
IP2 Red2
IP2 Red3
IP2 Red4

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Donde IPX y RedX son la IP y nombre de red de la máquina MaqX, respectivamente. El nombre de red de la máquina maestra no debe coincidir con el nombre de máquina.

Para copiar las llaves de una máquina a otra se puede utilizar el siguiente comando:

```
$ ssh-copy-id black@RedX
```

Alternativamente, se pueden añadir los contenidos de cada llave pública en id_rsa.pub a una nueva línea del archivo authorized_keys dentro de la carpeta /home/black/.ssh (este archivo guarda las llaves autorizadas para iniciar sesión por SSH de manera automática). Si el último no existe, se puede crear con un editor o con el comando touch.

Si la configuración de llaves SSH se ha efectuado correctamente, el siguiente comando debería iniciar una sesión en MaqX desde cualquier otra máquina:

```
$ ssh black@RedX
```

2.6.2 Configuración de Spark Modo Cluster

Dentro de la carpeta de instalación de Spark existe la carpeta conf/, la cual guarda todos los archivos de configuración de Spark. Se deben de añadir los nombres de red (los aliases en /etc/hosts) de todas las trabajadoras al archivo slaves de la carpeta de configuración:

```
# Nombres de Red en SPARK_HOME/conf/slaves
# A Spark Worker will be started on each of the machines listed below.
Red1
Red2
Red3
Red4
```

Este archivo se debe de editar en cada una de las máquinas del cluster.

Para verificar la instalación, se pude correr el shell script start-all.sh en la máquina maestra:

```
$ /usr/local/spark/sbin/start-all.sh
```

Este script inicializa la máquina maestra junto con todas las trabajadoras e indicará si alguna máquina no se puede inicializar o esta desconectada de la red. Se recomienda crear un alias para este comando y para el script stop-all.sh:

```
$ alias start-spark=/usr/local/spark/sbin/start-all.sh
$ alias stop-spark=/usr/local/spark/sbin/stop-all.sh
```

Para incializar una sesión de Spark en modo distribuido se debe especificar la máquina maestra en el comando spark-shell

```
$ spark-shell --master spark://Red1:7077
```

El puerto 7077 es el puerto defecto para sesiones de Spark y este se puede configurar. Dicha configuración esta fuera del alcance de este documento.

2.6.3 Configuración de Hadoop Modo Cluster

La herramienta de precios es capaz de ocupar el Filesystem distribuido de Hadoop (HDFS) para guardar y escribir archivos en un cluster. Dichos archivos son guardados por bloques en cada máquina del cluster, se supondrá que estos bloques serán guardados dentro de algún subdirectorio de /app/hadoop (Hadoop se encarga de decidir este subdirectorio). Se debe de crear el directorio que Hadoop utilizará para trabajar con los permisos correctos en cada máquina:

```
$ mkdir -p /app/hadoop/tmp
$ chown black:black /app/hadoop
$ chown black:black /app/hadoop/tmp
```

Las últimas instrucciones las debe de correr un superusuario.

La primera hoja de configuración a editar es dfs.include y se acostumbra colocarla dentro de /app/hadoop (esta ruta se puede configurar). Esta hoja debe de tener los nombres de red de cada máquina del cluster:

```
Red1
Red2
Red3
Red4
```

Esta es la lista de máquinas que guardan los archivos del cluster.

El resto de las hojas de configuración a editar de encuentran dentro de la carpeta de instalación de Hadoop, dentro de la subcarpeta etc/hadoop y son en su mayorúa archivos XML.

La hoja core-site.xml regula los datos referentes al "Namenode" del cluster, el cual se encarga de registrar donde están guardados los datos del HDFS en el cluster y de guardar el arbol de directorios del mismo HDFS. El Namenode puede ser la máquina maestra, como se muestra en la hoja siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- /usr/local/hadoop/etc/hadoop/core-site.xml -->
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>

<property>
  <name>fs.default.name</name>
  <value>hdfs://Red1:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>

<property>
  <name>fs.trash.interval</name>
  <value>1440</value>
  <description> Enables trash instead of immediate deletion,
  strongly recommended.</description>
</property>

</configuration>
```

El atributo hadoop.tmp.dir especifica la ruta donde se guardan los archivos del HDFS. fs.default.name especifica la URI con la que se hace interfaz con HDFS (se acostumbra usar el puerto 54310). fs.trash.interval se incluye para habilitar la papelera del HDFS, se vacía cada 1440 segundos (24 horas) y su inclusión previene que el comando rm de Hadoop borre permanentemente datos sin pasar antes por dicha papelera.

La hoja hdfs-site.xml regula los datos referentes a los "Datanodes", las máquinas del cluster que guardan los archivos del HDFS. En esencia, cada datanode es una máquina trabajadora para el HDFS. Un ejemplo de esta hoja se presenta a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- /usr/local/hadoop/etc/hadoop/hdfs-site.xml -->
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>dfs.replication</name>
  <value>2</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>

<property>
  <name>dfs.permissions.supergroup</name>
  <value>black</value>
  <description>The name of the group of super-users.</description>
</property>

<property>
  <name>dfs.hosts</name>
  <value>/app/hadoop/dfs.include</value>
  <description>Names a file that contains a list of hosts that are
  permitted to connect to the namenode. The full pathname of the file
  must be specified. If the value is empty, all hosts are
  permitted.</description>
</property>

</configuration>
```

dfs.replication regula la redundancia del guardado de archivos del HDFS; los

archivos del HDFS se dividen en bloques y estos mismos bloques se pueden replicar en distintos datanodes. En este caso cada bloque se guarda dos veces en el cluster, esta redundancia permite al cluster continuar funcionando en el evento de que una máquina falle o se desconecte del cluster. dfs.replication no debe de ser mayor al número de máquinas en el cluster. dfs.permissions.supergroup es el nombre el grupo de usuarios que pueden manipular los archivos del cluster con permisos de superusuario. dfs.hosts es la ruta de dfs.include, el cual tiene la lista de máquinas que se pueden conectar al namenode (la interfaz del HDFS) y fungir de trabajadoras.

La hoja mapred-site.xml regula datos referentes a las instrucciones que siguen el esquema MapReduce y su reporteo (HDFS utiliza este esquema de trabajo). En nuestro caso solamente hay que especificar el nombre de red de la máquina reporteadora y un puerto libre (se acostumbra el 54311):

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- /usr/local/hadoop/etc/hadoop/mapred-site.xml -->
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>mapred.job.tracker</name>
  <value>Red1:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>

</configuration>
```

Las variables presentadas en este documento para las hojas de configuración XML de Hadoop no son una lista exhaustiva. Se pueden encontrar todos los valores configurables en las siguientes ligas:
https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/core-default.xml
https://hadoop.apache.org/docs/r2.7.7/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml
https://hadoop.apache.org/docs/r2.7.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml

Las únicas hojas que quedan de editar son “masters” y “slaves.” La hoja de masters solo debe contener el nombre de la máquina maestra:

```
Red1
```

La hoja de slaves contiene la lista de todas las trabajadoras:

```
Red1
Red2
Red3
Red4
```

Las hojas de configuración se deben copiar en cada máquina que forma parte del cluster. Una vez terminada la edición y el copiado de las dichas se puede inciar el HDFS desde la maestra:

```
$ /usr/local/hadoop/sbin/start-dfs.sh
```

De nuevo se recomienda crear un alias para este script y otro alias para stop-dfs:

```
$ alias start-hdfs=/usr/local/hadoop/sbin/start-dfs.sh
$ alias stop-hdfs=/usr/local/hadoop/sbin/stop-dfs.sh
```

Antes de continuar trabajando con el cluster es necesario detenerlo.

El siguiente paso es formatear el HDFS para empezar su uso:

```
$ cd /usr/local/hadoop
$ ./hdfs namenode -format
```

El cluster no puede estar corriendo durante este proceso.

El HDFS debería de estar listo para su uso, se puede corroborar que este funcionando correctamente con las siguientes operaciones:

```
$ start-hdfs
$ hadoop fs -ls -h /
$ hadoop fs -mkdir /prueba
$ hadoop fs -ls -h /
```

Nótese que se ha utilizado el alias para el script del levantamiendo del cluster. El HDFS contiene casi todas las operaciones propias de un Filesystem común:

