UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

EXTRACTOR Y COMPARADOR DE CARACTERÍSTICAS PARA ESTABLECIMIENTOS TURÍSTICOS EMPLEANDO ANÁLISIS DE SENTIMIENTOS CON BIG DATA

Autor: Néstor Marín Siruela

Tutor: Luis Miguel Hernández Acosta

Titulación: Ingeniero de Telecomunicación

Fecha: Mayo 2017

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

EXTRACTOR Y COMPARADOR DE CARACTERÍSTICAS PARA ESTABLECIMIENTOS TURÍSTICOS EMPLEANDO ANÁLISIS DE SENTIMIENTOS CON BIG DATA

HOJA DE FIRMAS

Alumno/a

Fdo.: Néstor Marín Siruela

Tutor/a

Fdo.: Luis Miguel Hernández Acosta

Fecha: Mayo 2017

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

EXTRACTOR Y COMPARADOR DE CARACTERÍSTICAS PARA ESTABLECIMIENTOS TURÍSTICOS EMPLEANDO ANÁLISIS DE SENTIMIENTOS CON BIG DATA

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal **Secretario/a**

Fdo.: Fdo.:

Fecha: Mayo 2017

Índices

Tabla de Contenido

Índice de Ilustraciones

Índice de Tablas

Índice de Ecuaciones

Memoria

Capítulo 1: Introducción

1.1. Introducción

Pocas frases tan sencillas resultan tan significativas como "*El conocimiento es poder*"[1]. Y, sin embargo, las posibilidades que abarca el conocimiento son infinitas. Desde el conocimiento de uno mismo hasta el conocimiento del resto del mundo, no hay aspecto de este que no permita obtener una ventaja significativa con respecto a quien carece de él.

En esta Era de las Telecomunicaciones es posible obtener información con una facilidad que no se ha visto anteriormente en la historia de la humanidad. Las estadísticas no fallan: el uso de Internet a lo largo y ancho del planeta sólo crece, y crece a un ritmo increíble. Ya en 2014, la cantidad de personas con acceso a Internet superaba los tres billones de usuarios, y las personas con contratos de banda ancha para sus terminales móviles superaban los 3.4 billones[2], y en 2015, la estadística[3] indica que un 44% de la población mundial ya es internauta.

La capacidad de generar datos por parte de Internet desemboca en una cantidad masiva de información potencial que navega en la Red, la cual puede ser canalizada mediante herramientas que transforman esos datos en conocimiento real. Por ejemplo, pulseras que analicen el ritmo cardíaco de las personas y permitan generar un esquema diario de su salud física, mediciones del clima utilizando millones de sensores repartidos por todo el mundo, mantener la temperatura de un acuario a un valor óptimo realizando análisis de datos en tiempo real… Sin contar con todas las operaciones que pueden realizarse mediante métodos electrónicos y que no requieran hardware o tarea humana. Los canales de generación de datos son, por tanto, extremadamente numerosos y muy ricos en información.

Considerando, por tanto, el poder real que ofrece el conocimiento, es normal que empresas y gobiernos utilicen multitud de herramientas que permitan recabar dicho conocimiento de los datos disponibles, y en caso necesario, perfeccionar o personalizar las herramientas para su mayor beneficio. Sin embargo, realizar esta clase de algoritmos es bastante complejo por la increíble cantidad de datos a considerar.

Para poder manipular tantos datos y obtener información de estos, es preciso recurrir a técnicas como la **Minería de Datos**[4]. La forma en la que

están estructurados los datos, el origen de estos, para qué se utilizarán… Todo ello cobra una especial relevancia. El objetivo final es obtener un conocimiento de calidad y que resulte efectivo; la cantidad de datos recabados o procesados no es realmente importante si no ayudan al objetivo final.

La minería de datos permite encontrar patrones comunes en la información, los cuales sirven especialmente en negocios o empresas, como se está demostrando de forma espectacular en la actualidad, tanto en industria como en el sector servicios, y naturalmente en el Internet de las Cosas. El proceso más popular a la hora de realizar minería de datos es el conocido como **CRISP–DM**, el cual se puede resumir en los siguientes 6 pasos:

1. **Entendimiento del Negocio** – ¿Para qué va a utilizarse los datos?
2. **Entendimiento de los Datos** – ¿Qué son los datos?
3. **Preparación de los Datos** – ¿De qué forma se almacenan y analizan los datos?
4. **Modelado** – ¿Cómo se realiza la minería?
5. **Evaluación** – ¿Son los resultados correctos?
6. **Despliegue** – Ejecución de la minería de datos

Por tanto, para realizar una minería de datos apropiada solo es necesario tener nociones de programación, así como una cierta idea de los datos con los que se trabajará y la idea que se quiere obtener de estos.

No obstante, la minería de datos sigue suponiendo el manejo de una inmensa cantidad de información para ser utilizada en un tiempo razonable por un solo sistema. De ahí que sea necesario adentrarse en los conceptos de **Big Data**[5], y la manipulación de datos a escalas muchísimo mayores.

Big Data es un novedoso concepto que toma como premisa el hecho que, efectivamente, el exceso de datos a procesar es un problema, aunque uno con solución. Bien sea aplicar distintos algoritmos de navegación a través de los datos (sacrificando detalle por velocidad de procesado), o el procesado paralelo con múltiples sistemas enlazados entre sí, la realidad es que es utilizado cada vez más por las increíbles ventajas que aporta con un coste sorprendentemente reducido. El hecho que las grandes multinacionales del planeta hayan mostrado su completa

aceptación y hayan ratificado sus ventajas no deja lugar a dudas de la importancia de considerar sus conceptos.

Así mismo, dentro de Big Data se encuentran varias ramas, muy importantes, que versan sobre distintas formas de extraer los datos. Una que se tendrá muy en cuenta en este Proyecto de Fin de Carrera es la conocida como **Análisis de Sentimiento**[6]. Dentro del análisis de sentimiento se engloban una serie de algoritmos capaces de extraer si un término o un concepto provocan sentimientos positivos o negativos. También incluye la extracción de palabras clave de un texto, como los verbos o los sustantivos, e incluso analizar el idioma.

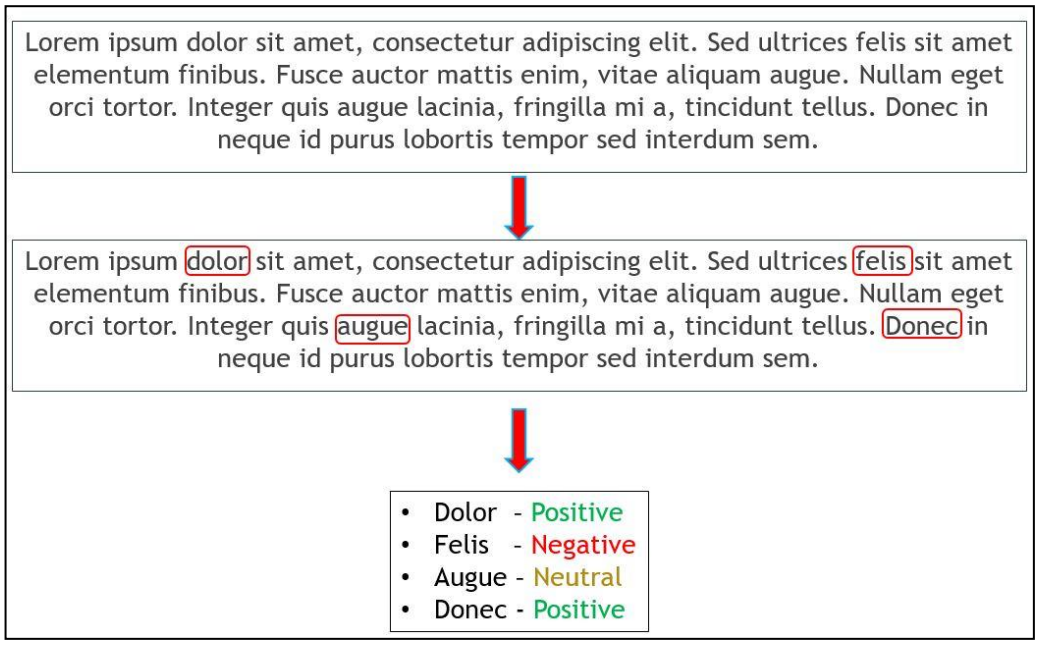


Ilustración 1 – Esquema del Análisis de Sentimiento

1.2. Objetivos del Proyecto

El objetivo final de este Proyecto Fin de Carrera (**E**xtractor y **C**omparador de **C**aracterísticas para **E**stablecimientos **T**urísticos empleando **A**nálisis de **S**entimientos con Big Data), o **ECCETAS**, es desarrollar un sistema que, utilizando el análisis de sentimiento como soporte de funcionamiento, permita recabar las características positivas o negativas de un establecimiento turístico (o varios) de forma gráfica, apoyándose en Big Data para el almacenamiento y gestión de los datos iniciales. A tal fin, será necesario cumplir los siguientes objetivos parciales:

1.2.1. Estudio del estado del arte.

Considerando la gran variedad de las herramientas aptas para el caso práctico del extractor y comparador, se deberán buscar aquellas que ofrezcan una mejor funcionalidad, bien por su rapidez, su robustez o su flexibilidad.

1.2.2. Análisis y selección de herramientas especializadas.

Dado que el presente proyecto tiene su núcleo en el análisis de sentimiento, será necesario considerarlo a la hora de realizar la selección de herramientas, también teniendo en mente las áreas de extracción y representación de datos.

1.2.3. Estudio, recolección y procesado de los datos disponibles en las redes sociales.

Se estudiarán las páginas web de establecimientos turísticos con el fin de tomar ideas iniciales de cuál será la forma más apropiada de obtener el conjunto de datos. Ello dependerá también de las herramientas escogidas en el paso anterior.

1.2.4. Diseño del extractor y comparador.

Disponiendo de una idea de la naturaleza de los datos y las herramientas apropiadas, se procederá a la generación del sistema completo, que enlace las herramientas entre sí para obtener el resultado final y permita un uso sencillo por parte del usuario.

1.2.5. Utilización del extractor y comparador y análisis de los resultados obtenidos.

Disponiendo de los datos y el sistema completo, se procederá a la obtención de resultados fiables y su representación, de forma que se verifique la utilidad de todo el sistema.

1.3. Metodología

Desde un inicio, la metodología de diseño de este Proyecto Fin de Carrera ha pasado por esquematizar y sistematizar, buscando no obtener una única caja negra que resuelva todos los problemas, sino disponer de un sistema dinámico y adaptable, de forma que sea posible actualizar sus partes con nuevas herramientas o conceptos.

A tal fin, el sistema general de Eccetas está dividido en 3 Bloques de Diseño:

- **Extracción**, donde se recogen los datos y se adaptan para los siguientes bloques.
- **Análisis**, donde se realiza el núcleo central del Proyecto y se obtienen resultados.
- **Representación**, donde se sintetizan los resultados anteriores en gráficas sencillas.

Por tanto, los pasos a realizar en el desarrollo son el diseño y creación de cada uno de los Bloques anteriormente mencionados, sin olvidar el enlace necesario con Big Data, que permitirá el almacenamiento y búsqueda de los datos iniciales del sistema.

Considerando lo expuesto anteriormente, se decidió optar por un diseño secuencial en bloques, comenzando por el caso de la Extracción. Se consideraron algunas fuentes básicas de información, y a raíz de ello, se realizó la búsqueda de una herramienta que permitiera la extracción de dichas fuentes, encontrándose la que se expondrá más adelante.

Así mismo, fue necesario sintetizar un código que permitiera una implementación de dicha herramienta, ya que por un lado era necesario automatizar la inyección de información, y por otro, almacenar correctamente los resultados de la herramienta. El sistema resultante permitía no solo el almacenamiento de los datos sino su inclusión en un sistema de Big Data cuya base de datos permitía el filtrado y selección de datos, tarea necesaria al ser unos archivos extremadamente extensos (la opción de realizar un histórico de los datos extraídos requiere la capacidad de manipular cantidades enormes de datos).

Disponiendo de esos datos iniciales, era necesario crear el núcleo del Bloque Análisis, por lo que se analizaron distintas opciones de herramientas de análisis de sentimiento hasta llegar a la herramienta elegida. Como en el caso anterior, era necesario generar un código que transformara los datos extraídos desde la base de datos al formato apropiado para la nueva herramienta, y otro código que recogiera los resultados y los almacenara de una forma práctica y legible por parte del usuario.

Llegados a este punto, era necesario encontrar una última herramienta que hiciera el trabajo del Bloque Representación, y transformara los datos del Bloque Análisis en imágenes que aportaran una información de mucha mayor calidad, y que no hiciera necesario aprender a interpretar los datos para poder obtener conocimiento de estos. Obtenida la herramienta apropiada, fue necesario integrarla al sistema, generando un Bloque adaptable e independiente de gran parte del resto de elementos, que genera el resultado final del sistema Eccetas.

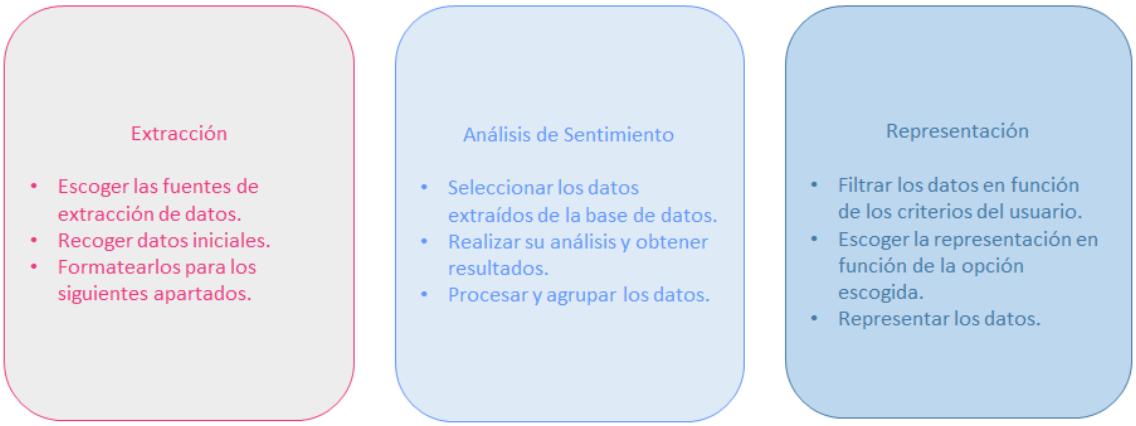


Ilustración 2 – Descripción de Bloques de Diseño

1.4. Peticionario

Actúa como peticionario de este proyecto la Escuela de Ingeniería de Telecomunicación y Electrónica (EITE), de la Universidad de las Palmas de Gran Canaria (ULPGC), siendo la realización de este Proyecto de Fin de Carrera requisito indispensable para la obtención del título de Ingeniero de Telecomunicación.

1.5. Estructura de la Memoria

La memoria de este Proyecto Fin de Carrera está separada en 8 capítulos, los cuales se describen brevemente a continuación.

Capítulo 1. Introducción: En este capítulo se explica por encima la premisa en la que se apoya el presente Proyecto de Fin de Carrera, indicando la importancia y el poder que provee el conocimiento, y cómo las tecnologías actuales requieren nuevos métodos para poder manipular cantidades tan grandes de este. Se expone la necesidad de utilizar técnicas de Minería de Datos, así como de Big Data, y la importancia que esta filosofía de diseño tendrá en el Proyecto. Así mismo, se habla del análisis de sentimiento, la piedra angular del Proyecto y lo que da potencial a este.

Capítulo 2. Estado del Arte: Se indican todas las tecnologías utilizadas en este Proyecto de Fin de Carrera, separadas en función de los Capítulos que se podrán ver a continuación, así como un conjunto general de tecnologías empleadas en todo el sistema. Así mismo, se expone la justificación de por qué las herramientas presentadas han sido escogidas.

Capítulo 3. Bloque Extracción: En este capítulo se exponen los criterios que se seguirán a la hora de tomar datos para el proyecto, asegurándose que tendrán una estructura útil y valor. Se indica, además, el caso particular para el que se utilizará este bloque, y se procede a presentar la herramienta Import.IO que permitirá obtener los datos. Así mismo, se acompañará del código que hace completo al bloque.

Capítulo 4. Subbloque Almacenamiento: En este breve capítulo se exponen los aspectos de Big Data utilizados en este Proyecto, concretamente, las capacidades para manipular y almacenar grandes cantidades de datos. Se definen las capacidades y utilidades del entorno Hadoop y se muestran unos pocos casos de uso aplicado.

Capítulo 5. Bloque Análisis: En este capítulo se indican las consideraciones que deberá tenerse con los resultados del análisis de sentimiento, teniendo en mente la subjetividad del método, obtener la mayor cantidad de resultados posibles será una de las mejores formas de actuar. También se presenta la herramienta AlchemyAPI, que como en el caso anterior será el corazón del bloque.

Capítulo 6. Bloque Representación: Se presenta la necesidad de representar los datos anteriores de una forma más cómoda y práctica, que no requiera un estudio previo del sistema completo, pero que mantenga el mismo nivel de información, y sea tan adaptable como se desee. A tal fin, se presenta la librería JFreeChart, la cual, acompañada con el código correspondiente, permite hacer realidad los aspectos anteriormente destacados y completar todo el sistema.

Capítulo 7. Resultados: Se presentan las trazas de los capítulos anteriores, mostrando así las formas en las que se obtienen los resultados que a continuación se describen, complementando los datos ya mostrados en los respectivos capítulos.

