Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Big data aplicado al análisis de opiniones sobre películas

Big data applied to the analysis of film reviews

Daniel Jiménez Rodríguez

La Laguna, 10 de Septiembre de 2019

D. **Jesús Manuel Jorge Santiso**, con N.I.F. 42.097.398-S profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

"Big data aplicado al análisis de opiniones sobre películas"

ha sido realizada bajo su dirección por D. **Daniel Jiménez Rodríguez**,

con N.I.F. 54.115.165-Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 10 de Septiembre de 2019

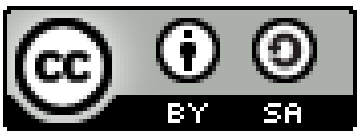
Agradecimientos

Quiero agradecer a mi familia por apoyarme durante toda mi formación educativa y en el resto de facetas de mi vida, no sería la persona que soy hoy si no los tuviera de mi lado.

Agradecimientos a Jesús Manuel Jorge Santiso por la ayuda prestada durante todo el desarrollo de este proyecto.

También agradecer a todo el alumnado de la carrera con el que en algún momento he colaborado, especialmente Iván y Raúl Ulises, sin su apoyo dudo que haya podido llegar hasta este punto.

Licencia



Resumen

La proliferación de las redes sociales, páginas web y los usuarios que la usan llevan aparejadas cantidades de datos en forma de mensajes, videos, imágenes... que son transferidas a través de internet cada día.

En las primeras décadas en las que se extendió el uso de internet, estos datos no han tenido ninguna relevancia más allá del entretenimiento de los usuarios, pero entrados en el siglo XXI se han realizado múltiples investigaciones en este campo, motivados por el potencial que se encuentra en saber ***calificar y evaluar los datos*** *para extraer información relevante de ellos, en otras palabras, la minería y análisis de datos.*

Esto desembocó en un aumento del uso de tecnologías ***Big Data****, que ofrecen múltiples programas orientados al manejo de grandes cantidades de información. A día de hoy existen múltiples alternativas, entre las que destaca* ***Hadoop*** *por encima de todas en cuanto a popularidad y cantidad de software que tiene asociado a su ecosistema.*

En este proyecto se adaptarán todos los conceptos anteriores a un mismo framework, para ofrecer una solución que permita obtener una visión global de la opinión que tiene la audiencia acerca de diferentes ***películas****, con el objetivo de poder prever su éxito y/o obtener información que sería difícil de inferir por medios normales.*

Para ello se hará uso de: Scripts Python, Flume, Hadoop, Hive y Apache Zeppelin.

Palabras clave: Análisis de sentimientos, Web-crawling, Big Data

Abstract

The proliferation of social networks, websites and the users who use them makes it so huge amounts of data in the form of messages, videos, images that are transferred through the internet every day.

In the first decades in which the use of the internet was extending, this type of data didn't had any relevance beyond the entertainment of their users, but since the start of the 21st century, multiple investigations have been carried out on this field, motivated by the potential that It's knowing how to ***qualify and evaluate the data*** *to extract relevant information from it, in other words, mining and data analysis.*

This led to an increase in the use of ***Big Data*** *technologies, which offer multiple programs aimed at handling large amounts of information. Today there are multiple alternatives, among which* ***Hadoop*** *stands out above all in terms of popularity and amount of software associated with its ecosystem.*

In this project, all the previous concepts will be adapted to the same framework to offer a solution that allows its user to obtain a global vision of the audience's opinion about different ***films****, with the aim of being able to predict their success or to obtain information that would be difficult to infer by normal means.*

For this, use will be made of: Python scripts, Flume, Hadoop, Hive and Apache Zeppelin.

Keywords: Sentiment Analysis, Web-crawling, Big Data

Índice general

Índice de figuras

Índice de tablas

Capítulo 1 Introducción

El hecho de que el uso de las tecnologías de la información y los dispositivos de comunicación se haya extendido en las últimas décadas ha dado lugar a un flujo de información sin precedentes: videos, mensajería, fotos…

Uno de los libros blancos de Cisco (*Cisco Visual Networking Index*[1]) proyecta que para el año 2022 el tráfico IP anual alcanzará los 396 exabytes (EB) por mes, esto es, 3,96e+8 terabytes.

Es debatible si en un futuro se podría transformar semejante volumen de datos en conocimiento útil, pero la realidad es que con las herramientas adecuadas la información puede ser extraída de la red y categorizada para aplicarla a aquellos campos que nos sean de interés, en vez de simplemente desecharla.

Big Data es el nombre de dicha práctica, que a día de hoy se encuentra en auge en múltiples industrias. Uno de los usos que se le puede dar a la tecnología, es la de obtener y procesar las opiniones de un grupo usuarios con el fin de representarlas de una manera sencilla que permita ver la recepción de un producto o servicio.

En este proyecto se abordará dicho uso, y **los productos que se estudiarán serán películas**, de manera que se pueda obtener información sobre: qué términos se usan frecuentemente al mencionar una película, el volumen de personas que habla acerca de la película a lo largo del tiempo….

1.1 Objetivos

1.1.1 Objetivo general

El **objetivo principal** consistiría en desarrollar una herramienta capaz de ofrecer de manera sencilla una visión global del éxito de una película de cara al público a partir de datos extraídos de diferentes páginas web. La herramienta realizaría tareas de **minería de textos y análisis de sentimientos** para llegar a tal objetivo.

El requisito para la elaboración del trabajo es que se debe realizar en una arquitectura Big Data, en la que se pueda trabajar en modo distribuido con múltiples equipos. Además, se utilizará **únicamente software libre** para su elaboración.

1.1.2 Objetivos específicos

- Estudio acerca de los diferentes **tipos de ecosistemas** Big Data.
- Aprender sobre las diferentes **herramientas de software libre** disponibles en cada ecosistema.
- Creación del ecosistema y **banco de pruebas inicial.**
- Estudiar acerca de las diferente maneras de **procesar datos.**
- Investigar sobre el **análisis de sentimientos** y diferentes procesos para llevarlo a cabo.
- Estudiar acerca de los **métodos de web crawling** disponibles en el ecosistema escogido.
- Indagar en diferentes herramientas de **visualización de datos**, para conocer las posibilidades que estas pueden llegar a ofrecer.
- Aprender cómo crear **interfaces gráficas** de usuario para Linux.

1.2 Interés del problema

La necesidad que esta aplicación cubre es la de conocer las **opiniones del público** con respecto a una determinada película, incluso en consumidores que no aportan su opinión directamente con reseñas o críticas.

El hecho de que la arquitectura que se emplee sea Big Data implica que se podrán recolectar y procesar una mayor cantidad de datos en poco tiempo usando múltiples equipos, lo que dará una visión más real de la opinión del público al aumentar el número de muestras.

Se podrá conocer información como:

- Si una determinada franquicia o película debería ser continuada.
- Cuales son los aspectos más hablados de ella.
- Como de relevante a sido la película a lo largo del tiempo.
- Qué películas son recomendadas usando este criterio.

  …

1.3 Big Data y gestión del conocimiento

Algunos conceptos fundamentales que estarían relacionados con este proyecto son: la gestión del conocimiento y Big Data

La gestión del conocimiento alude a un proceso para compartir información y generar conocimiento con el objetivo de crear valor con la que el usuario se puede beneficiar.

Su importancia reside en que al gestionar altos volúmenes de información relevante, las empresas pueden generar una visión del entorno en el que se mueven, y esta puede ser muy beneficiosa a la hora de tomar decisiones estratégicas.

Los dos factores más importantes a la hora de gestionar el conocimiento son: su transferencia y almacenamiento.

Cabe mencionar que, a diferencia de otros bienes, el conocimiento siempre crece con el tiempo. Sin embargo, para aprovecharlo al máximo, es necesario saber transmitirlo y manipularlo. Actualmente, la evolución tecnológica, y el surgimiento de herramientas digitales han permitido crear canales para la difusión del conocimiento de una forma más veloz y eficiente.

En este caso, el conocimiento que se maneja sería el conjunto baremado de todas las opiniones que se puedan obtener en relación a una película o serie determinada.

Ahora bien, cuando hablamos de **Big Data** nos referimos a conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, tales como bases de datos relacionales y paquetes estadísticos dentro del tiempo necesario para que sean útiles.

Aunque el tamaño utilizado para determinar si un conjunto de datos se considera Big Data no está firmemente definido y sigue cambiando con el tiempo, la mayoría de los analistas y profesionales actualmente se refieren a conjuntos de datos que van desde 30-50 Terabytes a varios Petabytes.

La naturaleza compleja del Big Data se debe principalmente a la naturaleza **no estructurada** de gran parte de los datos generados por las tecnologías modernas, como los weblogs, la identificación por radiofrecuencia (RFID), los sensores incorporados en dispositivos, la maquinaria, los vehículos, las búsquedas en Internet, las redes sociales como Facebook, computadoras portátiles, teléfonos inteligentes y otros teléfonos móviles, dispositivos GPS y registros de centros de llamadas.

En la mayoría de los casos, con el fin de utilizar eficazmente el Big Data, éste debe combinarse con datos estructurados (normalmente de una base de datos relacional) de una aplicación comercial más convencional, como un ERP (Enterprise Resource Planning) o un CRM (Customer Relationship Management).

1.4 Estado del arte

A día de hoy se han realizado multitud de estudios, memorias, soluciones… acerca de Big Data debido a su relevancia en los últimos años. Esto incluye a la Universidad de la Laguna, con memorias como:

- **Despliegue de un clúster Spark sobre Docker para Big Data**[2]: Donde se realiza un estudio preliminar de Docker y Spark de manera que se entienda sus beneficios y funcionamiento.
- **Análisis de ficheros log de la WiFi-ULL usando técnicas de Big Data**[3]: Donde se analizan las diferentes opciones software, lenguajes de programación... para una solución Big Data.
- **Big Data y la Visualización en el ámbito Educativo**[4]: Donde se estudian diferentes técnicas de visualización para la representación de los datos.

Aunque fuera de la universidad se encuentran estudios más relacionados con el enfoque del proyecto, véase por ejemplo:

- **Herramienta de Text Mining aplicado a textos cortos y redes sociales**[5]: En él se explica cómo crear un sistema que implemente técnicas para procesar, analizar y clasificar textos.
- **Sentiment analysis and opinion mining**[6]: Libro acerca de los avances y nuevas tecnologías para el análisis de sentimientos, procesado de lenguaje natural….
- **Social media competitive analysis and text mining: A case study in the pizza industry[7]**: Estudio en el que se analizan datos no estructurados usando técnicas de "*text mining*" para cadenas de comida rápida en Twitter y Facebook.

También hay ejemplos de soluciones profesionales que ya se encuentran en el mercado para problemas relacionados con el del proyecto, como: **Text mining solutions**[8] o **Chinetek Strategy**[9] en la que conociendo los tipos estudios que se desean realizar, estas compañias desarrollarán una solución a medida.

Algunos ejemplos de implementaciones de código abierto que se encuentran disponibles con temas similares:

- **Twitter-sentimental-analysis**[10]: programa que usa Mongodb, LingPipe tool kit y MapReduce para determinar si un tweet es positivo o negativo.
- **Sentiment Analysis**[11]: en donde se utilizan crawlers en varias páginas webs para recuperar comentarios, analizarlos gramaticalmente y asignarles una puntuación en base a si son positivos o negativos.

Mi proyecto se separará de soluciones similares no solo por el tema (análisis de sentimientos sobre películas) sino por el software implicado y las fuentes que se utilizan para adquirir los resultados.

1.4.1 Desarrollos parecidos

La mayoría de soluciones o desarrollos similares a este proyecto son de pago, sin embargo se pueden encontrar implementaciones similares en algunas partes de nuestra solución en los siguientes proyectos online:

- **Twitter Sentiment Analysis and N-Gram with Hadoop and Hive**[12]**:** En este tutorial de Github se puede observar una implementación parecida a la de este proyecto en cuanto al análisis de sentimientos.

- **Statistics and Data Mining (Hive)[13]:** Artículo en la página de hive que explica cómo utilizar las funciones ngrams y context_ngrams(), con una implementación parecida en el análisis de sentimientos a la de este proyecto usando como ejemplo a Twitter.

Capítulo 2 Fundamentos y tecnologías a utilizar

2.1 Text Mining y Análisis de sentimientos

2.1.1 Text Mining

El "*Text mining*" o minería de textos consiste en el análisis del lenguaje natural.

El beneficio del text mining se encuentra en que es capaz de ayudar a la organización a **recolectar información** potencialmente valiosa a través de textos que posee la empresa, como por ejemplo: los documentos de correo electrónico y publicaciones en redes sociales (como Facebook, Twitter…).

Hay bastantes tipos de aplicaciones que se pueden crear a partir de procesos text mining, algunos ejemplos son:

- Aplicaciones que analizan las respuestas de las encuestas abiertas.
- Procesamiento automático de mensajes, correos etc.
- Analizar la garantía/veracidad de reclamaciones en seguros.
- Investigar a los competidores obteniendo datos de sus páginas.

Para que estas aplicaciones funcionen se debe de analizar el texto y simplificarlo hasta un punto en el que se puedan obtener los datos en los que se está interesado de manera sencilla. Para ello, se pueden generar una matriz de frecuencias a través de un documento, en la cual se muestra la cantidad de veces que una palabra ha aparecido.

Sin embargo, hay muchas variaciones y palabras distintas que tienen el mismo significado, y contarlas por separado dificultará interpretar el texto, por eso existe un proceso llamado "*steming*" en el que se cuentan todas las variaciones y sinónimos de una palabra como una misma entrada (por ejemplo, palabras como: conducir, conducirá, dirigir, conduce … se contarían como si fueran la palabra conducir).

Además, las palabras que carezcan de significado por sí mismas no se contarían en la matriz. Algunos ejemplos de estas palabras serían: "a, los, por, allí …".

2.1.2 Análisis de sentimientos

En los últimos años la cantidad de datos que son generados y compartidos por la red ha aumentado exponencialmente, y una gran parte de estos datos son *posts* y comentarios de distintos usuarios en diferentes plataformas. Estos pueden llegar a interpretarse gracias al análisis de sentimientos, que posibilita discernir entre opiniones positivas, negativas y neutras acerca de cualquier producto o servicio.

Existen varias formas de llevar a cabo esta tarea, como simplemente ir a Twitter y buscar de forma manual menciones acerca de aquello que resulte de interés. Sin embargo, esto no es óptimo, debido a que para tener una idea realista acerca de la opinión de los usuarios hace falta analizar un volumen de datos lo más grande posible, y hacerlo manualmente consumiría demasiado tiempo.

Por ello se implementan procesos basados en datos escalables, que trabajen de manera exhaustiva mediante la realización de **minería de opiniones**.

Para minar las opiniones y analizarlas, se ha de convertir la información no estructurada en información estructurada. Esto implica **extraer los datos** que se consideran importantes y **formatearlos** de manera que se puedan procesar a todos de la misma manera.

El tipo de información más importante que se puede extraer en el análisis de sentimientos es la opinión de un usuario. Hay diferentes tipos de opiniones:

- Opiniones directas: que dan una opinión sobre una entidad directamente.
- Opiniones comparativas: que se expresan mediante la comparación de una entidad con otra.

  …

El tipo de opinión más difícil de analizar sería probablemente metáforas, porque la gran mayoría de programas que realizan estos tipos de análisis están pensados para analizar el texto literalmente, y no son capaces de interpretar expresiones o sarcasmos.

También existen análisis de sentimientos en los que se pueden convertir la simple polaridad de positivo y negativo en un espectro más amplio y detallado:

- Podemos clasificar emociones como la felicidad, frustración, tristeza….
- Podemos elegir analizar el sentimiento de cómo un cliente se siente con respecto a un solo aspecto de un producto/servicio.
- Tratar de descifrar la intención del usuario que escribió el texto.