```
$ hadoop fs -ls -h /directorio/*
$ hadoop fs -mkdir /dir1/dir2
$ hadoop fs -rm -r /dir1
$ hadoop fs -mv /dir/archivo1 /dir/archivo2
```

Desde uno de los nodos es posible subir archivos al HDFS,

```
$ hadoop fs -put archivo_local /alguna_ruta/archivo_en_hdfs
```

También es posible descargarlos

```
$ hadoop fs -get /ruta/archivo_en_hdfs destino_local
$ hadoop fs -getmerge /ruta/archivo_en_hdfs destino_local
```

Estos dos comandos no son equivalentes. El comando get se utiliza para obtener archivos que se guardan en un solo bloque, los archivos multibloques se obtienen por medio de *getmerge*. En la práctica *getmerge* funciona para los dos tipos de

archivos. Esta distinción se debe a que los archivos multibloques son guardados como un directorio cuyos contenidos son los bloques (archivos .part distribuidos sobre los datanodes) del archivo total, getmerge obtiene todos los bloques por medio de un solo comando. Advertencia: no hay un orden específico para los bloques, Spark+HDFS no se debe utilizar para guardar tablas donde el orden de los registros codifica información (a menos de que se tomen precauciones para guardar estas tablas en un solo bloque).

La herramienta de precios guarda las tablas de resultados y sus cabeceras de manera separada, así que para extraer resultados de la herramienta de precios se utiliza getmerge:

```
$ hadoop fs -getmerge /ruta/resultados/archivo.csv.plainheader \
/ruta/resultados/archivo.csv destino.csv
```

Este comando concatenará las cabeceras con la tabla guardada en el archivo .plainheader. En general, archivo.csv.plainheader serán siempre las cabeceras de archivo.csv. Los archivos .sparkheader son utilizados internamente por la herramienta, estos guardan metadatos detallados sobre la tabla en cuestión.

2.7 Instalación de la Herramienta

2.7.1 Carpeta de Datos

Antes de correr la herramienta se debe de preparar alguna carpeta donde se guarden los datos, a esta carpeta se le llamará $CHDT_PATH. En el equipo entregado al equipo de precios de Chedraui esta carpeta se ha preparado aquí: /home/black/Documents/chedraui/
Sobre esta carpeta deben de existir las carpetas inventarios/, ventas/, preciosCompetencia/ y resultados/:

```
$ CHDT_PATH="/home/black/Documents/chedraui/"
$ mkdir -p $CHDT_PATH
$ mkdir $CHDT_PATH/inventarios
$ mkdir $CHDT_PATH/ventas
$ mkdir $CHDT_PATH/preciosCompetencia
$ mkdir $CHDT_PATH/resultados
```

En caso de estar en modo cluster, se debe usar hadoop -mkdir. La carpeta $CHDT_PATH contiene las subcarpetas de datos y los catálogos, un ejemplo de esta carpeta con sus archivos y subcarpetas se presentan a contiuación:

```
$CHDT_PATH/
    cat_skus.csv
    cat_excepciones.csv
    ... (todos los archivos de catalogos)
    cat_matriz.csv
    inventarios/
        Inven_15Ene17.csv
```

```
        Inven_14Feb17.csv
        ... (todos los archivos de inventarios)
        Inven_22Dic17.csv
    ventas/
        ventas201702.csv
        ventas201703.csv
        ... (todos los archivos de ventas)
        ventas201751.csv
    preciosCompetencia/
        nielsen201702.csv
        nielsen201703.csv
        otroproveedor201703.csv
        ... (todos los archivos de precios)
        nielsen201750.csv
    resultados/
        ... (la herramienta deposita resultados)
```

La preparación de los catálogos y hojas de datos se cubren en **??**.

2.7.2 Carpeta de Trabajo de Spark

Junto con este documento, se debió de haber entregado la herramienta compilada, es un archivo .jar llamado herramienta-chedraui_nVersion.jar, donde nVersion contiene la versión de Scala con la fue compilado y la versión de la herramienta.

Spark genera archivos auxiliares durante su ejecución, así que se recomienda colocar el .jar en una carpeta donde solamente exista el jar. El compilado no debe estar en una carpeta donde se encuentren los datos. Dentro de la misma carpeta se puede correr Spark con la herramienta:

```
$ start-spark # si es que Spark no ha sido inicializado
$ spark-shell --jars herramienta-chedraui_2.11-1.0.jar \
  --driver-memory 8g
```

Driver memory se puede ajustar según el volumen de los datos, con 6 semanas de datos de precios 8 GB es suficiente. Los detalles del uso de la herramienta se cubren en la sección **??**.

2.8 Herramientas Auxiliares

2.8.1 LibreOffice

LibreOffice Calc es un programa parecido a Excel, para los fines de este documento se tratará como un programa que ayudará a transformar exceles a csv con separadores arbitrarios. El archivo de instalación se encuentra en la siguiente liga:
https://www.libreoffice.org/download/download/
Descargue el .deb o el .rpm según sea el caso.

2.8.2 MDB Tools

Durante la preparación de datos es necesario extraer tablas de archivos .accdb y .mdb. MDB Tools permite la extracción de estas tablas a archivos de texto plano.

MDB Tools forma parte de los repositorios de defecto de Ubuntu,

```
$ sudo apt-get install mdb-tools
```

También forma parte de EPEL para distribuciones basadas en Red Hat, a continuación se presenta una liga con instrucciones de instalación para cada distribución:
https://fedoraproject.org/wiki/EPEL
En cuanto a SUSE, es posible añadir el repositorio apropiado a su versión:

```
$ zypper addrepo $URL_REPOSITORIO
$ zypper refresh
$ zypper install mdbtools
```

Para SUSE 12.3 la liga $URL_REPOSITORIO es:
https://download.opensuse.org/repositories/openSUSE:12.3/standard/openSUSE:12.3.repo
La liga exacta del repositorio se puede obtener del siguiente sitio:
https://software.opensuse.org/download.html?project=openSUSE%3A12.3&package=mdbtools

2.9 Resumen de Variables de Entorno

Durante esta sección se ha cubierto la instalación de Java, Scala, Spark y Hadoop. Las instrucciones recomiendan o especifican declarar nuevas variables de entorno o modificar el $PATH de los usuarios.

La instalación de Java coloca los archivos de programa sobre rutas distintas según la distribución y versión del sistema operativo. Se puede obtener la ruta apropiada con el siguiente comando:

```
$ $(dirname $(dirname $(readlink -f $(which javac))))
```

Las rutas de instalación de Scala, Spark y Hadoop se pueden ajustar segun las necesidades de administración de la o las máquinas de trabajo.

Para facilidad del lector se presenta una muestra de un archivo .bashrc, suponiendo las rutas de instalación antes especificadas:

```
# Spark+Hadoop
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:/usr/local/scala/bin
export PATH=$PATH:/usr/local/spark/bin
export PATH=$PATH:/usr/local/hadoop/bin
```