Capítulo 8. Conclusiones y líneas futuras: Se vuelve a resaltar la importancia y potencial de las tecnologías actuales, para posteriormente analizar los objetivos marcados en el capítulo inicial y contrastarlos con los resultados obtenidos. Luego se procede con las conclusiones que abarcan la totalidad del Proyecto, y se terminan ilustrando las líneas futuras de desarrollo.

Se incluyen, como complemento a este proyecto, los siguientes documentos y anexos:

Referencias bibliográficas: Indica la bibliografía consultada durante la realización de este proyecto.

Pliego de condiciones: Expone con qué recursos y bajo qué condiciones se ha desarrollado este Proyecto Fin de Carrera. Se indican las especificaciones materiales y de equipos, y las condiciones para su ejecución. Así mismo se dan las instrucciones necesarias para que el material y la información contenidos en el DVD adjuntado a este proyecto puedan ser correctamente utilizados. Finalmente, se enumeran las condiciones legales que rodean este proyecto.

Programas. Se enumeran y se resumen los programas que han sido desarrollados en este proyecto, de una forma más breve que en los capítulos anteriores.

Presupuesto: Se enumeran y detallan los conceptos que componen el presupuesto de este proyecto.

Anexo 1. Contenido del DVD−R: Un DVD con la información generada en este trabajo, según el modelo normalizado de la EITE.

Capítulo 2: Estado del Arte

2.1. Introducción

Las herramientas y aplicaciones que utilizan tanto los conceptos de Big Data como los de Minería de Datos y Análisis de Sentimiento son cada vez más comunes en el panorama tecnológico actual. Las ventajas que ofrecen de cara a empresas y desarrolladores son sin duda críticas de cara a tomar una posición de liderazgo para con la competencia.

Durante la realización de este Proyecto Fin de Carrera, se han utilizado un buen número de estas tecnologías de cara a obtener el sistema final. A continuación, se presenta una serie de tecnologías comunes a todo el sistema, para después separar el resto en función de los tres bloques funcionales de Eccetas.

2.2. Tecnologías generales

2.2.1. Java

Java[7] es un lenguaje de programación orientado a objetos, de propósito general y concurrente, utilizado fundamentalmente en este Proyecto para vincular todo el conjunto de herramientas entre sí y dar al sistema una funcionalidad completa.

La premisa de Java es permitir a los desarrolladores diseñar un código que sea ejecutable en cualquier plataforma sin necesidad de ser recompilado. Así mismo, se utiliza notablemente para aplicaciones cliente–servidor de web (parte del uso que se le da en este Proyecto es para esa misma tarea).

La enorme popularidad de este lenguaje, la relativa facilidad de uso, y el hecho de que forma parte del plan de estudios de la carrera de Ingeniería de Telecomunicación, colocan a Java por encima de otros lenguajes de programación que podrían haber tomado su lugar en este Proyecto, como C o Python.

En este Proyecto, Java se ha utilizado para las siguientes tareas:

- Acceder a las distintas herramientas del sistema y obtener resultados de estas.
- Tratar con el usuario.

- Trasladar los resultados parciales a los siguientes bloques del sistema.

2.2.2. Big Data

El concepto de Big Data nace a partir de la realización de que la cantidad de datos de la que se dispone para una operación o desarrollo pueda tan grande que no sea posible utilizarlos, bien sea por la imposibilidad de almacenar los datos, procesarlos en un tiempo efectivo o que no aporte la precisión esperable. Con este "problema" en mente, surgen herramientas que trabajan de formas alternativas a las consideradas como tradicionales.

Ejemplos de este cambio en las técnicas son, por ejemplo, las bases de datos no relacionales[8], utilizar procesado en paralelo en clústeres de servidores, etc. Habitualmente se da el caso en el que se sacrifica detalle en las operaciones por practicidad y velocidad, llegándose al caso en el que gracias a la inmensa cantidad de datos procesados, resulta mucho más efectivo que el procedimiento antiguo. En este Proyecto, se han utilizado múltiples conceptos de Big Data, como el Análisis de Sentimiento, pero también herramientas especializadas como Hadoop[9], para almacenar los resultados parciales de la extracción de datos y trabajar sobre estos.

Big Data es más una filosofía de trabajo que una especificación concreta de programas o tareas. Es aceptar que con una cantidad de información y elementos tan masiva como la que se dispone hoy en día, no es posible procesarla en las formas habituales, por lo que han de tomarse nuevas interpretaciones y criterios de diseño. Así mismo es una corriente que no deja de reinventarse a sí misma; año tras año surgen nuevos avances, herramientas, y actualizaciones de estas, así como congresos y cursos. El uso de esta filosofía se está volviendo, rápidamente, un pilar crítico en numerosas empresas, que han descubierto en Big Data un apoyo increíble a sus intereses.

Una de las grandes capacidades de Big Data es su versatilidad. Al estar conformado por una gran variedad de programas y herramientas distintas, constantemente en desarrollo y actualización, ofrecen un entorno modular y flexible de trabajo. Por ello, es posible usar Big Data en prácticamente cualquier área que genere datos, desde la recepción del clima a la medicina, pasando por el deporte, la economía, empresas…

Utilizar Big Data efectivamente requiere un claro entendimiento de lo que desee realizarse con estos datos; fundamentalmente se describen 3 pasos necesarios para obtener conocimiento al utilizar Big Data, que son los siguientes:

- **Idea**: Disponer de datos no sirve de nada si no se tiene claro qué se desea hacer con ellos. La idea es el concepto más abstracto y difícil de obtener de los tres pasos. Por ejemplo, se puede disponer de una enorme red de sensores que midan la temperatura de un cierto mar. Un restaurante puede tener la idea de analizar dichas mediciones para saber cuándo es el momento ideal para comprar el marisco, que es cuando la temperatura es máxima, mientras que un barco investigador podría decidir no salir en busca de ciertas algas si detecta que el mal no propiciará su crecimiento a causa del cambio en su temperatura. En este Proyecto, la idea es generar conocimiento en base a los comentarios que envían los clientes de páginas web de viajes sobre los establecimientos que visitan.
- **Datos**: Evidentemente, es necesario disponer de datos, si bien no toda clase de dato vale de la misma forma. Son apreciados los datos estructurados (logs de GPS, bases de datos, etc.), los cuales permiten ser procesados a una velocidad mucho mayor y por un número superior de herramientas, aunque es posible utilizar otras para tratar datos no estructurados, que son la mayoría. En esencia, disponer de un adecuado conjunto de datos es vital para el correcto desarrollo del proceso. En este Proyecto, los datos son los comentarios mencionados anteriormente, los cuales representan un esquema no estructurado de datos (de ahí que sea necesario recurrir a Minería de Datos, como se explicará más adelante).
- **Herramientas**: Este es el concepto más abierto a cambios, y aquel en el que los desarrolladores han de emplear la mayoría de sus esfuerzos. Si bien hay una enorme cantidad de herramientas capaces de procesar datos de múltiples maneras, es necesario utilizar el conjunto apropiado de estas, realizando tanto el proceso de selección como el necesario trabajo para combinarlas en un programa sólido que permita hacer realidad la idea objetivo. En este Proyecto, el sistema resultante representa las herramientas.

2.2.3. JSON

JSON, el acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. Debido a ser una opción muy utilizada como alternativa a XML, se le considera un formato de lenguaje independiente, y se ha usado fundamentalmente en este Proyecto para almacenar la gran mayoría de los datos.

El hecho de que JSON sea un formato tan extendido y empleado en numerosas herramientas (como las que se utilizan en este Proyecto, un motivo adicional para su adopción), hace que su uso resulte muy sencillo. Del mismo modo, su estructura de llaves y palabras clave permite una fácil adopción y una vinculación sencilla a Java mediante librerías especializadas. Finalmente, su capacidad para trabajar con documentos extremadamente grandes lo hace un candidato óptimo para almacenar las ingentes cantidades de datos que se recabarán a lo largo de este Proyecto.

2.2.4. CSV

CSV, del inglés comma–separated values, indica a un tipo de documento en formato abierto sencillo, utilizado para representar datos en forma de tabla, donde las columnas se separan por comas (o por puntos y comas, donde la coma sea un separador decimal), y las filas por saltos de línea. Se ha utilizado en conjunto con JSON para almacenar datos en este Proyecto, fundamentalmente aquellos extraídos de las herramientas de Big Data.

Este formato, extremadamente sencillo aunque no estandarizado, es propenso a generar errores si no se tienen en cuenta caracteres particulares como dobles comillas o saltos de línea, siendo relativamente sencillo provocar un error si no se tienen en cuenta estos aspectos. Este formato engloba, además, a otros formatos de valores separados por delimitadores, como espacios o saltos de línea, lo cual también puede dificultar su uso. Sin embargo, su uso en Hadoop, uno de los principales estandartes de Big Data, hace que sea necesario recurrir a él si se desean extraer datos mediante las especializadas y potentes herramientas del sector. Así mismo, su popularidad en otras áreas hace posible que existan librerías en formato abierto de Java que permiten su fácil manipulación en el sistema.

2.2.5. Servicio web

Se define como servicio web a la tecnología que utiliza un conjunto de protocolos y estándares que permiten intercambiar datos entre aplicaciones, ampliamente utilizado en Internet al utilizar este estándares abiertos. Se utiliza en este Proyecto para transmitir las peticiones de extracción de datos y subsecuente análisis, a las herramientas apropiadas.

La necesidad de interactuar entre clientes y servidores mediante aplicaciones web, resulta obvia teniendo en mente que métodos similares son los que permiten a entidades bancarias realizar transacciones por todo el mundo, así como a usuarios únicos comprar entradas de cine o realizar una compra de alimentos a través de la Red.

Dado que los servicios web funcionan por encima de las plataformas sobre las que se instalen, fomentan los estándares y protocolos basados en texto y permiten que los servicios y el software ubicados en distintos lugares geográficos actúen como único, sus ventajas son muy importantes por encima de sus inconvenientes (Un menor rendimiento a otros modelos de computación distribuida al depender de un formato basado en texto, hay otros estándares abiertos de computación distribuida mucho más desarrollados, etc).

Este concepto, por tanto, será muy útil en la aplicación de este Proyecto Fin de Carrera, ya que es necesario acceder a varios servicios con el fin de obtener el resultado final. Incluso, como se indicará en capítulos posteriores, convertir todo este sistema en un servicio web sería una opción más que razonable.

2.3. **Extracción**

2.3.1. Minería de Datos

La minería de datos[10] es un campo interdisciplinar de las ciencias de la estadística y de la computación que define al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utilizando métodos interconectados de inteligencia artificial, aprendizaje de máquinas, estadística y

bases de datos, persigue extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior.

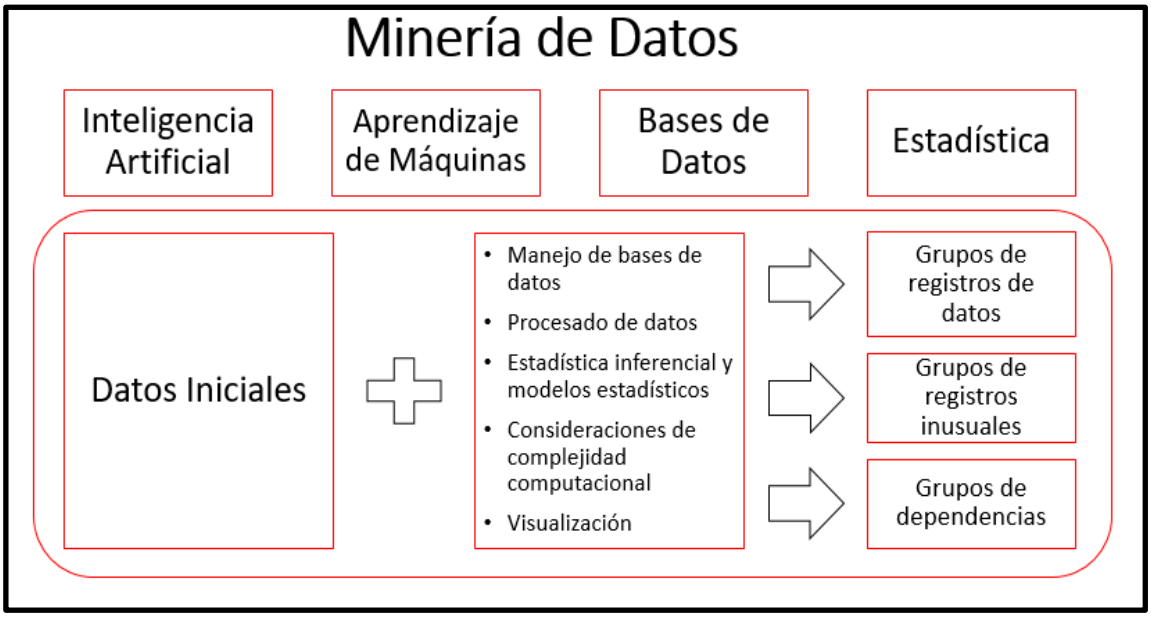


Ilustración 3 – Representación de la Minería de Datos

Además de las ramas mencionadas anteriormente, la minería tiene otros aspectos secundarios a considerar:

- **Bases de datos y manejo de bases de datos** – Trabajar con enormes cantidades de datos supondrá, necesariamente, disponerlos de alguna forma. Para ello, será necesario crear una base de datos y manipularla apropiadamente.
- **Procesado de datos.**
- **Consideraciones de modelos estadísticos y estadística inferencial** – Será necesario realizar aproximaciones estadísticas para poder analizar los datos.
- **Métricas de interés** – Resultará interesante obtener según qué resultados del análisis realizado.
- **Consideraciones de complejidad computacional** – De cara a trabajar con cantidades de datos extremadamente densos o complejos, la propia capacidad computacional será un aspecto a considerar, así como las mejores formas de resolver las operaciones.

- **Postprocesado de estructuras descubiertas.**
- **Visualización**.
- **Actualización online.**

En términos generales, hay tres clases de resultados que se pueden obtener del análisis automático o semiautomático de grandes cantidades de datos:

- **Grupos de registros de datos**, que desembocan en *análisis de grupos* (Agrupar un conjunto de objetos de forma que los miembros del mismo grupo sean más similares)
- **Grupos de registros inusuales,** que desembocan en *detector de anomalías* (Identificar registros que no se adaptan al patrón establecido o esperado)
- **Grupo de dependencias**, que desembocan en *reglas de asociación* (Encontrar hechos que ocurren en común dentro de un determinado conjunto de datos)

Se pueden ilustrar estos conceptos con un ejemplo, si se toma una base de datos de GPS de camiones a lo largo de un país, que disponga de los datos básicos como el identificador del vehículo, la carga, el conductor, el consumo de combustible, y las fechas y horas de viaje:

- El *análisis de grupos* permitiría agrupar a los conductores más eficientes, o detectar aquellos camiones que podrían necesitar una revisión en función del consumo de gasolina relacionado con su distancia.
- El *detector de anomalías* permitiría localizar a aquellos conductores que destacan de forma negativa o positiva en su trabajo.
- Las *reglas de asociación* podrían indicar qué rutas suelen ser más efectivas en función de las horas a las que se realizan, o bien en función de las fechas.

Por tanto, la minería de datos resulta una piedra angular en este proyecto, y una increíble rama multidisciplinar, que deberá requerir de una herramienta capaz de realizar la mayoría, sino todos los pasos mencionados anteriormente. Ante lo cual, además del código *Java* que será necesario, se utilizará la herramienta que se presenta a continuación.

2.3.2. Import.IO

Import.IO[11] es una potente herramienta que permite extraer información ordenada de páginas web, mediante la técnica conocida como web scraping. Esta se basa en analizar la estructura HTML de la página web en cuestión, aprendiendo así su estructura, para después poder tomar de esta los datos seleccionados. En este Proyecto Fin de Carrera, es la principal herramienta de extracción de datos.

Una de sus principales ventajas es su funcionalidad; se basa en una interfaz que analiza automáticamente las estructuras de las páginas web y ofrece en una forma de tabla todos los datos posibles, de forma que no es necesario generar código alguno para su uso apropiado. Así mismo, es posible llamar a esta herramienta desde *Java*, ya que dispone de una API basada en REST, lo que permite una fácil implementación.

Su potencial y una definición más completa se expondrá en el capítulo Extracción del presente Proyecto.

2.4. **Análisis de Sentimiento**

2.4.1. Procesamiento de Lenguajes Naturales (PLN)

El procesamiento de lenguajes naturales es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales (esto es, lenguajes cuyo único propósito es la comunicación, a diferencia de lenguajes de programación, lógica matemática, etc). El PLN trata de diseñar mecanismos para comunicarse que sean eficaces computacionalmente.

Hasta la década de 1980, la mayoría de los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, gracias a la mejora de la capacidad computacional de los ordenadores, hubo una revolución en PLN con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.

Es necesario considerar la dificultad que entraña el procesamiento de los lenguajes naturales:

- A nivel **léxico**, una misma palabra puede tener varios significados, y la selección del apropiado se debe deducir a partir del contexto oracional o conocimiento básico. Muchas investigaciones en el campo del procesamiento de lenguajes naturales han estudiado métodos de resolver las ambigüedades léxicas mediante diccionarios, gramáticas, bases de conocimiento y correlaciones estadísticas.
- A nivel **referencial**, la resolución de anáforas y catáforas implica determinar la entidad lingüística previa o posterior a que hacen referencia.
- A nivel **estructural**, se requiere de la semántica para entender la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos. Por ejemplo, en la frase “rompió el dibujo de un ataque de nervios”.
- A nivel **pragmático**, una oración, a menudo, no significa lo que realmente se está diciendo. Elementos tales como la ironía tienen un papel importante en la interpretación del mensaje.