  …

Hay dos formas principales en que podemos implementar el análisis de sentimientos:

La primera forma, consistiría en utilizar un **sistema basado en reglas** que realiza un análisis de sentimientos basado en un conjunto de parámetros especificados manualmente, estas reglas identifican la subjetividad y la polaridad al utilizar un diccionario.

Un **diccionario** es una tabla de palabras y expresiones positivas y negativas. En una columna se muestra la palabra, y en otra el valor que tiene asignado.

Se pueden crear desde cero, pero a día de hoy no hay necesidad debido a que hay muchos fácilmente accesibles online (hablamos de diccionarios del orden de 100.000 palabras en inglés con su polaridad asociada).

Aquellos métodos que se basan en usar un diccionario, cuentan las palabras polarizadas en una oración de manera que, por ejemplo, cada palabra negativa reste 1 al total y por cada positiva se sume 1.

Al ver el resultado final se podrá averiguar como de positiva o negativa fue la oración dependiendo de si el resultado total es positivo o negativo.

También es posible asignar a las palabras en el diccionario un valor que no sea +1 o -1, porque pueden haber palabras que sean más positivas o negativas que otras. Por ejemplo: bien (+1) o fantástico (+2).

El enfoque manual no requiere un proceso de capacitación y es más fácil de depurar, pero no es tan preciso con frases complejas como el automático.

La segunda forma es que el mismo programa **imponga las reglas a utilizar**, sin que el programador las defina con un diccionario. En este caso se usa algoritmos de inteligencia artificial (IA).

Esto se categoriza como un problema de calificación, en el que una vez que se haya entrenado a el modelo, éste sea capaz de funcionar por sí mismo. Este entrenamiento consistiría en inyectar palabras y designar su valor manualmente, de manera que el modelo vaya asociando si determinados tipos de palabras son positivas o negativas.

El enfoque automatizado es más fácil de escalar y, por lo general, ofrece resultados más precisos que el enfoque manual.

2.2 Ecosistema Big Data

Un ecosistema Big Data se define como cualquier sistema que permite almacenar, procesar, analizar y visualizar datos a gran escala.

Hay multitud de maneras y herramientas con las que formar un ecosistema Big Data, pero estas se pueden dividir en aproximadamente cuatro tipos:

- **Tecnologías basadas en Hadoop**: Hortonworks, MapR, Cloudera….

Estas son usadas generalmente cuando se necesitan procesar grandes cantidades de información y producir muchos tipos distintos de resultados. Generalmente trabajan como un sistema por lotes sin conexión, pero la arquitectura Hadoop también permite el procesamiento de datos en tiempo real.

- **Almacenes tradicionales de datos**: TeraData Aster, Greenplum, Oracle Exadata….

Este tipo de tecnología se ha usado por décadas, pero se siguen usando a día de hoy por su robustez y estabilidad. Al contrario que las tecnologías basadas en Hadoop, este tipo de bases de datos tradicionales no pueden procesar datos no estructurados, ya que requieren de mantener una estructura consistente para ser de utilidad.

- **Tecnologías noSQL**: Riak, Redis, MongoDB, Apache Hbase, Cassandra, Couchbase….

Las tecnologías noSQL están enfocadas a ser el back-end de otras aplicaciones, que usan datos en menor cantidad pero muy frecuentemente, por ejemplo, aplicaciones a gran escala en internet: Facebook, Ebay… que pueden ser usadas por cientos de miles de usuarios al mismo tiempo, ya que estos usuarios solo requieren cantidades de información muy pequeñas en comparación al total para usar la aplicación.

- **Bases de datos en memoria:** SAP HANA, Oracle TimesTen...

Tienen las ventajas de todas las anteriores: pueden producir terabytes de datos en respuesta, extremadamente rápidas (por debajo de un segundo)... pero tienen el problema de que necesitan altas cantidades de memoria RAM para ser utilizadas, y esto puede suponer en un costo gigantesco. Al contrario que el resto de alternativas, que se pueden adaptar a las características de los equipos en los que están siendo utilizados, las bases de datos en memoria requieren de equipos especializados para ser utilizadas efectivamente.

Como se va a trabajar desde un solo ordenador básico y procesando gran cantidad de datos, las tecnologías Hadoop se ajustan a los requerimientos que se tienen para este proyecto.

2.3 Hadoop y Map/Reduce

Apache Hadoop es un proyecto de software de código abierto que permite un procesamiento distribuido de grandes conjuntos de datos en grupos de servidores (clusters) que está diseñado para poder escalarse desde un solo servidor a miles de ellos con un grado muy alto de tolerancia a fallas a partir del sistema Mapreduce y sistema de archivos de Google. Su uso se ha extendido a muchas empresas como: Facebook, LinkedIn, Twitter…

Un aspecto clave de la resistencia de los clústeres de Hadoop es la capacidad del software para detectar y manejar fallas en la capa de aplicación (donde se encuentra Map/Reduce).

Hadoop tiene dos subproyectos principales:

- **MapReduce:** es el marco que monitoriza y reparte el trabajo a los nodos en el clúster y, en segundo lugar, HDFS para el sistema de archivos en el clúster de Hadoop para el almacenamiento de datos.

- **HDFS:** es el sistema capaz de combinar partes de archivos divididos entre distintos nodos locales para formar archivos completos, de manera que todos los nodos forman un sistema de archivos gigante.Existe la posibilidad de que puedan existir fallas en los archivos almacenados, por lo tanto, replican los datos a través de múltiples nodos, de manera que existan diferentes copias en caso de que una falle.

Hay muchas diferencias en como Hadoop almacena y estructura datos en comparación a bases de datos SQL tradicionales.

La primera diferencia es que Hadoop utiliza "***Schema on Read***" mientras que una base de datos SQL utiliza "***Schema on Write***".

Cuando movemos datos entre dos bases de datos que usan "*Schema on Write*", se necesita tener información a mano antes de escribir en la otra base de datos, cómo saber cuál es la estructura de en la base de datos destino y cómo adaptar los datos para que se ajusten a esa estructura. Además, tenemos que asegurarnos de que los datos que se transfieren cumplan con los tipos de datos que la base de datos espera. Si intentamos cargar algo que no cumple con el tipo de datos que se espera, empezarán a aparecer errores.

Ejemplo de uso del *"Schema on Write"*:

- Creación de tabla:

```
CREATE TABLE Ejemplo (x int, y varchar(8))...
```

- Añadir datos:

```
INSERT INTO Ejemplo FROM ....
```

- Seleccionar datos:

```
SELECT x FROM Ejemplo ....
```

En el caso de "*Schema on Read*", se tiene un proceso enfocado a la hora de leer dichos datos. Cuando escribimos datos en HDFS, simplemente los introducimos sin dictar ninguna regla de control de acceso. Luego, cuando queremos leer los datos, aplicamos reglas al código que lee los datos en lugar de configurar con anterioridad la estructura. Estas reglas son dadas por los *scripts* asignados al "*mapper*" y "*reducer*" de la plataforma, y se estructuran e interpretan a medida que son leídos.

Ejemplo de uso del *"Schema on Read"*:

- Cargar los datos a HDFS:

```
hdfs dfs -copyFromLocal <ruta_del_fichero_local> <destino_en_HDFS>
```

- Seleccionar los datos:

```
hadoop jar Hadoop-streaming.jar
     -mapper script1.py
```

```
    -reducer script2.py
    -input <ruta_del_fichero_en_HDFS>
    -output <destino_en_HDFS>
```

Esta no es la única diferencia importante, ya que mientras HDFS (Hadoop) almacena los datos como archivos comprimidos, SQL los estructuraría en filas y columnas.

Aparte de los proyectos mencionados, existen muchos otros que conforman el ecosistema de Apache en el que Hadoop se encuentra, tales como:Pig, Hive y Zookeeper, que extienden el valor de Hadoop y aumenta su usabilidad.

Hadoop ha cambiado la economía y la dinámica de la computación a gran escala debido a su:

- **Escalabilidad:** en la que nuevos nodos pueden ser añadidos a medida que hagan falta sin necesitar cambiar el formato de los datos, como se cargan, o modificar las aplicaciones que se sirven de ellos.
- **Rentabilidad:** Hadoop ofrece computación paralela masiva en grandes grupos de servidores. El resultado es una disminución considerable en el costo por terabyte de almacenamiento, que a su vez hace que el análisis de todos sus datos sea asequible.
- **Flexibilidad:** Hadoop no tiene esquema y puede absorber cualquier tipo de datos (estructurados o no) de cualquier número de fuentes. Los datos de múltiples fuentes se pueden unir y agregar de manera arbitraria, permitiendo análisis más profundos que otros sistemas puedan proporcionar.
- **Tolerancia a fallos.** Cuando un nodo falla, el sistema redirige el trabajo a otra ubicación en el clúster y continúa procesando sin perder un tiempo. Todo esto sucede sin que los programadores tengan que escribir un código especial o ser conscientes de la mecánica de la infraestructura y su procesamiento paralelo.

2.4 Flume

Flume es un mecanismo para mover grandes volúmenes de datos a Hadoop, de forma automática en respuesta a determinados eventos, como por ejemplo, el borrado o edición de un archivo.
Flume puede tomar sus datos de varias fuentes, incluyendo: archivos, registros… y enviarlos a distintos destinos, incluyendo Hadoop o HBase. Ofrece una variedad de diferentes configuraciones y topologías, lo que lo hace muy versátil.
Flume está formado por los siguientes seis componentes:

- **Las fuentes** (sources) aceptan datos de una aplicación o servidor.
- **Los sumideros** (sinks) reciben datos y los almacenan en un repositorio como HDFS o los reenvían a otra fuente.
- **Los canales** (channels) enlazan las fuentes a los sumideros y brindan varios tipos de servicio con diferentes opciones.

- **Los interceptores** (interceptors) transforman o eliminan los datos a medida que fluyen a través del sistema.
- **Los eventos** son las unidades de datos que se transfieren a través de un canal desde la fuente hasta el sumidero, generalmente son alrededor de 4 K.

Flume es un sistema distribuido que se ejecuta en varias máquinas. Puede recopilar grandes volúmenes de datos de muchas aplicaciones y sistemas. Se pueden diseñar flujos complejos de múltiples saltos donde los eventos viajan a través de varios agentes antes de llegar a un sistema de archivos de destino. Esto se hace vinculando un sumidero de un agente a una fuente de otro agente.

Las fuentes pueden enviar eventos a múltiples canales, lo que resulta en un mecanismo de despliegue. Múltiples fuentes pueden entregar eventos a un sumidero, lo que resulta en un mecanismo de entrada. Pasar un evento de una fuente a un sumidero se implementa como una transacción confiable. Los eventos no se eliminan de un canal hasta que se almacenan de forma segura en el siguiente canal o en el sistema de archivos de destino.

Los canales basados en memoria proporcionan una transferencia más rápida, pero pueden perder datos cuando falla un sistema. Un canal basado en JDBC (Java Database Connectivity) permite que el canal se recupere de las fallas del sistema. Flume incluye mecanismos para equilibrar la carga y la conmutación por error, y puede ampliarse y personalizarse de muchas maneras.

Flume es un sistema escalable, confiable, configurable y extensible para administrar la transferencia de grandes volúmenes de datos.

2.5 Hive

Apache Hive es uno de los componentes de almacenamiento de datos más populares en el panorama de Big Data, y se utiliza principalmente para complementar el sistema de archivos Hadoop. Este trabaja como una **capa de abstracción** por encima del sistema de ficheros distribuido **HDFS**.

Hive fue desarrollada en un principio por Facebook, y a día de hoy, se encuentra en propiedad de la Apache Software Foundation.

Hive es utilizado por empresas que basan gran parte de su modelo de negocio en ser capaces de almacenar grandes cantidades de información, y hacerla accesibles de manera fácil y rápida (como por ejemplo Netflix y Amazon).

La creación de Hive fue motivada por la necesidad de querer crear una **base estructurada** de datos dentro de Hadoop, que no solo fuera escalable sino que también sea rentable cuando se trata de procesar grandes volúmenes de información.

Estas bases estructuradas de datos (basadas en SQL) forman parte de un marco de trabajo con el que muchos desarrolladores están acostumbrados a trabajar. El problema

reside en que Hadoop tiene un marco de trabajo completamente diferente al de las bases de datos tradicionales, por lo que se creó Hive como punto medio para aplicar los conocimientos previamente adquiridos del lenguaje SQL al paradigma de Hadoop.

Hive proporciona la posibilidad de que sus usuarios utilicen el conocimiento que ya han adquirido de SQL para escribir consultas SQL llamadas **HQL (Hive Query Language)** para extraer datos de Hadoop.

Estas consultas HQL se pueden convertir posteriormente a trabajos Mapreduce, con los que Hive se puede comunicar con el sistema de archivos HDFS y el ecosistema de Hadoop.

Hive también se puede utilizar junto al procesamiento analítico de soluciones tipo OLAP. Las soluciones OLAP son escalables, rápidas y flexibles, y también se consideran el estándar para interactuar con grandes conjuntos de datos.

Es capaz de usar diferentes tipos de archivo para conformar su base de datos, algunos ejemplos son: Archivos secuenciales, archivos de texto, ORC…. Y dichos archivos, a su vez, pueden ser comprimidos de muchas maneras: gzip, snappy….

Los metadatos en la base de datos se encuentran almacenados en RDBMS (sistema de gestión de bases de datos relacionales). Además permite definir funciones, hacer joins… junto a otras operaciones especializadas que ayudan a mejorar el rendimiento de las consultas.

Sin embargo, hay determinadas situaciones en las que no se debería usar Hive:

- Hive no es una base de datos relacional, aunque contenga muchas de sus características, por lo que va a estar limitada a determinados tipos de trabajo.
- No se puede usar para el procesamiento de transacciones en línea (OLTP).
- No se puede usar para actualizaciones o consultas en tiempo real.
- No se puede utilizar para situaciones en las que se espera baja latencia a la hora de recuperar datos después de solicitarlos, porque existe una latencia en la conversión de los *scripts* de Hive a los *scripts* de MapReduce por parte de Hive, y este proceso puede llegar a extender considerablemente el tiempo de espera.

2.6 API REST

API significa interfaz de programación de aplicaciones y, básicamente, permite que un determinado software **intercambie información** con otro.

Hay muchos tipos diferentes de API, pero una de las más usadas son las API tipo REST. Estas son las más usadas por grandes compañías como Twitter o Google.

Sus siglas provienen de "*representational state transfer*" (transferencia de estado representacional) y está diseñado para aplicaciones en línea. Depende de un protocolo para la comunicación cliente-servidor (generalmente HTTP).

La API sería la **encargada de comunicar** al servidor lo que el cliente quiere, y **enviar la respuesta deseada** de vuelta.

Las APIs Web suelen enviar las respuestas a las solicitudes del cliente en formato JSON, que son datos estructurados y organizados de acuerdo a los valores a los que corresponden. Estos valores son llamados parámetros, y detallan el tipo de información que que se quiere del servidor. El formato JSON está extendido a muchos de los lenguajes de programación que se usan actualmente.

2.7 Web Crawling

Un "*web crawler*" es un bot de Internet capaz de **rastrear y extraer datos** de elementos web. Se pueden denominar también como araña, indexador automático o simplemente un rastreador.

Se pueden utilizar tanto para extraer datos de páginas web que no dispongan de una API, como para indexar páginas web en un buscador.

