SolidQ

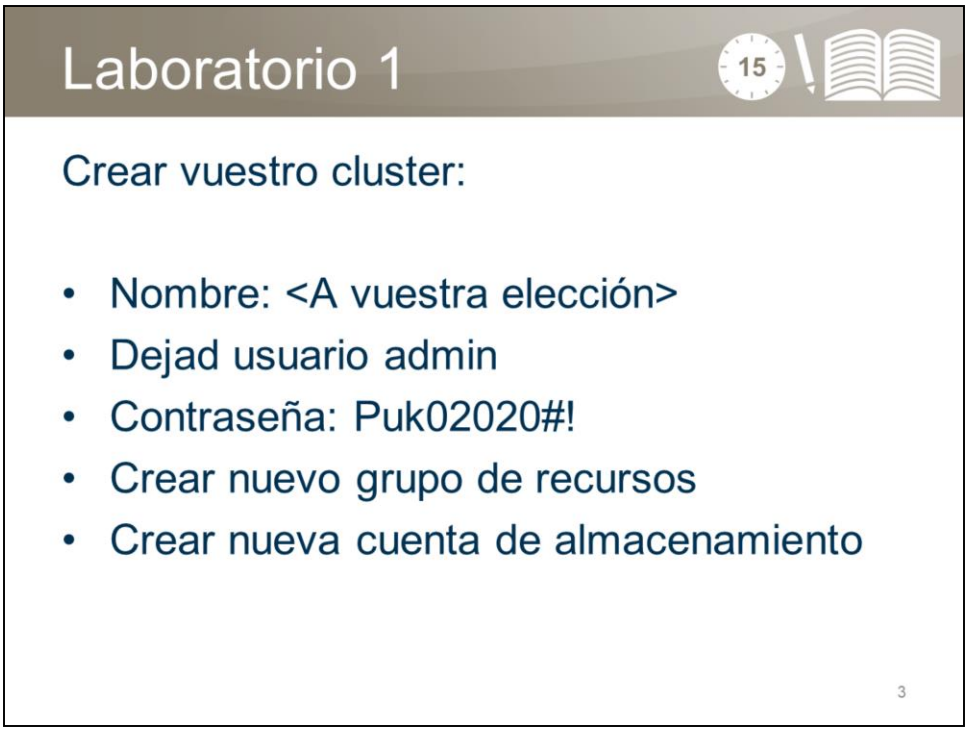
BIG DATA: Diseño y Arquitectura de Soluciones con Hadoop, Spark y R

Antonio Soto
CEO
asoto@solidq.com

Vitae

Laboratorios

- Laboratorio 1: Creación de un cluster HDInsight
- Laboratorio 2: HDFS
- Laboratorio 3: HIVE



En este laboratorio se creará un cluster Hadoop desde el portal de Azure y se verá las opciones para interactuar con el cluster.

Ejercicio 1: Crear el cluster

- Conecta al portal azure https://portal.azure.com
- Inicia sesión con el usuario que se te haya asignado
- Arriba a la izquierda tienes la opción *Crear un Recurso*, selecciónala
- En el cuadro de buscar, pon HDInsight. Selecciona entre los resultados la opción ***HDInisght***, y dale a *Crear*
- En *Nombre del cluster* pon tu nombre de usuario
- En *Tipo de cluster* selecciona Hadoop y dale a Seleccionar
- En contraseña, pon la contraseña asignada
- En grupo de recursos selecciona crear uno nuevo y dale un nombre, dale a *Siguiente*
- En Seleccionar una cuenta de storage seleccionar crear uno nuevo y dale un nombre .
- Se presentará el resumen en el que puedes darle a Crear para crear el

cluster. Fíjate que podrías descargar una plantilla de despliegue para poder desplegarlo de forma automatizada.