Para resolver estos tipos de ambigüedades y otros, el problema central en el PLN es la traducción de entradas en lenguaje natural a una representación interna sin ambigüedad, como árboles de análisis.

2.4.2. Análisis de Sentimiento

El Análisis de Sentimiento, también conocido como Minería de Opiniones, combina el uso del procesamiento de lenguajes naturales anteriormente indicado, el análisis de texto y la lingüística computacional para identificar y extraer información subjetiva de una fuente en concreto. En términos generales, permite determinar la actitud de una fuente respecto a un elemento concreto, o su polaridad completa. En este Proyecto, se utilizará este concepto con los datos recogidos por el bloque anterior para obtener valoraciones sobre las características de los establecimientos, en base a una metodología que se explicará más adelante.

Debido a la complejidad y a la magnitud de opciones que el análisis de sentimientos puede generar, se define una serie de tipos en función de los resultados que obtendrán:

- **Polaridad** – Definir la polaridad de un cierto texto de un documento, frase, o elemento, bien sea positiva, neutral o negativa. En esta rama se puede dar además un análisis más profundo, donde los resultados son sentimientos como "feliz" o "triste".
- **Gradual** – Utilizar un sistema de valoraciones, donde las palabras comúnmente asociadas a sentimientos neutrales, positivos o negativos se les fija un valor de una escala de –10 a +10 (mayor negativo a mayor positivo). De esta forma, se puede ajustar el sentimiento de un cierto elemento considerando su entorno, habitualmente la sentencia en la que se encuentre. De esta forma, es posible acumular los resultados y obtener una valoración general de todo un texto y su sentimiento, en lugar de simplemente obtener la polaridad de este.
- **Identificación subjetivo/objetivo** – Clasificar un texto, habitualmente una sola frase, en objetiva o subjetiva. Esta clasificación suele ser más compleja que la clasificación de polaridad, ya que la subjetividad de las palabras y las frases pueden depender de su contexto, y un documento objetivo puede contener frases subjetivas como citas.
- **Característica–aspecto** – Determinar la opinión o sentimiento basado en las características o aspectos de entidades, como pueden ser teléfonos móviles, coches o zapatos. Una característica o aspecto es un atributo o componente de una entidad, como puede ser la pantalla de un teléfono o las ruedas de un coche. La ventaja de este tipo de análisis es que permite extraer características sobre objetos de interés, las cuales varían naturalmente entre sí (un teléfono puede tener una batería mala, pero una buena pantalla). Este tipo requiere tratar varios subproblemas, como detectar las entidades relevantes, sus características, y analizar si la opinión es positiva, negativa o neutral. Este tipo es el que se usará en el presente Proyecto Fin de Carrera, ya que aporta exactamente los resultados que se desea obtener.

En cuanto a los métodos, existen fundamentalmente tres[12] categorías en las que pueden agruparse:

- **Técnicas 'Knowledge–based'** – Clasifican texto en categorías, en función de que contenga palabras inequívocas como "feliz", "aburrido", "tristeza", "miedo", etc… Las cuales están almacenadas en listas. Algunas de estas técnicas amplían sus listas, añadiendo además de las palabras obvias otras con una "afinidad" a emociones en particular.
- **Métodos estadísticos** – Se basan en elementos de aprendizaje de máquinas como el análisis de semántica latente, máquinas de vectores de soporte, y "bolsa de palabras", entre otros.
- **Aproximaciones híbridas** – Se apoyan tanto en aprendizaje de máquinas como elementos de representación del conocimiento (un área de la IA que facilita la inferencia), como pueden ser ontologías o redes semánticas para detectar semántica sutil, que no podría resolverse directamente.

Queda claro por tanto que el análisis de sentimiento está íntimamente ligado a la capacidad de las máquinas para interpretar e identificar la subjetividad del ser humano, y para poder ejecutar dicha capacidad, es requisito indispensable acudir a software y hardware especializado para ello. Aprendizaje de máquinas, análisis estadístico, procesamiento de texto natural… Hay que considerar, además, que es posible ejecutar análisis de sentimiento en contenido visual, lo cual agrega toda una capa de complejidad al proceso.

Finalmente, hay que considerar la efectividad de estos sistemas. En general, una máquina efectuará un apropiado análisis cuanto más se parezca al juicio humano. En términos generales, este juicio humano (cómo de acuerdo están las personas entre sí) suele ser de un 79%[13], por lo que un programa que alcance un 70% de efectividad puede ser considerado como exitoso, ya que el factor subjetivo y humano a la hora de tomar decisiones hace que un 100% sea absolutamente imposible, incluso para un analista humano. Por tanto, la evaluación del sistema siempre será un asunto complejo, regularmente quedando en manos de correlaciones y ajustes estadísticos. En este Proyecto, se hará esa misma consideración, precisando obtener un cierto número de valoraciones antes de ofrecer resultado alguno.

2.4.3. AlchemyAPI

AlchemyAPI forma parte del servicio AlchemyLanguage[14], una colección de analizadores de texto que obtienen información semántica de las fuentes analizadas. AlchemyAPI, en particular, trabaja con el procesamiento de lenguajes naturales, de forma que con recibir un texto, una estructura html, o una url pública, será posible ofrecer un análisis del contenido de alta calidad, incluyendo incluso la detección de entidades o palabras clave.

Este último punto, el de las palabras clave, será especialmente relevante en este Proyecto Fin de Carrera. AlchemyAPI es capaz de extraer las palabras relevantes de un texto, y al mismo tiempo, realizar el análisis de sentimiento, otorgando a cada palabra clave una polaridad (neutral–positivo–negativo) junto a su porcentaje relativo. Esto será absolutamente indispensable para el correcto procesado del sistema, aunque los resultados no son perfectos y será necesario realizar ajustes y filtrados apropiados.

Entre otras características, AlchemyAPI dispone de varios SDK gratuitos disponibles, que permiten vincular fácilmente la herramienta con cualquier código que se genere. Un análisis más detallado de su funcionalidad se expondrá en su correspondiente capítulo.

2.5. **Representación**

Tras los bloques anteriormente mencionados, se llega a la siguiente situación:

- El proceso genera muchos datos parciales, complejos de utilizar.
- El resultado final genera unos datos que, si bien correctos, son enormemente extensos y complejos de analizar.
- Es necesario disponer de conocimientos técnicos para el correcto aprovechamiento de los datos.

Dado que se desea que el resultado de este sistema sea sencillo, fácilmente comprensible y que no requiere conocimiento previo alguno, será necesario

recurrir a apropiados sistemas de representación que conviertan los resultados en gráficas personalizables y entendibles.

2.5.1. JFreeChart

Ante la necesidad de representar enormes cantidades de datos complejos, aplicar las anteriores tecnologías y conceptos mencionados, y obtener resultados legibles por cualquier usuario, transformar dichos resultados complejos en sencillas gráficas es una tarea que la librería JFreeChart[15] para *Java* permite.

JFreeChart es una librería 100% gratuita, ampliamente usada y con un diseño flexible y muy personalizable, que permite sintetizar las gráficas o directamente almacenarlas en formatos PNG o JPG. En este Proyecto Fin de Carrera, permite la representación del resultado completo del sistema, y en su correspondiente capítulo se extenderá su descripción.

Capítulo 3:

Extracción

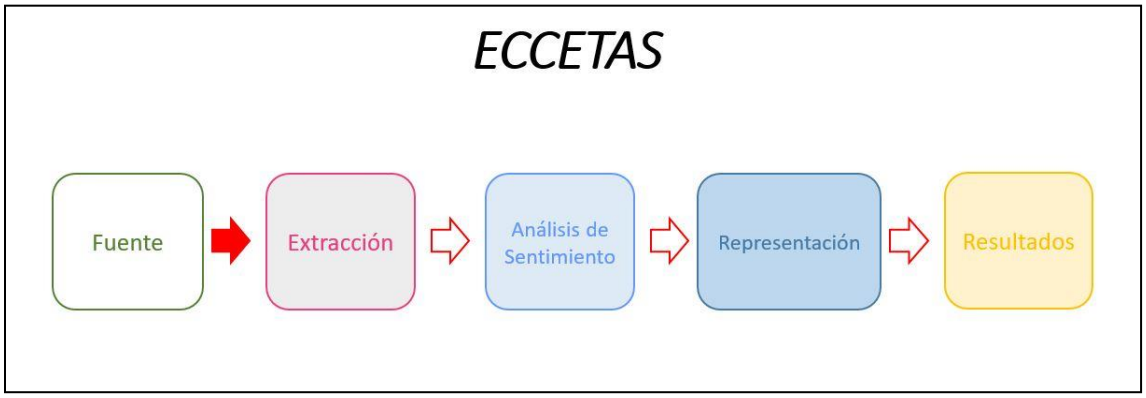


Ilustración 4 – Eccetas: Bloque Extracción

3.1. Introducción

La obtención de datos es, evidentemente, una parte crítica de cualquier sistema que requiera de estos, bien se utilicen para entrenamiento del propio sistema, o de cara a obtener resultados con ellos. No obstante, hay que considerar que disponer de datos no es todo lo que se desea; estos deben poseer alguna estructura apropiada, así como ofrecer información útil. No tiene sentido recabar grandes cantidades de datos convenientemente ordenados en tablas, si la información que aportan dichos datos no tiene utilidad particular.

El objetivo principal del **Bloque Extracción** será, por tanto, recabar datos, y a partir de estos datos en crudo, generar unos bloques estructurados de datos válidos que dispongan de interés al resto del sistema.

Por tanto, hay dos aspectos claros que deben cumplir los datos que se introduzcan al **Bloque Extracción**:

- **Estructura** – Los datos no estructurados son los más habituales a la hora de obtener datos del mundo real, por lo que será necesario manipularlos para darles una estructura que sea manipulable por el sistema.
- **Sentido** – Disponer de datos sin más no aporta ninguna utilidad al sistema. Afinar los criterios de selección, de forma que se dispongan de los datos apropiados, es un aspecto casi tan crítico como el de disponer de una estructura apropiada.

Considerando lo expuesto, se utilizarán los siguientes criterios:

3.1.1. Estructura

- De cara a utilizarlos en los siguientes bloques, se precisará que los datos se extraigan en un formato adecuado, como JSON, que permite almacenarse y manipularse desde casi cualquier otra herramienta.
- Se hará énfasis en disponer de los *datos críticos* apropiadamente estructurados, pero también será necesario disponer de los *datos secundarios* que permitan ordenar los datos (Por ejemplo, si el *dato crítico* es el contenido de una carta, el *dato secundario* sería el autor de esta, o su dirección).

3.1.2. Sentido

- Un aspecto a considerar a la hora de tomar los datos es el idioma. Las herramientas de análisis de texto (**Bloque Análisis**) no funcionan adecuadamente sin saber previamente el idioma que se va a analizar, por lo que será más sencillo para el sistema si el **Bloque Extracción** organiza sus datos en función del idioma.
- Teniendo en cuenta su aplicación en el **Bloque Representación**, será necesario que, a la hora de hacer comparativas entre datos, estos pertenezcan al mismo ámbito o dispongan de algunos criterios de valoración aceptables.

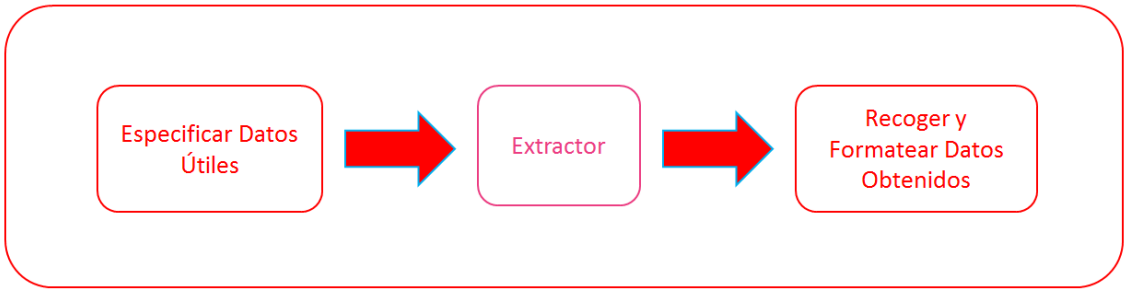


Ilustración 5 – Resumen General del Proceso de Extracción

3.1.3. Validez de los Datos

Hay que considerar que, en numerosas ocasiones, se cuenta con datos que no son completamente objetivos, sino que aportan subjetividad al sistema que trate con ellos. Suelen ser casos donde los datos son extremadamente escasos, o generados en base a la opinión de una persona. Siendo estos casos situaciones

claras donde el análisis realizado por un sistema automático puede no ser útil al carecer de la capacidad de gestionar la subjetividad.

Se puede considerar, no obstante, que un gran número de datos subjetivos pueden desembocar en un dato objetivo, con el adecuado análisis y siempre que estos datos converjan en el mismo resultado, y es lo que se considerará en este Proyecto Fin de Carrera.

3.1.4. Caso de Estudio

El caso de estudio que se presenta en el presente Proyecto Fin de Carrera y que permite ejemplificar lo expuesto anteriormente presenta las siguientes características:

- Se extrae información de páginas web, concretamente los *comentarios* sobre establecimientos turísticos.
- La información puede estar en múltiples idiomas.
- La cantidad de información puede resultar masiva, e imposible de extraer de forma manual.

A tal fin, se utilizará una herramienta que permita realizar la extracción de datos, sin importar su tamaño, de la cual se hablará a continuación. Por otro lado, será necesario generar código que permita que dicha herramienta reciba los datos, y los almacene en una estructura útil para los siguientes Bloques.

Finalmente, como resultado complementario, también se almacenarán las *valoraciones generales* que acompañan a los comentarios, con el fin de demostrar el potencial de la información obtenida en más de una forma, no solo con el análisis de los comentarios.

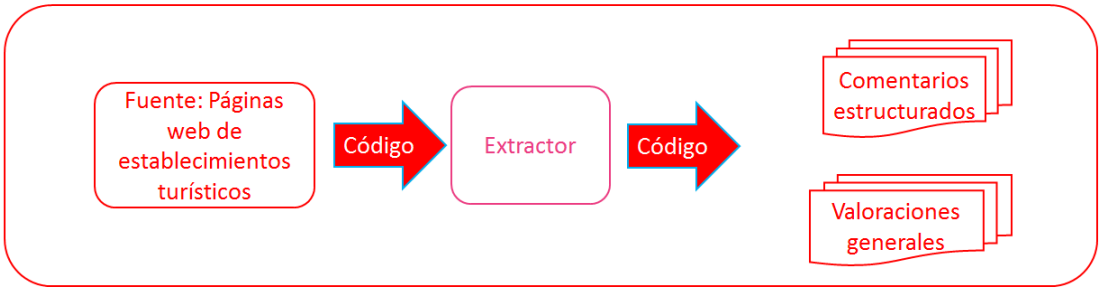


Ilustración 6 – Esquema del Caso de Estudio del Bloque Extracción

3.2. **Import.IO**

Esta es la principal herramienta utilizada por el **Bloque Extracción** de Eccetas. Permite analizar páginas web y obtener datos de estas en función de parámetros indicados por el usuario.

Import.IO permite realizar web scraping, esto es, es capaz de recopilar información de forma automática de Internet, analizando la estructura de las páginas indicadas y extrayendo de ellas los datos solicitados. Esta es una práctica cada vez más utilizada por la gran cantidad de información útil que puede obtenerse en un corto espacio de tiempo, aunque requiere de un correcto modo de funcionamiento (habitualmente, esperando un cierto tiempo antes de volver a solicitar un nuevo resultado, logrando así que los servidores web no consideren al sistema responsable del web scraping como un ataque de saturación de servicio).

Import.IO se ejecuta mediante un servicio Web, disponiendo a su vez de una API REST para sus llamadas mediante solicitudes, siendo esta la forma en la que se utilizará en este Bloque.

Si bien en versiones anteriores contaba con hasta 4 tipos distintos de *scripts*, actualmente solo posee uno, el *Extractor*, cuya interfaz puede verse en la Ilustración 7.

Este *script* es capaz de extraer los datos que hayan sido indicados por el usuario en uno o múltiples direcciones URL. Dicha extracción será posible por la interfaz de entrenamiento que ofrece Import.IO a la hora de construir el *Extractor*.

La limitación de este *script* es la necesidad de disponer de las URL previamente a su uso, ya que si bien posee un generador de URLS bastante potente, no es posible acceder a él mediante llamada a su API. Por tanto, será necesario que en este sistema, mediante código, se indiquen las URL de las páginas a visitar.

Ilustración 7 – Ejemplo de Extractor

A la hora de crear un *Extractor*, se realizarán los pasos siguientes:

1. Tomar una página web como ejemplo. El *script* solo funcionará con páginas web que cumplan la estructura de la página seleccionada.
2. La herramienta cargará la página web, generando una interfaz sobre esta y tomará unos datos iniciales que la propia herramienta considera útiles para definir el modelo de extracción de datos. Así mismo, permitirá al usuario seleccionar qué datos quiere extraer, y de qué tipo (texto, número, enlaces html, imágenes, etc.…).
3. La herramienta es inteligente, por lo que seleccionará, por ejemplo, todos los elementos de una lista si se ha tomado uno ya, entendiendo que el usuario desea tomar todos los datos similares.
4. Finalizado el proceso, se guardará el *script*, quedando disponible para su uso con cualquier página web que cumpla la estructura de la página de ejemplo.