Cuando se utilizan para indexar páginas web, se rastrea una página a través de un sitio web hasta que todas las páginas se han indexado. Esto ayuda a recopilar información sobre un sitio web y los enlaces relacionados con ellos, y también ayudan a validar el código HTML y los hipervínculos.

Las arañas son capaces de recopilar información como la URL del sitio web, la información en los tags utilizados, el contenido de la página web, los enlaces en la página web, los destinos que derivan de esos enlaces, el título de la página web y cualquier otra información relevante. Realizan un seguimiento de las URL que ya se han descargado para evitar volver a descargar la misma página.

Las arañas son los componentes clave de los motores de búsqueda y los sistemas que analizan las páginas web. Ayudan a indexar las entradas web y permiten a los usuarios enviar consultas contra el índice y también proporcionan las páginas web que coinciden con las consultas. Otro uso para las arañas es el "*web filing*", que involucra grandes conjuntos de páginas web que se recopilan y archivan periódicamente. Los rastreadores web también se utilizan en la minería de datos, en donde las páginas se analizan en busca de diferentes propiedades, como estadísticas, y luego se realizan análisis de datos en ellas.

Este último uso para las arañas es el que se emplea para la aplicación que se va a desarrollar en este proyecto.

Capítulo 3 Solución desarrollada

3.1 Esquema del funcionamiento del programa

En este esquema se pueden observar los distintos programas que intervienen en la arquitectura Big Data de este proyecto:

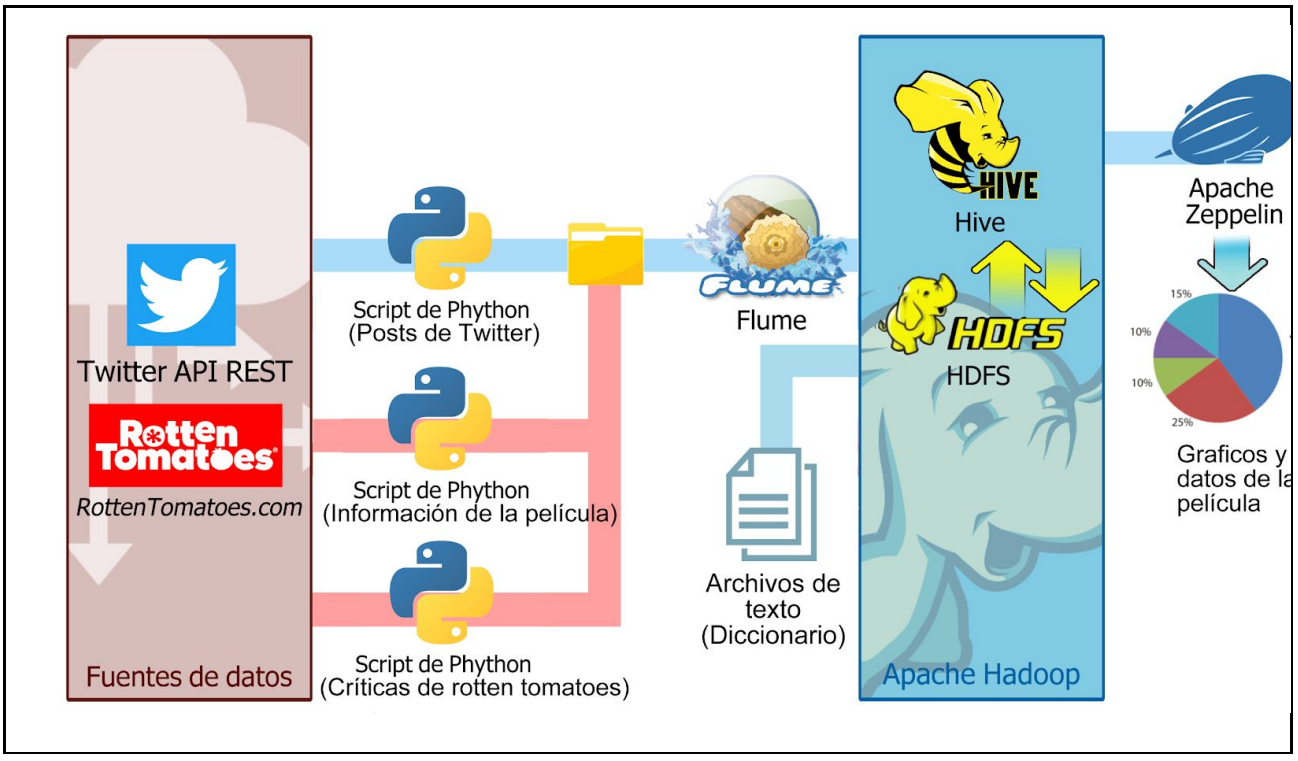


Figura 3.1: Esquema de componentes y funcionamiento del proyecto.

A continuación se explicará cada componente del esquema en detalle, y de qué manera están relacionados los unos con los otros.

3.2 Fuentes de datos:

Los datos que se utilizarán para la realización de este proyecto provienen de dos páginas distintas: Twitter y Rotten Tomatoes.

3.2.1 Twitter

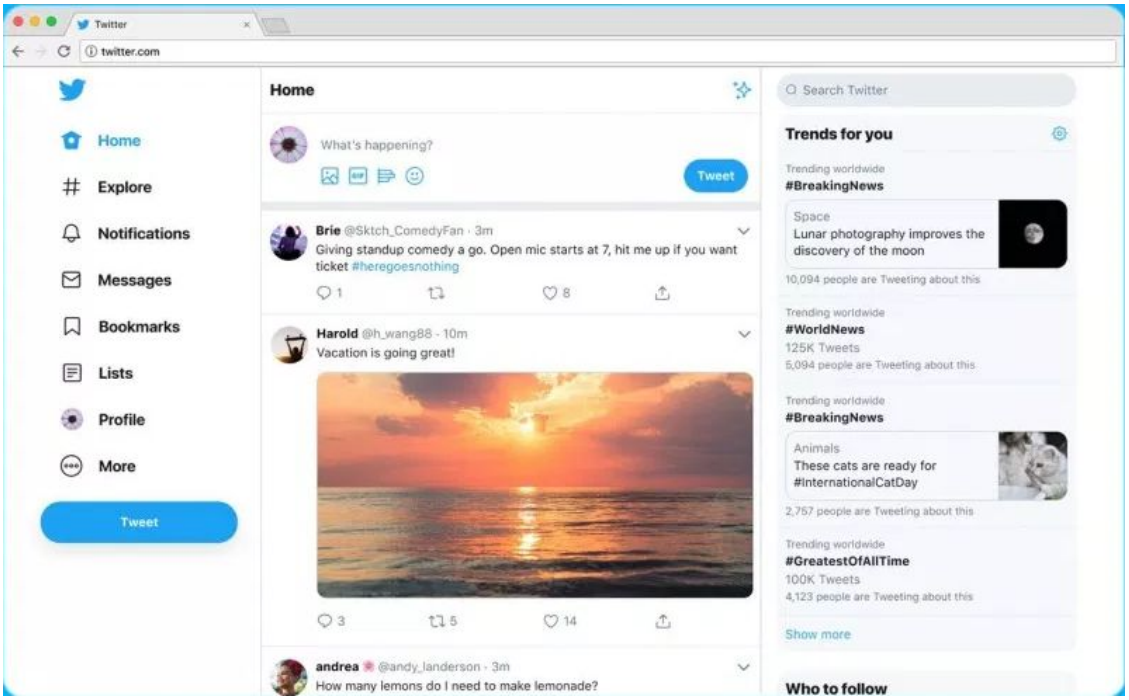


Figura 3.2.1: Ejemplo de la página de inicio de Twitter

Twitter[13] es una red social que permite usar mensajes cortos a todos los usuarios que se hayan registrado en la plataforma para enviar mensajes a sus amigos o seguidores en forma de *tweets* que no tengan más de 140 caracteres.

Los *tweets* o mensajes que se publican pueden incluir un enlace web, fotos, videos, *gifs*, encuestas….

Los usuarios pueden seguir la cuenta de otros dentro de la plataforma. Seguir ayuda a leer o responder a nuevos *tweets* de dichos usuarios y compartir *tweets* que sus seguidores o el mismo usuario hacen, lo que también se conoce como *retweet*.

La mayoría de la terminología básica de Twitter consiste en:

- **Tweets:** La forma más básica de un mensaje en el que se puede escribir y publicar textos para los seguidores del usuario. Los *tweets* se deben escribir en menos de 140 caracteres.

- **Hashtags:** Los términos de búsqueda en Twitter que también existen en otros sitios web, como Facebook o Instagram.Si agrega un hashtag a las publicaciones, los usuarios que realicen una búsqueda de esa palabra clave encontrarán *tweets* asociados a ella.
- **Retweet:** Funcionan de la misma forma que el botón para compartir en Facebook. Cuando el usuario vea el tweet de otra persona, puede compartirlo con sus seguidores e integrarlo dentro de su timeline.
- **Timeline:** El conjunto de todos los *tweets, retweets* y demás contenido multimedia con el que el usuario haya interactuado. La *timeline* del usuario se muestra al pulsar sobre sus nombres, junto al resto de la descripción del usuario.
- **Comentarios:** Funcionan de la misma manera que en cualquier otra red sociales. El botón de comentar se usa para iniciar una conversación o para enviar una respuesta a cualquier *tweet* específico que el usuario haya encontrado. Toda la cadena de comentarios que responden a un *tweet* es visible para el resto de usuarios, debajo del *tweet* donde respondieron y en la *timeline* del mismo.
- **Menciones:** Consiste en cualquier tipo de publicación en la que se añada el nombre de un usuario con el que está relacionado.
- **Mensajería directa (DM):** Otra característica básica de Twitter son los *DMs*, utilizados en caso de que se quiera enviar a un usuario específico un mensaje privado que solo pueda ser visto por ellos y no por el resto de seguidores.Para que otro usuario reciba y envíe mensajes directos, ese usuario tiene que seguir o aceptar la solicitud de enviar un mensaje de vuelta en base al primer mensaje que se le haya enviado. Aunque también se encuentra la opción de bloquear todos los mensajes directos.

Además de usuarios particulares, Twitter alberga múltiples negocios que usan plataformas para:

- Promocionarse a sí mismos.
- Construir relaciones con los clientes.
- Obtener opiniones de la audiencia/clientes.
- Gestión de su reputación en línea.

  …

3.2.2 Rotten Tomatoes

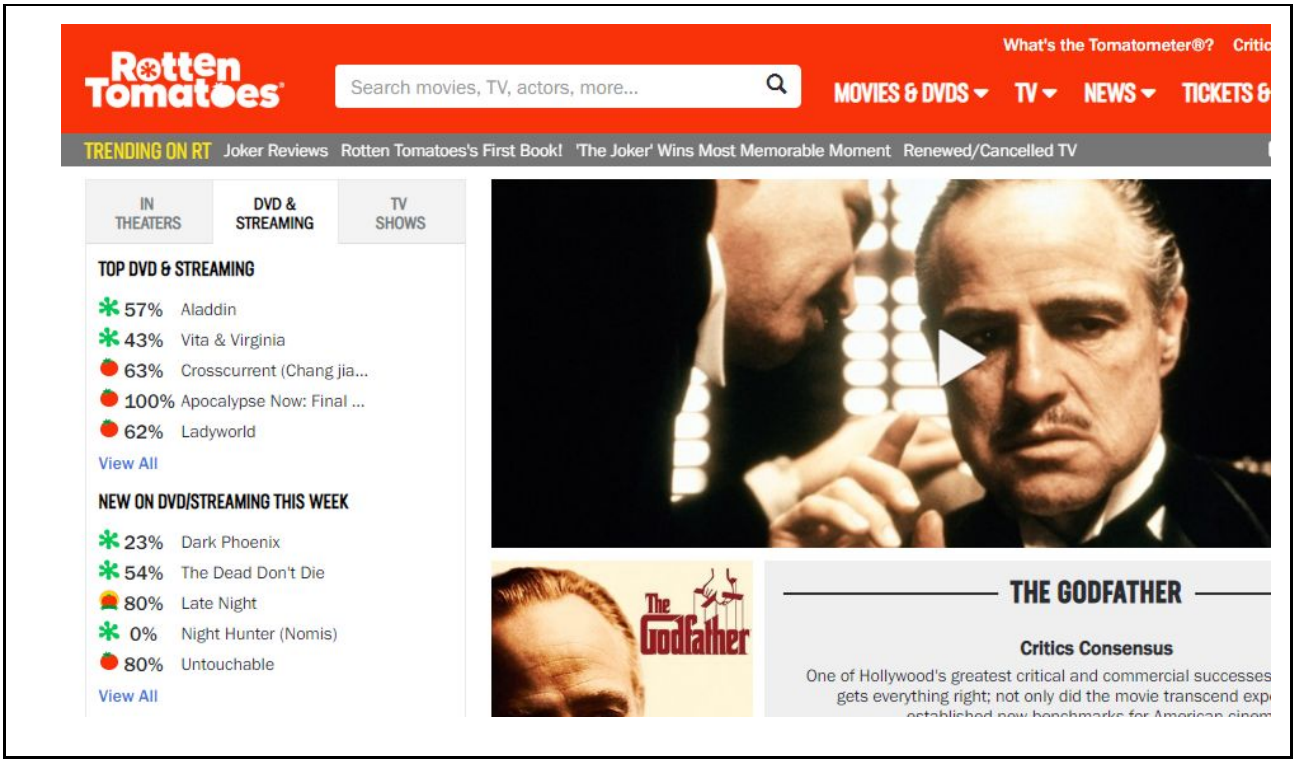


Figura 3.2.2 Ejemplo de una película en Rotten Tomatoes

Rotten Tomatoes[14] fue lanzado en 1998, como una página web que reúne opiniones de los críticos de cine y televisión y asigna una puntuación que puede ser: "*fresh*" (positivo) o "*rotten*" (negativo) a las diferentes películas.

Cuando obtuvo un nivel elevado de importancia, fue adquirida por *Fandango*, un sitio web que vende entradas anticipadas a películas para muchas de las principales cadenas de cine.

La puntuación que Rotten Tomatoes asigna a una película corresponde al porcentaje de críticos que han juzgado la película como "*certified fresh*", lo que significa que su opinión es más positiva que negativa. La idea es ofrecer rápidamente a los espectadores un consenso crítico.

Las opiniones de aproximadamente 3,000 críticos, también conocidos como "Críticos aprobados por el tomatómetro" que han cumplido con los criterios establecidos por Rotten Tomatoes, se incluyen en las puntuaciones del sitio, aunque no todos los críticos revisan cada película, por lo que cualquier calificación dada se deriva más típicamente de unos pocos cientos de críticos, o incluso menos.

Las puntuaciones no incluyen a cualquiera que se haga llamar crítico o tenga un blog de películas; Rotten Tomatoes solo agrupa a los críticos que han estado publicando regularmente reseñas de películas con un medio de información razonablemente leído durante al menos dos años, y esos críticos deben estar en activo, lo que significa que han publicado al menos una crítica en el último año.

El sitio considera que este subconjunto de usuarios son los más importantes, dado que sus opiniones aparecen al lado de la ficha y descripción de la película, y calcula una puntuación separada que sólo los incluye a ellos.

A medida que se acumulan las opiniones de críticos acerca de una película determinada, Rotten Tomatoes mide el porcentaje de opiniones positivas frente a las negativas y asigna una calificación total:

- Las puntuaciones que tengan **más de 60% de votos positivos** se consideran frescas.
- Las puntuaciones que tengan **menos del 59% de votos positivos** se consideran podridas.
- Los usuarios normales también pueden dar su opinión y la emiten especificando un **número de estrellas que varía entre 1 y 5**.