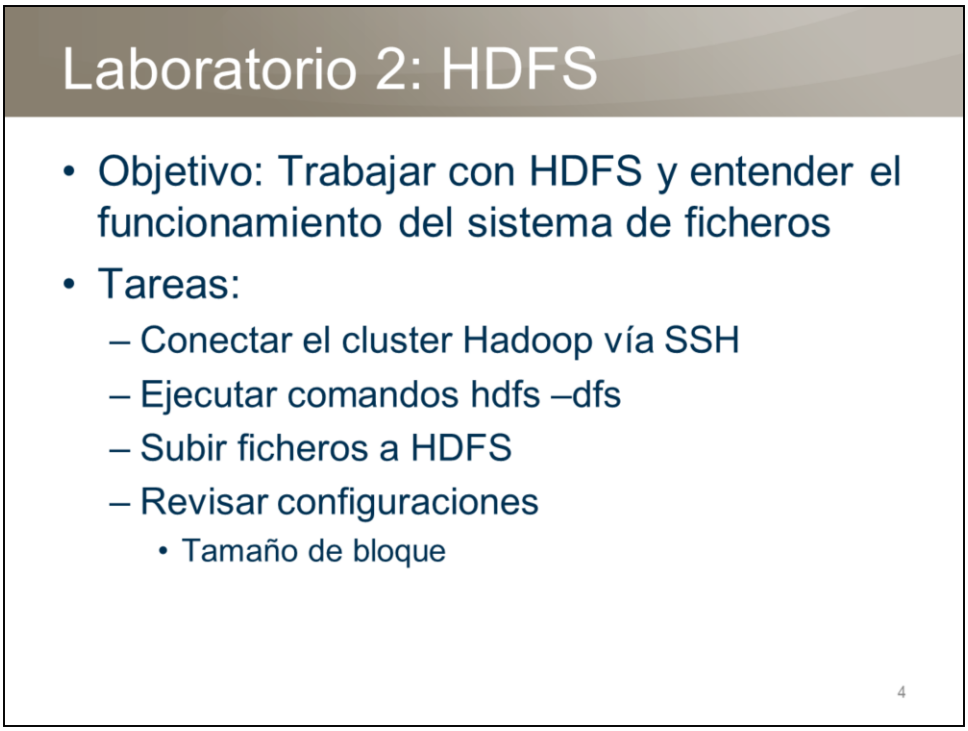
Ejercicio 2: Entender el entorno

Una vez desplegado el cluster, vamos a ver que opciones tenemos en el entorno Azure y posteriormente en nuestro cluster ya creado. En el menú de la izquierda busca *Todos los recursos* y busca tu cluster de HDInsight y hazle clic. SE abrirán las opciones de administración del recurso Azure. Podemos ver información sobre los nodos del cluster, cores disponibles, etc.

- En la parte izquierda buscamos la opción ***SSH e inicio de sesión del cluster***. Seleccionamos el nodo que nos aparece y copiamos la información de conexión.
- Desde un cliente SSH conectamos con esas credenciales.

Ejercicio 3: Configurando nuestro equipo para acceso SSH y Tunnel

Sigue las instrucciones del documento https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-linux-ambari-ssh-tunnel para configurar un túnel SSH y poder acceder a todas las opciones de administración del portal Ambari



En este segundo laboratorio, explorarás las opciones disponibles en HDFS para gestionar ficheros y el almacenamiento HDFS.

Ejercicio 1: El Comando hdfs dfs

Familiarízate con el comando hdfs dfs:

- Ejecuta hdfs dfs y revisa las opciones
- Ejecuta hdfs dfs –help ls
- Crea un directorio llamado curso y comprueba que se ha creado
- Explora donde se encuentra realmente el almacenamiento en el nodo al que estás conectado. Vete a /Hadoop/hdfs/namenode

Ejercicio 2: Copiando ficheros

Al trabajar con una estructura “virtual” HDFS es necesario copiar las ficheros desde su almacenamiento original al HDFS y vicecersa . Para ello utilizaremos el comando –copyFromLocal para subir a HDFS ficheros y el –copyToLocal para bajar ficheros desde HDFS al nodo local:

1. Comprobamos que existe un fichero .log en /home/hive
    1. ls /home/hive
2. Copiamos el fichero
    1. hdfs dfs –copyFromLocal /home/hive/*.* /curso/
3. Comprobamos que se ha copiado
    1. hdfs dfs –ls /curso