Llegados a este punto, puede verse claramente que realizar el entrenamiento inicial, introducir las direcciones de origen, y ejecutar el sistema, puede resultar tedioso o complejo, especialmente si es necesario ejecutar múltiples *scripts* a la vez. Por ello, además de los *scripts* de Import.IO a ejecutar, será necesario generar código que llame a los *script* de Import.IO que sean necesarios.

Habiendo obtenido los datos deseados, habitualmente presentados en un fichero JSON, se verá que, en etapas posteriores de este Bloque, no son procesables directamente, por lo que será necesario generar código que permita tomar estos datos y formatearlos de forma que el sistema pueda utilizarlos cómodamente.

3.3. Estructura y Descripción del Bloque

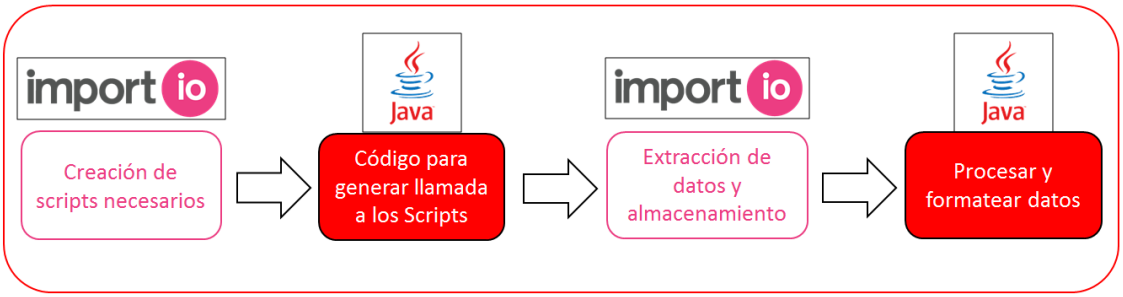


Ilustración 8 – Estructura del Bloque Extracción

Siguiendo las explicaciones anteriores, el **Bloque Extracción** estará formado por los siguientes subbloques:

- Generación de los *scripts* de Import.IO que sean necesarios.
- Código Java para generar la llamada a los *scripts* creados anteriormente.
- Uso de los *scripts* de Import.IO y almacenamiento de datos en crudo.
- Código Java para procesar los datos y formatearlos convenientemente.

La limitación principal de este **Bloque** será, por tanto, el origen de los datos de entrada, que obligarán a crear *scripts* determinados de Import.IO por cada modelo existente, los cuales a su vez dependerán de un código de ejecución y procesado adaptado. Sin embargo, el concepto general se mantiene y el sistema será igual en cualquier modelo utilizado.

Así mismo, se hará la siguiente consideración a la hora de recabar los datos, de forma que habrá dos tipos de extracción posibles:

- **Caso Específico** – El usuario desea la información de un solo establecimiento.
- **Caso General** – El usuario desea la información de todos los establecimientos disponibles de un área determinada, como un país, y en un idioma en concreto. Por defecto, se tomará España como país.

3.3.1. Generación de *scripts* de Import.IO

Para obtener los resultados deseados, será necesario realizar un *script* que recoja los datos apropiados de las páginas web que serán la fuente de información. Debido a que los *script* dependerán en gran medida de la fuente de información, estos serán únicos y utilizables solo en esas fuentes de información en concreto.

En este Proyecto Fin de Carrera, con la fuente de datos escogida, será necesario ver que un *Extractor* no será capaz de extraer la información **de contacto** del establecimiento (país, dirección, nombre) a la vez que la información **crítica** (comentario, fecha, valoración general, etc.), por lo que será necesario utilizar al menos dos, como se ve en la Ilustración 9. Con el **Caso General** mencionado anteriormente, además, será necesario generar otro *script* para indicar al sistema el número de establecimientos que se obtendrán. Esto se explicará con mayor detalle en el siguiente subbloque.

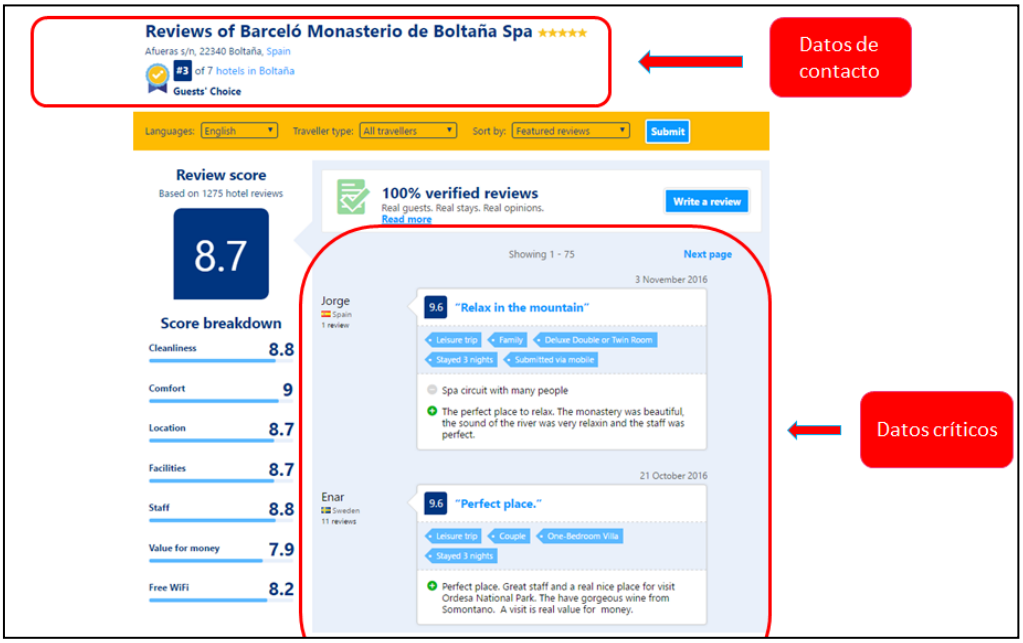


Ilustración 9 – Ejemplo de datos de contacto y datos críticos

A continuación, se presentan los Extractores realizados para este Proyecto:

3.3.1.1 Extractor: Main.

El *script* central del **Bloque Extracción**, utilizará los enlaces proporcionados para extraer todos los datos **críticos** indicados en la interfaz de Import.IO, como:

- Nombre del autor
- Fecha
- Número de comentarios
- Valoración general
- Comentario (Negativo y positivo, en este caso)
- Detalles
- Etc.

Para poder ser ejecutado, este *Extractor* necesitará los siguientes datos:

- Enlace del establecimiento a visitar.

- Número de páginas a visitar (**Profundidad**) – Dado que el sistema funciona con solo una página por vez, será necesario indicar al *Extractor* que hay más páginas a visitar. En el caso que se ocupa, por ejemplo, cada página ofrece 75 comentarios. Si el establecimiento tiene 750 comentarios, será necesario indicar una Profundidad de 10.
- Nombre del establecimiento.

Con los datos correctos, el sistema recibirá los resultados de los servidores de Import.IO, y los almacenará directamente en un archivo (denominado como **RAW**, al ser los datos directos), asegurando así que, en caso de existir algún error, se podrá rescatar la mayor cantidad de información posible. Sin embargo, estos datos almacenados poseen una estructura, como puede verse en la Ilustración 10, que no resulta apropiada para su uso directo, por lo que será modificada en el subbloque final.

{"Fecha":[{"text":"8 November 2016"}],"Nombre":[{"text":"Johann74"}],"Pais":[{"text":"Germany"}],"Comentarios":[{"text":"22"}],"Valoracion":[{"text":"8.8"}],"Detalles":[{"text":"• Leisure trip"},{"text":"• Family"},{"text":"• 2 rooms"},{"text":"• Stayed 1 night"}],"Texto_neg":[{"text":"The parking is too far The double bed is too small"}],"Texto_pos":[{"text":"Very good position, right in the center, the rooms are ok (we had 3 rooms), nice staff, good WIFI, breakfast ok for 3 stars"}]},{"Fecha":[{"text":"7 November 2016"}],"Nombre":[{"text":"Theofanis"}],"Pais":[{"text":"Belgium"}],"Comentarios":[{"text":"2"}],"Valoracion":[{"text":"8.3"}],"Detalles":[{"text":"• Leisure trip"},{"text":"• Couple"},{"text":"• Double or Twin Room"},{"text":"• Stayed 2 nights"}],"Texto_neg":[{"text":"The room and the bed were small so I wouldn't choose it for more than two nights but the value for money comparing to other options in Madrid is very good."}],"Texto_pos":[{"text":"Good location and very friendly staff. Clean etc., with very good breakfast. Shoe shine sponge came in very handy."}]}

Ilustración 10 – Elemento de Fichero RAW

3.3.1.2 Extractor: ECA

Este *Extractor* tiene la tarea de obtener los datos **de contacto** del establecimiento, que el *Extractor* Main no puede obtener:

- Nombre del Establecimiento
- País del Establecimiento
- Dirección del Establecimiento

Si bien es más sencillo de ejecutar y solo requiere el enlace del establecimiento, no tiene utilidad sin vincularlo al resto de los datos extraídos por el *Extractor* anterior, lo cual hace más compleja su implantación en el código de procesado.

3.3.1.3 Extractor: Obtener Número Límite.

Utilizado en el **Caso General**, donde se requiere el número total de establecimientos que se extraerán a la vez. En el caso de estudio, este *Extractor* tiene como tarea obtener el número exacto de establecimientos turísticos existentes en España.

Para ser utilizado, requiere del enlace apropiado con el país en cuestión del que se desee obtener datos. En este Proyecto Fin de Carrera, el enlace es el de España, pudiendo elegirse entre español, inglés o alemán como idiomas de extracción de datos.

Ilustración 11 – Ejemplo del número máximo de establecimientos en Booking

3.3.1.4 Extractor: Obtener Enlaces Establecimientos.

Este *script*, utilizado solamente en el **Caso General**, utilizará el valor obtenido por el *Extractor* anterior para generar el número apropiado de peticiones a la página web para recopilar los enlaces web donde se encuentran los comentarios, indicados por las flechas en la Ilustración 12. En el Caso de Estudio, los establecimientos se encuentran agrupados en un cierto número de enlaces por página, siendo necesario extraer todos y cada uno de esos enlaces.

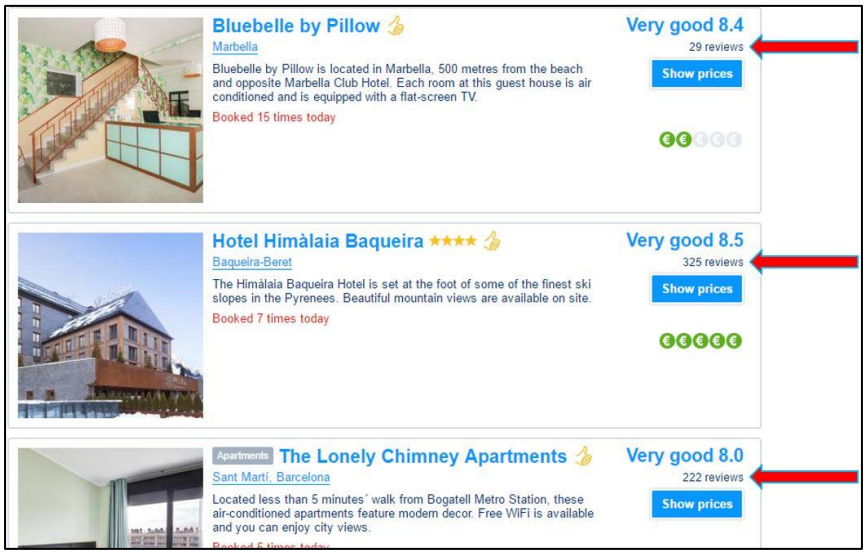


Ilustración 12 – Indicación de la posición de los enlaces web de comentarios

3.3.2. Generar llamadas a Import.IO

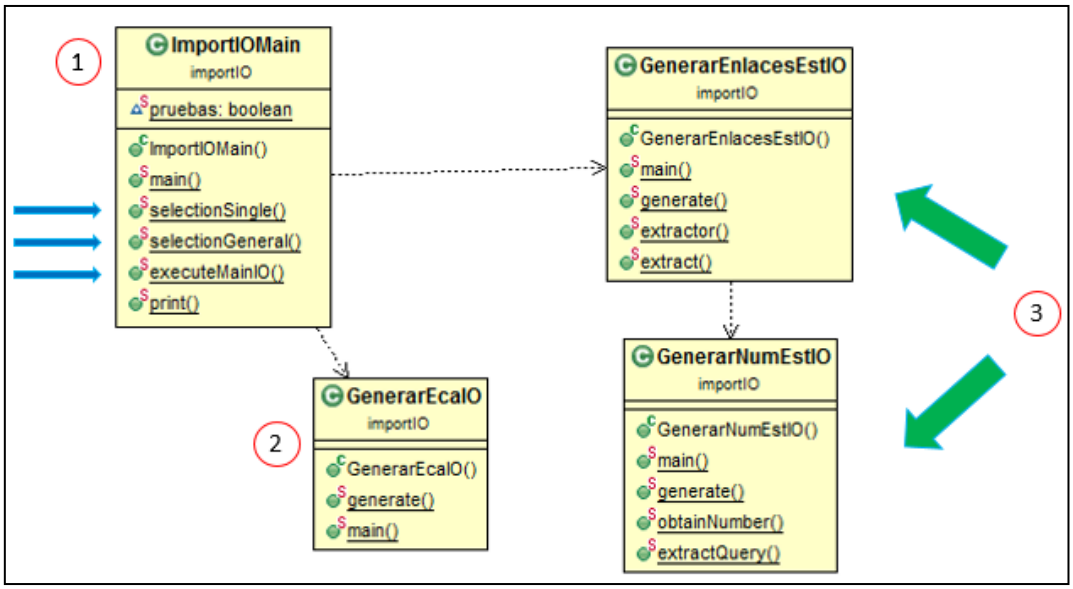


Ilustración 13 – Representación del bloque de generación de llamada a Import.IO

Habiendo ya considerado las fuentes de entrada, y disponiendo de los *scripts* mencionados anteriormente, se estará en disposición de generar el código representado por la Ilustración 13, cuyas funciones principales serán las siguientes:

1. Gestiona el proceso principal. Recoge los datos de entrada, en función del caso que vaya a realizarse (**General** o **Específico**), y ejecuta los *Extractores Main* y *ECA*. Requiere que el usuario indique la URL y el nombre del establecimiento si es en un caso individual, o las siglas del idioma si es un caso general. En cualquiera de los dos casos, también será necesario proveer de la Profundidad mencionada anteriormente.
2. Genera el archivo *ECA*, en función de la opción escogida previamente. Si es el **Caso General**, el archivo dispondrá de una serie de objetos JSON con los tres valores mencionados anteriormente. Si es **Específico**, solo será un objeto.
3. En caso que se haya tomado la opción **General**, será necesario llamar a los otros *Extractores* mencionados anteriormente, los cuales generarán sus ficheros resultantes, antes de ofrecer a los *Extractores Main* y *ECA* la lista de enlaces correspondientes.

Como se ha comentado previamente, la ejecución del *Extractor* dependerá del caso que se tome. A continuación, se exponen ambos casos y sus representaciones:

3.3.2.1 Caso General

Se solicita a Import.IO extraer todos los datos de los establecimientos del país indicado (por defecto, España), y en el idioma seleccionado (por defecto, español, alemán o inglés).

1. Se ejecuta el *Extractor Obtener Número Límite*, proveyendo al sistema del número de establecimientos a extraer.
2. Se ejecuta el *Extractor Obtener Enlaces Establecimientos*, que aprovechará el resultado anterior para generar la lista de URLs para el resto del sistema. En este punto, dado que es habitual que se produzcan errores, el sistema se ha

configurado para repetirse un cierto número de veces con los casos erróneos, con el fin de obtener el mayor número de enlaces posible.

3. Se ejecutan los *Extractores Main* y *ECA* tantas veces como se haya designado por la variable *Profundidad*, y se repetirá el bucle hasta agotar la lista de enlaces generada.

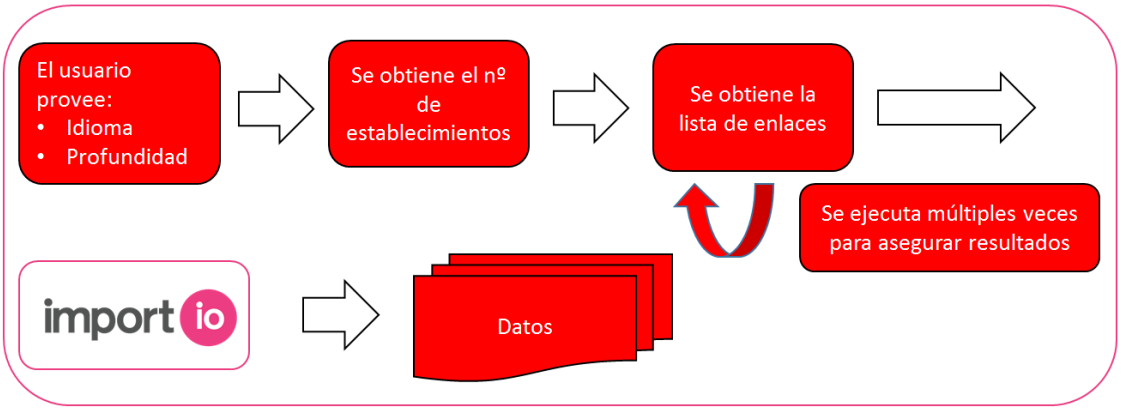


Ilustración 14 – Ejemplo de llamada de Caso General

3.3.2.2 Caso Específico

Ilustración 15 – Ejemplo de llamada de Caso Específico

En esta opción, el usuario solicita al sistema los datos de contacto del establecimiento del que se va a extraer información, por lo que simplemente se ejecuta el *Extractor ECA* una vez, y tantas veces el *Main* como se haya indicado por la variable *Profundidad*.