Para obtener el sello de "*certified fresh*", una película necesita al menos 40 opiniones de críticos, de las cuales, el 75% deberían ser nuevas y cinco de ellas ser de críticos de alto rango dentro de la página.

Las puntuaciones que se le asignan a una película en Rotten Tomatoes pueden tener distintos significados:

- Si un usuario le diera a una película una opinión mixta, que por lo general, pueda ser considerada positiva (que equivaldría entre 3 y 5 estrellas), esta opinión recibe el mismo peso que cualquier otra crítica.
- El hecho de que una película tenga como puntuación "*rotten*" tampoco implica que sea una mala película, ya que películas con una media de 5 (del 1 al 10), entrarían también dentro de esa categoría.

Es importante tener en cuenta que Rotten Tomatoes como entidad, nunca valora las películas por si misma, solo calculan el **consenso de todos los usuarios de la plataforma**.

3.3 Adquisición, integración y almacenamiento de datos

Para adquirir los datos de las fuentes especificadas anteriormente se utilizarán tres *scripts* escritos en Python que, de distintas maneras, extraerán y limpiarán los datos para posteriormente ser almacenados de manera estructurada en archivos de texto.

3.3.1 Script Python para Twitter

Este *script* utiliza la versión estándar de la API REST de Twitter para realizar búsquedas de *tweets* relacionados con diferentes películas.

La versión estándar de la API de Twitter solo permite recabar *tweets* de hasta 7 días en el pasado. Esto hace que el *script* esté limitado a menos que se page por la versión Premium, que puede llegar hasta 30 días atrás, o Enterprise que permite un acceso total a los tweets de la plataforma.

Sin embargo en su versión estándar, sirve perfectamente para generar un banco de datos de prueba o para conseguir información de películas que hayan salido al cine recientemente, ya que mucha gente estará hablando de ellas y se encontrarán más *tweets* dentro de ese espacio de tiempo.

Los datos que se recabarán de cada tweet serán:

- La cantidad de *likes* que tiene.
- La cantidad de *retweets que tiene.*
- Si el tweet ha sido publicado por una cuenta verificada.
- El texto publicado en el tweet.
- La fecha en la que ha sido publicado.

Además, se añadirá a cada tweet un identificador y el nombre de la película que se utilizó para encontrarlo.

Funcionamiento:

El *script* funciona usando python, y se puede utilizar por sí mismo a través de la terminal de linux con la ayuda de este comando:

```
# python3.6 tweetssearch2.py godfather 1 2 3 all
```

- **python3.6:** versión de python utilizada para el *script*.
- **tweetsearch2.py:** nombre del *script*.
- **godfather:** película de ejemplo que se utilizara para hacer la búsqueda.
- **1:** número mínimo de *likes* que debe tener el tweet para ser escogido.
- **2:** número mínimo de *retweets* que debe tener el tweet para ser escogido.
- **3:** número máximo de *tweets* que se quieren recuperar.
- **all:** tipo de cuentas aceptadas para ser escogidas. Si el valor es "*all*" significa que todas valen, pero si es "*verified*" solamente las cuentas verificadas se podrán recuperar.

Al usarlo se mostraran los diferentes ids de los *tweets* que se van procesando por la consola, y los que cumplan los requerimientos se guardarán en:

```
/home/user/FlumetoHive/tweets.txt
```

```
100001  it_chapter_two  1754    311     verified        “I waas on Saturday Night Live. That’s like the worst show to be on if you have anxiety.” Bill Hader talks Barry, anxiety, and “IT Chapter Two”:
https://t.co/Py5S1WHZVm         2019-09-05
100002  it_chapter_two  2278    361     verified        You don’t want to miss IT. Get tickets now: https://t.co/h2T4K6Ti08 #ITMovie    2019-09-05
100003  it_chapter_two  808     79      verified        Pretty much sums it up... Hey go see our movie IT: CHAPTER TWO in theatres NOW!! 🎈🎈  2019-09-05
100004  it_chapter_two  50      19      verified        Dare to experience the end of #ITChapter2 in the most immersive way possible? Reserve your seat to @ITMovieOfficial in #IMAX theatres and prepar
feel like you’re trapped inside a pitch black, inescapable house of horror. Now playing. https://t.co/3GV6d1YSIK        2019-09-05
100005  it_chapter_two  178     27      unverified      """I had no idea, but when that scene came about, it was like, of course. I don't want to give too much away, but I think it made a lot of sense
because a lot of kids go through that. I thought it was really smart."" Finn Wolfhard about Richie's reveal 📖: https://t.co/XZ0hnsF61T "       2019-09-05
100006  it_chapter_two  65      20      verified        Really dig this @SassyMamainLA chat with Finn Wolfhard and Bill Hader—truly my favorite actors in 'It Chapter Two'—for @GQMagazine      2019-09-
100007  it_chapter_two  268     41      verified        After 'It Chapter Two' and 'Stranger Things,' Finn Wolfhard is an accidental horror star        2019-09-05
100008  it_chapter_two  153     46      verified        [Interview] One of the Original Losers from ‘IT’ 1990 Makes a Cameo Appearance in ‘IT: Chapter Two’!    2019-09-05
100009  it_chapter_two  35      11      unverified      STOP SPOILING IT chapter two. the movie JUST came out and not everyone has seen it, almost no one. don’t ruin someone else’s experience just bec
you’re too excited about it. we’ve waited almost two years for the movie. or at least put in your tweets that are spoilers thanks       2019-09-05
100010  it_chapter_two  216     23      unverified      IT CHAPTER TWO is an overstuffed grab-bag of sincere Stephen King love, weird nods to other movies, jokes that don't work, and fucking bonkers
imagery. It is a whole lot of strange for a mainstream blockbuster.     2019-09-05
100011  it_chapter_two  96      18      unverified      Highest YouTube trailer view counts for upcoming 2019 releases (as of this past Saturday): Frozen II - 104.4M Joker - 87.0M IT Chapter Two - 67.
Star Wars: The Rise of Skywalker - 52.7M Jumanji: The Next Level - 50.9M Terminator: Dark Fate - 28.7M Gemini Man - 28.1M       2019-09-05
100012  it_chapter_two  380     134     unverified      stop spoiling it chapter two. the movie JUST came out and not everyone has seen it, almost no one. don’t ruin someone else’s experience just bec
you’re too excited about it. we’ve waited almost two years for the movie. thanks        2019-09-05
100013  it_chapter_two  79      23      unverified      pov: you spoil IT chapter two for me because i don’t watch until sunday and i’m beating your ass        2019-09-05
100014  it_chapter_two  157     45      unverified      Check out a fantastic new interview and photoshoot with Finn Wolfhard (@FinnSkata) for the @latimes! Finn talks about his unintentional success
the horror genre, @ITMovieOfficial, @GoldfinchMovie, @Stranger_Things and @TurningMovie 🎬📷: https://t.co/8tQwqXyeWA  2019-09-05
100015  it_chapter_two  39      19      verified        IT CHAPTER TWO Eyeing Record $200 Million Global Box Office Opening! https://t.co/g6GB496Aj2    2019-09-05
100016  it_chapter_two  64      12      unverified      GUYS YES IT IS INDEED POSSIBLE TO SPOIL A BOOK ADAPTATION.... R U DUMB.....? THERE WAS SOOO MUCH ADDED TO CHAPTER TWO SO DONT GIVE ANY OF THAT "
BOOKS BEEN OUT FOR 30 yrs” BULLSHIT     2019-09-05
```

Figura 3.3.1: Contenido de *tweets*.txt

3.3.2 Script Python para las opiniones de Rotten Tomatoes

Este *script* utiliza librerías orientadas al *web-crawling*, lo que le permite extraer información de cualquier página que haya sido especificada. La más importante de ellas es *“Beautifulsoup”*, que crea un árbol con todos los elementos del documento y los parsea de manera sencilla.

El *script* obtiene información de dos tipos de página distintas:

- Las páginas para las review de los críticos:

*"https://www.rottentomatoes.com/m/"+**line**+"/reviews/?page="+**str(page_number)**+"&sort="*

- Las páginas para las review de los usuarios:

"https://www.rottentomatoes.com/m/"+**line**+"/reviews/?page="+**str(page_number)**+"&type=user"

Dónde *“line”* es el nombre de la película, y *“str(page_number)”* es el número de la página donde se encuentra.

Este *script* va página por página extrayendo opiniones hasta que llegue a la página indicada por los argumentos en la que debe parar, o hasta que la página devuelva al *script* un estado que no sea “200”, es decir, que ya no existan más páginas de las cuales extraer la información.

Los datos que se obtendrán de cada crítica serán:

- El tipo de usuario que publicó la crítica (crítico o usuario normal).
- En caso de que sea crítico, la calificación que le dio: *“rotten”* (mala) o *“fresh”* (buena).
- En el caso que sea un usuario normal, la valoración que le dio a la película en estrellas.
- El texto de la crítica.
- El día en que fue publicada.

Además, se añadirá a cada crítica un identificador y el nombre de la película que se utilizó para encontrarlo.

La gran mayoría de datos extraídos de las críticas no hace falta transformarlos, a excepción de la fecha, que se debe ajustar al formato especificado por la tabla:

Fecha original en la web: *“July 12, 2016”* -> Fecha transformada: *“2016-07-12”*

Funcionamiento:

El *script* funciona usando python, y se puede utilizar por sí mismo a través de la terminal de linux con la ayuda de este comando:

```
# python3.6 rottencrit.py godfather 200 300
```

- **python3.6:** versión de python utilizada para el *script.*
- **rottencrit.py:** nombre del *script.*
- **godfather:** película de ejemplo que se utilizara para hacer la búsqueda.
- **200:** número de críticas escritas por críticos profesionales que se quieren (200 críticas implica recorrer 10 páginas, ya que cada página de críticos contiene 20 opiniones).
- **300:** número de críticas escritas por usuarios que se quieren (300 críticas implica recorrer 30 páginas, ya que cada página de críticos contiene 10 opiniones).

Aquellas críticas que se vayan obteniendo se irán almacenando en el fichero:

```
/home/user/FlumetoHive/rottenrev.txt
```

```
1	godfather	Critic	Fresh	NULL	If ever there was a great example of how the best popular movies come out of a merger of commerce and art, "The Godfather" is it.	2019-04-15
2	godfather	Critic	Fresh	NULL	Casting in these roles has been exceptional, particularly with Al Pacino, a relative new comer as Michael. James Caan, too, deserves special mention, to
his role as Sonny... But the prime extra ingredient of the film is Brando.	2019-02-06
3	godfather	Critic	Fresh	NULL	This is a curious film. One comes to understand, even to condone, the activities of the Godfather and his clan. 2017-03-15
4	godfather	Critic	Fresh	NULL	Brando is the strong magnet that will draw fans to The Godfather. But behind-the-scenes creativity is of equal value to this film of towering
proportions.	2015/02/22
5	godfather	Critic	Rotten	NULL	I don't see how any gifted actor could have done less than Brando does here. His resident power, his sheer innate force, has rarely seemed weaker.
2015-02-22
6	godfather	Critic	Fresh	NULL	The Godfather is overflowing with life, rich with all the grand emotions and vital juices of existence, up to and including blood.	2014/02/26
7	godfather	Critic	Fresh	NULL	These films are as elegant as they are expansive, acutely perceptive and operatic in their high emotions.	2019/07/09
8	godfather	Critic	Fresh	NULL	The movie is preposterously entertaining, telling Puzo's compendium of old-time Mafia anecdotes with all the gravity of Old Testament epic.	2018-03-
9	godfather	Critic	Fresh	NULL	Francis Ford Coppola's adaptation of Mario Puzo's bestseller remains the great American epic of the immigrant dream turned family business.	2018-02-
10	godfather	Critic	Rotten	NULL	I found that flogging about for three hours in that quagmire was spiritually debilitating and a crazy waste of time.	2018-01-30
11	godfather	Critic	Fresh	NULL	What else can you say about the movie 45 years after it changed the face of cinema? The actors really nailed their parts? Some of the dialogue feels lik
might enter the lexicon and never leave?	2017-05-23
12	godfather	Critic	Fresh	NULL	It is a long (three hours), often exciting and always well-directed film about the struggle for survival of one of the five Mafia 'families,' the Corleo
in late 1940s New York. 2015-03-11
```

Figura 3.3.2: Contenido de rottenrev.txt

3.3.3 Script Python para la información de la película

Este es el *script* más sencillo de todos, ya que toda la información necesaria para que funcione se encuentra en el mismo lugar en la misma página, y no hace falta transformar ni parsear los datos de ninguna manera para que funcione. Al igual que el *script* anterior, este utiliza *“Beautfiul Soup”* para su funcionamiento.

La página que utiliza el *script* es del tipo:
"https://www.rottentomatoes.com/m/"+**line**"
Dónde "*line*" es el nombre de la película de la cual se extrae la información.

Los datos que se obtendrán de cada película serán:

- Nombre de la película.
- A qué porcentaje de críticos les gustó la película.
- A qué porcentaje de la audiencia le gustó la película.
- Sinopsis de la película.
- Edad recomendada para ver la película:
    - para todos los públicos (G).
    - con supervisión paternal(PG).
    - para mayores de 13 años(PG-13).
    - para mayores de 18 (R).
- Género de la película.
- Nombre del director.
- Nombre del escritor.
- Día que salio la película al cine.
- Día que salió el disco de la película a la venta.
- Duración de la película.
- Estudio que produjo la película.

En caso de que se esté buscando por una serie en vez de una película, información como "día en que salió al cine" o "duración de la película" no estarán disponibles.

Funcionamiento:

El *script* funciona usando python, y se puede utilizar por sí mismo a través de la terminal de linux con la ayuda de este comando:

```
# python3.6 rottendesc.py godfather
```

- **python3.6:** version de python utilizada para el *script.*
- **rottencrit.py:** nombre del *script.*
- **godfather:** película de ejemplo que se utilizara para hacer la búsqueda.

Aquellos datos que se vayan obteniendo se irán almacenando en el fichero:

```
/home/user/FlumetoHive/rottendesc.txt
```

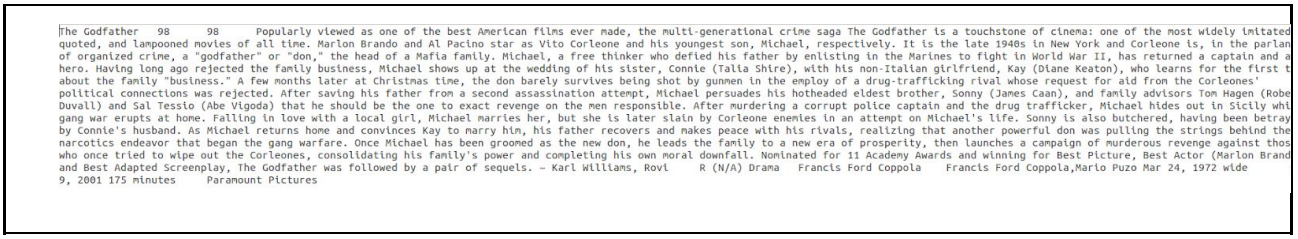


Figura 3.3.3: Contenido de rottendesc.txt

3.4 Transformación de los datos con Flume

El trabajo de Flume es consiste en estar monitorizando continuamente los archivos que están cambiando debido al uso de los *scripts*, y pasar sus contenidos directamente a las tablas correspondientes en Hive. Su función sería la de un **disparador**.

Este no es generalmente el propósito de Flume, ya que fue creado para interactuar con HDFS solamente, pero existen varios conectores para extender su uso a otras bases de datos, incluyendo Hive.