```
# Start scripts
alias start-spark=/usr/local/spark/sbin/start-all.sh
alias start-hdfs=/usr/local/hadoop/sbin/start-dfs.sh
alias stop-spark=/usr/local/spark/sbin/stop-all.sh
alias stop-hdfs=/usr/local/hadoop/sbin/stop-dfs.sh
```

Este fragmento puede servir de punto de partida para colocar las declaraciones de variables de entorno para su caso específico.

3 Datos de Entrada y su Preparación

Los datos de entrada (inputs) de la herramienta se deben presentar en texto plano separado por pipes (el caracter "|") y extensión csv, de preferencia codificado en UTF-8. Estos datos consisten de diversos catálogos de Chedraui, tablas de ventas, tablas de inventarios y tablas de precios de la competencia (por ejemplo Nielsen).

Al final de esta sección se presentarán métodos para transformar tablas de Microsoft Excel (xls, xlsx), Microsoft Access (accdb) y Microsoft DataBase (mdb) a texto plano separado por pipes.

El listado de catálogos es como sigue:

Archivo	Descripción
cat_tiendas.csv	Tiendas
cat_skus.csv	SKUs
cat_impuestos.csv	Impuestos
cat_matriz.csv	Matriz de competencia
cat_promos.csv	Umbral de detección de promociones
cat_margenes.csv	Márgenes objetivo por clase
cat_deciles_nacional.csv	Deciles nacionales
cat_deciles.csv	Deciles por tienda
cat_excepciones.csv	Excepciones
cat_grupos.csv	Grupos

La otra fuente de información son los datos de ventas, precios e inventario. Las ventas y precios se extraen de manera semanal, el inventario se extrae en días determinados. Se presentan estas fuentes en forma tabular:

Archivo Ejemplo	Descripción
ventas201704.csv	Ventas de Chedraui
nielsen201708.csv	Precios de la competencia
Inven_03Ene18.csv	Inventario en un día dado

Nótese que la convención de nombres de los archivos de inventarios difiere del resto. Para los inventarios se sigue la convención que utiliza Chedraui para nombrar las tablas de inventario.

3.1 Catálogos

A continuación se describen las variables (campos) que contiene cada catálogo. Es de suma importancia que los títulos de las variables se escriban tal y como se indica, de lo contrario se generará un error y no se podrá leer el archivo.

3.1.1 Tiendas

La función de este catálogo es dar especificaciones de cada centro de Chedraui de acuerdo a su identificador numérico.

Campo	Descripción
Tienda	Identificador numérico
Nombre_tda	Nombre del centro
Formato	Formato del centro
Region	Identificador alfanumérico
Zona	Identificador alfanumérico
Tipo_Tda	Tipo del centro

3.1.2 SKUs

La función de este catálogo es dar especificaciones de cada artículo de Chedraui de acuerdo a su SKU. El código de barras (UPC) se extrae de este catálogo. Este catálogo proviene de una extracción de información de la MARA.

Campo	Descripción
SKU	Identificador numérico
UPC	Código de barras
Principal	Campo alfanumérico
Descripcion	Nombre del artículo
GrupoArticulos	Campo numérico
Estatus	Campo alfanumérico
Depto	Identificador numérico
DescripDepto	Nombre del departamento
SubDepto	Identificador numérico
DescripSubdepto	Nombre del subdepartamento
Clase	Identificador numérico
DescripClase	Nombre de la clase
SubClase	Identificador numérico
DescripSubCl	Nombre de la subclase
Num_Dpt	Campo numérico

3.1.3 Deciles Nacionales

Catálogo por artículo de deciles Nielsen nacionales.

Campo	Descripción
SKU	Identificador numérico
UPC	Código de barras
DecilNacional	Campo numérico, puede tener como prefijo "D"

3.1.4 Deciles por Tienda

Catálogo por artículo de deciles por tienda Chedraui. El formato de las fechas deberá ser formato numérico consistente con SQL, ya sea AAMMDD, AAAMMDD o AAAAMMDD (por ejemplo 170321, 2170321, o 20170321).

Campo	Descripción
InicioVigencia	Fecha en formato numérico SQL
FinVigencia	Fecha en formato numérico SQL
Tienda	Identificador numérico
SKU	Identificador numérico
Decil	Campo numérico

3.1.5 Impuestos

Catálogo por artículo de IVA e IEPS para calcular precios con impuestos añadidos. El valor numérico deberá ser el porcentaje de impuestos por un factor de 10. Por ejemplo, 16% de IVA deberá tener un valor de 160.00.

Campo	Descripción
SKU	Identificador numérico
IVA	Campo numérico
IEPS	Campo numérico

3.1.6 Excepciones

Tabla de excepciones por artículo, las excepciones deben de tener una descripción no vacía.

Campo	Descripción
SKU	Identificador numérico
Excepcion	Campo alfanumérico, no puede ser vacío

3.1.7 Grupos de Artículos

Tabla de grupos por artículo, cada grupo debe ser identificado por un campo numérico.

Campo	Descripción
SKU	Identificador numérico
GrupoArticulo	Campo numérico

3.1.8 Umbrales de Detección de Promociones

La herramienta cuenta con funcionalidad de detección de promociones de manera estadística. Por cada Subdepartamento es posible especificar bajo que porcentaje de desviación de precio se considera un precio como promocional. Por ejemplo un porcentaje del 12% simplemente se representará como 12 en esta tabla.

Campo	Descripción
Depto	Identificador numérico
SubDepto	Identificador numérico
CambioPromocion	Porcentaje, campo numérico

3.1.9 Márgenes Objetivo por Clase

Margen objetivo de cada clase con el fin de regulación de precios. Por el momento, este campo no se utiliza y se puede dar un catálogo vacío, solamente con columnas.

Campo	Descripción
Depto	Identificador numérico
SubDepto	Identificador numérico
Clase	Identificador numérico
MargenObjetivo	Porcentaje, campo numérico

3.1.10 Matriz de Competencia

Tabla de tiendas junto con su competencia. Es posible asignar varias tiendas de competencia a un centro Chedraui, la columna Prioridad asigna un nivel de importancia a cada tienda en caso de que se encuentre más de un precio en las tiendas de competencia. La herramienta tomará el primer precio no vacío de prioridad más alta en la matriz de competencia. Prioridad 1 es el valor más alto, seguido de 2, 3, etc.

Campo	Descripción
Tienda	Identificador numérico
Prioridad	Campo numérico menor a 100
TiendaCompetencia	Identificador numérico
FuenteCompetencia	Campo alfanumérico
CadenaCompetencia	Campo alfanumérico

3.2 Datos Periódicos

En esta sección nos enfocaremos en la parte de los datos, para cada archivo se indica qué variables se deben tener como mínimo, es decir, cada archivo puede tener más columnas de las indicadas, pero al menos debe tener las que se indican en cada caso. Al igual que con los catálogos, es necesario que los nombres de las variables se escriban tal cual aparecen en la descripción de cada archivo.

3.2.1 Ventas de Chedraui

Estos archivos contienen los datos de ventas semanales de Chedraui. El nombre del archivo debe seguir el siguiente formato:

ventas<AñoCompleto><SemanaDosDigitos>.csv

Por ejemplo, las ventas de la semana 2 del año 2018 se deben guardar con el siguiente nombre y dentro de la carpeta ventas:

ventas201802.csv

El campo/columna EAN_UPC puede tener como nombre UPC, la herramienta es capaz de detectar esta columna con cualquiera de estos dos nombres.

Campo	Descripción
SKU	Identificador numérico
EAN_UPC	Código de barras
Depto	Identificador numérico
SubDepto	Identificador numérico
Clase	Identificador numérico
SubClase	Identificador numérico
Semana	Campo numérico
Tienda	Identificador numérico
VentaUni	Campo numérico
VentaPesos	Campo numérico
VentaCosto	Campo numérico

3.2.2 Inventarios

Estos archivos contienen los inventarios de Chedraui a nivel día y centro. El nombre del archivo debe seguir el siguiente formato:

Inven_<Día><NombreMes><AñoDosDigitos>.csv

El nombre del mes puede tomar los siguientes valores, aunque la herramienta es capaz de aceptar estos nombres en inglés:

Ene, Feb, Mar, Abr, May, Jun, Jul, Ago, Sep, Oct, Nov, Dic.

Por ejemplo, el inventario del día 1 de Febrero del año 2018 se debe guardar con el siguiente nombre y dentro de la carpeta inventarios:

Inven_01Feb18.csv

Nótese que el nombre de este archivo es consistente con la convención que toma Chedraui en su extracción de inventarios a archivos Access. Los campos también son idénticos a dicha extracción.

Campo	Descripción
SKU	Identificador numérico
FechaInven	Campo de fecha
Tienda	Identificador numérico
InvFinUni	Campo numérico
InvFinVta	Campo numérico
InvFinCto	Campo numérico

3.2.3 Precios de la Competencia

Estos archivos contienen los datos de precios semanales de la competencia de Chedraui. El nombre del archivo debe seguir el siguiente formato:

<Proveedor><AñoCompleto><SemanaDosDigitos>.csv

Donde el nombre del proveedor esta an minúsculas, sin acentos ni espacios. Por ejemplo, los precios de la semana 2 del año 2018 reportados por Nielsen se deben guardar con el siguiente nombre y dentro de la carpeta preciosCompetencia:

nielsen201802.csv

Las columnas de los precios de la competencia pueden tomar dos nombres, un conjunto de nombres genéricos o los nombres que toma Nielsen. Cualquiera de los dos nombres para las columnas es válido.

Campo	Descripción
shop (o Tienda)	Identificador numérico
barcode (o UPC)	Identificador numérico
pr (o Precio)	Campo numérico

3.3 Conversión de archivos a formato csv

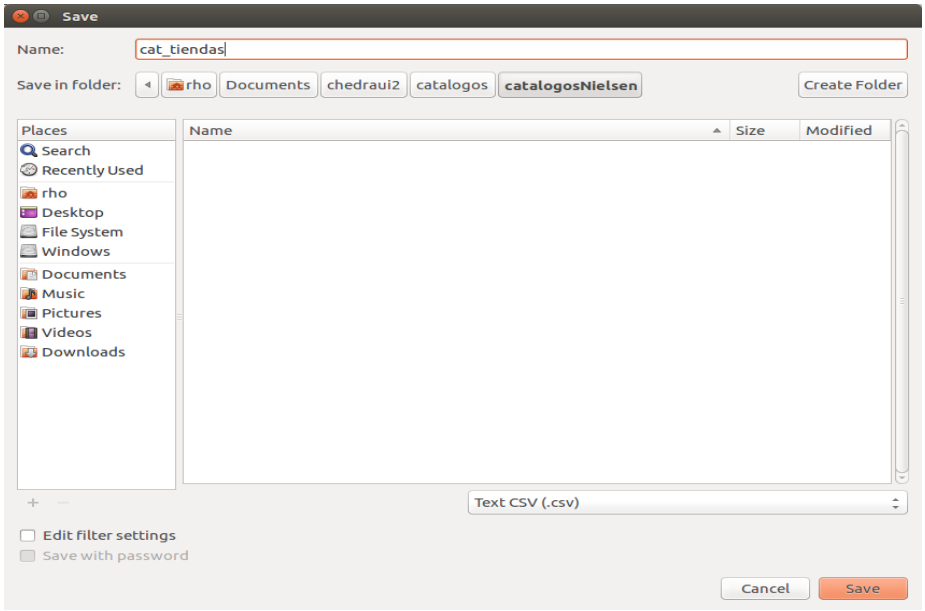
Los archivos en excel, mdb, access, etc. deben convertirse a '.csv' y el separador que se debe utilizar es el pipe '|'. A continuación se describe el procedimiento para realizar la conversión.

3.3.1 Excel a csv

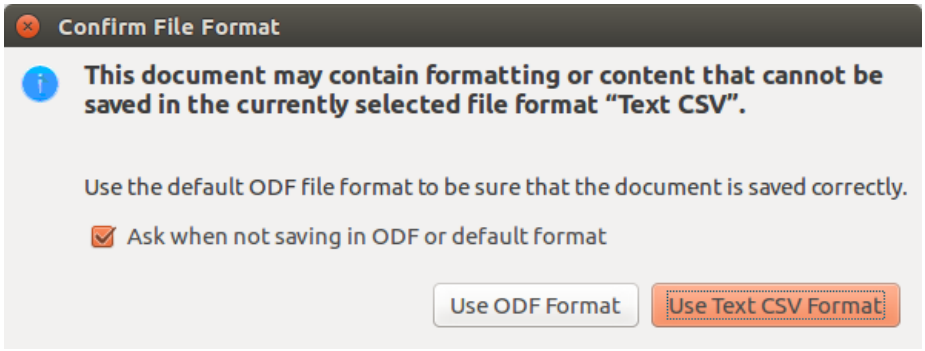
La máquina de trabajo provista por BlackTrust debe de tener instalada la herramienta LibreOffice Calc junto con el sistema operativo Ubuntu. LibreOffice Calc es extremadamente parecido a Excel, puede leer archivos de Excel y tiene una funcionalidad para transformar hojas de cálculo texto plano.

Primero se debe de abrir el archivo de Excel con LibreOffice. Esto se puede hacer navegando al archivo, luego click derecho, después "Abrir con" y seleccionar LibreOffice Calc.

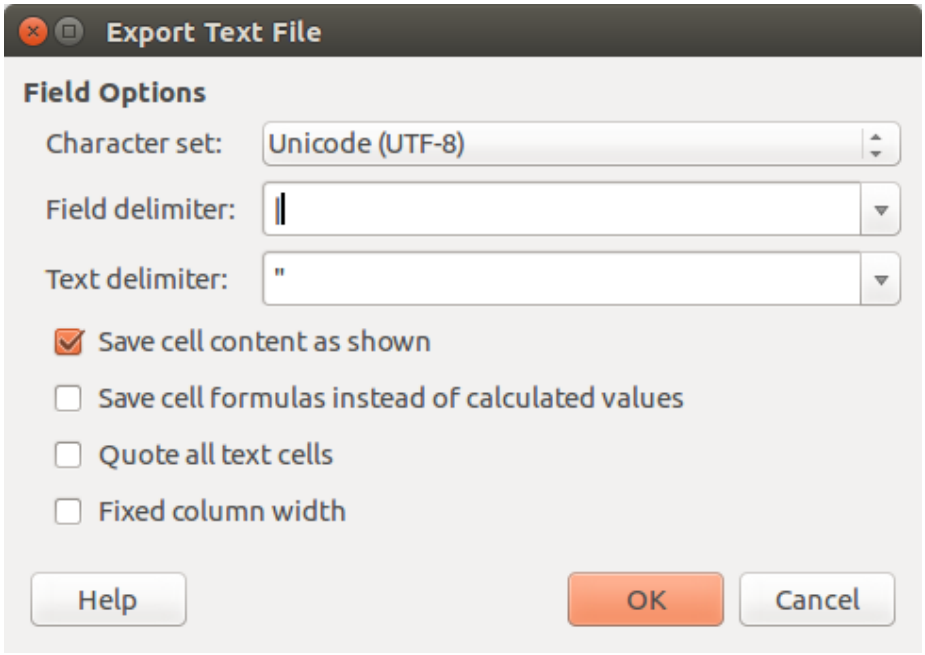
El segundo paso consiste en ir al menú superior Archivo -> Guardar Como. Se escribe el nombre del archivo deseado sin extensión y en la parte inferior del menú se selecciona "Archivo CSV" como lo indica la figura.



Dependiendo de la configuración de la máquina, LibreOffice Calc pedirá una confirmación para cambiar el formato del archivo, se le indica a LibreOffice Calc que se desea proceder con formato CSV.



Surgirá una ventana en la cuál se puede especificar la codificación (de preferencia UTF-8) y el separador. El separador debe ser pipe (el caracter “|”) como lo indica la imagen.



Esto concluye el proceso de transformación Excel a csv. Nótese que este proceso solamente guardará la hoja que esta seleccionada en ese momento.

3.3.2 Access/MDB a csv

El procedimiento para transformar archivos Access a csv es idéntico al mismo para MDB a csv. Se debe utilizar la terminal (consola) para este proceso. Por convención, cualquier línea que empiece con $ es un comando introducido en la terminal. Una línea terminada por
indica que la línea en cuestión continua en la siguiente y se ha partido para un mejor despliegue en este documento. Se pueden autocompletar comandos y nombres de archivos con la tecla de tabular (tab).

La máquina de trabajo provista por BlackTrust debe de tener instalada la herramienta mdb-tools junto con el sistema operativo Ubuntu. En caso de no ser así, se puede abrir una terminal para introducir el siguiente comando:

```
$ sudo apt-get install mdb-tools
```

La mejor manera de explicar este proceso es con un ejemplo. Supóngase que se tiene el archivo "Inventario_22Dic17.accdb" dentro de carpeta /home/black/Documents/datos y que se desea extraer la tabla de inventarios de ella.

Primero se le pide a la máquina navegar a la carpeta deseada,

```
$ cd /home/black/Documents/datos
```

Listado de tablas en el archivo, mdb-tables es el comando que permite conocer las tablas,

```
$ mdb-tables Inventario_22Dic17.accdb
Inven_22Dic17
```

Extracción de la tabla con separador pipe a un archivo csv,

```
$ mdb-export -d "|" -R "\n" -X "\\" Inventario_22Dic17.accdb \
Inven_22Dic17 > Inven_22Dic17.csv
```

Este comando quiere decir "Extrae del archivo Inventario_22Dic17.accdb la tabla Inven_22Dic17 y guárdala en Inven_22Dic17.csv, usa como separador pipe y fin de línea reconocida por Linux."

La máquina provista por BlackTrust debe de tener un alias para esta extracción llamado mdb-export2, el cuál permite omitir las opciones,

```
$ mdb-export2 \
Inventario_22Dic17.accdb Inven_22Dic17 > Inven_22Dic17.csv
```

Este comando es equivalente al anterior. Una vez terminado el proceso debería de existir el archivo deseado, esto se puede corroborar con el comando ls o directamente en el navegador de archivos.

4 Funciones de la Herramienta

La herramienta hace los cálculos utilizando las siguientes funciones:

1. **leeCatalogos**
2. **calcCompetencia**
3. **calcDatos**
4. **calcPrecios**
5. **calcImpactoPrecios**

Adicionalmente, se incluyen funciones auxiliares para el proceso de datos voluminosos. Estas funciones no son utilizadas por las funciones principales de la herramienta y su inclusión busca facilitar el trabajo del personal de Chedraui:

1. **sumaVentas**
2. **listaUPCsCompetencia**

Cada función depende de ciertas variables que el usuario debe ingresar, de acuerdo a lo que desee procesar. En general, las funciones deben de conocer el directorio raíz de los datos, así como el año y semana que se desea procesar.

4.1 Funciones Principales

4.1.1 leeCatalogos

Declaración de la función:

```
def leeCatalogos(rootDir: String,
        matriz: String = "cat_matriz.csv"):
        Map[String,DataFrame]
```

Ejemplo de uso:

```
val cats = leeCatalogos("/home/black/Documents/chedraui", matriz =
    "matriz_noviembre.csv")
```

Como su nombre lo indica, esta función lee los catálogos y limpia la información que contienen, regresa un Map de Strings (etiquetas) a DataFrames (tablas). El resultado de este función se utiliza en el resto de las funciones de la herramienta.

A continuación se describen los argumentos de esta función:

- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- matriz - Nombre de archivo de la matriz de competencia, toma el valor “cat_matriz.csv" por defecto. Cadena de caracteres, opcional.

4.1.2 calcCompetencia

Declaración de la función:

```
def calcCompetencia(myYear: Int,
        myWeek: Int,
        cats: Map[String, DataFrame],
        rootDir: String,
        uri: String = ""):
        Unit
```

Ejemplo de uso:

```
calcCompetencia(2017, 51, cats, "/home/black/Documents/chedraui")
```

Esta función genera tablas auxiliares con precios de la competencia para el uso interno de la herramienta. Esta función genera archivos que comienzan con "comp" y contiene precios y medidas estadísticas de estos precios.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.
- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.

4.1.3 calcDatos

Declaración de la función:

```
def calcDatos(myYear: Int,
        myWeek: Int,
        cats: Map[String, DataFrame],
        rootDir: String,
        uri: String = "",
        inven: String = ""):
        Unit
```

Ejemplo de uso:

```
calcDatos(2017, 51, cats, "/home/black/Documents/chedraui", inven =
    "Inven_02ene2017.csv")
```

Esta función genera la tabla maestra de datos que la herramienta utiliza para sugerencias de precios. La lógica interna de esta función esta fuera del alcance de esta descripción, a grandes rasgos busca precios de la competencia, calcula precios de regulación central, precios promedio y les añade impuestos. Adicionalmente, calcula cantidades y banderas auxiliares para el uso interno

de la herramienta. Esta función genera archivos que comienzan con "datos" y generalmente son los archivos más voluminosos generados por la herramienta y por ello se recomienda reiniciar la consola después de su uso.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.
- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.
- inven - Nombre del archivo de inventario en la carpeta de inventarios que se desea usar como fuente de SKUs y Tiendas, toma el valor vacío por defecto. Cadena de caracteres, opcional.

4.1.4 calcPrecios

Declaración de la función:

```
def calcPrecios(myYear: Int,
        myWeek: Int,
        cats: Map[String, DataFrame],
        rootDir: String,
        uri: String = ""):
        Unit
```

Ejemplo de uso:

```
calcPrecios(2017, 51, cats, "/home/black/Documents/chedraui")
```

Esta función genera la tabla de precios sugeridos. La lógica interna de esta función esta fuera del alcance de esta descripción, a grandes rasgos elimina precios marcados como remates o promocionales, aplica lógica de igualación de precios de grupos, aplica lógica de eliminación de excepciones y redondea precios de acuerdo a las especificaciones de Chedraui. Esta función genera archivos que comienzan con "precios" y contienen las sugerencias de precios antes de su cruce con inventario.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.

- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.

4.1.5 calcImpactoPrecios

Declaración de la función:

```
def calcImpactoPrecios(myYear: Int,
        myWeek: Int,
        cats: Map[String, DataFrame],
        inven: String,
        rootDir: String,
        uri: String = "",
        ventas: String = ""):
        Unit
```

Ejemplo de uso:

```
calcDatos(2017, 51, cats, "Inven_02ene2017.csv", "/home/black/
    Documents/chedraui", ventas = "ventas201801.csv")
```

Esta función genera la tabla de impactos de cambio de precio. La lógica interna de esta función esta fuera del alcance de esta descripción, a grandes rasgos utiliza los resultados de calcPrecios para sugerir precios dado un inventario y un archivo de ventas (este último es opcional). Adicionalmente, genera reportes útiles en el diagnóstico de cálculos. Esta función genera archivos que comienzan con "impacto" y se recomienda utilizar estos archivos para sugerir precios.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.
- inven - Nombre del archivo de inventario en la carpeta de inventarios que se desea usar como fuente de SKUs y Tiendas, toma el valor vacío por defecto. Cadena de caracteres.
- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.

- ventas - En caso de empezar con “ventas", es el nombre del archivo de ventas en la carpeta de ventas que se desea usar como fuente de SKUs y Tiendas. En caso de comenzar con “sumaVentas", es el nombre del archivo de suma de ventas generado por sumaVentas en la carpeta de resultados. En caso de ser vacío, la herramienta tomará el archivo de ventas correspondiente a la semana y año con la que se llamó la función. Vacío por defecto. Cadena de caracteres, opcional.

4.2 Funciones Auxiliares

4.2.1 sumaVentas

Declaración de la función:

```
def sumaVentas(myYear: Int,
        myWeek: Int,
        weeks: Int,
        cats: Map[String, DataFrame],
        rootDir: String,
        uri: String = ""):
        Unit
```

Ejemplo de uso:

```
sumaVentas(2017, 51, 11, cats, "/home/black/Documents/chedraui")
```

Esta función suma todas las semanas de ventas desde el año y semana especificadas hasta un número de semanas especificado hacia atrás en el tiempo. Por ejemplo, con myYear=2017, myWeek=50 y weeks=12 se suman las ventas desde 201738 hasta 201750. Esta función genera archivos que comienzan con “sumaVentas” y contienen las ventas totales de las semanas especificadas.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- weeks - Número de semanas a considerar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.
- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.

4.2.2 listaUPCsCompetencia

Declaración de la función:

```
def listaUPCsCompetencia(myYear: Int,
        myWeek: Int,
        weeks: Int,
        cats: Map[String, DataFrame],
        rootDir: String,
        uri: String = ""):
        Unit
```

Ejemplo de uso:

```
listaUPCsCompetencia(2017, 51, 11, cats, "/home/black/Documents/
    chedraui")
```

Esta función lista los UPCs presentes en las tablas de precios de la competencia desde el año y semana especificadas hasta un número de semanas especificado hacia atrás en el tiempo. Por ejemplo, con myYear=2017, myWeek=50 y weeks=12 se listan los UPCs desde 201738 hasta 201750. Esta función genera archivos que comienzan con "listaUPC" y contienen los UPCs presentes en las semanas especificadas.

A continuación se describen los argumentos de esta función:

- myYear - Año que se desea procesar. Entero.
- myWeek - Semana que se desea procesar. Entero.
- weeks - Número de semanas a considerar. Entero.
- cats - Colección de catálogos generado por leeCatalogos. Mapa de String a DataFrame.
- rootDir - Ruta donde se localizan los archivos a procesar. Cadena de caracteres.
- uri - Dirección única de recurso en caso de utilizar un sistema de archivos HDFS, solamente necesario al usar un cluster con HDFS, toma el valor vacío por defecto. Cadena de caracteres, opcional.

5 Resultados

La herramienta nos proporcionará los archivos que se describen en esta sección. Generalmente, los nombres de los archivos de resultados de la herramienta toman la forma:

<nombre>_<AñoCompleto><SemanaDosDigitos>.csv

Donde <nombre> es el nombre básico del archivo, <AñoCompleto> es el año en formato completo y <SemanaDosDigitos> es el número de semana con un cero a la izquierda si esta es menor que 10. Por ejemplo:

impacto_201809.csv

Este es el archivo "impacto" correspondiente a la semana 9 del 2018. La gran mayoría de los archivos resultantes siguen este formato, se indicará cuando este no sea el caso.

5.1 compHist

Es un archivo auxiliar con medidas estadísticas de las últimas 6 semanas, por artículo y por tienda. El nombre del archivo debe tener el siguiente formato:

compHist_<AñoCompleto><Semana>ċsv

Campo	Descripción
UPC	Código de barras
TiendaCompetencia	Identificador numérico
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
PrecioCompetenciaHistProm	Promedio del precio del UPC en la tienda competencia a lo largo del tiempo
PrecioCompetenciaHistDesv	Desviación estándar del precio del UPC en la tienda competencia a lo largo del tiempo
PrecioCompetenciaHistMin	Precio mínimo del UPC en la tienda competencia a lo largo del tiempo
PrecioCompetenciaHistMax	Precio máximo del UPC en la tienda competencia a lo largo del tiempo
PrecioCompetenciaHistMedn	Mediana del precio del UPC en la tienda competencia a lo largo del tiempo
PrecioCompetenciaHistPr10	Es el valor del percentil 10 de precios
PrecioCompetenciaHistPr20	Es el valor del percentil 20 de precios
PrecioCompetenciaHistPr80	Es el valor del percentil 80 de precios
PrecioCompetenciaHistPr90	Es el valor del percentil 90 de precios

5.2 compNacional

Son medidas estadísticas de precios de productos en la semana cero y la semana -2, esto con respecto a la semana contemplada. El nombre del archivo debe tener el siguiente formato:

compNacional_<AñoCompleto><Semana>.csv

Campo	Descripción
UPC	Código de barras
PrecioMercadoProm	Promedio del precio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia dos semanas anteriores con respecto a la semana contemplada
PrecioMercadoDesv	Desviación estándar del precio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia dos semanas anteriores con respecto a la semana contemplada
PrecioMercadoMedn	Mediana del precio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia dos semanas anteriores con respecto a la semana contemplada
PrecioMercadoFrec	Es la frecuencia de la moda (PrecioMercadoModa)
PrecioMercadoModa	Es la moda del precio en la semana -2
PrecioMercadoActualProm	Es el precio promedio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia en la semana contemplada
PrecioMercadoActualDesv	Desviación estándar del precio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia en la semana contemplada
PrecioMercadoActualMedn	Mediana del precio del UPC sobre todas las tiendas de la competencia contempladas en la matriz de competencia en la semana contemplada
PrecioMercadoActualFrec	Es la frecuencia de la moda (PrecioMercadoActualModa)
PrecioMercadoActualModa	Es la moda del precio en la semana actual

5.3 compRcntNacional

Son medidas estadísticas de precios de la competencia.

El nombre del archivo debe tener el siguiente formato:
compRcntNacional_<AñoCompleto><Semana>.csv

Campo	Descripción
SemanaMercadoReciente	Es la semana en la que se encontró el precio más reciente
UPC	Código de barras
PrecioMercadoRecienteProm	Es el promedio del precio de mercado más reciente
PrecioMercadoRecienteMedn	Es la mediana del precio de mercado más reciente
PrecioMercadoRecienteFrec	Es la frecuencia de la moda (PrecioMercadoRecienteModa)
PrecioMercadoRecienteModa	Es la moda del precio de mercado más reciente

5.4 compRcnt

Son los precios de la competencia.

El nombre del archivo debe tener el siguiente formato:
compRcnt_<AñoCompleto><Semana>.csv

Campo	Descripción
UPC	Código de barras
TiendaCompetencia	Identificador numérico
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
SemanaReciente	Es la semana más reciente donde se encontraron los precios
PrecioCompetenciaReciente	Es el precio más reciente encontrado

5.5 preciosAux

El nombre del archivo debe tener el siguiente formato:
preciosAux_<AñoCompleto><Semana>.csv

Campo	Descripción
SKU	Código de barras
UPC	Código de barras
Depto	Identificador numérico
SubDepto	Identificador numérico
Descripcion	Nombre del artículo
Tienda	Identificador numérico
Nombre_tda	Nombre del centro
PrecioSugerido	Es el precio sugerido por la herramienta
FuenteSugerido	Nombre del proveedor de donde proviene el precio
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
PrecioPuntual	Es el precio del producto en la tienda de la competencia
TiendaCompetencia	Identificador numérico
CadenaCompetencia	Es la cadena a la que pertenece la tienda de la competencia
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
MedianaTienda	Es la mediana del precio en la tienda
MedianaNacional	Es la mediana del precio en la semana -2
ModaNacional	Es la moda del precio en la semana -2

5.6 preciosExcepciones

Contiene las sugerencias de precios para artículos con excepción. El nombre del archivo debe tener el siguiente formato:

preciosExcepciones_<AñoCompleto><Semana>.csv

Campo	Descripción
GrupoArticulo	Campo numérico
SKU	Identificador numérico
UPC	Código de barras
Tienda	Identificador numérico
PrecioSugerido	Precio sugerido para la semana contemplada
FuenteSugerido	Nombre del proveedor de donde proviene el precio
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
PrecioPuntual	Es el precio del producto en la tienda de la competencia
TiendaCompetencia	Identificador numérico
CadenaCompetencia	Es la cadena a la que pertenece la tienda de la competencia
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
MedianaTienda	Es la mediana del precio en la tienda
MedianaNacional	Es la mediana del precio en la semana -2
ModaNacional	Es la moda del precio en la semana -2
PrecioGrupo	Precio mínimo de ese grupo de precios en la tienda
MedianaGrupo	Es el mínimo de las medianas de los artículos que pertenecen al GrupoArticulo
ModaGrupo	Es el mínimo de las modas en el grupo
Excepcion	Descripción de la excepción del artículo
ExcepcionGrupoSKU	SKU del artículo que presenta la excepción del grupo
ExcepcionGrupo	Primera excepción presentada en alguno de los artículos del grupo

5.7 precios

Contiene las sugerencias de precios para artículos sin excepción.

El nombre del archivo debe tener el siguiente formato:
precios_<AñoCompleto><Semana>.csv

Campo	Descripción
GrupoArticulo	Grupo al que pertenece el artículo en cuestión
SKU	Identificador numérico
UPC	Código de barras
Tienda	Identificador numérico
PrecioSugerido	Precio sugerido para la semana contemplada
FuenteSugerido	Nombre del proveedor de donde proviene el precio
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
PrecioPuntual	Es el precio del producto en la tienda de la competencia
TiendaCompetencia	Identificador numérico
CadenaCompetencia	Es la cadena a la que pertenece la tienda de la competencia
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
MedianaTienda	Es la mediana del precio en la tienda
MedianaNacional	Es la mediana del precio en la semana -2
ModaNacional	Es la moda del precio en la semana -2
PrecioGrupo	Precio mínimo de ese grupo de precios en la tienda
MedianaGrupo	Es el mínimo de las medianas de los artículos que pertenecen al GrupoArticulo
ModaGrupo	Es el mínimo de las modas en el grupo
Excepcion	Descripción de la excepción del artículo
SKUGrupo	SKU del artículo que presenta la excepción del grupo
ExcepcionGrupo	Primera excepción presentada en alguno de los artículos del grupo

5.8 impacto

Contiene sugerencias de precios y los impactos con respecto al precio de regulación central y al precio promedio. El nombre del archivo debe tener el siguiente formato:

impacto_<AñoCompleto><Semana>.csv

Campo	Descripción
Tienda	Identificador numérico
SKU	Identificador numérico
GrupoArticulo	Grupo al que pertenece el artículo en cuestión
InvFinUni	Contiene el número de unidades en inventario
InvFinVta	Campo numérico
PrecioCentralTotal	Precio de regulación central con impuestos incluidos
PrecioPromedioTotal	Precio promedio de venta con impuestos incluidos al que se vendió ese artículo en la semana contemplada
UPC	Código de barras
PrecioSugerido	Precio sugerido para la semana contemplada