Ejercicio 3: Generando HAR

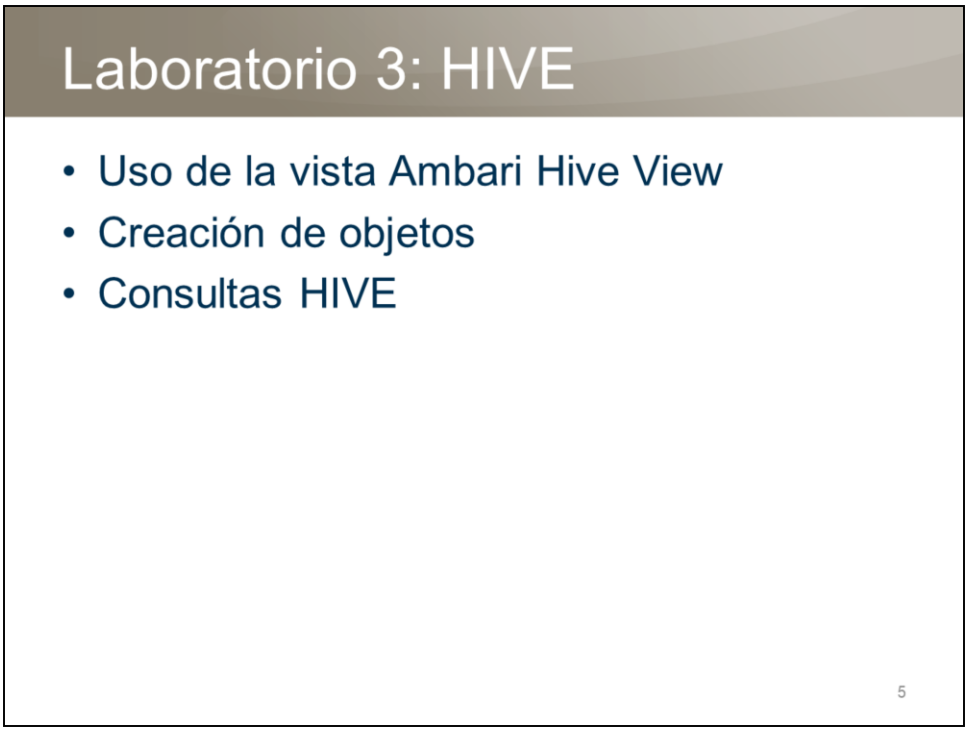
Con lo visto durante el módulo, genera un archivo HAR a partir de los archivos que se encuentran en /example/data. Almacena el resultado en la carpeta “archivos”

Ejercicio 4: Revisando configuración

Revisa el fichero /etc/Hadoop/conf/hdfs-site.xml

Ejercicio 5: Aprovechando Azure

- En el portal Azure busca la cuenta de almacenamiento
- Vete a blob
- Ahí verás la estructura HDFS
- Carga el fichero weblogs.csv en una carpeta llamada /curso/weblogs



En este laboratorio de HIVE veremos como interactuar con el entorno para crear objetos, consultar objetos etc.

Para esta laboratorio es necesario que el archivo weblogs.csv esté en el directorio data

Ejercicio 1: Las bases de HIVE

**** *Es necesario cargar previamente el fichero weblogs.csv en la carpeta /curso/weblogs de la infraestructura HDFS* ****

Realizaremos este laboratorio desde la Vista HIVE de Ambari, que nos permite interactuar gráficamente con el entorno. Conéctate y lanza la vista de usuario. Desde ahí ejecuta:

```
DROP DATABASE IF EXISTS HDILABDB CASCADE;
CREATE DATABASE HDILABDB;
Use HDILABDB;
CREATE EXTERNAL TABLE IF NOT EXISTS weblogs(
```

```
    TransactionDate varchar(50) ,
    CustomerId varchar(50) ,
    BookId varchar(50) ,
    PurchaseType varchar(50) ,
    TransactionId varchar(50) ,
    OrderId varchar(50) ,
    BookName varchar(50) ,
    CategoryName varchar(50) ,
    Quantity varchar(50) ,
    ShippingAmount varchar(50) ,
    InvoiceNumber varchar(50) ,
    InvoiceStatus varchar(50) ,
    PaymentAmount varchar(50)
) ROW FORMAT DELIMITED FIELDS TERMINATED by ',' lines
TERMINATED by '\n'
STORED AS TEXTFILE LOCATION ‘/curso/weblogs’;
```

Con esto, tendríamos una tabla externa. Comprueba que el fichero continúa en su ubicación original, y que puedes consultar el contenido de la tabla weblogs.

