

# HOTEL ADVISOR



**FERNANDO MÁRQUEZ ARROYO**



**KSCHOOL**

# HOTEL ADVISOR - PFM



1. Me apasiona viajar (soy blogger de viajes).

<http://www.pinchandoelglobo.com>

2. Me encanta la informática.

3. Estoy estudiando un Máster en Arquitectura Big Data.

¿Por qué no tratar de integrarlo todo?

# HOTEL ADVISOR - PFM



- \* Sistema de recomendación de hoteles.
- \* Uso de tecnologías, técnicas y herramientas de Big Data.
- \* Basado en las preferencias de usuario.



# HOTEL ADVISOR - DATASETS



1.000.000 de hoteles en todos los continentes,  
proporcionados por **Booking.com**



14.000.000 de puntos de interés (POIs) en todos los  
continentes, extraídos de



**KSCHOOL**

# HOTEL ADVISOR – DATASETS II



Ficheros TSV (campos separados por tabulador) con el formato:

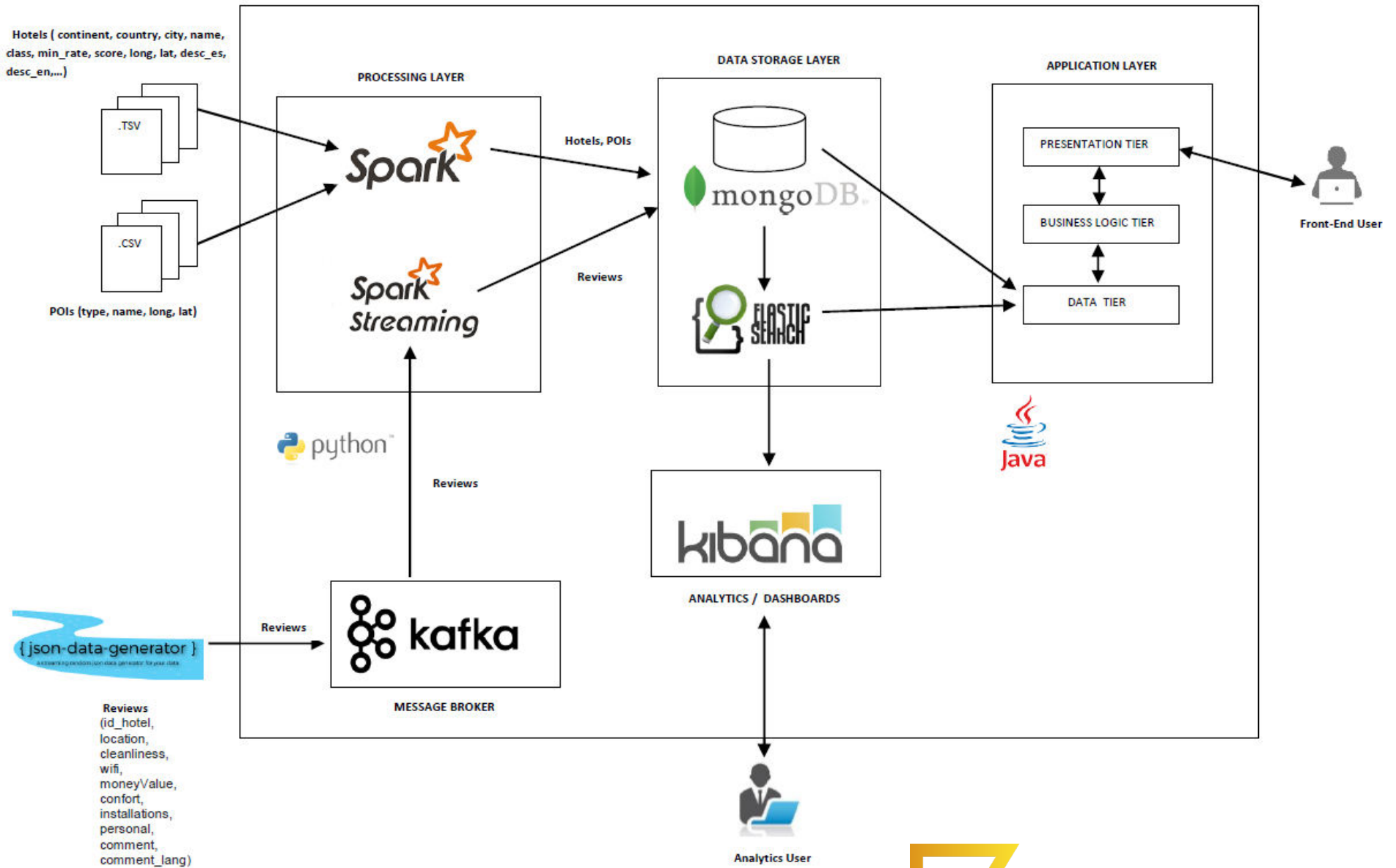
id **name** address zip city\_hotel cc1 ufi **class** currencycode  
**minrate** maxrate preferred nr\_rooms **longitude** **latitude**  
public\_ranking **hotel\_url** **photo\_url** **desc\_en** **desc\_es**  
city\_unique city\_preferred continent\_id **review\_score** **review\_nr**



Ficheros CSV (campos separados por | ) con el formato:

**category\_name** osmid **latitude** **longitude** **poi\_name**

## HOTEL ADVISOR – ARCHITECTURE DIAGRAM



# HOTEL ADVISOR – PROCESSING LAYER (BATCH)



Capa encargada del parseo, limpieza y filtrado de los datasets de entrada. El sistema utiliza la tecnología Spark para el procesamiento, dada su madurez y altas prestaciones.



