Práctica Bigdata-Architecture (César Portales Alonso).

# Tabla de Contenidos

[**Tabla de Contenidos**](#_gsjbcic16bgg) **1**

[**Estrategia.**](#_cvm59pfe24o4) **1**

[**Arquitectura**](#_ix1l7l2hv7hf) **2**

[**Operativa**](#_n2x2tcm22spw) **3**

[**Desarrollo**](#_6bmkaf70fcr7) **4**

# 

# Estrategia.

Una empresa americana especializada en gestionar viajes a España desea mostrar en su sitio web una página que contenga una serie de cuadros de mando que proporcionen información sobre hospedaje en varias ciudades españolas (Madrid, Barcelona, Palma de Mallorca.

Utilizaremos como principal tecnología Google Cloud Platform, tanto para el tratamiento de los datos como para su almacenamiento.

Para ello contactaremos con varias empresas que gestionan alojamientos en las ciudades requeridas, y contrataremos API’s para acceder a la información precisa (Airbnb, Booking, y Rumbo). En el presente documento nos centraremos en la gestión de la información proporcionada por airbnb.

Procesaremos estos datos (filtrado, limpieza, etc), los cargaremos en Google cloud y mediante Tableau crearemos los cuadros de mando necesarios.

En este documento nos centraremos en airbnb. En este caso, hemos contratado una API que nos permitirá descargar un archivo (con extensión csv) con toda la información de los alojamientos de que disponen.

# Arquitectura

* **Google Cloud Platform**. Tanto para el tratamiento como para el almacenamiento de la información. Para ello montaremos un cluster con tres nodos.
* **Hadoop MapReduce con Word Count**. Esto nos permitirá poder hacer un procesamiento distribuído de la información. Los datos intermedios se almacenarán en HDFS y los “definitivos” (los que se mostrarán en el cuadro de mando) se guardarán en Google Cloud.
* **MongoDb**. El archivo proporcionado por airbnb contiene multitud de información que no nos va a ser útil (porque el cliente no la precisa) y, en cambio, hay información que nos han pedido que mostremos (por ejemplo, actividades que se pueden realizar en la ciudad).

De modo que, en fase de preproducción tendremos que realizar un tratamiento de datos que consistirá en dividir la información en dos ficheros diferentes. Uno contendrá información que entendemos que no es probable que sufra cambios (o al menos que no lo haga con frecuencia (el ID, el nombre, la descripción, la ciudad, la dirección, la geolocalización, …) y otro que contenga información que es lógico que pueda cambiar (obviamente la disponibilidad, pero también datos como el nombre del propietario, el teléfono, número de camas o de inquilinos, etc.).

Con esto esperamos reducir los tiempos de actualización de la información. Con vistas a las futuras actualizaciones los dos ficheros deberán incluir los campos “Scrape ID” y “Last Scraped” para actualizar solo los alojamientos que hayan sido actualizados.

Uno de los campos que habrá que añadir a la tabla secundaria sería el precio en dólares. Dado que la información se va a mostrar en una página de Estados Unidos, el cliente ha solicitado que, aparte de en euros, los precios se muestren en dólares.

El valor de cambio lo obtendremos haciendo scraping sobre la página de Expansión para obtener la última cotización.

* **Scraping**: Complementaremos el data-lake con datos que obtendremos realizando scraping sobre las páginas web que nos puedan proporcionar la información adicional que nos solicita el cliente y que no incluye el fichero de airbnb (por ejemplo, la información de las actividades que se pueden realizar). Para este caso en concreto haremos scraping de la página <http://www.minube.com> .
* **ElasticSearch**. Lo usaremos para hacer ETL. el fichero de airbnb contiene multitud de errores: valores nulos, códigos postales incorrectos, etc. Por ejemplo, la información de campos como Ciudad o Estado presenta multitud de incoherencias. Por ejemplo, en el campo ciudad, hablando solo de datos de Madrid, aparece “Madrid”, “Madrid Centro”, “Chueca”, “Villa verde bajo”, “Centro”...

Elegimos ElasticSearh por la posibilidad de hacer búsquedas semánticas y por la rapidez de procesar grandes volúmenes de datos (aunque se ralentice el almacenamiento). Con esta herramienta recogeremos los datos, haremos limpieza, y trataremos de dar a la información la mayor coherencia posible.

* **Geocoding Api (de Google)**. La utilizaremos fundamentalmente en fase de pre-producción con dos finalidades: Uno de los campos más importantes entendemos que es la direcciòn (campo Street), que es uno de los campos que aparecen en el fichero que están más “sucios” . Direcciones con todo tipo de formatos, direcciones inexistentes, o multitud de direcciones del tipo “*Madrid, Comunidad de Madrid 28007, Spain*”. Mediante esta API y a través del campo “*Geolocation*” conseguiremos la dirección (lo más exacta posible) que se corresponda con las coordenadas de longitud y latitud.

También la utilizaremos para completar nuestro Data-lake con datos que nos ha solicitado el cliente: Distancia y tiempo de desplazamiento al aeropuerto, a las estaciones de Atocha y Chamartín, y a alguno de los principales monumentos (museo del Prado, Palacio Real, etc).

* **Apache Hive**. Se utilizará para hacer consultas con diferentes agrupamientos.
* **Tableau Desktop**. Se utilizará esta herramienta para generar los cuadros de mando que el cliente quiere visualizar en su página web.

# Operativa

* en la fase de pre-producción :
  + Creación de la infraestructura necesaria
    - Creación del cluster de Google Cloud
    - Generación de todo el código necesario (para descargar csv, para el scraping, etc.)
  + Preparación del data lake
    - Se importa y se carga el archivo csv con los datos de airbnb en la nube.
    - se filtran los datos necesarios (ciudades y campos requeridos).
    - Se hace limpieza de los datos con Elastic.
    - Se crearán los csv de datos principales y secundarios.
    - Se actualizará la información del campo Street en el fichero principal mediante Geocoding Api**.**
    - Se actualiza la información del precio en dólares.
    - Se cargarán los archivos csv con la información obtenida de scrapping.
  + Creación de los cuadros de mando con Tableau.
* en fase de producción.
  + Proceso de ETL diaria de los ficheros secundarios.
  + Scraping de <http://www.minube.com> y carga de resultados miércoles y sábados
  + Actualización del fichero principal. Si hay alojamientos nuevos, actualizar datos con la API de geocoding. Los domingos

(En todos los casos deberá actualizarse la salida de los datos en los cuadros de mando de Tableau. No se establecen horarios para que el cliente los determine en función de en qué franja horario se encontrarán los servidores que mostrarán los cuadros de mando).

# Desarrollo

* El scrapy se realiza con la librería “**beautifulsoup4**” en lugar de con scrapy, por investigar y porque parece más ligera.
* Los cálculos de las cotizaciones del dólar respecto al euro se han realizado con un script que hace un scraping en la página de Expansión para buscar la última cotización.