FuenteSugerido	Nombre del proveedor de donde proviene el precio
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
PrecioPuntual	Es el precio del producto en la tienda de la competencia
TiendaCompetencia	Identificador numérico
CadenaCompetencia	Es la cadena a la que pertenece la tienda de la competencia
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
MedianaTienda	Es la mediana del precio en la tienda
MedianaNacional	Es la mediana del precio en la semana -2
ModaNacional	Es la moda del precio en la semana -2
PrecioGrupo	Precio mínimo de ese grupo de precios en la tienda
MedianaGrupo	Es el mínimo de las medianas de los artículos que pertenecen al GrupoArticulo
ModaGrupo	Es el mínimo de las modas en el grupo
ImpactoConCentral	Es igual a InvFinUni * (PrecioSugerido - PrecioCentralTotal)
ImpactoConPromedio	Es igual a InvFinUni * (PrecioSugerido - PrecioPromedioTotal)
ExcepcionGrupo	Descripción de la excepción del grupo

Campo	Descripción
ExcepcionGrupoSKU	Es el SKU del artículo que presenta la excepción del grupo
Excepcion	Descripción de la excepción del artículo
Descripcion	Nombre del artículo
GrupoArticulos	Campo numérico
Estatus	Campo alfanumérico
Depto	Identificador numérico
DescripDepto	Nombre del departamento
SubDepto	Identificador numérico
DescripSubdepto	Nombre del subdepartamento
Clase	Identificador numérico
DescripClase	Nombre de la clase
SubClase	Identificador numérico
DescripSubCl	Nombre de la subclase
Num_Dpt	Campo numérico
DecilNacional	Campo numérico, puede tener como prefijo "D"
Nombre_tda	Nombre del centro
Formato	Formato del centro
Region	Identificador alfanumérico
Zona	Identificador alfanumérico
Tipo_Tda	Tipo del centro

5.9 impactoPromos

Es la lista de impactos para los artículos que tuvieron un precio de promoción en la competencia.

El nombre del archivo debe tener el siguiente formato:
impactoPromos_<AñoCompleto><Semana>.csv

Campo	Descripción
Tienda	Identificador numérico
SKU	Identificador numérico
GrupoArticulo	Grupo al que pertenece el artículo en cuestión
InvFinUni	Contiene el número de unidades en inventario
InvFinVta	Campo numérico
PrecioCentralTotal	Precio de regulación central con impuestos incluidos
PrecioPromedioTotal	Precio promedio de venta con impuestos incluidos al que se vendió ese artículo en la semana contemplada
UPC	Código de barras
PrecioSugerido	Precio sugerido para la semana contemplada
FuenteSugerido	Nombre del proveedor de donde proviene el precio

Campo	Descripción
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
PrecioPuntual	Es el precio del producto en la tienda de la competencia
TiendaCompetencia	Identificador numérico
CadenaCompetencia	Es la cadena a la que pertenece la tienda de la competencia
FuenteCompetencia	Nombre del proveedor de los precios de competencia, por ejemplo Nielsen. El nombre del proveedor de precios va en minúsculas
MedianaTienda	Es la mediana del precio en la tienda
MedianaNacional	Es la mediana del precio en la semana -2
ModaNacional	Es la moda del precio en la semana -2
PrecioGrupo	Precio mínimo de ese grupo de precios en la tienda
MedianaGrupo	Es el mínimo de las medianas de los artículos que pertenecen al GrupoArticulo
ModaGrupo	Es el mínimo de las modas en el grupo
ImpactoConCentral	Es la suma de InvFinUni * (PrecioSugerido - PrecioCentralTotal) para el SKU en cuestión
ImpactoConPromedio	Es la suma de InvFinUni * (PrecioSugerido - PrecioPromedioTotal) para el SKU en cuestión
ExcepcionGrupo	Descripción de la excepción del grupo
ExcepcionGrupoSKU	Es el SKU del artículo que presenta la excepción del grupo
Excepcion	Descripción de la excepción del artículo
Descripcion	Nombre del artículo
GrupoArticulos	Campo numérico
Estatus	Campo alfanumérico
Depto	Identificador numérico
DescripDepto	Nombre del departamento
SubDepto	Identificador numérico
DescripSubdepto	Nombre del subdepartamento
Clase	Identificador numérico
DescripClase	Nombre de la clase
SubClase	Identificador numérico
DescripSubCl	Nombre de la subclase
Num_Dpt	Campo numérico
DecilNacional	Campo numérico, puede tener como prefijo "D"
Nombre_tda	Nombre del centro
Formato	Formato del centro
Region	Identificador alfanumérico
Zona	Identificador alfanumérico

Campo	Descripción
Tipo_Tda	Tipo del centro

5.10 impactoConteos

Es un reporte de cuántos precios se están regulando.

El nombre del archivo debe tener el siguiente formato:
impactoConteos_<AñoCompleto><Semana>.csv

Campo	Descripción
Tienda	Identificador numérico
Prioridad	Es la prioridad de la tienda de donde se obtiene el precio
FuenteSugerido	Nombre del proveedor de donde proviene el precio
Conteo	indica cuántos artículos tienen precio sugerido

5.11 impactoNulos

Es la lista de artículos para los cuales no hubo sugerencias de precios. El nombre del archivo debe tener el siguiente formato:
impactoNulos_<AñoCompleto><Semana>.csv

Campo	Descripción
SKU	Identificador numérico
GrupoArticulo	Grupo al que pertenece el artículo en cuestión
TodosNulos	Esta variable toma dos valores, 0 y 1. Toma el valor 0 si en al menos una tienda se pudo sugerir un precio. Toma el valor 1 si en ninguna tienda se pudo sugerir precio
ExcepcionGrupo	Descripción de la excepción del grupo
ExcepcionGrupoSKU	Es el SKU del artículo que presenta la excepción del grupo
UPC	Código de barras
Descripcion	Nombre del artículo
GrupoArticulos	Campo numérico
Estatus	Campo alfanumérico
Depto	Identificador numérico
DescripDepto	Nombre del departamento
SubDepto	Identificador numérico
DescripSubdepto	Nombre del subdepartamento
Clase	Identificador numérico
DescripClase	Nombre de la clase
SubClase	Identificador numérico
DescripSubCl	Nombre de la subclase
Num_Dpt	Campo numérico

5.12 impactoSKU

Es el archivo que contiene la suma de impactos para cada SKU.

El nombre del archivo debe tener el siguiente formato:
impactoSKU_<AñoCompleto><Semana>.csv

Campo	Descripción
SKU	Identificador numérico
UPC	Código de barras
ImpactoConCentral	Es la suma de InvFinUni * (PrecioSugerido - PrecioCentralTotal) para el SKU en cuestión
ImpactoConPromedio	Es la suma de InvFinUni * (PrecioSugerido - PrecioPromedioTotal) para el SKU en cuestión
Descripcion	Nombre del artículo
GrupoArticulos	Campo numérico
Estatus	Campo alfanumérico
Depto	Identificador numérico
DescripDepto	Nombre del departamento
SubDepto	Identificador numérico
DescripSubdepto	Nombre del subdepartamento
Clase	Identificador numérico
DescripClase	Nombre de la clase
SubClase	Identificador numérico
DescripSubCl	Nombre de la subclase
Num_Dpt	Campo numérico
DecilNacional	Campo numérico, puede tener como prefijo "D"

5.13 impactoSubDepto

Guarda la suma de los impactos por subdepartamento.

El nombre del archivo debe tener el siguiente formato:
impactoSubDepto_<AñoCompleto><Semana>.csv

Campo	Descripción
Tienda	Identificador numérico
Depto	Identificador numérico
SubDepto	Identificador numérico
NoNulos	es el número de artículos con precio sugerido
Nulos	es el número de artículos sin precio sugerido
PorcentajeRegulado	es el porcentaje de artículos con precio sugerido
ImpactoConCentral	Es la suma de InvFinUni * (PrecioSugerido - PrecioCentralTotal) para el SKU en cuestión
ImpactoConPromedio	Es la suma de InvFinUni * (PrecioSugerido - PrecioPromedioTotal) para el SKU en cuestión

5.14 impactoTienda

Es el archivo que contiene la suma de impactos para cada tienda.

El nombre del archivo debe tener el siguiente formato:
impactoTienda_<AñoCompleto><Semana>.csv

Campo	Descripción
Tienda	Identificador numérico
ImpactoConCentral	Es la suma de InvFinUni * (PrecioSugerido - PrecioCentralTotal) para la tienda en cuestión
ImpactoConPromedio	Es la suma de InvFinUni * (PrecioSugerido - PrecioPromedioTotal) para la tienda en cuestión
Nombre_tda	Nombre del centro
Formato	Formato del centro
Region	Identificador alfanumérico
Zona	Identificador alfanumérico
Tipo_Tda	Tipo del centro

5.15 sumaVentas

Es la suma de las ventas en las semanas indicadas (semana inicio y semana final).

El nombre del archivo debe tener el siguiente formato, se define YearWeek como la combinación de año y semana (por ejemplo 201802):

sumaVentas_<YearWeek1>a<YearWeek2>.csv

Campo	Descripción
SKU	Identificador numérico
Tienda	Identificador numérico
TotalVta	Es la suma de las ventas en pesos en las semanas indicadas
TotalUni	Es la suma de las unidades vendidas en las semanas indicadas
Depto	Identificador numérico
SubDepto	Identificador numérico
Clase	Identificador numérico
SubClase	Identificador numérico

6 Procesos

6.1 Estructura de Archivos y Carpetas para la Herramienta

La herramienta tiene como entradas los catálogos, así como los datos de ventas, inventarios y precios de competencia. Estos datos deben de estar bajo cualquier carpeta del sistema, la herramienta solo debe conocer la ubicación de la carpeta durante su funcionamiento.

Con el fin de ser concretos, supondremos que los archivos están guardados en la carpeta /home/black/Documents/chedraui. Dentro de esta carpeta se deberán colocar todos los catálogos, y se deberán crear las carpetas inventarios/, ventas/ y preciosCompetencia/. Dentro de las últimas se deben colocar los archivos de inventarios, ventas y precios de competencia respectivamente. Adicionalmente, la herramienta deposita sus resultados dentro de la carpeta llamada resultados/, se recomienda que esta carpeta sea creada antes de correr cualquier función de la herramienta.

Una carpeta con todas las entradas necesarias para la herramienta tendrán como ejemplo los siguientes archivos:

```
/home/black/Documents/chedraui/
    cat_skus.csv
    cat_excepciones.csv
    ... (todos los archivos de catalogos)
    cat_matriz.csv
    inventarios/
        Inven_15Ene17.csv
        Inven_14Feb17.csv
        ... (todos los archivos de inventarios)
        Inven_22Dic17.csv
    ventas/
        ventas201702.csv
        ventas201703.csv
        ... (todos los archivos de ventas)
        ventas201751.csv
    preciosCompetencia/
        nielsen201702.csv
        nielsen201703.csv
        otroproveedor201703.csv
        ... (todos los archivos de precios)
        nielsen201750.csv
    resultados/
        ... (la herramienta deposita resultados)
```

6.2 Sugerencias de Precios

Una vez que los datos de entrada y los catálogos están colocados en una carpeta con la estructura especificada, la herramienta puede hacer su trabajo. El proceso

que se sigue con la herramienta para generar sugerencias de precios generalmente sigue el siguiente esquema:

1. Importación de funciones de la herramienta.
2. Lectura de catálogos con la función leeCatalogos.
3. Construcción de tablas de precios de la competencia con la función calcCompetencia.
4. Generación de tabla maestra datos_<Anio><Semana>.csv con la función calcDatos.
5. Generación de sugerencias de precios asociadas a la tabla maestra con la función calcPrecios.
6. Cruce con inventario y ventas con la función calcImpactoPrecios, en este paso también se calculan los impactos de cambio de precios.

Por ejemplo, supóngase que los datos son colocados en la carpeta /home/black/Documents/chedraui/ y se desea generar sugerencias de precios con datos de la semana 201751 utilizando el inventario del 9 de Enero del 2018. Spark debe estar instalado en la máquina proporcionada por BlackTrust, su consola se debe iniciar junto con el archivo jar de la herramienta,