El lenguaje de programación utilizado para la implementación de esta capa es Python, comunicándose con Spark a través del Spark Python API (PySpark).



El almacenamiento en la base de datos Mongo DB se realiza a través del API de PyMongo.

# HOTEL ADVISOR – PROCESSING LAYER (REAL TIME)

Para simular la entrada de comentarios y puntuaciones de hoteles desde un sistema externo, se ha dotado al sistema de un módulo de procesamiento en tiempo real.



Los eventos con las notas aleatorias y comentarios de hoteles se generan con el simulador de eventos: Json Data Generator, introduciendo los eventos en un topic de Kafka.



El sistema utiliza una cola de mensajes Kafka, para encolar en un topic, los eventos generados mediante el simulador de eventos. Esta tecnología permite gestionar streams de datos, eficientemente y en tiempo real.



El sistema se apoya en la tecnología Spark Streaming, para conectarse al topic de Kafka al que llegan las notas y comentarios de los hoteles, y realizar el procesamiento en tiempo real de dicha información.



# HOTEL ADVISOR – PROCESSING LAYER (REAL TIME) II



El lenguaje de programación utilizado para la implementación de esta capa es Python, comunicándose con Spark a través del Spark Python API (PySpark).



El almacenamiento en la base de datos Mongo DB se realiza a través del API de PyMongo.

# HOTEL ADVISOR – DATA STORAGE LAYER

Capa encargada del almacenamiento persistente de la información utilizada por la aplicación.



El sistema utiliza MongoDB, 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. Almacena información de hoteles, pois y usuarios.

El sistema consta de un conector de MongoDB (Mongo Connector) que permite nutrir de información al motor de búsqueda de Elastic Search.



El sistema emplea el motor de búsqueda de Elastic Search para ofrecer al usuario la posibilidad de realizar búsquedas de texto libre sobre la información almacenada en el sistema.



# HOTEL ADVISOR – APPLICATION LAYER



Capa encargada de la explotación de los datos almacenados en el sistema y su visualización por parte del usuario final, en base a sus preferencias.



El lenguaje de programación utilizado para la implementación de esta capa es Java.



La visualización de los hoteles y POIs recomendados se realiza en forma de tablas utilizando la tecnología Java Swing.

# HOTEL ADVISOR – APPLICATION LAYER II

Está conformada por tres módulos:



**Data Tier:** Encargado de gestionar la persistencia de la aplicación comunicándose con la capa de Data Storage.

Funciones:



- \* Conexión y creación de objetos en la base de datos MongoDB.



- \* Conexión con el motor de búsqueda de Elastic Search.

- \* Encapsular métodos de utilidad para realizar consultas sobre la base de datos MongoDB y sobre el motor de Elastic Search.

# HOTEL ADVISOR – APPLICATION LAYER III



**Business Logic Tier:** Encargado de gestionar la lógica de negocio de la aplicación.

Funciones:

- \* Clase principal, encargada de mostrar los menús y gestionar toda la lógica de la aplicación apoyándose en el resto de clases.
- \* Gestionar las operaciones de entrada/salida sobre la consola.
- \* Control de errores y excepciones.



# HOTEL ADVISOR – APPLICATION LAYER IV



**Presentation Tier:** Encargado de presentar los hoteles, pois y reviews al usuario final. Funciones:

- \* Búsqueda de hoteles (país, ciudad, puntuación, categoría, precio máximo por noche).
- \* Búsqueda avanzada de hoteles (búsqueda textual en descripción, en comentarios, búsqueda de hoteles similares).
- \* Búsqueda de POIs cercanos a un hotel determinado.
- \* Búsqueda de hoteles cercanos a un POI determinado.
- \* Top Hoteles (+ puntuados, + comentarios, + baratos, + grandes)
- \* Crear reviews de hoteles. Listar reviews personales.
- \* Modificar preferencias de usuario (POIs de interés, número de POIs a mostrar, datos personales,...)

# HOTEL ADVISOR – ANALYTICS/DASHBOARDS



Capa encargada del análisis de los datos almacenados en el sistema y su visualización por parte de un usuario de perfil más analítico.



El sistema utiliza Kibana, herramienta de visualización de datos indexados por Elastic Search. Permite navegar y explorar por los datos mediante la creación de gráficas, mapas y dashboards:

- \* Mapa geográfico de hoteles/pois.
- \* Puntuación media de los hoteles en base a la edad de usuarios.
- \* Número de reviews por día, semana, mes , año, ...
- \* Puntuación de los hoteles distribuída por país.
- \* ...

# HOTEL ADVISOR – DUDAS/PREGUNTAS

