**Datos del alumno**

|  |  |
| --- | --- |
| **Nombre** | Fernando |
| **Apellidos** | Márquez Arroyo |
| **Teléfono** | 606851493 |
| **E-mail** | [fmarquezarroyo@gmail.com](mailto:fmarquezarroyo@gmail.com) |

**Datos del proyecto**

|  |  |
| --- | --- |
| **Título** | **Hotel Advisor - Recomendador de hoteles** |
| **Tutor** | *Rubén Casado* |
| **Descripción** | Me apasiona viajar (soy blogger de viajes) y recientemente al ser partner de la empresa Booking.com he tenido acceso a información de 1.000.000 de hoteles en los 5 continentes (ciudad, país, longitud, latitud, precio por noche, categoría, descripciones en varios idiomas, ...). Por otro lado, existen datasets con información geolocalizada de alrededor de 14.000.000  de puntos de interés (POIs) de todo el mundo (restaurantes, tiendas, monumentos turísticos, museos, hospitales, paradas de autobús,...) obtenidos a partir de la herramienta Open Street Maps (.osm).  El proyecto consiste en utilizar técnicas de Big Data para cruzar información procedente de ambos conjuntos de datos para poder ofrecer al usuario una recomendación de hoteles de acuerdo a sus preferencias. El programa dispondrá de varias opciones:  1) Búsqueda tradicional de hoteles basado en: ciudad, precio mínimo por noche, puntuación mínima de hotel, categoría del hotel, ... El programa mostrará los hoteles que cumplen los filtros  especificados, así como un listado de POIs cercanos a cada uno de los hoteles, de acuerdo a sus preferencias (número máximo de POIs a mostrar por hotel, tipos de POIs, ...)  2) Búsqueda de POIs cercanos a un hotel dado, de acuerdo a las preferencias del usuario.  3) Búsqueda de hoteles a partir de un POI determinado. El programa devolverá los hoteles que se encuentran entre una distancia mínima y máxima del POI elegido, de acuerdo a las preferencias del usuario.(p. ej. hoteles a menos de 500 metros del Santiago Bernabeu, hoteles ubicados entre 500 metros y 1 km de las Pirámides de Giza, ...)  4) Críticas (reviews) de hoteles. El usuario podrá realizar una crítica de un hotel determinado, asignando notas entre 0 y 10 a diversos aspectos (localización, limpieza, confort, instalaciones, personal, wifi, relación calidad/precio), así como un comentario general de texto libre, en el idioma deseado, acerca de su estancia en el hotel. Estas notas afectarán a la puntuación total del hotel para los filtros por puntuación.  Asimismo, cuando se presentan los hoteles recomendados, se mostrará el número de comentarios del hotel, y se tendrá posibilidad de acceder al detalle de cada uno de los comentarios sobre el hotel.  5) TOP Hoteles. El sistema mostrará cuáles son los hoteles más caros, con más comentarios, con la puntuación más alta,...  Adicionalmente, dado que el dataset de hoteles no dispone de información acerca de comentarios ni puntuaciones de los mismos, y dado el elevado número de hoteles para escribir las reviews de forma manual, se precisa de un módulo encargado de generar comentarios y notas aleatorias sobre los hoteles, simulando eventos de entrada al sistema a partir de un sistema externo. |
| **Solución propuesta** | El sistema contará con dos conjuntos de datos de entrada: Hoteles y POIs.  Hoteles: Se dispone de varios ficheros .TSV (campos separados por tabuladores) por continente, con la siguiente información:  id name address zip city\_hotel cc1 ufi class currencycode minrate maxrate preferred nr\_rooms longitude latitude public\_ranking hotel\_url photo\_url desc\_en desc\_fr desc\_es desc\_de desc\_nl desc\_it desc\_pt desc\_ja desc\_zh desc\_pl desc\_ru desc\_sv desc\_ar desc\_el desc\_no city\_unique city\_preferred continent\_id review\_score review\_nr  NOTA: review\_score (puntuación) y review\_nr (número de comentarios) vienen a 1 para todos los hoteles.  POIs: Se dispone de un fichero .CSV (campos separados por el símbolo |) por continente, con la siguiente información:  category\_name osmid latitude longitude poi\_name  Para la parte de Batch Processing, encargada del parseo de estos ficheros y almacenamiento en base de datos, se ha optado por la tecnología **Spark**, dada su madurez y altas prestaciones. Concretamente se utilizará **PySpark**, utilizando **Python**como lenguaje de programación.    En cuanto a la capa de almacenamiento de información, se ha optado por la utilización de la base de datos **MongoDB**. Es una base de datos intuitiva, orientada a documentos y flexible. Ofrece fácil integración tanto con Python como con Java y dispone de un potente módulo para poder lanzar consultas geoposicionadas. Adicionalmente, a través de un conector, puede  replicar la información a Solr o ElasticSearch, si se quisiera ampliar el proyecto para permitir realizar búsquedas de texto libre sobre la información almacenada.  El almacenamiento de la información de hoteles y POIs en MongoDB desde Spark, se realizará mediante el driver de **PyMongo**.  Respecto al módulo encargado de generar comentarios y notas aleatorias para popular la base de datos de opiniones, se ha decidido utilizar una cola de mensajes **Kafka**. Esta tecnología permite gestionar streams de datos, eficientemente y en tiempo real. Los eventos con las notas aleatorias y comentarios de hoteles se generarán con un simulador del estilo del Json Data Generator introduciendo los eventos en la cola. Asimismo, se utilizará **Spark Streaming** para conectarse a la cola Kafka donde llegan los eventos, y guardar las opiniones sobre los hoteles en MongoDB a través de PyMongo.  Respecto a la parte de explotación de los datos, la idea es utilizar **Java**y los frameworks de agregación y consultas geoposicionadas de MongoDB. La presentación de hoteles, pois, comentarios se realizará en formato tabular mediante **Java Swing**.  Se valorará el empleo de tecnologías adicionales, según vaya avanzando el máster, ya que aún faltan algunos módulos, como por ejemplo, el de visualización de datos. |