



Big Data Architecture

Big Data Machine Learning Bootcamp

Enunciado de la práctica:

Diseñar, especificar y desplegar un datalake para el procesamiento de datos provenientes de fuentes de datos no estructurados extraídos mediante técnicas de scraping/crawling de sitios de dominio público.

PARTE 1: Enunciado y diagrama

- **IDEA DEL PROYECTO:**

Vamos a cruzar los datos del dataset inicial de airbnb con los museos de Madrid. De esta forma podremos recomendar los pisos más próximos al museo elegido, más baratos y mejor valorados.

- **DEFINICIÓN DE LA ESTRATEGIA DEL DATA ANALYSIS AS A SERVICE (DAaaS):**

Se parte de un dataset ya **scrapeado** de la web de airbnb con los pisos de Madrid. Se añade otro fichero de entrada que se obtiene haciendo scraping en la API REST del ayuntamiento de Madrid (datos.madrid.es). Este contiene el listado de museos de la ciudad de Madrid.

Ambos ficheros (extensión csv) **se suben a la nube**: Google Cloud Platform (GCP) y allí **se almacenan en un segmento** (Google Cloud Storage), **se depuran** para limpiarlos (Google Cloud Dataprep by Trifacta) **y se procesan con HIVE en un clúster de Hadoop** (Google Cloud Dataproc).

- **ARQUITECTURA DEL DAaaS:**

Para hacer el scraping se usa un Google Collaboratory que llama al API REST y almacena los datos en un fichero csv.

Ambos ficheros de entrada (airbnb y museos) se suben a un segmento del GCS.

A continuación, se usa el Dataprep de Trifacta (dentro de GCP) para limpiar los datos iniciales. El resultado se guarda en otro directorio dentro del mismo segmento del GCS.

Estos datos se cargan en el clúster de Hadoop para poder ser procesados. Dicho clúster está formado por 3 nodos: uno máster y dos de trabajo. El máster contiene un Resource Manager de YARN y un NameNode de HDFS además de los controladores de tareas. Por su parte los nodos de trabajo tienen un NodeManager de YARN y un DataNode de HDFS.

Al tratarse de datos estructurados (.csv) se usa HIVE, de esta forma se pueden crear y cargar las tablas, y se puede interactuar con ellos para hacer consultas. Estos resultados finales vuelven a almacenarse en el mismo segmento de trabajo de GCS.

● **MODELO OPERATIVO DEL DAaaS:**

Los ficheros de entrada los obtenemos por dos vías diferentes:

- El de airbnb está disponible en el link facilitado en el enunciado de la práctica ([dataset_airbnb](#)). En este caso se utiliza la versión corta de 58 MB (solo 14780 grabaciones seleccionadas) ya que el fichero completo de 1.8 GB no tiene ninguno de dicha ciudad. Son pisos principalmente de Amsterdam (Holanda) y Barcelona (España).
- El de museos de Madrid se obtiene ejecutando de forma manual un Google Collaboratory. Se ha codificado en python un pequeño código que realiza un scraping del API REST del ayuntamiento de Madrid (datos.madrid.es) y devuelve los datos en un fichero csv.

Ambos ficheros se suben de manera manual a un segmento del GCS. Para ello se usan los directorios `'gs://segmento-airbnb-museos-madrid/input_airbnb'` y `'gs://segmento-airbnb-museos-madrid/input_museos'` respectivamente.

A continuación se usa un flujo de Trifacta (Dataprep en GCP). Se aplican unas recetas y de esta forma depuramos ambos ficheros. Se eliminan todos los pisos que no pertenecen a Madrid y todos los museos que tienen un error de formato. Estos ficheros resultantes se vuelven a guardar dentro del mismo segmento en un directorio llamado `'gs://segmento-airbnb-museos-madrid/ficheros_depurados'`. Además, por motivos de seguridad y eficiencia se guarda una copia de todos los ficheros que hemos tratado hasta ahora (los dos originales de entrada y los dos depurados). Esto se realiza en los directorios del segmento: `'gs://segmento-airbnb-museos-madrid/backup_entradas'` y `'gs://segmento-airbnb-museos-madrid/backup_depurados'`