```
$ start-spark
$ spark-shell --jars herramienta-chedraui_2.11-1.0.jar \
  --driver-memory 8g
```

es posible que el nombre del archivo jar sea ligeramente distinto y esta línea se debe ajustar según ese nombre.

En la práctica se recomienda partir el proceso en dos, la generación de la tabla maestra y luego el cálculo de impactos. Para la primera parte del proceso, los comandos a seguir en la consola de Spark son:

```
// Importar funciones
import org.btrust.chedrauiHerramienta.Catalogues.leeCatalogos
import org.btrust.chedrauiHerramienta.Calc.{calcCompetencia,
    calcDatos,calcPrecios,calcImpactoPrecios}
spark.conf.set("spark.sql.autoBroadcastJoinThreshold",-1)

// Carpeta de datos
val rootDir = "/home/black/Documents/chedraui"

// Agno, semana, inventario
val myYear = 2017
val myWeek = 51
val inven  = "Inven_09Ene18.csv"

// Leer catalogos
val cats  = leeCatalogos(rootDir,matriz="matriz_09Ene2018.csv")

// Generar tablas de competencia
calcCompetencia(myYear,myWeek,cats,rootDir,uri)
```

```
// Generar tabla maestra de datos
calcDatos(myYear,myWeek,cats,rootDir,uri=uri,inven=inven)
```

En este punto se recomienda cerrar la consola con Control+D y reiniciarla, el colector de basura se puede saturar después de calcDatos. Hasta ahora se han generado los archivos con prefijos "comp" y "datos." La segunda parte del proceso es generar la tabla de impactos,

```
// Importar funciones
import org.btrust.chedrauiHerramienta.Catalogues.leeCatalogos
import org.btrust.chedrauiHerramienta.Calc.{calcCompetencia,
    calcDatos,calcPrecios,calcImpactoPrecios}
spark.conf.set("spark.sql.autoBroadcastJoinThreshold",-1)

// Carpeta de datos
val rootDir = "/home/black/Documents/chedraui"

// Agno, semana, inventario
val myYear = 2017
val myWeek = 51
val inven  = "Inven_09Ene18.csv"

// Leer catalogos
val cats  = leeCatalogos(rootDir,matriz="matriz_09Ene2018.csv")

// Generar tabla de precios sugeridos
calcPrecios(myYear,myWeek,cats,rootDir,uri)

// Generar tabla de impactos, cruce de precios con inventario
calcImpactoPrecios(myYear,myWeek,cats,inven,rootDir)
```

Como en este caso se reinició la consola, se tienen que declarar las variables y leer los catálogos de nuevo. Al terminar el proceso, se generan los archivos con prefijo "precios" e "impactos."

6.3 Extracción de Resultados

Debido al funcionamiento interno de la herramienta y sus dependencias los resultados se depositan en el disco duro en particiones (archivos part-). Los archivos CSV resultantes son en realidad carpetas que contienen las particiones y los encabezados de estos archivos.

Junto con la herramienta, se ha incluido una colección de scripts que permiten la extracción de estas carpetas en un solo archivo consolidado. Los scripts consisten en la lectura y consolidación de las particiones y encabezados de los resultados.

Los scripts son sc_listaCSV.bash, sc_extraeCSV.bash y sc_extraeTodo.bash. El archivo README incluido con ellos detalla el uso de estos scripts en la terminal.