Ahora:

- Elimina la tabla
- Créala de nuevo, pero como tabla interna
- Carga el fichero con el comando:

```
LOAD DATA INPATH ‘/curso/weblogs/weblogs.csv' INTO TABLE
HDILABDB.weblogs
```

Lo que creará la tabla weblogs en la base de datos HDILABDB. Puedes comprobar que se ha creado ejecutando:

```
hdfs dfs -ls /hive/warehouse
```

Ejercicio 2: Algunas consultas

Ejecuta las siguientes consultas:

```
SELECT COUNT(*) FROM HDILABDB.weblogs;

SELECT * FROM HDILABDB.weblogs LIMIT 5;

SELECT * FROM HDILABDB.weblogs WHERE orderid='107';
```

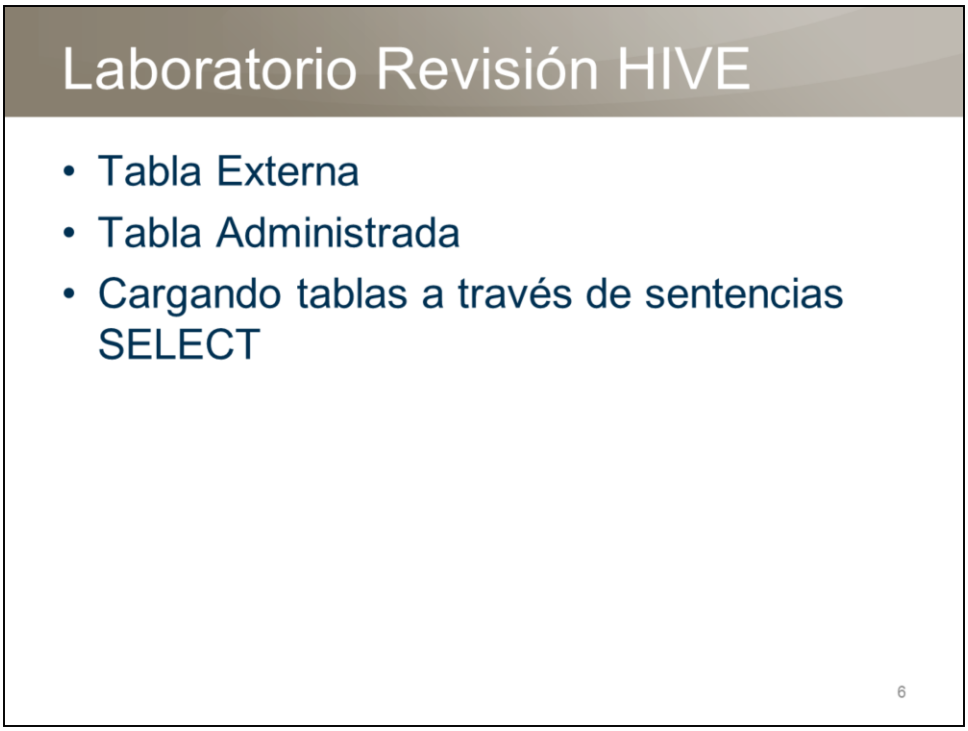
Crea un nuevo *Worksheet* y ejecuta:

```
SELECT DISTINCT bookname FROM HDILABDB.weblogs WHERE
orderid='107';

SELECT bookname,COUNT(*) FROM HDILABDB.weblogs GROUP BY
bookname;
```

Ejercicio 3: Problemas

1. Ejecuta una consulta que devuelva el total pagado para cada categoría por mes
2. Ejecuta una consulta que devuelva la cantidad total pagada y la cantidad vendida de cada libro
3. Escribe una consulta que devuelva el top 3 de libros vistos por los usuarios que también visitaron **THE BOOK OF WITNESSES**



En este laboratorio vamos a repasar los conceptos básicos de HIVE. Para ello utilizaremos el fichero de ejemplo del directorio /HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog.

Ejercicio 1: Tabla Externa

- Revisamos desde la consola el contenido del directorio con el comando hdfs dfs –ls /HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog
- Revisamos el contenido del almacén de HIVE con el comando hdfs dfs –ls /hive/warehouse
- Creamos la tabla externa a través del comando:

```
CREATE EXTERNAL TABLE samplelog(
  fecha date,
  time varchar(200),
  sitename varchar(200),
  method varchar(200),
  uristem varchar(200),
```

```
  uriquery varchar(200),
  port varchar(200),
  username varchar(200),
  ip varchar(200),
  UserAgent varchar(200),
  Cookie varchar(200),
  Referer varchar(200),
  host varchar(200),
  status varchar(200),
  substatus varchar(200),
  win32substatus varchar(200),
  scbytes int,
  csbytes int,
  timetaken int)
  ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '
STORED AS TEXTFILE LOCATION
'/HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog'
tblproperties ("skip.header.line.count"="2");
```