El siguiente paso es crear manualmente un clúster de Hadoop con 3 nodos (uno maestro y dos trabajadores) y asociarlo con dicho segmento donde están guardados los datos depurados. A continuación se crean manualmente tareas de HIVE o directamente se puede usar una consola en la nube para utilizar beeline. De esta forma se crean y cargan las tablas, así como se ejecuta la consulta que cruza los museos con los pisos más cercanos, más baratos y mejor valorados. Estos resultados se almacenan en un directorio del mismo segmento llamado: `'gs://segmento-airbnb-museos-madrid/outputs'`.

Llegados a este punto se elimina el clúster de la CGP para no consumir recursos, pero se tienen disponibles los resultados en el segmento para que se puedan usar en cualquier momento.

- **DESARROLLO DEL DAaaS:**

Los datos de entrada son los siguientes:

- [Dataset airbnb](#)
- Listado de museos de Madrid obtenido del siguiente scraping: [Scraping data madrid Museos.ipynb](#)

Se suben a la nube, en concreto al segmento “segmento-airbnb-museos-madrid” (he dado permisos a ricardovegas@gmail.com). [Link del segmento](#)

Dicho segmento tiene la siguiente estructura de directorios:

<input type="checkbox"/>	Nombre	Tamaño	Tipo
<input type="checkbox"/>	backup_depurados/	—	Carpeta
<input type="checkbox"/>	backup_entradas/	—	Carpeta
<input type="checkbox"/>	ficheros_depurados/	—	Carpeta
<input type="checkbox"/>	google-cloud-dataproc-metainfo/	—	Carpeta
<input type="checkbox"/>	input_airbnb/	—	Carpeta
<input type="checkbox"/>	input_museos/	—	Carpeta
<input type="checkbox"/>	output_wordcount/	—	Carpeta
<input type="checkbox"/>	outputs/	—	Carpeta

Los ficheros originales de entrada se guardan en /input_airbnb e /input_museos.

Los ficheros depurados en dataprep se almacenan en /ficheros_depurados.

Todos estos ficheros tienen una copia de seguridad en /backup_entradas y /backup_depurados.

Por último, las salidas de las consultas de HIVE están disponibles en /outputs.

El trabajo en HIVE se desarrolla de la siguiente manera:

- Se crea la tabla museos y se cargan los datos depurados:

```
CREATE TABLE museos (Nombre STRING, URL_info STRING, Ciudad STRING,
Codigo_postal STRING, Direccion STRING, Latitud STRING, Longitud STRING) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
LOAD                                DATA                                INPATH
'gs://segmento-airbnb-museos-madrid/ficheros_depurados/Copia de Copia de
museos depurado' INTO TABLE museos;
```