3.3.3. Procesar y Formatear los Datos.

Como pudo verse en la definición de los *Extractores* principales, si bien los resultados del *script* poseen una cierta estructura, no es una estructura útil, ni

práctica de utilizar. Por tanto, en este subbloque se procederá a formatear dichos datos, en la forma que se ilustra en la Ilustración 16:

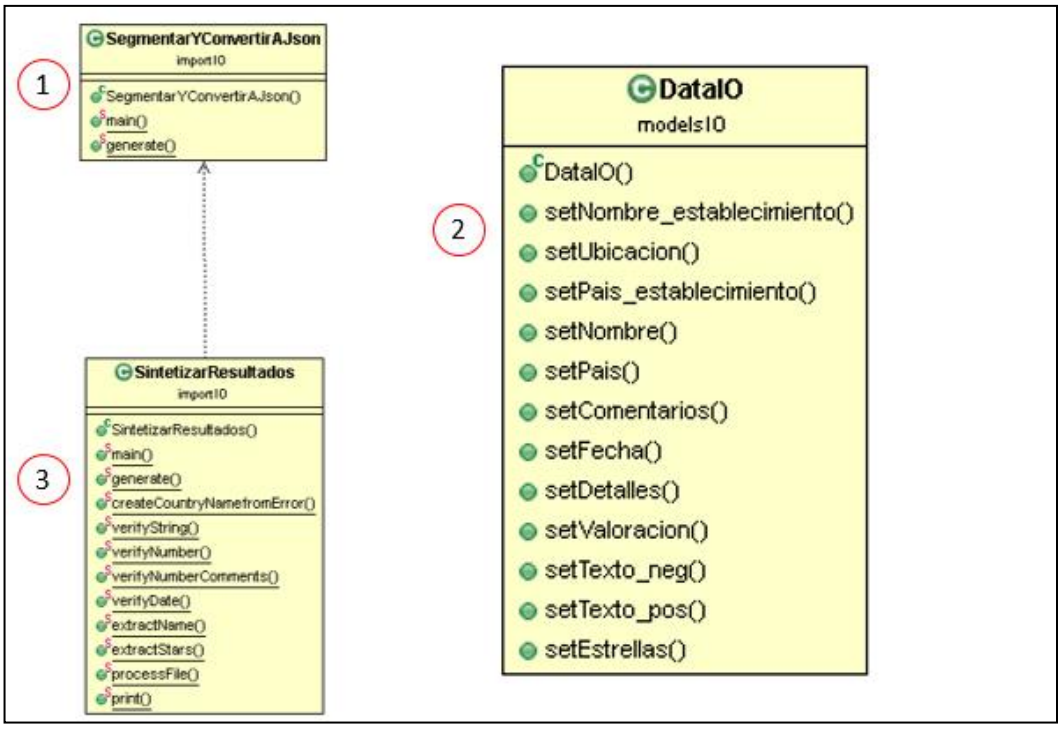


Ilustración 16 – Representación del bloque de procesado y formateado de datos

1. El primer paso será segmentar el archivo de entrada, con el fin de permitir que el sistema pueda trabajar sobre archivos cuyo tamaño no cause inconvenientes o errores de memoria, siempre asegurándose que ningún dato quede a medias entre dos archivos parciales. Estos ficheros, denominados **JSON**, están adecuadamente formateados para su uso.
2. Después, se procederá a generar un array de archivos *DataIO*, como el presentado en la imagen anterior, que poseerá los datos resultantes del *Extractor* en un formato estructurado y legible.
3. Así mismo, deberá comprobarse que los datos leídos son correctos, por lo que el sistema realizará un cierto número de comprobaciones y tareas secundarias, como sustituir o eliminar caracteres erróneos que puedan interferir con el resto de elementos del sistema (formato JSON, adaptable a BD, etc.), comprobar que los valores existen (en caso contrario, devuelve valores nulos que sean leíbles por el sistema y no provoquen errores), etc.

```
{
   "nombre_establecimiento": "Hotel Europa",
   "estrellas": "3",
   "ubicacion": "Carmen, 4, Madrid City Centre, 28013 Madrid",
   "pais_establecimiento": "Spain",
   "nombre": "Victoria",
   "pais": "United Kingdom",
   "comentarios": "5",
   "fecha": "2016-05-23",
   "detalles": "* Leisure trip * Couple * Double or Twin Room * Stayed 3 nights * Submitted via mobile",
   "valoracion": 8.3,
   "texto_neg": "It could get noisy at night when you're trying to sleep, but that wasn't really a problem for us.
   There was also a lot of work going on Sunday morning, again meaning lots of noise.",
   "texto_pos": "Excellent location, right inbetween Sol and Grand Via. Great value for money."
}
```


Ilustración 17 – Elemento de fichero CLEAN

El resultado final puede verse en la Ilustración 17, donde los comentarios se almacenarán en un array de elementos JSON, fácilmente integrables con una BD. Este array no proporciona conocimiento directo al usuario, ya que el fichero resultante, denominado **CLEAN** (en contrapartida al fichero **RAW** anteriormente mencionado), puede contener miles de estos comentarios. Por ello, será necesario proceder con los siguientes Bloques.

3.4. Resumen del Bloque

La extracción depende en parte de la propia fuente de datos. No obstante, hay elementos comunes en todas las posibles fuentes que permiten que este Bloque posea una estructura común y utilizable en todas ellas.

La versión actual del caso de estudio permite obtener comentarios por establecimientos individuales, o bien tomar todos los comentarios de un país, con tres posibles idiomas. Sin embargo, resulta obvio considerar que es posible modificar la búsqueda, por ejemplo, tomando múltiples países, o bien realizando una búsqueda por área mediante código postal. Así mismo, si se solventan las

limitaciones por idioma en el resto de herramientas, podría realizarse una extracción total de todos los archivos, o si se disponen de herramientas fiables de traducción, emplearlas para convertir todas las distintas opiniones a un solo idioma.

Por tanto, aun considerando sus potenciales modificaciones, quedará probada la utilidad de este Bloque en función de los resultados que se verán en los sucesivos capítulos de este documento.

Capítulo 4:

Almacenamiento

4.1. Introducción

En capítulos anteriores se ha expresado la importancia del concepto de Big Data, tanto por su capacidad de almacenar y manipular grandes cantidades de datos, como por sentar las bases del Análisis de Sentimientos. Debido a su importancia en este Proyecto Fin de Carrera, se hace necesario dedicar un breve capítulo para describir los conceptos aplicados en Eccetas.

4.2. Usos

4.2.1. Almacenamiento

Tal y como se ha visto en el anterior capítulo, el resultado del **Bloque Extracción** es un archivo JSON de elementos *DataIO*, como se ve en la Ilustración 18. Considerando que se pueden obtener miles o decenas de miles de estos elementos, trabajar con estas unidades de datos puede ser realmente complejo, particularmente a la hora de almacenar apropiadamente los datos.

A tal fin, se hará uso de las herramientas de Big Data, de forma que sea posible almacenar los datos y acceder a estos de forma segura y efectiva.

Ilustración 18 – Estructura DataIO

4.2.2. Gestión de Datos

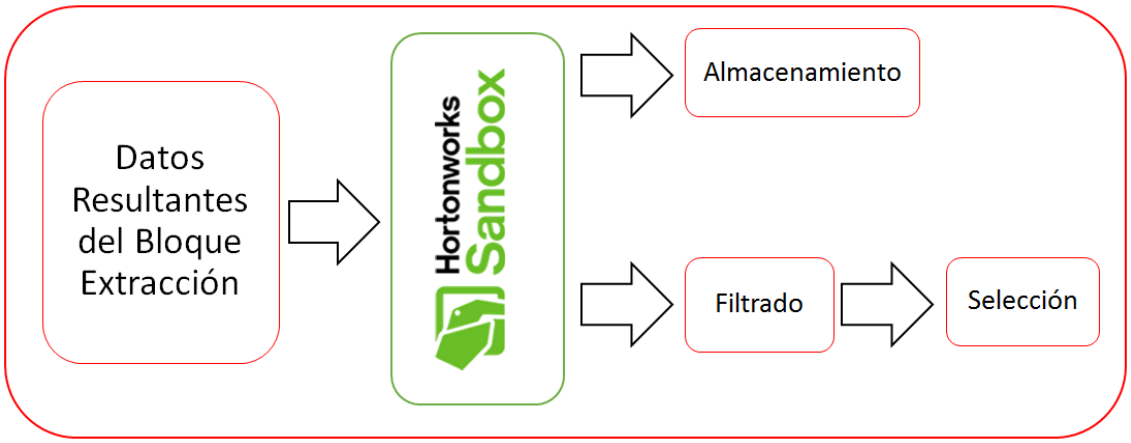


Ilustración 19 – Representación de las opciones de Gestión de datos

Disponiendo de los datos almacenados, será necesario poder trabajar sobre estos en un tiempo y a una velocidad razonable. A tal fin, existen herramientas que permiten modificar, cargar o suprimir grandes cantidades de datos en un tiempo aceptable y asegurando una ejecución efectiva y sin errores.

Particularmente, se hará necesario disponer de una herramienta que permita filtrar y seleccionar datos, bien basándose en algún atributo en concreto o premisa impuesta por el usuario, y a tal fin, se empleará el conjunto de herramientas Hadoop[16] (Particularmente, la herramienta Apache Hive), el cual se definirá a continuación.

4.3. Hadoop

Hadoop es una de las respuestas posibles a la siguiente cuestión:

¿Cómo es posible trabajar con un petabyte de información?

Ante esta pregunta, se exponen una serie de opciones:

- Utilizar muchos discos duros todos a la vez.
- Aplicando redundancia, ya que los discos duros tienden a fallar.
- Utilizar muchos núcleos de CPU todos a la vez.
- Aplicando reintentos, ya que los errores en la red tienden a sucederse.

¿Qué proporciona, por tanto, Hadoop?

- **Escalabilidad** – Muchos servidores con muchos núcleos y discos duros.
- **Fiabilidad** – Detecta errores, y almacena de forma redundante.
- **Resistente a fallos** – Auto–reintento, reparación de datos.
- **Simpleza** – Se usan muchos servidores como si fueran un solo gran ordenador.

Por tanto, en términos lógicos, Hadoop es un clúster de ordenadores que proporciona una capa de almacenamiento, y otra capa de ejecución.

Como uno de los principales framework de software que existen para Big Data, Hadoop es una de las piedras angulares del manejo de datos, permitiendo a las aplicaciones trabajar con miles de nodos y una gigantesca cantidad de datos.

Originariamente inspirado en Google y su computación, es un proyecto de alto nivel Apache, construido por una comunidad global de contribuyentes, haciéndolo un claro ejemplo del potencial del software libre.

Realizado en Java, el uso de Hadoop es masivo en todo el mundo, con numerosos casos de uso, así como eventos y cursos, que lo posicionan como un framework actual y en constante desarrollo, útil tanto para iniciados como para veteranos.

4.3.1. HDFS

Hadoop Distributed File System[17], o HDFS, es principalmente la capa de almacenamiento de Hadoop. Su funcionamiento, por encima del sistema operativo sobre el que ejecute, se basa en bloques de tamaño fijo, regularmente de 64 megabytes. Estos bloques, basados en el formato "*write once, read multiple times*", son además replicados hasta 3 veces a la hora de ser utilizados por alguna tarea. Si no se obtiene el mismo resultado las 3 veces, se descartarán y se volverá a empezar. De esta forma se asegura la fiabilidad.

4.3.2. Ambari

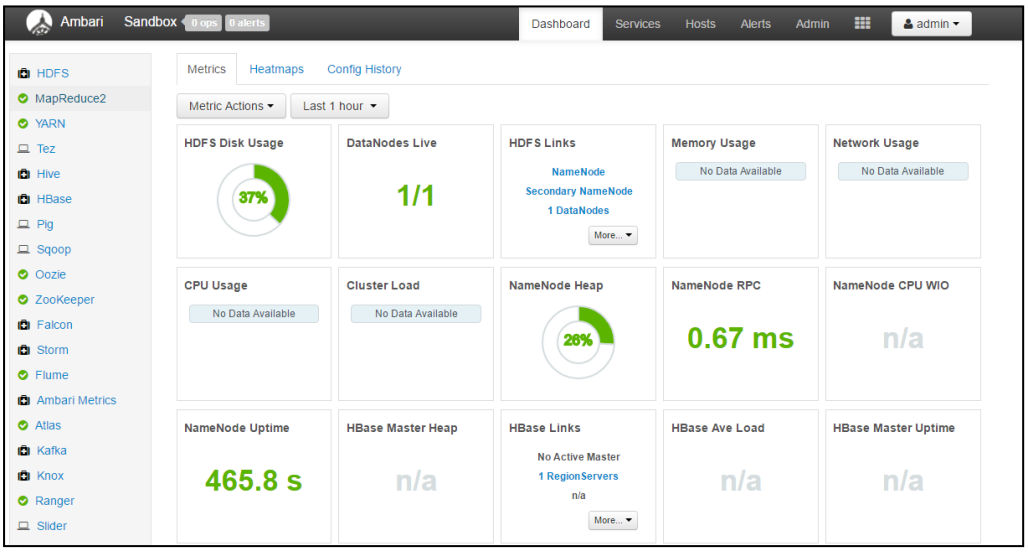


Ilustración 20 – Pantalla principal de Ambari

Ambari es un gestor de plataformas, escrito en código abierto, utilizado para el mantenimiento, monitorización, seguridad y aprovisionamiento de los clústeres de Hadoop. En esencia, remueve muchas de las dificultades que emanan de operar con Hadoop.

Como parte del Hortonworks Data Platform[18], facilita la instalación y configuración del HDP, tanto en su versión *REST API* como por su *Interfaz Web*, lo cual permite:

- Facilidad para configuración, instalación y mantenimiento.
- Configuración de seguridad centralizada.
- Visibilidad del estado del clúster.
- Altamente customizable y ampliable.

4.3.3. Hive

El estándar de facto para las peticiones SQL en Hadoop, Hive[19] permite procesar petabytes de datos en un tiempo y forma aceptables.

Dado que Hadoop fue construido para almacenar y organizar masivas cantidades de datos de todas las formas, tamaños y formatos, Hive debe ser capaz de procesar, resumir, explorar y analizar dichos datos, para después transformarlos en conocimiento útil. Todo ello puede resumirse en cuatro características:

- **Familiar** – Hacer llamadas a datos con un lenguaje basado en SQL. El formato que puede verse en la Ilustración 21 es relativamente común al empleado con SQL.
- **Rápido** – Tiempos de respuesta interactivos, incluso con cantidades de datos colosales.
- **Escalable y extensible** – En función de los datos y de su tamaño, el sistema puede ofrecer una mayor capacidad y respuesta.
- **Compatible** – Funciona con la integración tradicional de datos y las herramientas de análisis de estos.

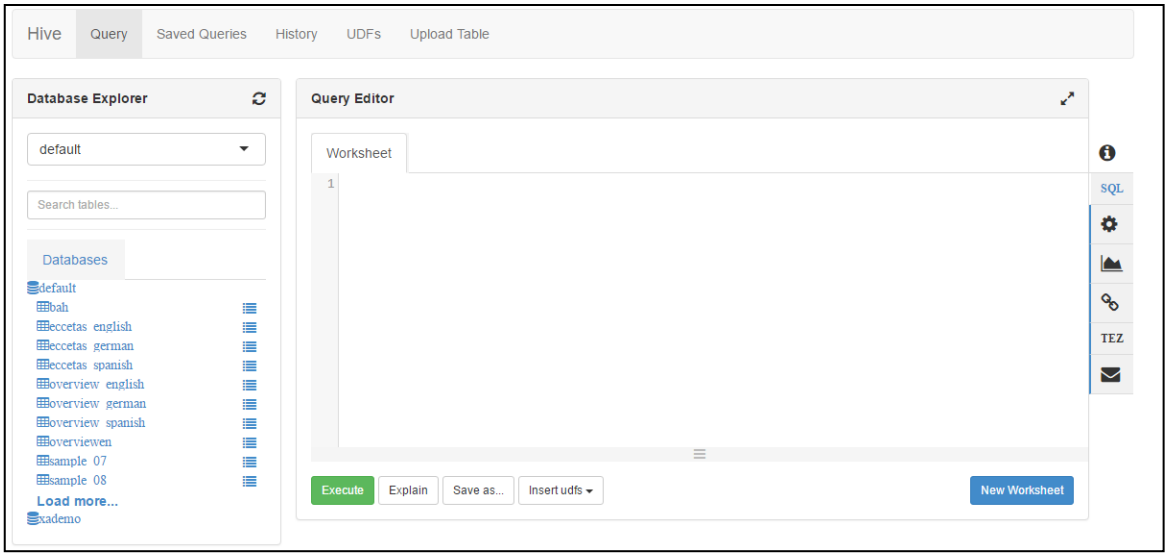


Ilustración 21 – Editor de peticiones de Hive

Las tablas de Hive son similares a las tablas en las bases de datos relacionales. Las bases de datos se comprimen en tablas, que están hechas de particiones. Es posible acceder a los datos por un lenguaje de peticiones sencillo, y Hive soporta sobreescribir o añadir datos al final.