- Ejecutamos SELECT * FROM samplelog; y obtenemos el resultado deseado
- Ejecutamos SELECT COUNT(*) from samplelog; y anotamos el número de filas
- Copiamos el fichero de origen con el comando hdfs dfs -cp /HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog/909f2b.log /HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog/909f2c.log
- Ejecutamos de nuevo SELECT COUNT(*) from samplelog; y obtendremos el doble de filas
- Si volvemos a consultar el contenido de los directorios, tanto del almacen de hive, como el origen, nada ha cambiado
- Eliminamos el fichero que hemos copiado /HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog/909f2c.log
- Ejecutamos de nuevo SELECT (COUNT*) from samplelog; y ha desaparecido el contenido de ese fichero

- Eliminamos la tabla externa DROP TABLE samplelog;

Como hemos podido observar la tabla externa no realiza ninguna operación con los datos origen. Tan solo muestra en formato tabla el contenido de un directorio. Los datos se “cargan” en el momento en el que se consultan, aplicando las “transformaciones” que se hayan definido en la consulta CREATE TABLE

Ejercicio 2: Tabla Administrada

Vamos a crear ahora una tabla administrada sobre los mismos datos de ejemplo. Para ello ejecutamos el comando:

```
CREATE TABLE samplelogadmin(
  fecha date,
  time varchar(200),
  sitename varchar(200),
  method varchar(200),
  uristem varchar(200),
  uriquery varchar(200),
  port varchar(200),
  username varchar(200),
  ip varchar(200),
  UserAgent varchar(200),
  Cookie varchar(200),
  Referer varchar(200),
  host varchar(200),
  status varchar(200),
  substatus varchar(200),
  win32substatus varchar(200),
  scbytes int,
  csbytes int,
  timetaken int)
```

ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '
tblproperties ("skip.header.line.count"="2");

- Consultamos el contenido de la tabla y vemos que está vacía
- Ejecutamos: LOAD DATA INPATH '/HdiSamples/HdiSamples/WebsiteLogSampleData/SampleLog/909f2b.log' INTO TABLE samplelogadmin
- Consultamos los directorios de origen y fichero y de almacén de Hive (hive/warehouse) ¿Qué ha ocurrido?

Ejercicio 3: Cargando tablas desde sentencias SELECT

En este ejercicio crearemos dos tablas externas sobre las que después generaremos otras dos tablas con contenido agregado que cargaremos a través de sentencias CREATE TABLE----- SELECT------

Ejecuta el siguiente script:

DROP TABLE IF EXISTS hvac;

--crear la tabla externa hvac sobre el csv
CREATE EXTERNAL TABLE hvac(dates STRING, time STRING, targettemp BIGINT,
actualtemp BIGINT, system BIGINT, systemage BIGINT, buildingid BIGINT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/HdiSamples/HdiSamples/SensorSampleData/hvac/';

DROP TABLE IF EXISTS building;

--crear la tabla externa building sobre el csv
CREATE EXTERNAL TABLE building(buildingid BIGINT, buildingmgr STRING,
buildingage BIGINT, hvacproduct

```
STRING, country STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION
'/HdiSamples/HdiSamples/SensorSampleData/building/';

DROP TABLE IF EXISTS hvac_temperatures;

--crear la tabla administrada hvac_temperatures desde la tabla hvac. Mostrar
donde se almacena esa tabla
CREATE TABLE hvac_temperatures AS
SELECT *, targettemp - actualtemp AS temp_diff,
        IF((targettemp - actualtemp) > 5, 'COLD',
        IF((targettemp - actualtemp) < -5, 'HOT', 'NORMAL')) AS
temprange,
        IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp)
< -5, '1', 0)) AS extremetemp
FROM hvac;

DROP TABLE IF EXISTS hvac_building;

-- crear la tabla hvac_building haciendo join entre las tablas de building y
hvac_temperatures
CREATE TABLE hvac_building AS
SELECT h.*, b.country, b.hvacproduct, b.buildingage, b.buildingmgr
FROM building b JOIN hvac_temperatures h ON b.buildingid = h.buildingid;
```

Comprueba que se han cargado datos en las dos últimas tablas a partir del resultado de las consultas SELECT. ¿Dónde se han almacenado?

Limpiamos el entorno:

```
DROP TABLE hvac;
DROP TABLE building;
```

```
DROP TABLE hvac_temperaturas;
DROP TABLE hvac_Building;
```


SolidQ
www.solidq.com
info@solidq.com