- Se crea la tabla airbnb y se cargan los datos depurados:

```
CREATE TABLE airbnb (ID INT, Listing_Url STRING, Scrape_ID STRING,
Last_Scraped STRING, Name STRING, Summary STRING, Space STRING, Description
STRING, Experiences_Offered STRING, Neighborhood_Overview STRING, Notes
STRING, Transit STRING, Access STRING, Interaction STRING, House_Rules
STRING, Thumbnail_Url STRING, Medium_Url STRING, Picture_Url STRING,
XL_Picture_Url STRING, Host_ID STRING, Host_URL STRING, Host_Name STRING,
Host_Since STRING, Host_Location STRING, Host_About STRING,
Host_Response_Time STRING, Host_Response_Rate STRING, Host_Acceptance_Rate
STRING, Host_Thumbnail_Url STRING, Host_Picture_Url STRING,
Host_Neighbourhood STRING, Host_Listings_Count STRING,
Host_Total_Listings_Count STRING, Host_Verifications STRING, Street STRING,
Neighbourhood STRING, Neighbourhood_Cleansed STRING,
Neighbourhood_Group_Cleansed STRING, City STRING, State STRING, Zipcode
STRING, Market STRING, Smart_Location STRING, Country_Code STRING, Country
STRING, Latitude STRING, Longitude STRING, Property_Type STRING, Room_Type
STRING, Accommodates STRING, Bathrooms STRING, Bedrooms STRING, Beds STRING,
Bed_Type STRING, Amenities STRING, Square_Feet STRING, Price FLOAT,
Weekly_Price STRING, Monthly_Price STRING, Security_Deposit STRING,
Cleaning_Fee STRING, Guests_Included STRING, Extra_People STRING,
Minimum_Nights INT, Maximum_Nights INT, Calendar_Updated STRING,
Has_Availability STRING, Availability_30 STRING, Availability_60 STRING,
Availability_90 STRING, Availability_365 STRING, Calendar_last_Scraped
STRING, Number_of_Reviews STRING, First_Review STRING, Last_Review STRING,
Review_Scores_Rating STRING, Review_Scores_Accuracy STRING,
Review_Scores_Cleanliness STRING, Review_Scores_Checkin STRING,
Review_Scores_Communication STRING, Review_Scores_Location STRING,
Review_Scores_Value STRING, License STRING, Jurisdiction_Names STRING,
Cancellation_Policy STRING, Calculated_host_listings_count STRING,
Reviews_per_Month STRING, Geolocation STRING, Features STRING) ROW FORMAT
DELIMITED FIELDS TERMINATED BY ':::';
```

```
LOAD                                DATA                                INPATH
'gs://segmento-airbnb-museos-madrid/ficheros_depurados/Copia de Copia de
airbnb-listings-depurado.csv' INTO TABLE airbnb;
```

En este punto he de decir que he tenido muchos problemas para trabajar con la tabla `airbnb`. Después de numerosas pruebas con el fichero original, con el depurado, con varios tipos de separadores, con las comillas como envoltorio de los campos, etcétera no consigo una tabla legible en todos sus campos, por lo que me es imposible hacer ningún tipo de JOIN. Mi idea es usar los campos `airbnb.city` y `museos.ciudad` para cruzar dichas tablas. De esta forma, usando las latitudes y longitudes de pisos y museos podría calcular la distancia entre ellos. Consulta tipo:

```
select 2 * 6371000 * asin(sqrt(pow(sin(radians((cast(airbnb.latitude as
decimal(20,10)) - cast(museos.latitud as decimal(20,10))) / 2)),2)
+      cos(radians(cast(museos.latitud as decimal(20,10)))) *
cos(radians(cast(airbnb.latitude as decimal(20,10)))) *
pow(sin(radians((cast(airbnb.longitude as decimal(20,10)) -
cast(museos.longitud as decimal(20,10))) / 2)), 2))) as distance from airbnb
where airbnb.latitude is not null;
```

Al campo distancia habría que añadir otros como los nombres, los identificadores, el precio y las valoraciones para hacer la consulta final y de ahí obtener el top de pisos deseados. El resultado de dicha sentencia se guarda en el segmento de trabajo de la siguiente forma. A modo de ejemplo:

```
INSERT OVERWRITE DIRECTORY 'gs://segmento-airbnb-museos-madrid/outputs/' ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT a.id, a.latitude,
a.longitude, m.nombre, m.Latitud, m.Longitud from airbnb a left join museos
m on a.City = m.Ciudad where a.id>1;
```

- **DIAGRAMA:**

[Diagrama práctica BDA](#)

PARTE 2: Scraper

Se ha creado un scraper en Google Collaboratory a partir del API REST de la web del ayuntamiento de Madrid. De aquí se obtiene un archivo `.csv` que contiene el listado de museos de dicha ciudad.