Finalmente, Hive soporta todos los formatos primitivos de datos como BIGINT, BINARY, BOOLEAN, CHAR, DECIMAL, DOUBLE, FLOAT, INT, SMALLINT, STRING, TIMESTAMP, and TINYINT. Además, se pueden crear tipos más complejos de datos como estructuras, mapas y arrays.

4.4. Caso de Uso

Para poder utilizar los archivos resultantes del **Bloque Extracción**, ha sido necesario utilizar una serie de librerías[20] para que Hive sea capaz de parsear apropiadamente los ficheros que se le han suministrado. A tal fin, será necesario utilizar una librería para JSON, y otra para CSV, las cuales han sido incluidas en el entorno Hadoop.

4.4.1. Almacenar archivos CLEAN del Bloque Extracción


```
{
    "nombre_establecimiento": "Hotel Europa",
    "estrellas": "3",
    "ubicacion": "Carmen, 4, Madrid City Centre, 28013 Madrid",
    "pais_establecimiento": "Spain",
    "nombre": "Victoria",
    "pais": "United Kingdom",
    "comentarios": "5",
    "fecha": "2016-05-23",
    "detalles": "* Leisure trip * Couple * Double or Twin Room * Stayed 3 nights * Submitted via mobile",
    "valoracion": 8.3,
    "texto_neg": "It could get noisy at night when you're trying to sleep, but that wasn't really a problem for us.
    There was also a lot of work going on Sunday morning, again meaning lots of noise.",
    "texto_pos": "Excellent location, right inbetween Sol and Grand Via. Great value for money."
}
```


Ilustración 22 – Elemento DataIO de un fichero CLEAN

En el Caso General del **Bloque Extracción**, o en algunos resultados del Caso Específico, se cuentan con miles o decenas de miles de elementos como el mostrado en la Ilustración 22. Es posible almacenarlos de forma segura en Hive, como una sucesión de JSONs, el cual cumplen los ficheros **CLEAN**. Por ello, solo será necesario crear la tabla con una petición tan simple como:

```
CREATE TABLE eccetas_EXAMPLE (
nombre_establecimiento string,
estrellas string,
ubicacion string,
pais_establecimiento string,
nombre string,
pais string,
comentarios string,
fecha string,
detalles string,
valoracion string,
texto_neg string,
texto_pos string
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
STORED AS TEXTFILE;
```

Ilustración 23 – Petición SQL para almacenar ficheros CLEAN

4.4.2. Buscar entre archivos CLEAN (Extraer datos)

Habiendo almacenado los datos como se ha visto previamente, también es posible navegar entre estos con cierta facilidad. Bastará con escoger un criterio para el filtrado de los datos, como por ejemplo:

```
Select * from eccetas_EXAMPLE where ubicación RLIKE “35007” and
comentarios > “1.0” limit 100;
```

Ilustración 24 – Petición de búsqueda de resultados en Hive

Esta sentencia obtendría 100 resultados de toda la tabla anterior, cuyo código postal sea el 35007, perteneciente a la ciudad de Las Palmas, y como requerimiento, que el usuario habrá debido de haber escrito más de 1 comentario en la web.

La cantidad de criterios y resultados es muy alta, y el sistema es altamente flexible en ese aspecto. Utilizar Hive permite que navegar entre bases de datos extremadamente densas sea una tarea relativamente sencilla si se tiene un conocimiento de bases de datos SQL básico.

Capítulo 5:

Análisis

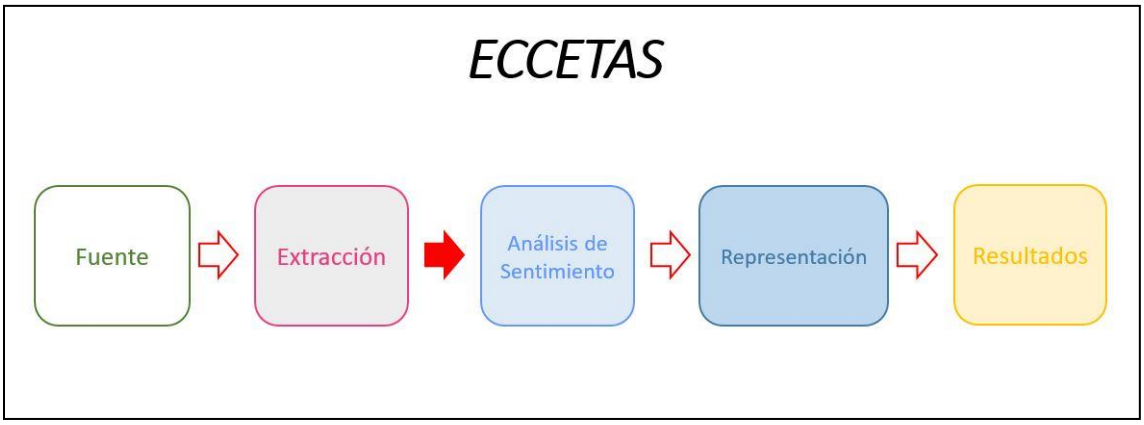


Ilustración 25 – Eccetas: Bloque Análisis

5.1. Introducción

Una vez obtenidos los datos sobre los que se debe trabajar, resulta claro que el siguiente paso es el procesado de estos. En el presente Proyecto Fin de Carrera, se les aplicará análisis de sentimiento. Esto es, es posible extraer la opinión o la valoración de los diversos datos de los que se disponen, utilizando para ello herramientas capaces de procesar el lenguaje natural, de forma que es posible analizar una gran cantidad de texto de forma automática.

No obstante, la propia subjetividad de los datos con los que se trabajan hace que el resultado de este bloque no pueda tomarse de forma completamente objetiva. Por ello, será necesario realizar las siguientes consideraciones:

- **Efectividad** – Se considerará que un resultado del análisis de sentimiento es efectivo cuando este se repita de forma consistente, considerando los datos que se disponen.
- **Idioma** – Por la gramática de las palabras y las limitaciones de las herramientas de procesamiento de texto, será necesario separar claramente los datos en función de su idioma, para no provocar errores en los resultados.

En esencia, será necesario considerar la subjetividad de los datos de entrada y de las operaciones de análisis, y obtener con ellos medidores de certeza, que permitan dar valor a los resultados finales.

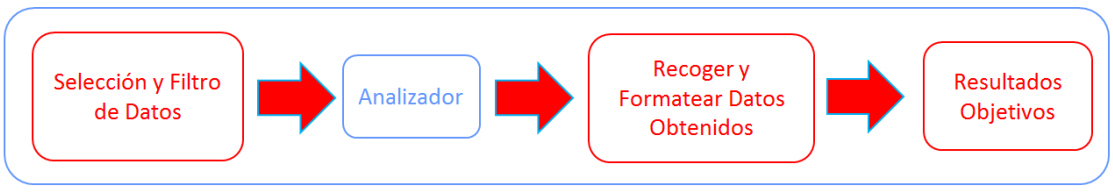


Ilustración 26 – Resumen General del Bloque Análisis

5.1.1. Caso de Estudio

El caso de estudio que se muestra a continuación para especificar y clarificar lo expuesto anteriormente presenta las siguientes características:

- Los datos de entrada al Bloque son comentarios en texto plano, almacenados bien en una BD de un entorno Hadoop como se ha indicado en el **Bloque Almacenamiento**, en formato CSV, o directamente generados por el **Bloque Extracción**, en formato JSON.
- De cara al **Bloque Representación**, será adecuado agrupar el resultado del análisis de sentimiento con la fecha correspondiente al comentario.
- Debido a las limitaciones de la herramienta de análisis de sentimiento, solo podrán usarse los textos en lenguaje **Inglés**.

Considerados estos hechos, será necesario disponer de una herramienta que realice el análisis de sentimiento y cuyos resultados, aun siendo subjetivos en su origen, puedan llegar a ser objetivos en base a todo el proceso realizado. Así mismo será necesario generar un código que de soporte a dicha herramienta, tanto para recibir los datos como para generar los resultados estructurados.

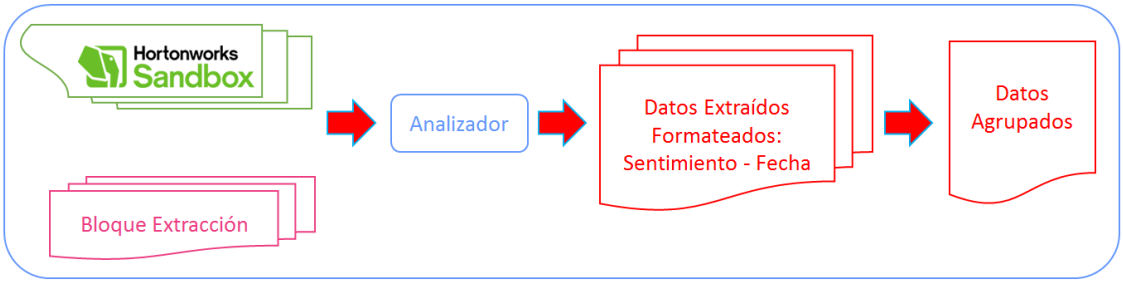


Ilustración 27 – Esquema del Caso de Estudio del Bloque Análisis

5.2. AlchemyAPI

AlchemyAPI es más que una herramienta; es una colección de scripts que permiten obtener una gran y variada cantidad de resultados de procesado de lenguaje, desde el sentimiento del texto hasta las palabras clave de este, pasando por el idioma o las entidades. Solo hay que aportar al script los datos que se desee analizar, y especificar el tipo de resultado que se desea obtener.

AlchemyAPI permite, por tanto, realizar el eje del trabajo de Eccetas. Con su script para extraer de un texto sus palabras clave, o *Keywords*, y a la vez obtener el sentimiento de estas, será posible construir la base principal de este Proyecto de Fin de Carrera: en base a los datos resultantes de los dos Bloques anteriores, será posible obtener el sentimiento sobre ciertas características de los establecimientos (dichas características se identificarán mediante las *Keywords*), pudiendo obtener un cierto nivel de objetividad en base a la cantidad de repeticiones en las que sea posible encontrar las K*eywords*. Esto es, se considerará que una característica analizada que se haya repetido un cierto número de veces en los comentarios será objetiva y no un dato sin valor.

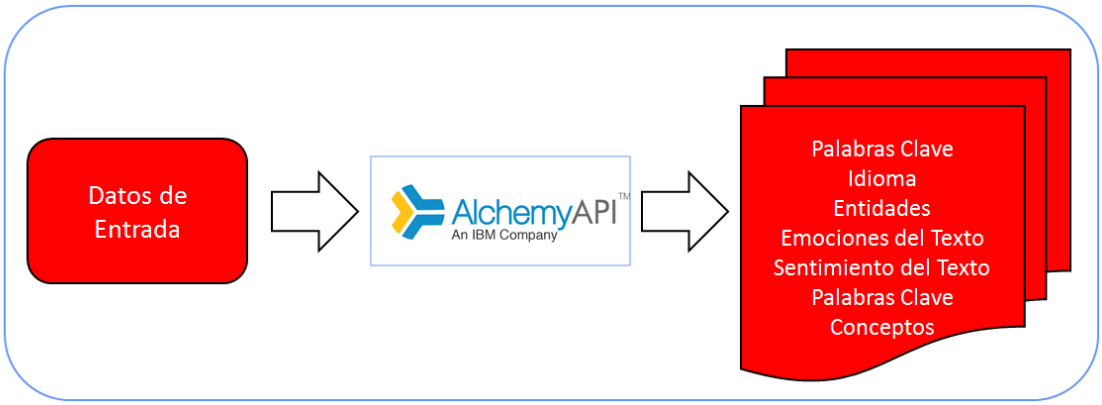


Ilustración 28 – Ejemplo de funcionamiento de AlchemyAPI

La ejecución de AlchemyAPI puede realizarse de forma manual mediante un entorno Web, o mediante unas librerías Java (u otros lenguajes de programación, como C o Python), que contactan con sus servidores para obtener resultados. El funcionamiento general para la obtención de las *Keywords* (cualquier otro resultado puede obtenerse, utilizando en ese caso la apropiada clase en el código), es el siguiente:

1. Se genera la clase de AlchemyLanguage con la *apikey* del usuario. Es posible marcar otros elementos, como el idioma, facilitando así el proceso de análisis.
2. Se crea un objeto clase *Map* que contendrá dos elementos. El primero, es el texto que se desea analizar. El segundo elemento debe indicar si se desea el sentimiento del texto a analizar.
3. Se lanza la petición de obtención de las *Keywords*, que devolverá un objeto Java con las *Keyword* y el sentimiento asociado a estas.

Estos resultados parciales se almacenarán para después ser procesados por otro código que permite agrupar las *Keywords* entre sí, de forma que el **Bloque Representación** dispondrá tanto de los datos parciales que se necesitan para realizar las gráficas pertinentes, como de los datos agrupados, que permiten su selección o filtrado.

Ha de tenerse en cuenta que debido al funcionamiento de la herramienta, se obtendrán muchos resultados sin sentido; la extracción de Keywords de un texto no es una tarea sencilla, por lo que habrá que descartar muchos resultados, así como agrupar aquellas palabras clave que sean necesarias (por ejemplo, "comfy bed", "small bed" y "great bed", deberán agruparse en un único término "bed", que tendrá un valor positivo, uno negativo y otro positivo, respectivamente).

Las únicas restricciones que aporta esta herramienta son:

- El número de idiomas soportados.
- El número de líneas de texto que pueda procesar AlchemyAPI, número que depende de la suscripción que se posea.

5.3. Estructura y Descripción del Bloque

Ilustración 29 – Estructura del Bloque Análisis

El **Bloque Análisis**, por tanto, estará conformado por los siguientes sub–bloques:

- Código Java para filtrar los datos entrantes y preparar la llamada de AlchemyAPI.
- Script de AlchemyAPI.

- Código Java para procesar los datos analizados, agruparlos, revisarlos y formatearlos convenientemente.

Considerando la enorme versatilidad de las numerosas API a escoger, y la simpleza de los datos de entrada, el **Bloque Análisis** es, por tanto, tremendamente adaptable y configurable, permitiendo añadir otras funciones adicionales que las mostradas en este documento.

5.3.1. Filtrar Datos y generar la llamada

La facilidad de uso de AlchemyAPI hace que sus datos de entrada tengan una estructura absolutamente simple. Solo hará falta entregar al sistema líneas de texto (al que se le anexará luego la fecha), por lo que podrán utilizarse las siguientes fuentes existentes en este desarrollo:

- Datos extraídos de la DB de Hadoop, ya convertidos por el **Bloque Extracción**, en formato CSV, que ya se presupone filtrados, quedándose solamente con los textos planos y sus correspondientes fechas.
- Datos extraídos directamente del **Bloque Extracción**, en formato JSON. Se realizará un filtro, extrayendo el texto de los comentarios y sus fechas asociadas, descartando el resto.

  Al final, sea cual sea la fuente, se dispondrá de un fichero en formato CSV que contendrá comentarios en texto plano, junto a sus fechas correspondientes.

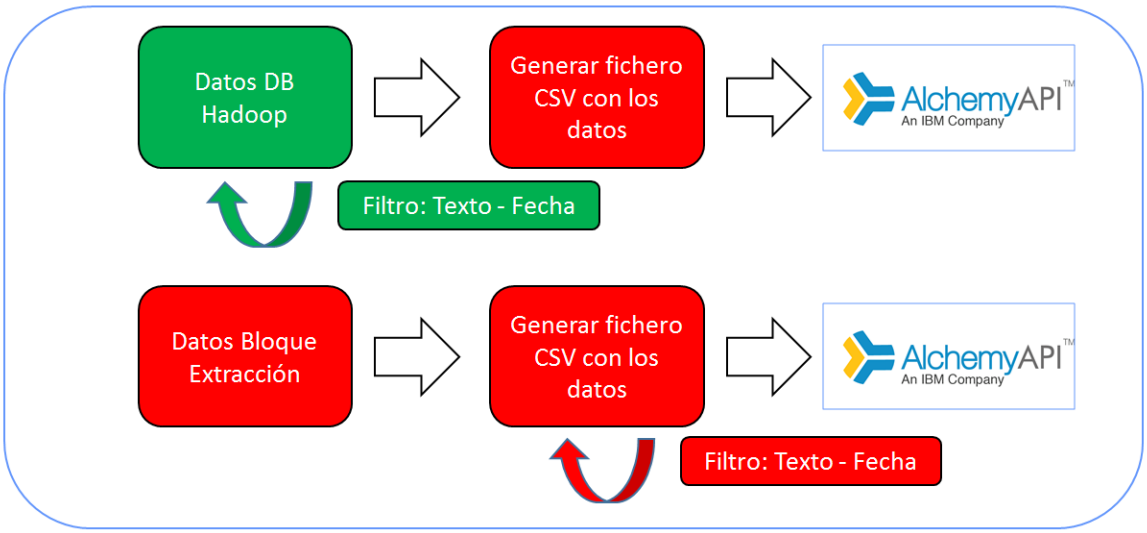


Ilustración 30 – Opciones de obtención de datos

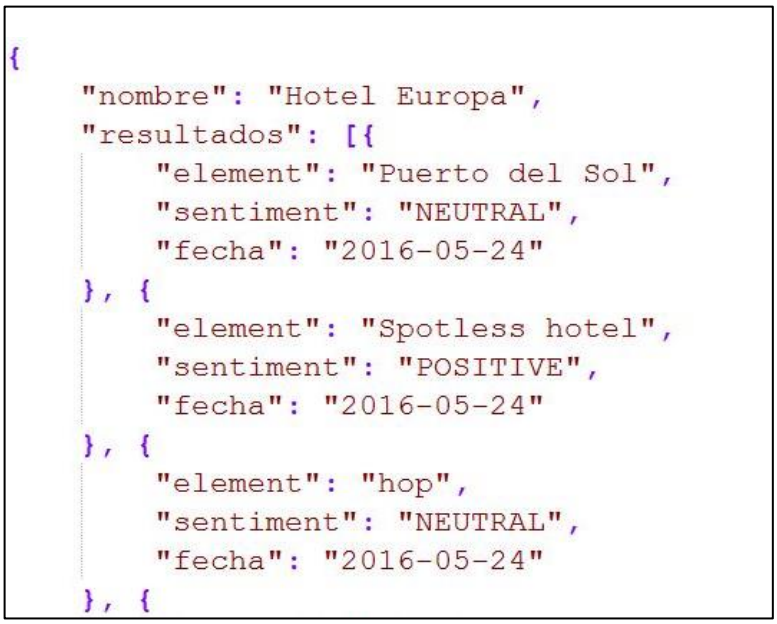
Es necesario en este punto hacer la siguiente consideración: en caso que no se disponga de una licencia de AlchemyAPI capaz de soportar todo el análisis (como ha sucedido durante la realización de este Proyecto Fin de Carrera, al recurrir a una licencia gratuita de 1000 procesamientos diarios) los ficheros CSV que se envían al script deberán de ser, como máximo, 500 líneas de texto (ya que solicitar sentimiento duplica el coste de procesado, llegando así al límite diario). En dicho caso, se utilizarán los códigos apropiados para agrupar todos los resultados una vez se hayan analizado las distintas partes, de cara a realizar el resto del proceso.