El conector que se usará en este caso es **HiveSink**, y cambia ligeramente la sintaxis y el tipo de datos que se deben aportar a Flume para su correcto funcionamiento.

Directorios involucrados:

Flume estará monitorizando continuamente el directorio "*FlumetoHive*" que se encuentra dentro del directorio *home*, donde los *scripts* apuntan a la hora de guardar sus contenidos. Y dentro del directorio, Flume se activará al detectar cambios en:

- ***/home/user/FlumetoHive/tomatodesc.txt***
  Archivo resultante al recoger información de la película en Rotten Tomatoes.
- ***/home/user/FlumetoHive/tomatorev.txt***
  Archivo resultante del *script* para las críticas en Rotten Tomatoes.
- ***/home/user/FlumetoHive/tweets.txt***
  Archivo resultante del *script* para los tweets de Twitter.

Archivo de configuración y HiveSink:

Flume realiza su trabajo en base al archivo de configuración que tenga asignado al iniciarlo. En este caso se están enlazando tres archivos distintos a tres tablas distintas dentro de Hive.

El archivo que se usará es **hivesink.conf**, e incluyen dentro de él:

- **Un agente (*Agent*):** en este caso se llamará "*FileAgent".*
- **Fuentes (*Sources*):** se creará una para cada *script*, tres en total (*catrotten, inforotten* y *cattwitter*).

- **Sumideros (*Sinks*):** serán tres de ellos, porque cada resultado de cada *script* acaba en una tabla distinta (*k1, k2, k3*).
- **Canales (*Channels*):** tres canales para conectar cada uno de los *scripts* a su tabla correspondiente en Hive. (*c1, c2 y c3*).

Estos elementos se configuran de la siguiente manera:

```
Tipo de fuente, en este caso, se obtienen los datos a partir de la consola
usando "cat" sobre el archivo a insertar, y el canal por donde pasarán los
datos al finalizar
FileAgent.sources.catrotten.type = exec
FileAgent.sources.catrotten.command = cat /home/TFG/FlumetoHive/tomatorev.txt
FileAgent.sources.catrotten.channels = c1
Tipo de base de datos, en este caso, Hive
FileAgent.sinks.k1.type = hive
El canal por el que hive obtendrá los datos
FileAgent.sinks.k1.channel = c1
Donde se encuentra la "metastore" de Hive, en este caso, apunta a la que se
tiene almacenada en MySQL
FileAgent.sinks.k1.hive.metastore = thrift://localhost:9083
Base de datos en la que se van a guardar
FileAgent.sinks.k1.hive.database = analisis
Tabla dentro de la base de datos en la que se van a guardar
FileAgent.sinks.k1.hive.table = rotten
Se especifica a Flume que a la hora de insertar los contenidos en la tabla,
divide los datos a partir de las tabulaciones
FileAgent.sinks.k1.serializer = DELIMITED
FileAgent.sinks.k1.serializer.delimiter = "\t"
FileAgent.sinks.k1.serializer.serdeSeparator = '\t'

Las diferentes columnas en las que se van a insertar, separando los datos por
tabulaciones.

FileAgent.sinks.k1.serializer.fieldnames =
id,film,review_type,rot_fresh,review_stars,text,review_date
```

Este es el comando que se utiliza para iniciar Flume:

```
flume-ng agent --conf conf --conf-file $FLUME_HOME/conf/hivesink.conf
--name FileAgent --classpath
"/home/dani/hive/hcatalog/share/hcatalog/*":"/home/dani/hive/lib/*"
-Dflume.root.logger=DEBUG.console
```

Sus parámetros son:

--conf: indica que se usará un archivo de configuración.
--conf-file: indica dónde se encuentra el archivo de configuración.
--name: indica el nombre del agente que se va a iniciar dentro del fichero.

--classpath: indica la ubicación de las librerías para la base de datos de Hive.
-Dflume.root.logger: activa el modo depuración.

3.5 Estructura de la base de datos

3.5.1 HDFS

HDFS en este proyecto tendrá un papel menor, y se usará de manera directa en limitadas ocasiones. Su utilidad consiste en **almacenar datos** que se vayan a usar posteriormente en las tablas de Hive. El mejor ejemplo de esto es la tabla “diccionario”, que no cambia después de cada ejecución como las demás.

Aparte de lo anteriormente mencionado, se estará utilizando HDFS de manera **indirecta** para guardar el almacén de Hive (warehouse), por lo que las dos rutas relevantes para HDFS son:

- *“/dictionary.tsv”*
- *“/user/hive/warehouse”*

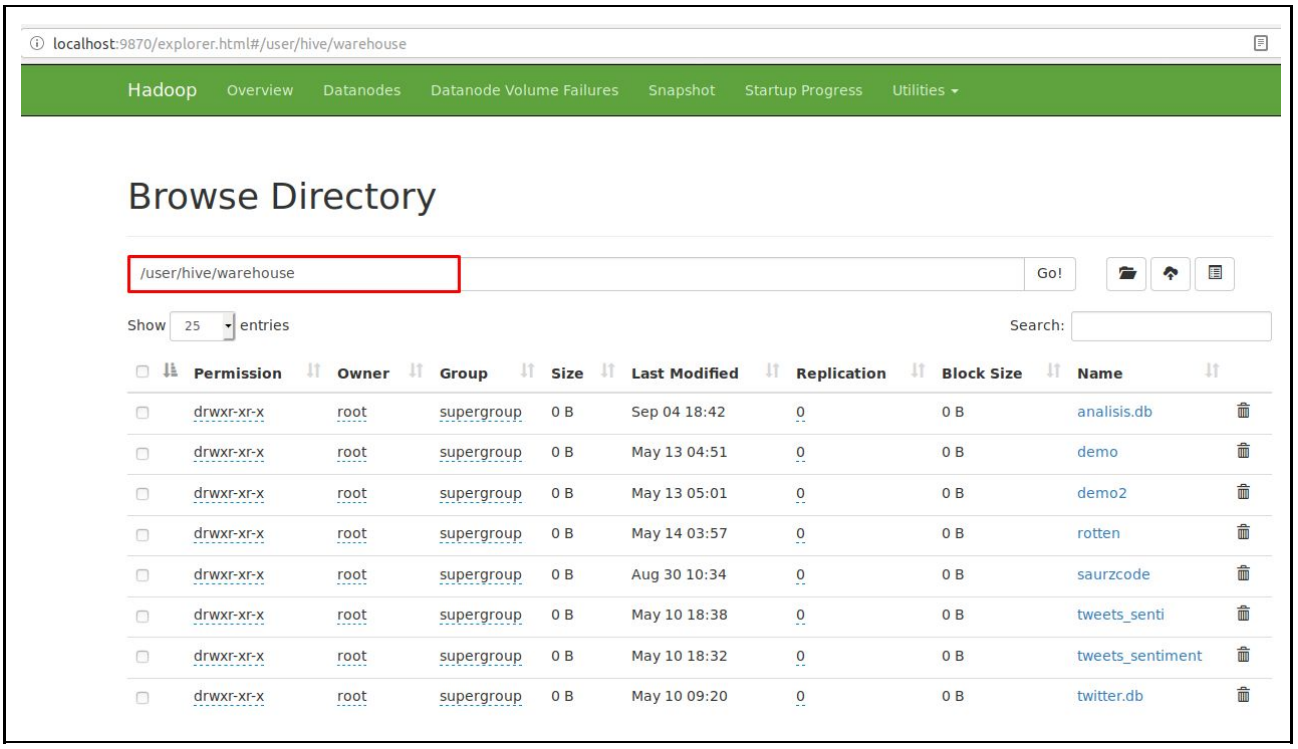


Figura 3.5.1: Almacén (Warehouse) de Hive dentro de HDFS, mostrando todos los datos de las tablas creadas hasta el momento.

3.5.2 Hive

Dado que este proyecto está orientado al procesamiento de grandes cantidades de datos, se escogió Hive no solo porque su rendimiento es mejor en este tipo de casos en comparación a otras bases de datos SQL[15], también se escogió porque combinar Hadoop y Hive suele dar los mejores resultados incluso en otros ecosistemas Big Data similares.

Todas las tablas se encuentran en la misma base de datos, llamada “analisis” dentro de Hive, y se encuentran almacenadas como ficheros tipo ORC, un formato diseñado para la carga de trabajos en Hadoop mucho más eficiente y ligero que el resto[16].

Tabla de información sobre la película

```
CREATE TABLE infofilm (
film string,
tatometer int,
audiencescore int,
description string,
rating string,
genre string,
directed string,
written string,
release string,
disc string,
duration string,
studio string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS ORC
TBLPROPERTIES ('transactional' = 'true') ;
```

- **film:** nombre de la película.
- **tatometer:** porcentaje de a cuantos críticos les gustó la película.
- **audiencescore:** puntuación de la audiencia.
- **description:** descripción de la película.
- **rating:** para que clase de público está hecha la película.
- **genre:** género de la película.
- **directed:** nombre del director.
- **written:** nombre del escritor.
- **release:** día que salio la película al cine.
- **disc:** día que salió el disco de la película a la venta.
- **duration:** duración de la película.
- **studio:** estudio que produjo la película.

Tabla de opiniones rotten tomatoes

```
CREATE TABLE rotten (
id int,
film string,
review_type string,
rot_fresh string,
review_stars int,
text string,
review_date date)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS ORC
TBLPROPERTIES ('transactional' = 'true') ;
```

- **id:** identificador de la crítica.
- **film:** nombre de la película.
- **review_type:** si la crítica fue hecha por un crítico (Critic) o por un usuario (User).
- **rot_fresh:** puntuación usada por los críticos.
- **review_stars:** puntuación usada por los usuarios.
- **text:** texto de la crítica.
- **review_date:** día que la crítica fue publicada.

Tabla de *tweets*

```
CREATE TABLE twitter (
id int,
film string,
favorite_count int,
retweet_count int,
verified string,
text string,
post_date string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS ORC
TBLPROPERTIES ('transactional' = 'true') ;
```

- **id:** identificador de los *tweets.*
- **film:** nombre de la película.
- **favorite_count:** número de *likes* en el *tweet.*
- **retweet_count:** número de *retweets.*
- **verified string:** si el usuario está verificado (*verified*) o no (*unverified*).
- **text:** texto del *tweet.*

- **post_date:** día que fue escrito.

Tabla diccionario

```
CREATE TABLE dictionary (
word string,
polarity string
)
PARTITIONED BY (type string)
CLUSTERED BY (polarity) INTO 4 BUCKETS
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS ORC
TBLPROPERTIES ('transactional' = 'true') ;
```

- **type:** cuánta importancia tiene la palabra dentro del texto, en lo que se refiere a polaridad:
  - weaksubj : no muy positiva o negativa.
  - strongsubj : muy positiva o muy negativa.
- **word:** la palabra en sí misma.
- **polarity:** si la palabra es positiva o negativa.

En esta última tabla se está añadiendo dos propiedades distintas a las demás tablas:

```
PARTITIONED BY (type string)
CLUSTERED BY (polarity) INTO 4 BUCKETS
```

"Partitioning"* y *"Bucketing" son dos conceptos importantes dentro de Hive.
Dado que Hive trabaja por encima de HDFS, y guarda las tablas divididas como ficheros, se le pueden dar instrucciones a Hive para que divida la tabla de un determinado modo, en eso consiste *"Partitioning" y "Bucketing"*.

La tabla "*dictionary*" va a tener alrededor de 8200 filas, y al contrario que las demás, no se van a cambiar sus contenidos en ningún momento y solo se va a recorrer para encontrar palabras con las que comparar el texto, lo que la hace perfecta para este procedimiento.

"Partitioning" va a dividir la tabla y almacenarla en carpetas usando la columna *"type"*, como type solo puede tener dos valores "*strongsubj*" y "*weaksubj*", eso implica que se crearán dos carpetas en la que se almacenarán todas las filas que tengan ese valor en común. A su vez, *"Bucketing"* va a dividir todas las filas escogidas para cada carpeta en ficheros en base al valor que se haya asignado, en este caso, cuatro ficheros.

Este proceso ayuda a su vez al *Map/Reduce* cuando hayan *queries* que requieran mucho tiempo para procesar, ya que hive sabe que debe dividir el trabajo en base a cómo se haya dividido la tabla con *"Partitioning" y "Bucketing",* disminuyendo los tiempos que se tarda en buscar en la tabla.

3.6 Análisis de sentimientos

3.6.1 Funcionamiento

El análisis de sentimientos se realiza dentro de *Hive*. Esta base de datos opera de una manera muy similar a una base de datos tipo SQL, pero con algunas limitaciones.

Sin embargo, se puede realizar todas las operaciones necesarias para el análisis usando recursos como las vistas y la función *explode()* disponible en *Hive*.

La función explode() es capaz de tokenizar las cadenas de texto en base a los argumentos que se le pasen a la hora de crear la vista o tabla.

Es **importante** tener en cuenta que, mientras que en la tabla y vistas de Twitter se utilizan para determinar positividad **solamente** en el texto de los *tweets*, en las críticas de Rotten Tomatoes se está teniendo en cuenta no solo la positividad del texto, sino la **puntuación que los usuarios y críticos** hayan asignado a la película. Esto ayudará a que la valoración del texto pueda ser más exacta.

3.6.2 Vistas

En la primera vista de ambas tablas: "*rottenview*" y "*twitterview*" se va a dividir el texto, que está guardado como una variable de tipo *string*. De esta manera se obtiene un array de palabras identificado por la columna "*id*". *"LATERAL VIEW"* posibilita crear copias de los datos en las columnas colindantes que no vayan a ser transformadas por el uso de la función explode().

```
CREATE VIEW rottenview AS
SELECT id, words
FROM rotten LATERAL VIEW explode(sentences(lower(text))) dummy AS
words;
```

```
CREATE VIEW twitterview AS
SELECT id, words
FROM twitter LATERAL VIEW explode(sentences(lower(text))) dummy AS
words;
```

Figura 3.6.1: Contenidos de la vista "*twitterview*"

En "*twitterview2*" y "*rottenview2*" se va a dividir la cadena de texto mostrada anteriormente de manera que quede una sola palabra por fila, junto al identificador que indica de qué texto proviene.

```
CREATE VIEW twitterview2 AS
SELECT id, word
FROM twitterview LATERAL VIEW explode(words) dummy AS word;
```

```
CREATE VIEW rottenview2 AS
SELECT id, word
FROM rottenview LATERAL VIEW explode(words) dummy AS word;
```

Figura 3.6.2: Contenidos de la vista "*twitterview2*"

En “*twitterview3*” y “*rottenview3*” se comparan cada fila de las vistas “*twitterview2*” y “*rottenview2*” con las filas del diccionario. Y si alguna palabra del texto se encuentra en él, se le asigna un valor dependiendo de si se trata de una palabra positiva o una palabra negativa.

Además, se tiene en cuenta la columna “*type*” dentro del diccionario a la hora de asignar una puntuación, ya que determina cómo de importante es la palabra para la polaridad del texto. Por ejemplo, no es tan importante la palabra “sospechoso” como la palabra “odio”, aunque ambas sean negativas la segunda denota aún más negatividad.