[Scraping data madrid Museos.ipynb](#)

PARTE 3: Proveedor de Cloud y clúster

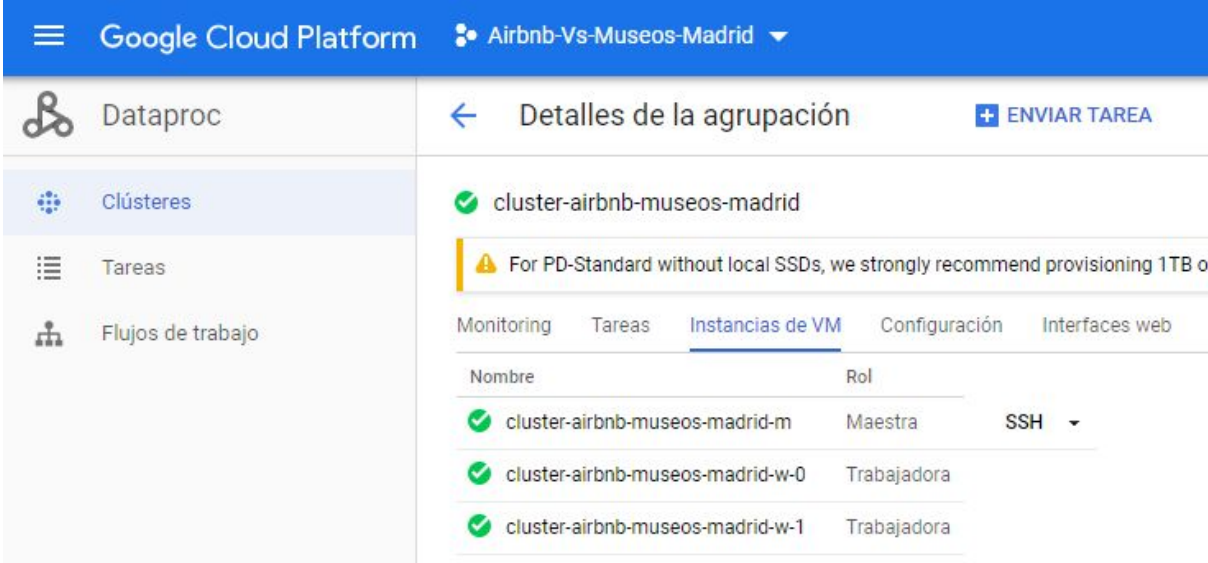
Estamos trabajando en la nube con Google Cloud Platform (GCP) y gracias a Dataproc hemos creado un clúster de 3 nodos: uno maestro y dos de trabajo.

El nodo maestro contiene un YARN Resource Manager, un HDFS NameNode y todos los controladores de tareas. Por su parte los de trabajo tienen un YARN NodeManager y un HDFS DataNode cada uno.

Además le hemos asignado el segmento que ya hemos creado previamente y en el que tenemos almacenados los datos de entrada de airbnb y museos Madrid.

A continuación añadimos una regla de cortafuegos (red de VPC del GCP) donde indicamos la ip de nuestra máquina y los puertos 8088 y 9870 (YARN y HDFS) del clúster para que podamos entrar así en las vistas de administración del clúster.

Clúster creado: “cluster-airbnb-museos-madrid”



The screenshot shows the Google Cloud Platform console interface. The top navigation bar includes the Google Cloud Platform logo and the project name 'Airbnb-Vs-Museos-Madrid'. The left sidebar contains navigation links for 'Dataproc', 'Clústeres', 'Tareas', and 'Flujos de trabajo'. The main content area is titled 'Detalles de la agrupación' and shows the cluster 'cluster-airbnb-museos-madrid' with a green status icon. A warning message states: 'For PD-Standard without local SSDs, we strongly recommend provisioning 1TB o'. Below this, there are tabs for 'Monitoring', 'Tareas', 'Instancias de VM', 'Configuración', and 'Interfaces web'. The 'Instancias de VM' tab is active, displaying a table of VM instances.

Nombre	Rol
cluster-airbnb-museos-madrid-m	Maestra
cluster-airbnb-museos-madrid-w-0	Trabajadora
cluster-airbnb-museos-madrid-w-1	Trabajadora

PARTE 4: Hadoop (HDFS, YARN y MAPREDUCE)

Como ya hemos visto en el apartado anterior, al crear el clúster lo hemos asociado con el segmento donde tenemos almacenados los ficheros con los que vamos a trabajar. De esta forma no nos hace falta insertarlos personalmente en el HDFS, sino que prácticamente es transparente para nosotros y de forma automática los ficheros del segmento pasan al HDFS del clúster en el que estamos trabajando.