5.3.2. Uso de AlchemyAPI

Como se ha indicado anteriormente, los pasos para utilizar AlchemyAPI son relativamente sencillos. Una vez se dispone de los datos, solo será necesario generar un objeto *Map* que se enviará a los servidores de AlchemyAPI, configurado como se describió anteriormente. Realizado esto, se extraerán las *Keywords*, el sentimiento asociado a estas y se almacenará en un objeto *SentimentElement* que contendrá a ambos elementos obtenidos, así como la fecha de redacción del texto.

Ilustración 31 – Funcionamiento básico de AlchemyAPI

Los resultados, llegados a este nivel, pueden representarse como se observa en las Ilustraciones 32 y 33. Son indicativos hasta cierto punto; siguen siendo cientos o miles de valores separados, agrupados en un fichero **SENT**, que pueden resultar incluso contradictorios entre sí (es perfectamente viable que hayan opiniones negativas o positivas del mismo elemento el mismo día), por lo que será necesario dar un paso más, y realizar un agrupamiento que permita dar mayor coherencia a los resultados obtenidos.



```
{
    "nombre": "Hotel Europa",
    "resultados": [{
        "element": "Puerto del Sol",
        "sentiment": "NEUTRAL",
        "fecha": "2016-05-24"
    }, {
        "element": "Spotless hotel",
        "sentiment": "POSITIVE",
        "fecha": "2016-05-24"
    }, {
        "element": "hop",
        "sentiment": "NEUTRAL",
        "fecha": "2016-05-24"
    }, {
```


Ilustración 33 – Ejemplo de elementos de Fichero SENT

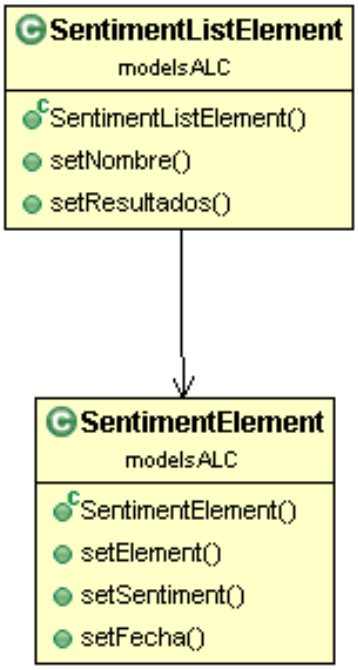
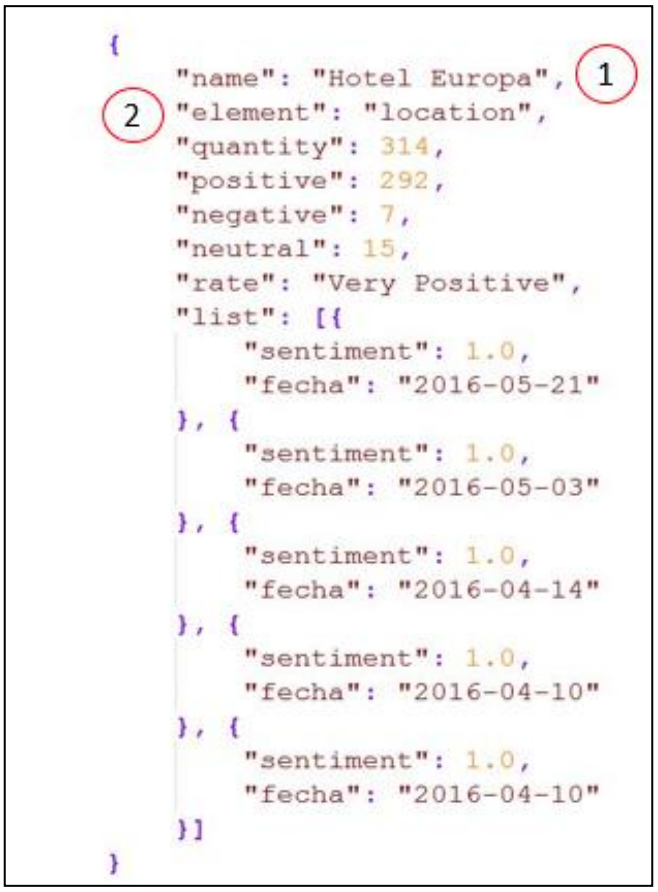


Ilustración 32 – Estructura SentimentList y su unidad



```
{
    "name": "Hotel Europa",
    "element": "location",
    "quantity": 314,
    "positive": 292,
    "negative": 7,
    "neutral": 15,
    "rate": "Very Positive",
    "list": [{
        "sentiment": 1.0,
        "fecha": "2016-05-21"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-05-03"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-14"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-10"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-10"
    }]
}
```


Ilustración 35 – Ejemplo de SentimentDataRow

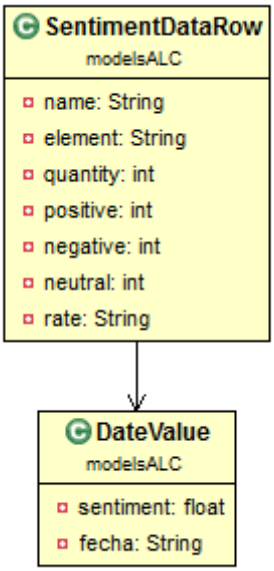


Ilustración 34 – Estructura SentimentDataRow y su unidad parcial

5.3.3. Procesado de los datos

Por lo expuesto anteriormente, será necesario considerar que los objetos *SentimentElement* de por sí no son aceptables. Es necesario poder agrupar los términos encontrados por sus *Keyword* para obtener un resultado sólido, y al mismo tiempo, conservar los resultados parciales para disponer de los distintos valores y fechas, datos vitales para el **Bloque Representación**. Ante esto, se propone utilizar la siguiente estructura, *SentimentDataRow*, que contiene una lista de elementos *DateValue*, como se observa en las Ilustraciones 34 y 35:

Esta estructura no solo agrupa apropiadamente los datos parciales, sino que aporta una gran cantidad de información al usuario. Así mismo, es fácilmente manejable de nuevo por Hadoop y representativa tanto del establecimiento (1) como de la característica en cuestión (2), como se ve en la Ilustración 34, permitiendo una organización mucho más práctica, y ofreciendo por tanto las unidades con las que el **Bloque Representación** podrá hacer sus comparativas.

Para obtener estos resultados, no obstante, ha sido necesario realizar una serie de procesos sobre las *Keyword* obtenidas por AlchemyAPI, ejecutados por el código que se representa en la Ilustración 36:

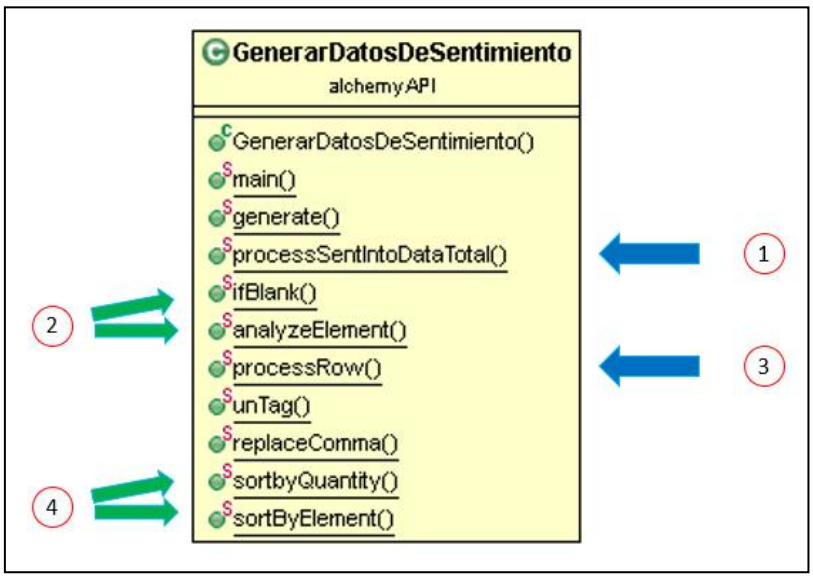


Ilustración 36 – Representación del bloque de procesamiento y agrupación de datos

1. Agrupar las *Keywords* que definan al mismo elemento en una única *SentimentDataRow* (sumando los *DateValues* correspondientes y actualizando

los valores ya existentes), y crear nuevas *SentimentDataRow* al encontrar *Keywords* nuevas.

2. Analizar las *Keywords* mediante un método de análisis por diccionario[21] para retirar adjetivos y otros elementos que impidan una agrupación apropiada de las *Keywords*, en caso de que encuentre que la *Keyword* en particular sea un término compuesto y no una sola palabra. Lo mismo se aplica con los términos plurales.
   - El análisis de diccionario mencionado aprovecha el hecho de que es necesario trabajar en el lenguaje inglés para realizar un análisis con una librería que tras analizar el texto, le agrega etiquetas ("tags") en función de si es un sustantivo, un adjetivo, un verbo… De forma que como se expuso anteriormente, "small bed", "great bed" y "comfy bed" pueden unirse en un solo *SentimentDataRow* "bed".
   - Idealmente, también resulta interesante combinar términos plurales como "rooms" en "room".
3. Una vez generadas todas las *SentimentDataRow*, calcular el número de opiniones negativas, positivas y neutras, y almacenar los resultados en los atributos del *SentimentDataRow* correspondiente.
4. Finalmente, el archivo **DATA** que contiene todas las *SentimentDataRow* estará organizado por el nombre de los elementos y el atributo *Quantity* de dichos objetos, primando este último criterio.

5.4. Resumen del Bloque

Siendo todos los Bloques de este Proyecto Fin de Carrera necesarios para el correcto desarrollo, este resulta ser el centro y núcleo, ofreciendo datos que podrían considerarse finales, y ofreciendo conocimiento si se compara con los datos obtenidos por el **Bloque Extracción**.

Sin embargo, el **Bloque Análisis** también aporta una enorme cantidad de subjetividad al sistema. El hecho que, por cada *Keyword* única obtenida, se genere una *SentimentDataRow*, hace que se generen varias *SentimentDataRow* de un solo elemento que no aportan ninguna información objetiva, como muestra la Ilustración 37:

```
        {"name":"Moderno","element":"key","quantity":1,"positive":1,"negative":0,"neutral":0,
"rate":"Very Positive","list":[{"sentiment":1.0,"fecha":"2015-10-07"}]}

        {"name":"Moderno","element":"kid","quantity":1,"positive":0,"negative":1,"neutral":0,
"rate":"Very Negative","list":[{"sentiment":-1.0,"fecha":"2016-01-03"}]}

        {"name":"Moderno","element":"men","quantity":1,"positive":0,"negative":1,"neutral":0,
"rate":"Very Negative","list":[{"sentiment":-1.0,"fecha":"2016-01-12"}]}

        {"name":"Moderno","element":"par","quantity":1,"positive":0,"negative":0,"neutral":1,
"rate":"Neutral","list":[{"sentiment":0.0,"fecha":"2016-05-27"}]}

        {"name":"Moderno","element":"tv","quantity":1,"positive":0,"negative":1,"neutral":0,"
rate":"Very Negative","list":[{"sentiment":-1.0,"fecha":"2014-10-06"}]}
```


Ilustración 37 – Elementos SentimentDataRow no útiles de fichero DATA

Por tanto, será necesario tener en cuenta la primera consideración expuesta en la introducción de este **Bloque**, y realizar un filtrado de los *SentimentDataRow* en función de su término *Quantity*, esto es, cuantas veces se ha opinado de la misma característica. Según se desee una mayor precisión, este valor deberá ser más elevado, filtrando dramáticamente los resultados, pero aportando una veracidad que, de otro caso, no sería posible obtener. Pero esta filtración se realizará en el siguiente Bloque, donde quedará a discreción del usuario.

Capítulo 6:

Representación

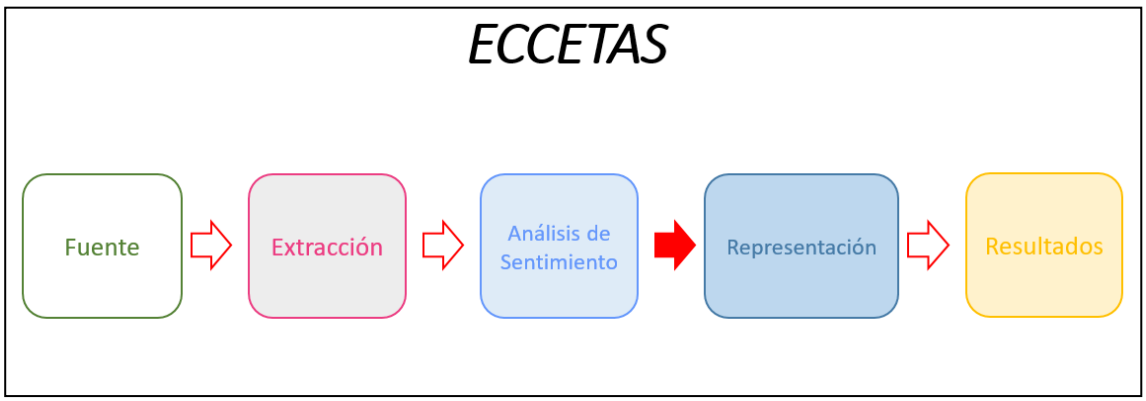


Ilustración 38 – Eccetas: Bloque Representación

6.1. Introducción

En el bloque anterior ya se disponía de datos que aportaban una cierta información útil. Sin embargo, estos datos requieren disponer de un conocimiento previo antes de poder obtener estimaciones de ellos, o sencillamente analizarlos requiere de un tiempo que los usuarios no están dispuestos a invertir, especialmente a la hora de hacer comparativas. Es a raíz de ello que se genera el **Bloque Representación**.

El objetivo de este Bloque es el de presentar los datos obtenidos por Eccetas con gráficas personalizables, que serán capaces de mostrar datos generales y específicos con la misma calidad que resultaría de analizar los datos directamente, con el tiempo extra que ello conlleva.

Las consideraciones que habrá que recalcar en este Bloque son, por tanto, las siguientes:

- **Precisión** – Las gráficas deberán aportar al menos la misma información que la que aportarían los datos en los que se basan, de ser analizados directamente.
- **Adaptabilidad** – Las gráficas deberán ser capaces de mostrar desde datos genéricos a valores específicos, adaptándose así a las necesidades del usuario.
- **Efectividad** – Considerando la subjetividad aportada por el **Bloque Análisis**, este Bloque deberá ser capaz de asignar criterios objetivos que eliminen la mayoría de la subjetividad, como por ejemplo, utilizar solamente las gráficas que posean un mayor número de resultados.

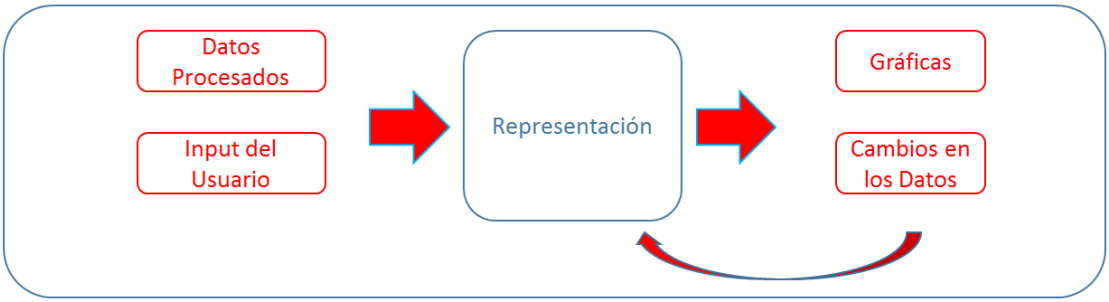


Ilustración 39 – Resumen General del Bloque Representación

6.1.1. Caso de Estudio

El caso de estudio dispone de las siguientes características:

- Los datos de entrada son un gran número de objetos *SentimentDataRow*, organizados por archivos en función del establecimiento.
- Los *SentimentDataRow* pueden ordenarse por establecimiento y característica. También disponen de los resultados parciales (fechas y números que indican el sentimiento) que se utilizarán para el diseño de las gráficas.