Las puntuaciones asignadas a cada palabra se realizan de acuerdo al esquema especificado en las siguientes vistas.

```
CREATE VIEW twitterview3 AS SELECT
    id,
    t2.word,
    CASE
      WHEN d.polarity='negative' AND d.type='weaksubj' THEN -1
      WHEN d.polarity='positive' AND d.type='weaksubj' THEN 1
      WHEN d.polarity='negative' AND d.type='strongsubj' THEN -2
      WHEN d.polarity='positive' AND d.type='strongsubj' THEN 2
      ELSE 0
    END AS polarity
 FROM twitterview2 t2 LEFT OUTER JOIN dictionary d ON t2.word = d.word;
```

```
CREATE VIEW rottenview3 AS SELECT
    id,
    r2.word,
    CASE
      WHEN d.polarity='negative' AND d.type='weaksubj' THEN -1
      WHEN d.polarity='positive' AND d.type='weaksubj' THEN 1
      WHEN d.polarity='negative' AND d.type='strongsubj' THEN -2
      WHEN d.polarity='positive' AND d.type='strongsubj' THEN 2
      ELSE 0
    END AS polarity
 FROM rottenview2 r2 LEFT OUTER JOIN dictionary d ON r2.word = d.word;
```

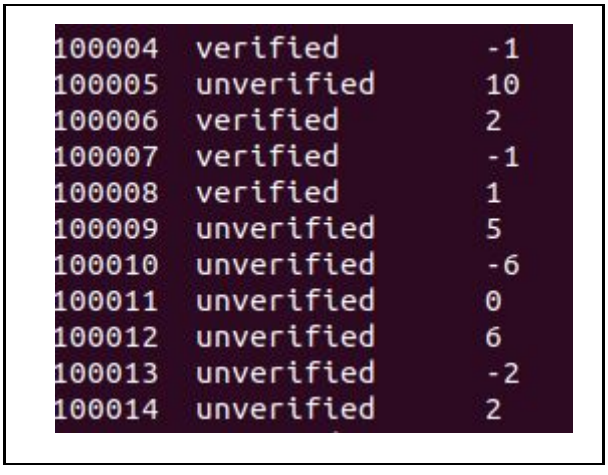
```
100006  chapter 0
100006  with    0
100007  stranger        -2
100007  wolfhard        0
100007  accidental      -1
100007  after   0
100007  an      0
100007  and     0
100007  two     0
100007  chapter 0
100007  things  0
100007  star    2
```

Figura 3.6.3: Contenidos de la vista "*twitterview3*"

En "*twitterview4*" y "*rottenview4*" se suman todas las puntuaciones en cada fila, de manera que se vea finalmente la puntuación total en cada una de las opiniones.

```
CREATE VIEW twitterview4 AS SELECT t.id, t.verified, SUM(t3.polarity)
AS value
FROM twitter t LEFT OUTER JOIN twitterview3 t3 ON t.id = t3.id
GROUP BY t.id, t.verified;
```

```
CREATE VIEW rottenview4 AS SELECT r.id, r.review_type, SUM(r3.polarity)
AS value
FROM rotten r LEFT OUTER JOIN rottenview3 r3 ON r.id = r3.id
GROUP BY r.id, r.review_type;
```



```
100004  verified        -1
100005  unverified      10
100006  verified        2
100007  verified        -1
100008  verified        1
100009  unverified      5
100010  unverified      -6
100011  unverified      0
100012  unverified      6
100013  unverified      -2
100014  unverified      2
```

Figura 3.6.4: Contenidos de la vista "*twitterview4*"

En “*twitterpolarity*” y “*rottenpolarity*” se van a contar y dividir todos los *tweets* y críticas en tres columnas, para observar cuántos de ellos son considerados: positivos, negativos y neutros.

Sin embargo, no se están contando todas las opiniones de la misma manera, ya que hay algunas opiniones con más **relevancia** que otras:

- Si una opinión en Twitter ha sido publicada por una cuenta verificada, esa opinión equivale a **dos opiniones** del resto de usuarios.
- Si una opinión en Rotten Tomatoes ha sido publicada por un crítico, esa opinión equivale a **tres opiniones** del resto de usuarios.

Esta ponderación mejora significativamente los resultados, ya que los usuarios verificados y los críticos suelen tener opiniones de mejor calidad a su nombre.

```
CREATE VIEW twitterpolarity AS
SELECT (t4.POSITIVOS+t4.POSITIVOS2*2) AS positivos,
(t4.NEGATIVOS+t4.NEGATIVOS2*2) AS negativos, (t4.NEUTROS+t4.NEUTROS2*2)
AS neutros
FROM
(SELECT
SUM(CASE WHEN value > 0 and verified = 'unverified' THEN 1 ELSE 0 END)
POSITIVOS,
SUM(CASE WHEN value < 0 and verified = 'unverified' THEN 1 ELSE 0 END)
NEGATIVOS,
SUM(CASE WHEN value == 0 and verified = 'unverified' THEN 1 ELSE 0 END)
NEUTROS,
SUM(CASE WHEN value > 0 and verified = 'verified' THEN 1 ELSE 0 END)
POSITIVOS2,
SUM(CASE WHEN value < 0 and verified = 'verified' THEN 1 ELSE 0 END)
NEGATIVOS2,
SUM(CASE WHEN value == 0 and verified = 'verified' THEN 1 ELSE 0 END)
NEUTROS2
FROM twitterview4) t4;
```

```
CREATE VIEW rottenpolarity AS
SELECT (r4.POSITIVOS+r4.POSITIVOS2*3) AS POSITIVOS,
(r4.NEGATIVOS+r4.NEGATIVOS2*3) AS NEGATIVOS, (r4.NEUTROS+r4.NEUTROS2*3)
AS NEUTROS
FROM
(SELECT
SUM(CASE WHEN value > 0 and review_type = 'User' THEN 1 ELSE 0 END)
```

```
POSITIVOS,
SUM(CASE WHEN value < 0 and review_type = 'User' THEN 1 ELSE 0 END)
NEGATIVOS,
SUM(CASE WHEN value == 0 and review_type = 'User' THEN 1 ELSE 0 END)
NEUTROS,
SUM(CASE WHEN value > 0 and review_type = 'Critic' THEN 1 ELSE 0 END)
POSITIVOS2,
SUM(CASE WHEN value < 0 and review_type = 'Critic' THEN 1 ELSE 0 END)
NEGATIVOS2,
SUM(CASE WHEN value == 0 and review_type = 'Critic' THEN 1 ELSE 0 END)
NEUTROS2
FROM rottenview4) r4;
```

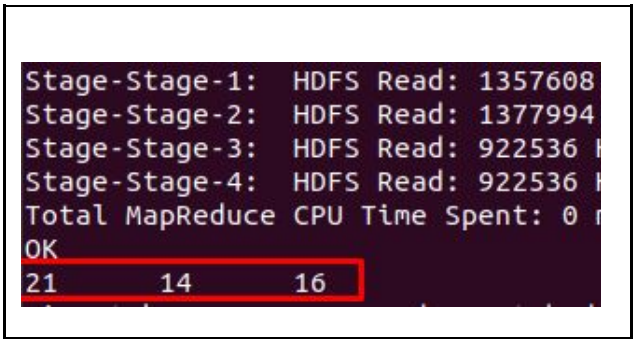


Figura 3.6.5: Contenidos de la vista "*twitterpolarity*"

3.6.3 Queries

Estas son las *queries* que se utilizan dentro de Hive para extraer la información relevante de los usuarios de Twitter y Rotten Tomatoes a través de las tablas y vistas previamente establecidas:

Palabras más utilizadas en los *tweets*:

```
SELECT
word, count(word) as number FROM twitterview3
WHERE polarity != 0
GROUP BY word
ORDER BY number DESC
LIMIT 3;
```

Palabras más utilizadas en las críticas:

```
SELECT
word, count(word) as number FROM rottenview3
WHERE polarity != 0
GROUP BY word
ORDER BY number DESC
LIMIT 3;
```

Distribución de opiniones positivas, negativas y neutras en Twitter:

```
SELECT * FROM twitterpolarity
```

Distribución de opiniones positivas, negativas y neutras en Rotten Tomatoes:

```
SELECT * FROM rottenpolarity
```

Distribución de opiniones positivas, negativas y neutras de todas las plataformas:

```
SELECT
(t.positivos+r.positivos) AS positivos,
(t.negativos+r.negativos) AS negativos,
(t.neutros+r.neutros) AS neutros
FROM twitterpolarity t, rottenpolarity r
```

Número de *tweets* sobre la película por fecha:

```
SELECT COUNT(id) FROM twitter GROUP BY post_date
```

Número de críticas sobre la película por fecha:

```
SELECT COUNT(id) FROM rotten GROUP BY review_date
```

Los tres comentarios más positivos en Twitter:

```
SELECT t.text, t4.value FROM twitterview4 t4, twitter t WHERE t.id =
t4.id ORDER BY t4.value DESC LIMIT 3
```

Los tres comentarios más positivos en Rotten Tomatoes:

```
SELECT r.text, r4.value FROM rottenview4 r4, rotten r WHERE r.id =
r4.id ORDER BY r4.value DESC LIMIT 3
```

Los tres comentarios más negativos en Twitter:

```
SELECT t.text, t4.value FROM twitterview4 t4, twitter t WHERE t.id =
t4.id ORDER BY t4.value ASC LIMIT 3
```

Los tres comentarios más negativos en Rotten Tomatoes:

```
SELECT r.text, r4.value FROM rottenview4 r4, rotten r WHERE r.id =
r4.id ORDER BY r4.value ASC LIMIT 3
```

3.7 Interfáz gráfica de la aplicación

3.7.1 Cliente de la aplicación

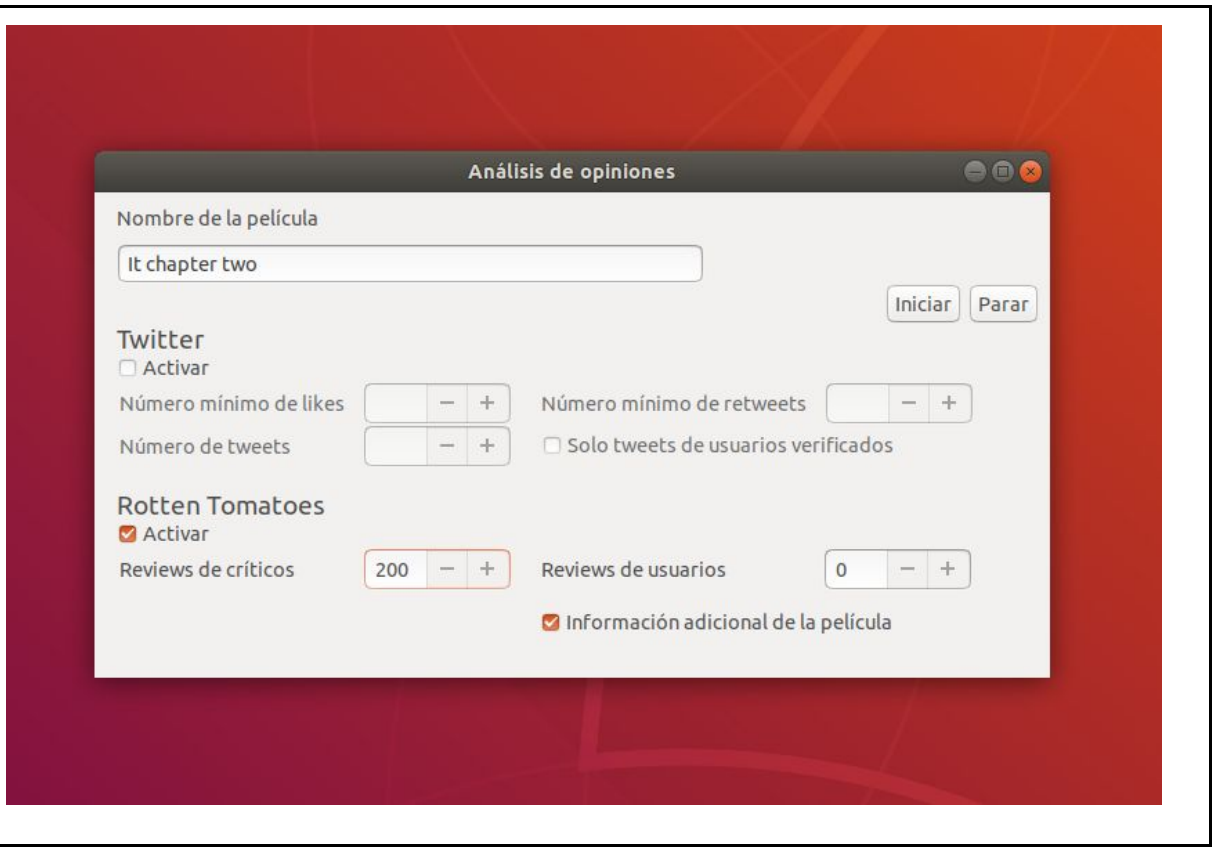


Figura 3.7.1: Cliente de la aplicación del proyecto

Opciones disponibles en la interfaz de usuario

- Entrada de texto para especificar la película a analizar.

- Opción a poder adquirir *tweets* para el análisis:
    - Mínimo número de *likes* que debe tener un *tweet* para ser escogido.
    - Mínimo número de *retweets* que debe tener el *tweet* para ser escogido.
    - Cantidad de *tweets* que se desean para el análisis.
    - Opción para escoger solamente *tweets* publicados por cuentas verificadas.

- Opción a poder adquirir reviews de Rotten Tomatoes para el análisis:
    - Cuantas *reviews* provenientes de críticos se desean para el análisis.
    - Cuantas *reviews* provenientes de usuarios se desean para el análisis.
    - Opción a incorporar información adicional de la película, como por ejemplo: nombre del director, sinopsis de la película….

Funcionamiento

Una vez se hace *click* en el botón de iniciar:

1) La ventana bloquea las entradas de textos y *checkboxes*.
2) Comprueba que todos los programas y procesos están iniciados, y en caso que no lo estén, los ejecuta.
3) Se limpian todas las tablas de la base de datos que lo necesiten.
4) Se activan los *scripts* para obtener las opiniones.
5) Cuando estos acaban su trabajo, el cliente activa Flume y carga los archivos resultantes de los *scripts* dentro de las tablas de Hive.
6) El cliente abre una ventana con el navegador predeterminado del sistema hacia Zeppelin.
7) Los datos resultantes del análisis se cargaran junto a Zeppelin en el navegador.

El proceso completo tarda en completarse en función a la cantidad de datos que se haya indicado procesar.

3.7.2 Zeppelin

Consiste en un cuaderno multipropósito, capaz de extraer información de diferentes bases de datos y visualizarla en multitud de maneras, tales como: tablas, histogramas, gráficos circulares… a partir de *queries*.