Segmento creado: “segmento-airbnb-museos-madrid”

Google Cloud Platform Airbnb-Vs-Museos-Madrid

Storage

Navegador

Transferencia

Transferencia de datos in situ

Transfer Appliance

Configuración

← Detalles del segmento EDITAR SEGMENTO

segmento-airbnb-museos-madrid

Objetos Visión general Permisos Bloqueo del segmento

Subir archivos Subir carpeta Crear carpeta Administrar retenciones Eliminar

Filtrar por prefijo...

Segmentos / segmento-airbnb-museos-madrid

<input type="checkbox"/>	Nombre	Tamaño	Tipo
<input type="checkbox"/>	backup_depurados/	—	Carpeta
<input type="checkbox"/>	backup_entradas/	—	Carpeta
<input type="checkbox"/>	ficheros_depurados/	—	Carpeta
<input type="checkbox"/>	google-cloud-dataproc-metainfo/	—	Carpeta
<input type="checkbox"/>	input_airbnb/	—	Carpeta
<input type="checkbox"/>	input_museos/	—	Carpeta
<input type="checkbox"/>	output_wordcount/	—	Carpeta
<input type="checkbox"/>	outputs/	—	Carpeta

Vinculación entre clúster y segmento:

Google Cloud Platform Airbnb-Vs-Museos-Madrid

Dataproc

Clústeres

CREAR CLÚSTER ACTUALIZAR ELIMINAR REGIONES

Pulsa Intro para buscar clusters

<input type="checkbox"/>	Nombre	Región	Zona	Número total de nodos de trabajo	Eliminación programada	Segmento de aplicación de fases de Cloud Storage
<input checked="" type="checkbox"/>	cluster-airbnb-museos-madrid	europe-west1	europe-west1-b	2	Desactivado	segmento-airbnb-museos-madrid

A continuación, a modo de ejemplo, utilizamos WordCount para ejecutar una tarea dentro de Dataproc de GCP.

Creación de la tarea:

Google Cloud Platform

Airbnb-Vs-Museos-Madrid

Dataprocc

Clústeres

Tareas

Flujos de trabajo

←

Enviar una tarea

ID de tarea

wordcount-airbnb

Región

europa-west1

Clúster

cluster-airbnb-museos-madrid

Tipo de tarea

Hadoop

Clase principal o .jar

file:///usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar

Argumentos (Opcional)

wordcount

gs://segmento-airbnb-museos-madrid/ficheros_depurados/jobrun/input_airbnb.csv

gs://segmento-airbnb-museos-madrid/output_wordcount

Pulsa <Intro> para añadir más argumentos

Archivos .jar (Opcional)

Introduce la ruta del archivo, por ejemplo hdfs://example/example.jar

Propiedades (Opcional)

+ Añadir elemento

Etiquetas (Opcional)

+ Añadir etiqueta

N.º máximo de reinicios por hora (Opcional)

Déjalo en blanco si no quieres permitir que se lleven a cabo reinicios automáticos cuando fallen las tareas. Más información

1-10

Enviar

Cancelar

Resultados de la tarea (dentro del segmento):

Google Cloud Platform

Airbnb-Vs-Museos-Madrid

Storage

Navegador

Transferencia

Transferencia de datos in situ

Transfer Appliance

Configuración

← Detalles del segmento

EDITAR SEGMENTO

segmento-airbnb-museos-madrid

Objetos

Visión general

Permisos

Bloqueo del segmento

Subir archivos









Subir carpeta

Crear carpeta

Administrar retenciones

Filtrar por prefijo...

Segmentos / segmento-airbnb-museos-madrid / output_wordcount

<input type="checkbox"/>	Nombre	Tamaño	Tipo
<input type="checkbox"/>	 _SUCCESS	0 B	application/octet-stream
<input type="checkbox"/>	 part-r-00000	2,33 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00001	2,32 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00002	2,32 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00003	2,37 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00004	2,4 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00005	2,38 MB	application/octet-stream
<input type="checkbox"/>	 part-r-00006	2,33 MB	application/octet-stream