Configuración

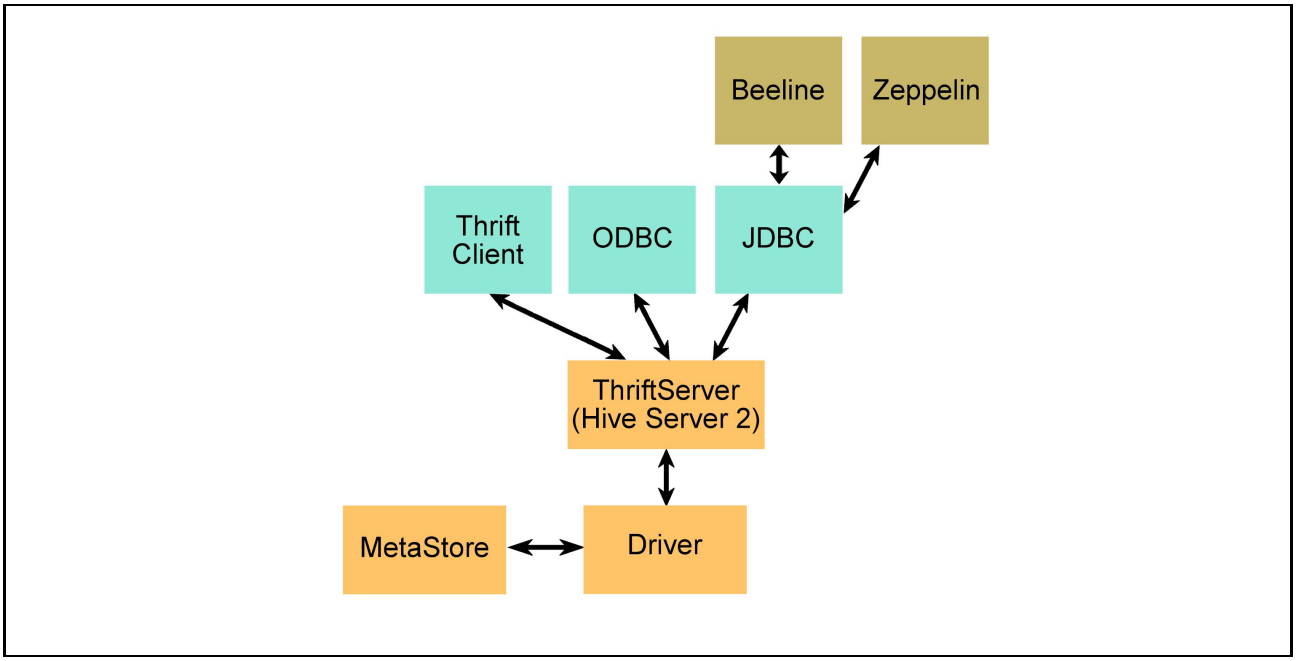


Figura 3.7.2: Conexión entre Hive y Zeppelin en el proyecto

Tal como se ve en el esquema, Zeppelin está conectado por medio de un interpretador JDBC (API) al controlador JDBC de Hive. Este es el mismo método de conexión que usa Beeline, el cliente nativo de Hive para conectarse al Hive Server 2.

Hive Server 2 es un servicio incorporado a Hive que permite a diferentes clientes utilizar la plataforma sin estar conectados directamente a ella. Hay muchas manera de conectarse a Hive Server 2, y estas se puede especificar en el fichero *"hive-site.xml"*, dentro de los archivos de configuración de Hive.

La conexión en este caso usa como protocolo de transporte HTML y sin ningún medio de autenticación (noSASL). Esto es debido a que solo se ha estado utilizando Zeppelin de manera local por medio del *localhost*, por lo que no hay riesgos para la seguridad.

Aparte, para conectarse al servidor de Hive hace falta configurar el interpretador por el lado de Zeppelin. Para ello se creó un perfil dentro del interpretador JDBC para Hive, y se especificaron los siguientes datos:

```
hive.url=jdbc:hive2://127.0.0.1:10002/analisis;transportMode=http;httpP
ath=cliservice
hive.user=hiveuser
hive.password=hivepassword
hive.driver=org.apache.hive.jdbc.HiveDriver
```

- **hive.url:** consistiría en la dirección con la cual conectarse al servidor Hive Server 2.
- **hive.user:** usuario que se desea usar para acceder a la base de datos.
- **hive.password:** contraseña para utilizar el usuario.
- **hive.driver:** El *driver* necesario para interpretar Hive Server 2.

Interfaz de Zeppelin

A continuación se mostrarán los resultados obtenidos de forma gráfica a través de Zeppelin para la película "*Scary Stories to tell in the dark*":

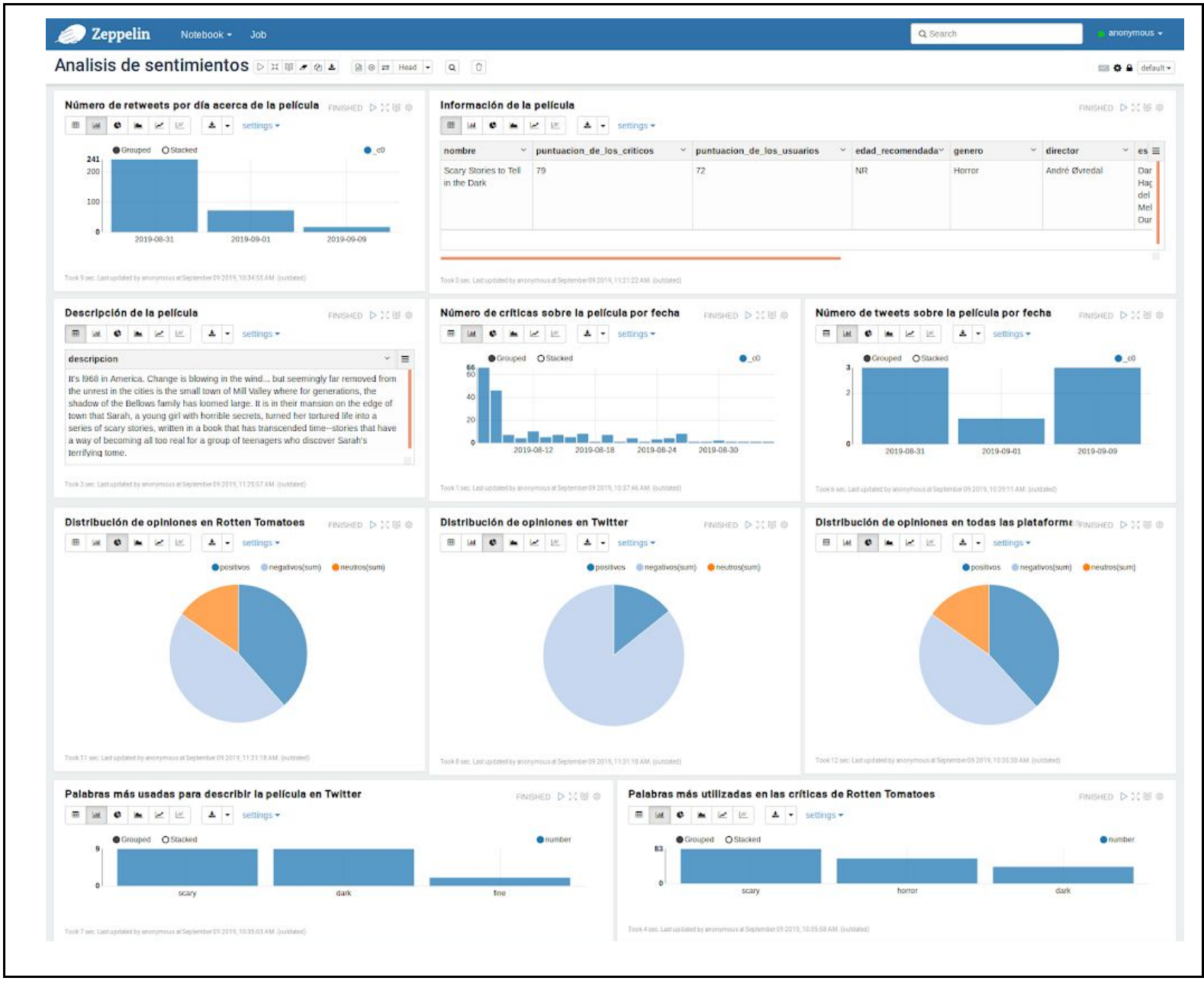


Figura 3.7.3: Primera parte de la página de Zeppelin

Datos mostrados:

- Número de *retweets* ordenados por día en forma de histograma de frecuencias.
- Datos relacionados con la película analizada:
  - Nombre de la película.
  - Porcentaje de opiniones positivas por parte de los críticos.
  - Porcentaje de opiniones positivas por parte de la audiencia.

    - Edad recomendada para ver la película.
    - Género.
    - Nombre del director.
    - Nombre del escritor.
    - Día de lanzamiento de la película al cine.
    - Día de lanzamiento del disco.
    - Duración de la película.
    - Estudio que la produjo.
- Sinopsis de la película analizada.
- Número de críticas sobre la película por fecha.
- Número de *tweets* sobre la película por fecha.
- Distribución de opiniones positivas, negativas y neutras en Rotten Tomatoes.
- Distribución de opiniones positivas, negativas y neutras en Twitter.
- Distribución de opiniones positivas, negativas y neutras de todas las plataformas.
- Palabras más utilizadas en las críticas.
- Palabras más utilizadas en los *tweets*.

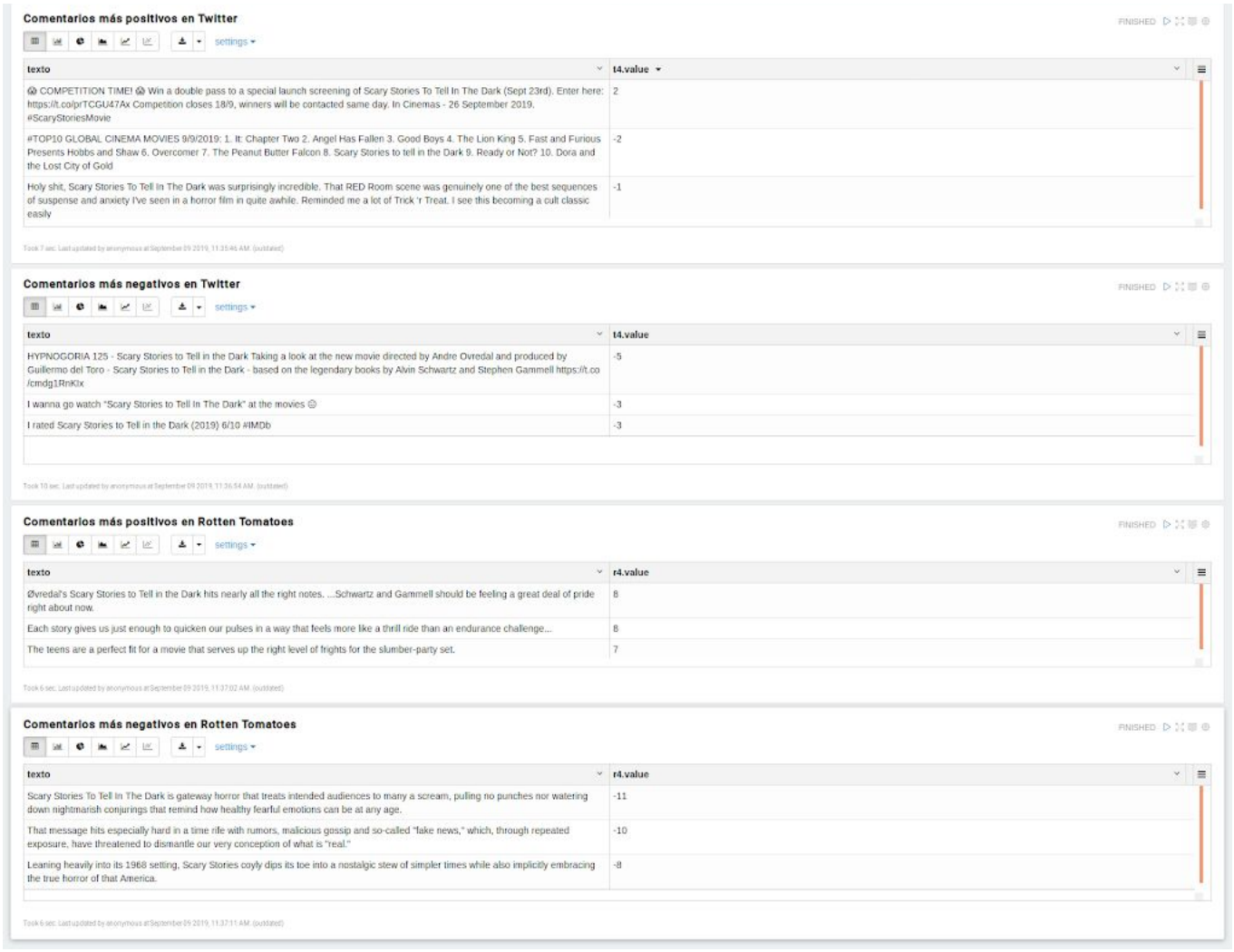


Figura 3.7.4: Segunda parte de la página de Zeppelin

Datos mostrados:

- Los tres comentarios más positivos en Twitter.
- Los tres comentarios más negativos en Twitter.
- Los tres comentarios más positivos en Rotten Tomatoes.
- Los tres comentarios más negativos en Rotten Tomatoes.

Capítulo 4 Resultados obtenidos

En este apartado se comparan los resultados obtenidos a partir de diferentes películas, y se relacionarán con otros factores asociados a las mismas, en este caso:

- Cantidad de dinero ganado por la película.
- Cantidad de dinero gastado en producir la película.
- Si han recibido algún premio.

(Nota: Tener en cuenta que debido a las limitaciones del equipo que se usó para el proyecto, y el hecho de que la API gratuita de Twitter no deja recabar tantas entradas de datos como se quisiera, las muestras utilizadas para el programa podrían no ser lo suficientemente grandes como para determinar un resultado preciso).

4.1 Tablas

En las tablas se mostrarán campos con datos provenientes de dos fuentes distintas:

- Película, Estreno, Premios, Recaudación, Coste y Recepción de Rotten Tomatoes pertenecen a diferentes páginas web.
- Distribución de opiniones global, Distribución en Twitter, Distribución en Rotten Tomatoes y todos los campos relacionados con los comentarios han sido elaborados a partir del análisis de sentimientos.

A continuación se mostrarán el análisis de dos películas ("*Furious 7*" y "*Scary Stories to tell in the dark*"), que dejan ver como el análisis de sentimientos puede llegar a fallar en ciertas ocasiones:

Leyenda			
P/ = Positivo	N/ = Negativo	E/= Neutro	RTs = Rotten Tomatoes

Tabla 4.1: Leyenda para las tablas de resultados

Información de la película		*Análisis de sentimientos*	
Película	Furious 7	**Distribución de opiniones global**	P/ 88% N/ 3% E/ 9%
Estreno	3 de Abril de 2015	**Distribución en Twitter**	P/ 35% N/ 40% E/ 25%
Premios	Ninguno	**Distribución en Rotten Tomatoes**	P/ 88% N/ 2% E/ 10%
Recaudación	$1,516,045,911		
Coste	$190,000,000		
Recepción en Rotten Tomatoes	Críticos= 81/100 Audiencia= 82/100		
Comentario más positivo (Twitter)	"#Furious7 is the best in the series, with thrilling action, an engaging story, a fantastic villain in Jason Statham, and a beautiful sendoff to Paul Walker whom we lost too soon"		
Comentario más negativo (Twitter)	"For God's sake! He just drove off a damn mountain. How is he not dead?! #Furious7"		
Comentario más positivo (RTs)	Great and familiar cast, excellent script, superb production values and the direction is tight as a drum. Can't say this is the best film of the franchise, because so many of them have been excellent. But it's easily the most daring.		
Comentario más negativo (RTs)	"Gleefully sexist (butttttts!), stupefyingly loud and heart-hurtingly ridiculous, you at least can't accuse Furious 7 of being boring. Or nuanced, or aware of the basic laws of science. But it sure is angry."		
Nº de Tweets	50	**Nº de críticas de RTs**	5322

Tabla 4.2: Resultados de la película: "*Furious 7*"

La información fue extraída de:

https://www.rottentomatoes.com/m/furious_7

https://www.boxofficemojo.com/movies/?id=fast7.htm

Se puede observar que las opiniones de Rotten Tomatoes están bastante cerca en este caso del 81-82% determinado por la plataforma si se realiza un análisis de sentimientos sobre el texto, sin embargo los resultados de Twitter no se ajustan al resto de resultados debido a que de esos 50 *tweets* que se recogieron, la mayoría estaban hablando acerca de cómo la película había sido superada por el Rey León (2019) en récords de audiencia, lo que explica el 40% de comentarios neutros.

Información de la película		*Análisis de sentimientos*	
Película	Scary Stories to tell in the dark	**Distribución de opiniones global**	P/ 36% N/ 48% E/ 16%
Estreno	9 de Agosto de 2019	**Distribución en Twitter**	P/ 27% N/ 36,5% E/ 36,5%
Premios	Ninguno	**Distribución en Rotten Tomatoes**	P/ 36% N/ 48% E/ 16%
Recaudación	$88,854,347		
Coste	$25,000,000		
Recepción en Rotten Tomatoes	Críticos= 79/100 Audiencia= 72/100		
Comentario más positivo (Twitter)	"😱 COMPETITION TIME! 😱 Win a double pass to a special launch screening of Scary Stories To Tell In The Dark (Sept 23rd). Enter here: https://t.co/prTCGU47Ax Competition closes 18/9, winners will be contacted same day. In Cinemas - 26 September 2019. #ScaryStoriesMovie"		
Comentario más negativo (Twitter)	"HYPNOGORIA 125 - Scary Stories to Tell in the Dark Taking a look at the new movie directed by Andre Ovredal and produced by Guillermo del Toro - Scary Stories to Tell in the Dark - based on the legendary books by Alvin Schwartz and Stephen Gammell https://t.co/cmdg1RnKIx"		
Comentario más positivo (RTs)	"Øvredal's Scary Stories to Tell in the Dark hits nearly all the right notes. ...Schwartz and Gammell should be feeling a great deal of pride right about now."		
Comentario más negativo (RTs)	"Scary Stories To Tell In The Dark is gateway horror that treats intended audiences to many a scream, pulling no punches nor watering down nightmarish conjurings that remind how healthy fearful emotions can be at any age."		
Nº de Tweets	64	**Nº de críticas de RTs**	2386

Tabla 4.3: Resultados de la película: "*Scary Stories to tell in the dark*"

La información fue extraída de:

https://www.rottentomatoes.com/m/scary_stories_to_tell_in_the_dark/

https://www.boxofficemojo.com/movies/?id=scarystoriestotellinthedark.htm

En este caso, ninguno de los análisis en los textos de Twitter y Rotten Tomatoes llegan a acercarse a el promedio presente en Rotten Tomatoes (72-79%). Esto se debe a que la película en este caso pertenece al género del horror, y aquellos que la critiquen, incluso si no es negativamente, utilizaran palabras que el diccionario del proyecto determina como negativas, por ejemplo: "*scream*", "*fearful*", "*scary*"... entre otras.

4.2 Gráficas

Se realizaron 15 análisis con distintas películas que se estrenaron durante el mes de Agosto de 2019, y se comparó la puntuación global de todos los críticos y usuarios de la plataforma de Rotten Tomatoes con los resultados del análisis de sentimientos en este proyecto:

	Rotten Tomatoes		**Análisis de Sentimientos**		
Película	**Críticos**	**Usuarios**	**Positivo**	**Neutro**	**Negativo**
FAST & FURIOUS PRESENTS: HOBBS & SHAW	66	88	91	4	5
SCARY STORIES TO TELL IN THE DARK	79	72	36	16	48
GOOD BOYS	79	86	80	0	20
READY OR NOT	87	79	33,5	21	45,5
THE KITCHEN	21	69	75	6	19
ALADDIN	57	94	64	0	36
JOHN WICK: CHAPTER 3 - PARABELLUM	90	87	81	5	14
MEN IN BLACK INTERNATIONAL	22	66	52,5	13,5	34
TONE-DEAF	36	38	11,5	0	88,5
BE NATURAL: THE UNTOLD STORY OF ALICE GUY-BLACHÉ	96	97	90	2,5	7,5
BOOKSMART	97	77	88	0	12
JACOB'S LADDER	5	26	7	15	78
THE DEAD DON'T DIE	54	38	13,5	23	63,5
ECHO IN THE CANYON	94	92	78	8,5	13,5
THE TOMORROW MAN	43	45	32	39	29

Tabla 4.4: Películas con sus diferentes valoraciones en el mismo orden que en las gráficas

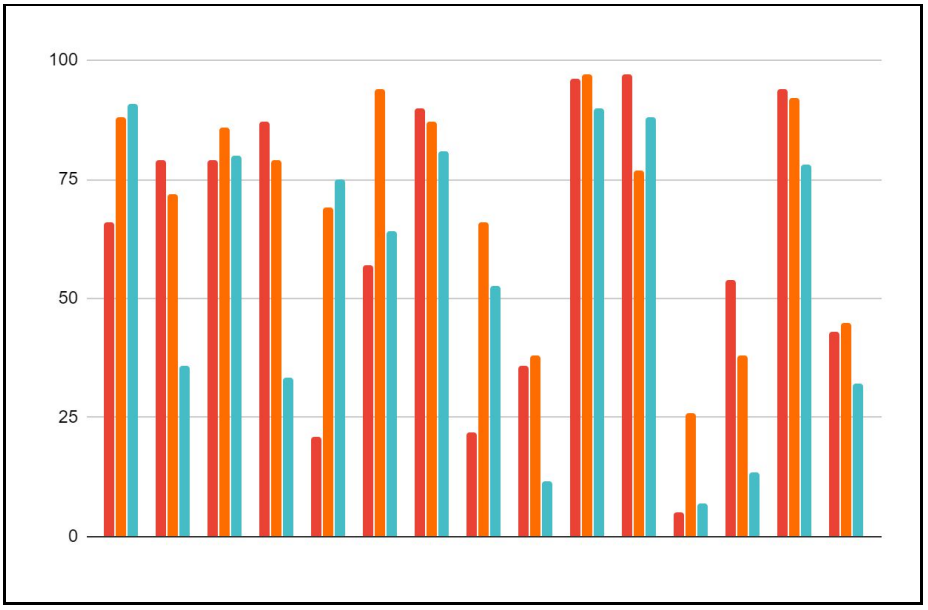


Figura 4.1: Gráfica para comparar las métricas de Rotten Tomatoes con el análisis de sentimientos

- Rojo: Valoración positiva de los críticos.
- Naranja: Valoración positiva de los usuarios.
- Azul: Valoración positiva calculada a partir del análisis.

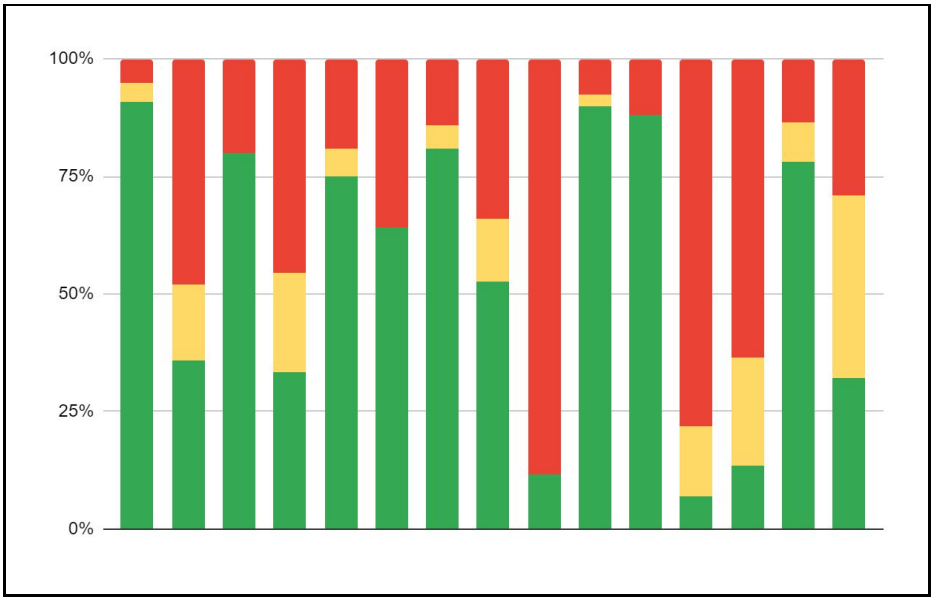


Figura 4.2: Distribución de opiniones positivas, neutras y negativas por película analizada

- Verde = Valoración positiva a partir del análisis.
- Amarillo = Valoración neutra a partir del análisis.
- Rojo = Valoración negativa a partir del análisis.

Se puede observar en la figura 4.1, que los resultados suelen ser, la mayoría de las veces, bastante cercanos a la cifra estimada por Rotten Tomatoes.
Sin embargo, el análisis siempre valora a las películas de miedo con una puntuación mucho menor a la cifra estimada debido a las palabras que se usan para describirlas.
Y también se da el caso contrario, aquellas películas que son del género de comedia son valoradas por encima del resto, porque se definen con palabras como: "*happy*", "*joyful*", "*hilarious*"... que el diccionario considera como muy positivas.

Capítulo 5 Conclusiones y líneas futuras

La elaboración de este TFG ha requerido del estudio de múltiples ámbitos de la informática, especialmente en todo lo relacionado al ecosistemas Big Data. Gran parte del tiempo invertido en este proyecto se usó para explorar las diferentes alternativas y opciones posibles para su desarrollo, hasta el punto que las herramientas que había planteado usar en un principio fueron reemplazados por otras que se ajustaban de mejor manera al trabajo que quería realizar.

Debido a la naturaleza de este proyecto, tuve que mejorar mis habilidades con Python para crear *scripts* capaces de hacer *web-crawling* e intercambiar información con diferentes APIs. También fue necesario aprender acerca de las diferentes maneras que existen de realizar un análisis de sentimientos y cuál se adaptaría mejor al ecosistema Big Data.

Aprendí a usar Map/Reduce como técnica de procesamiento de tareas, y los distintos usos del sistema de ficheros de Hadoop, que funciona de una manera totalmente distinta a las bases de datos SQL tradicionales.

El proyecto final que conseguí crear, aún siendo funcional, se puede mejorar en bastantes aspectos:

- Se podrían tener varias tablas con distintos diccionarios para ser capaz de analizar textos en otros idiomas más allá del inglés.
- Sería posible incorporar un corrector de faltas ortográficas al diseño, para que palabras mal escritas puedan seguir siendo analizadas.
- También se podrían añadir diccionarios específicos para cada género cinematográfico, y así evitar errores de cálculos como en algunos de los ejemplos anteriores.

  ...

La oportunidad que he tenido de trabajar en este proyecto me ha dado una nueva perspectiva en cuanto a las utilidades y versatilidad que poseen los entornos Big Data, y en como apostar e invertir en este tipo de tecnologías es beneficioso hoy en día.

Capítulo 6 Summary and Conclusions

The development of this TFG has required the study of multiple areas of information technology, especially when it comes to Big Data ecosystems. Much of the time invested in this project was used to explore the different alternatives and possible options for its development, to the point in which the tools I thought of using at the beginning were replaced by others that fit better he job I wanted them to perform .

Due to the nature of this project, I had to improve my skills with Python to create scripts capable of doing web-crawling and exchanging information with different APIs. It was also necessary to learn about the different ways that exist to perform a sentiment analysis and which one would best adapt to the Big Data ecosystem.

I learned to use Map/Reduce as a task processing technique, and the different uses of the Hadoop file system, which works in a totally different way than traditional SQL databases.

The final project I managed to create, while still functional, can be improved in many ways:

- You could have several tables with different dictionaries to be able to analyze texts in other languages beyond English.
- It would be possible to incorporate a spell checker to the design, so that misspelled words can still be analyzed.
- You could also add specific dictionaries for each film genre, and thus avoid calculation errors as in some of the previous examples.

  ...

The opportunity I had to work on this project has given me a new perspective regarding the utilities and versatility of Big Data environments, and how betting and investing in this type of technology is beneficial nowadays.

Capítulo 7 Presupuesto

En este apartado se muestra una tabla con el presupuesto de todos los elementos que se recomienda tener para llevar a cabo el desarrollo de este proyecto:

7.1 Hardware

Tipo	Descripción	Precio
Servidor (Nodo Maestro)	- Red de 1 Gbps - Procesador de 8 cores - Memoria RAM de 24-48GB (ECC) - Fuente de alimentación redundante - 6 discos duros de 512GB SSD - Discos en RAID	3000€
Servidor (Nodo Esclavo)	- 4 Discos duros de 1TB SATA - Red de 1 Gbps - Procesador de 4 cores - Memoria RAM de 30 GB	1500€
Monitor	- Monitor pnp genérico de 1920x1080	300€
Ratón	- Ratón Logitech	60€
Total		4860€

Tabla 7.1: Presupuesto del proyecto

Se pudo desarrollar el ecosistema Big Data y obtener resultados utilizando el modo pseudo distribuido de Hadoop, sin embargo, en un caso real en el que se utilizarían grandes cantidades de datos para recabar información, sería preferible contar con un entorno distribuido como el mencionado en la tabla, con múltiples equipos y escalable en caso de que se necesiten más servidores.

7.2 Recursos humanos

Ha continuación se muestra el coste total y el número de horas requeridas para llevar este proyecto a cabo:

Personal	Horas de trabajo	Coste/Hora	Total
1	350	18€	6300€

Tabla 7.2: Coste de los recursos humanos

Bibliografía

[1] "*Cisco Visual Networking Index: Forecast and Trends, 2017–2022 ....*"
https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.
[2] "*Despliegue de un clúster Spark sobre Docker para Big Data*"
https://riull.ull.es/xmlui/bitstream/handle/915/3088/Despliegue%20de%20un%20cluster%20Spark%20sobre%20Docker%20para%20Big%20Data.pdf?sequence=1&isAllowed=y
[3] "*Análisis de ficheros log de la WiFi-ULL usando técnicas de Big Data*"
https://riull.ull.es/xmlui/bitstream/handle/915/1412/Analisis%20de%20ficheros%20log%20de%20la%20WiFi-ULL%20usando%20tecnicas%20de%20Big%20Data..pdf?sequence=1&isAllowed=y
[4] "*Big Data y la Visualización en el ámbito Educativo*"
https://riull.ull.es/xmlui/bitstream/handle/915/5860/Big%20Data%20y%20la%20Visualizacion%20en%20el%20ambito%20Educativo.pdf?sequence=1&isAllowed=y
[5] "*Herramienta de Text Mining aplicado a textos cortos y redes sociales*"
https://repositorio.unican.es/xmlui/bitstream/handle/10902/10706/Callejo%20Gonzalez%20Javier.pdf?sequence=1&isAllowed=y
[6] "*Sentiment analysis and opinion mining*"
https://www.morganclaypool.com/doi/abs/10.2200/s00416ed1v01y201204hlt016
[7] "*Social media competitive analysis and text mining: A case study in the pizza industry*"
https://www.sciencedirect.com/science/article/pii/S0268401213000030
[8] "*Text mining solutions*"
https://www.textminingsolutions.co.uk
[9] "*Chinetek Strategy*"
http://chinetekstrategy.com
[10] "*Twitter-sentimental-analysis*"
https://github.com/kailashjoshi/Twitter-sentimental-analysis
[11] "*Sentiment Analysis*"
https://github.com/kjahan/opinion-mining
[12] "*Twitter Sentiment Analysis and n-gram with Hadoop and Hive SQL*"
https://gist.github.com/umbertogriffo/a512baaf63ce0797e175
[13] "*Twitter*": https://twitter.com

[14] "*Rotten Tomatoes*": https://www.rottentomatoes.com

[15] "Sethy, Rotsnarani & Dash, Santosh & Panda, Mrutyunjaya. (2017). *Performance Comparison Between Apache Hive and Oracle SQL for Big Data Analytics*. 130-141. 10.1007/978-3-319-60618-7_14."

[16] *ORC: Intelligent Big Data file format Hadoop Hive*

https://www.semantikoz.com/blog/orc-intelligent-big-data-file-format-hadoop-hive/