Por tanto, será posible hacer comparativas entre establecimientos, o analizar las características de un solo establecimiento con mayor precisión, todo ello a lo largo del tiempo, gracias a que se dispone de los resultados parciales.

Considerando lo expuesto, será conveniente disponer de una herramienta que permita recoger los datos de los que se dispone, y generar las gráficas que cumplan los criterios anteriormente mencionados, así como disponer de un código que haga todos los cambios y filtrados que el usuario considere necesarios a los datos disponibles.

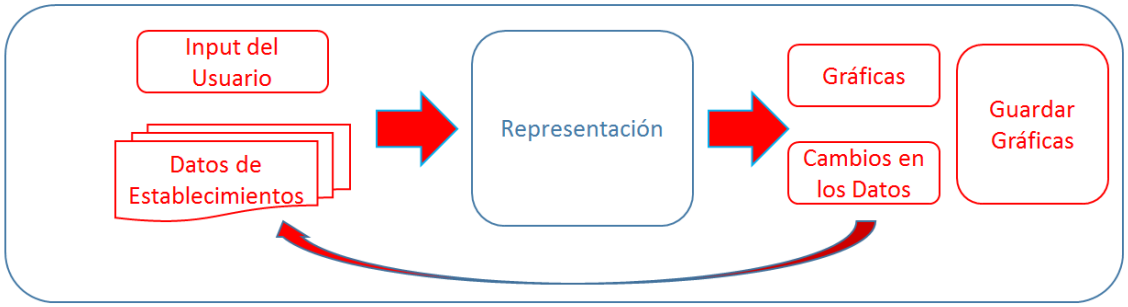


Ilustración 40 – Esquema del Caso de Estudio del Bloque Representación

6.2. JFreeChart

JFreeChart es una librería Java bajo licencia LGPL[22], que permite su uso gratuito en soluciones de cualquier tipo. Permite generar una gran variedad de distintas gráficas, altamente personalizables, requiriendo de un proceso de aprendizaje relativamente rápido para introducir en sus clases particulares los datos apropiados.

Donde Import.IO funciona mediante una REST API accesible desde múltiples entornos, y AlchemyAPI posee varias librerías en múltiples lenguajes de programación para acceder a su API, JFreeChart es una librería exclusiva de Java. A cambio, aporta la ventaja de ser completamente gratuita, y su reducido tamaño hace que sea relativamente sencillo aplicar esta librería en casi cualquier solución.

La estructura de JFreeChart engloba dos elementos fundamentales: *Datasets*, y *Charts*.

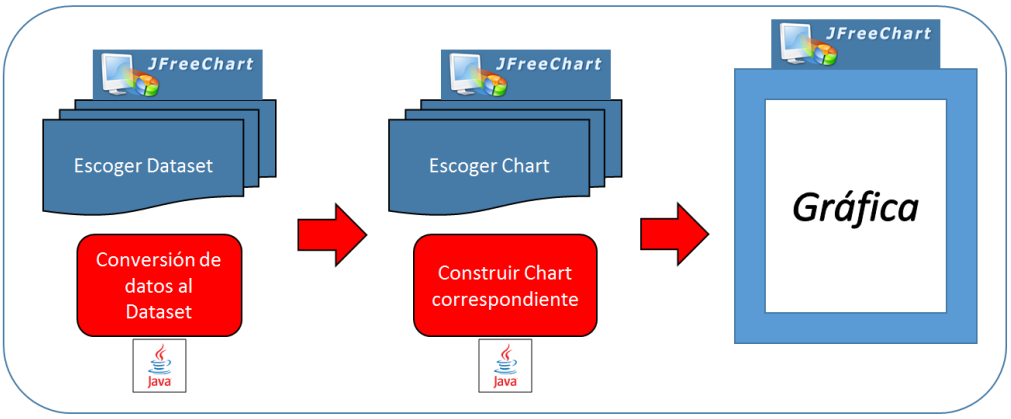


Ilustración 41 – Construcción de una gráfica JFreeChart

Un *Dataset* son los datos que tendrá la gráfica, pudiendo disponer de varios tipos distintos (*CategoryDataset*, *SeriesDataset*, *PieDataset*, *XYDataset*, etc.), los cuales se utilizarán en función del tipo de gráfica que se desee, siendo necesario adaptar los datos de origen a la estructura del *Dataset* para su correcto funcionamiento.

Un *Chart*, por otro lado, es el "lienzo" sobre el que se pintarán los datos que posee el *Dataset*. Los *Chart* también disponen de numerosas formas (*BarChart*, *LineChart*, *PieChart*, *TimeSeriesChart*, etc.), que solo funcionarán si se les

introduce el *Dataset* adecuado. Así mismo, son altamente configurables, desde el grosor del trazo hasta el color utilizado, pasando por añadir etiquetas a los datos, etc. En la Ilustración 42 se observan dos ejemplos: El primero es un caso de *BarChart*, mientras que el segundo corresponde a un *PieChart*.

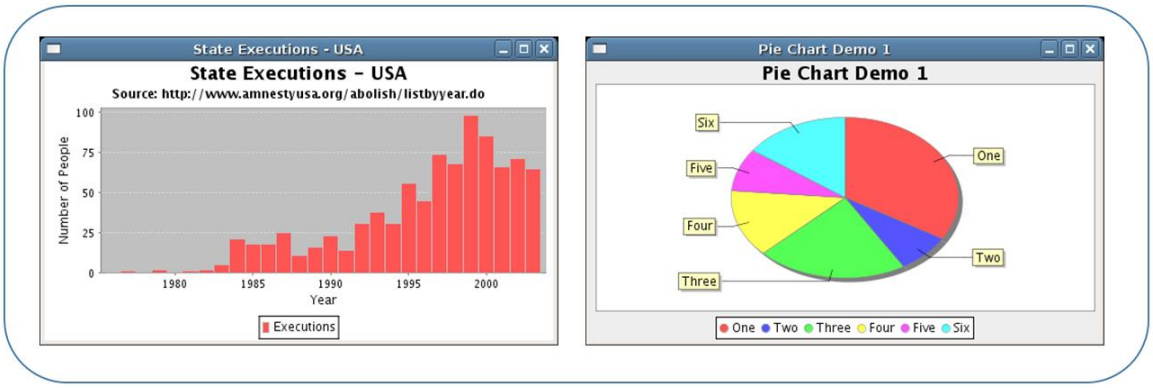


Ilustración 42 – Ejemplos de *Chart*

El proceso de uso de JFreeChart se describe a continuación:

- Se formatean los datos de entrada en el *Dataset* que vaya a utilizarse.
- Se genera el objeto *Chart* apropiado, y se le vincula el *Dataset* creado.
- Se lanza el *Chart*, generándose la gráfica apropiada, o se almacena directamente como archivo de imagen.

Por ello, y dadas las diferentes opciones que proporciona esta librería, será posible cumplir todas las consideraciones indicadas anteriormente, de la forma en la que se describe a continuación.

6.3. Estructura y Descripción del Bloque

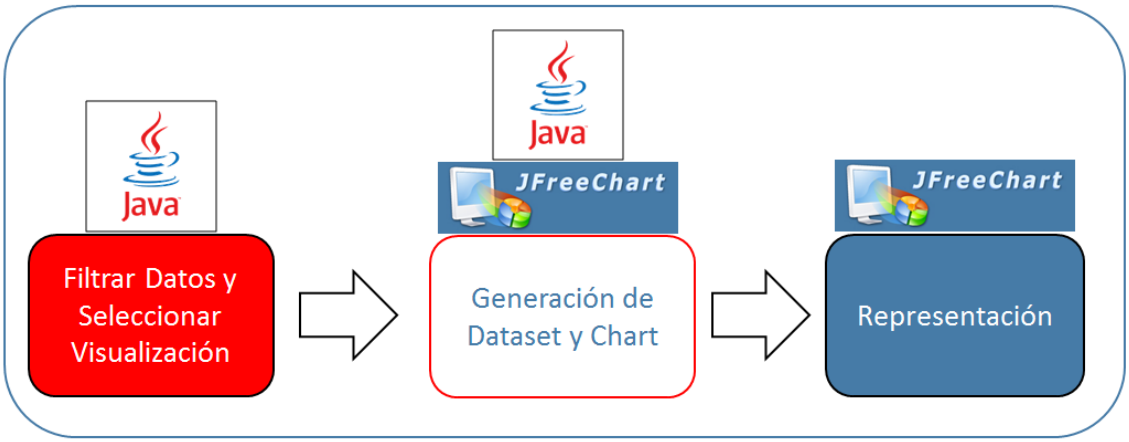


Ilustración 43 – Estructura del Bloque Representación

El **Bloque Representación**, por tanto, estará conformado de la siguiente manera:

- Código Java para filtrar los datos y seleccionar qué clase de gráfica se generará.
- Generación del *Dataset* y *Chart* correspondientes.
- Representación de los datos finales.

Si bien este **Bloque** depende de los resultados obtenidos del **Bloque Análisis**, sin embargo posee una enorme versatilidad debido al número de *Datasets* y *Charts* disponibles, que permiten realizar múltiples representaciones distintas.

6.3.1. Filtrar Datos y Seleccionar Gráfica

Inicialmente, antes de solicitar el input al usuario, será preciso generar la base de datos sobre la que se apoyará este Bloque. Esta tarea, automática e independiente del usuario, generará unos nuevos ficheros, filtrados de los datos generales en función a un número concreto de comentarios. El usuario podrá solicitar un filtrado más o menos riguroso, como se indica a continuación:

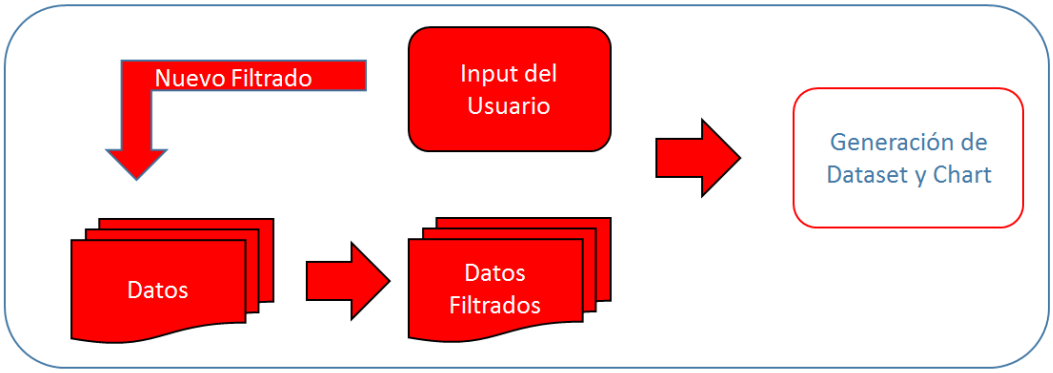


Ilustración 44 – Esquema del filtrado y selección de datos

En este Proyecto de Fin de Carrera, se encuentran habilitadas las siguientes opciones para el usuario:

1. Visualizar una característica común en todos los establecimientos.
2. Elegir un establecimiento, y una característica concreta de este.
3. Visualizar la valoración general de un establecimiento.

4. Visualizar las características negativas de un establecimiento, en función de una fecha concreta.
5. Escoger entre visualizar directamente la gráfica, o almacenarla.
6. Cambiar el valor mínimo para filtrar los datos iniciales (Rehace la base de datos con el nuevo valor).
7. Cambiar el valor mínimo para considerar características como negativas (Parámetro de la 4º opción)

En este paso será donde se pueda fijar la condición de efectividad mencionada durante el resumen final del **Bloque Análisis** (Opción 6º), esto es, descartar todas las *SentimentDataRow* cuyos valores *Quantity* no alcancen el valor fijado, limitándose así a información más objetiva, o bien asignar un valor menor y tomar más datos, a coste de una mayor subjetividad.

Así mismo, será aquí donde, en función de la opción escogida, se generará el par de *Dataset–Chart* correspondiente.

6.3.2. Generación de Dataset–Chart

En función de las opciones escogidas anteriormente, se procederá a la creación de los objetos Java necesarios para JFreeChart, así como el cálculo de los valores que se cargarán en dichos objetos. Los pasos son los siguientes:

6.3.2.1 Dataset

- Acceder al fichero (o ficheros) JSON con las *SentimentDataRow* filtrados según su *Quantity*.
- Obtener las *SentimentDataRow* apropiadas, en función de la selección del usuario.
- Generar el *Dataset*, rellenándolo con los datos correctos y realizando las operaciones que sean convenientes (sustituir las fechas completas de los datos parciales por solo mes y año, sumar resultados parciales para dar estimaciones totales, etc.), siempre adaptándose al formato del *Dataset* escogido.

6.3.2.2 Chart

- Generar un C*hart* del tipo correspondiente al Dataset creado anteriormente.
- Fijar todas las características en función de la gráfica escogida; estilo de gráfica, formato de fecha, color, etc.

6.3.3. Representación

En este punto, se dispondrá de las gráficas, que pueden ser almacenadas en el disco duro o representadas en pantalla. A continuación, se presenta un ejemplo de las gráficas que ofrece el presente Proyecto Fin de Carrera:

6.3.3.1 ClusteredData

Esta gráfica permite representar los datos de sentimiento sobre una característica y un establecimiento turístico en concreto. Permite también hacer una comparativa con las características comunes a múltiples establecimientos, adaptando su eje de fechas apropiadamente según cuando fueron tomados los valores en cada caso. Un ejemplo se muestra en la Ilustración 45. Su leyenda es la siguiente:

1. Tipo de Gráfica
2. Característica
3. Establecimiento
4. Leyenda

6.3.3.2 NegativeData

La finalidad de esta gráfica es mostrar las características negativas, dada una fecha en concreto. Principalmente, permite analizar los últimos datos de los que se dispongan, con el fin de localizar rápidamente aspectos críticos en un establecimiento turístico. Un ejemplo se muestra en la Ilustración 46. Su leyenda es la siguiente:

1. Tipo de Gráfica
2. Fecha de origen de los datos

3. Característica negativa y el nº de opiniones
4. Establecimiento

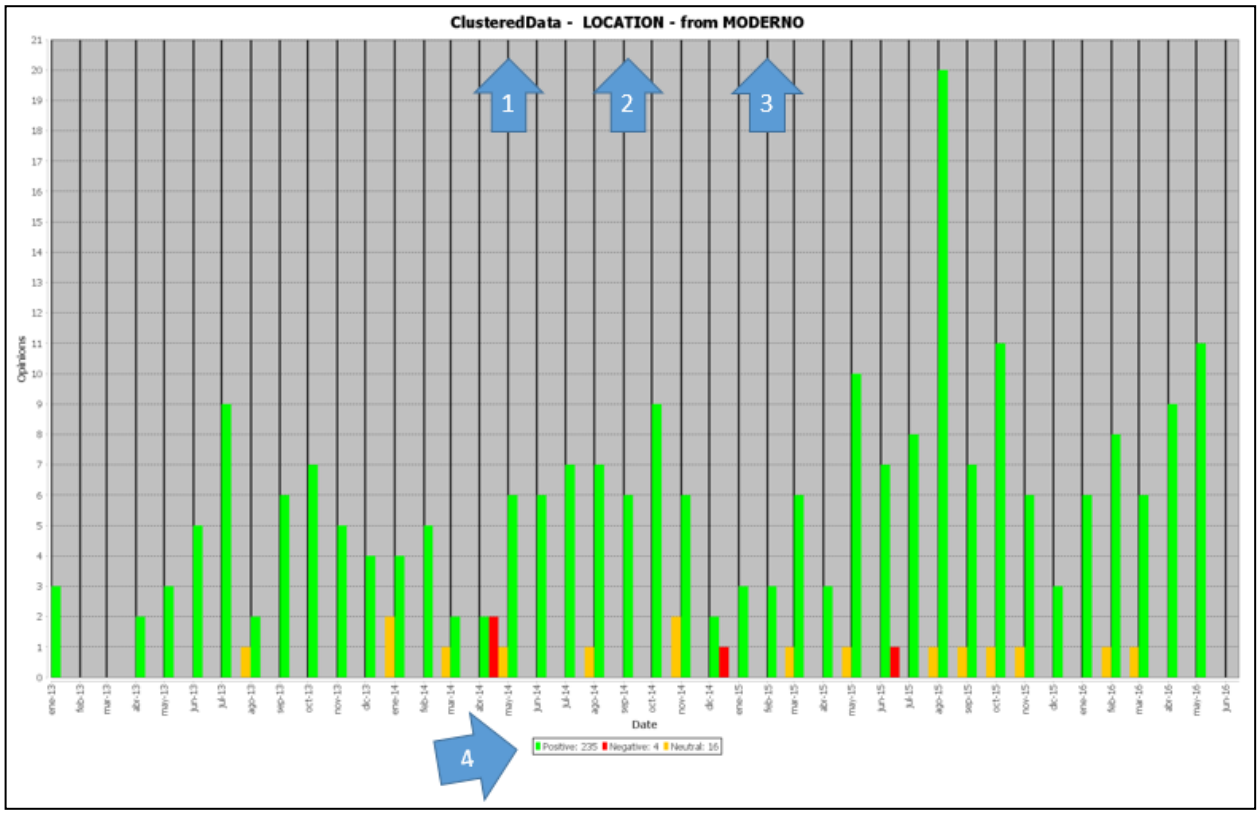


Ilustración 45 – Ejemplo de gráfica ClusteredData

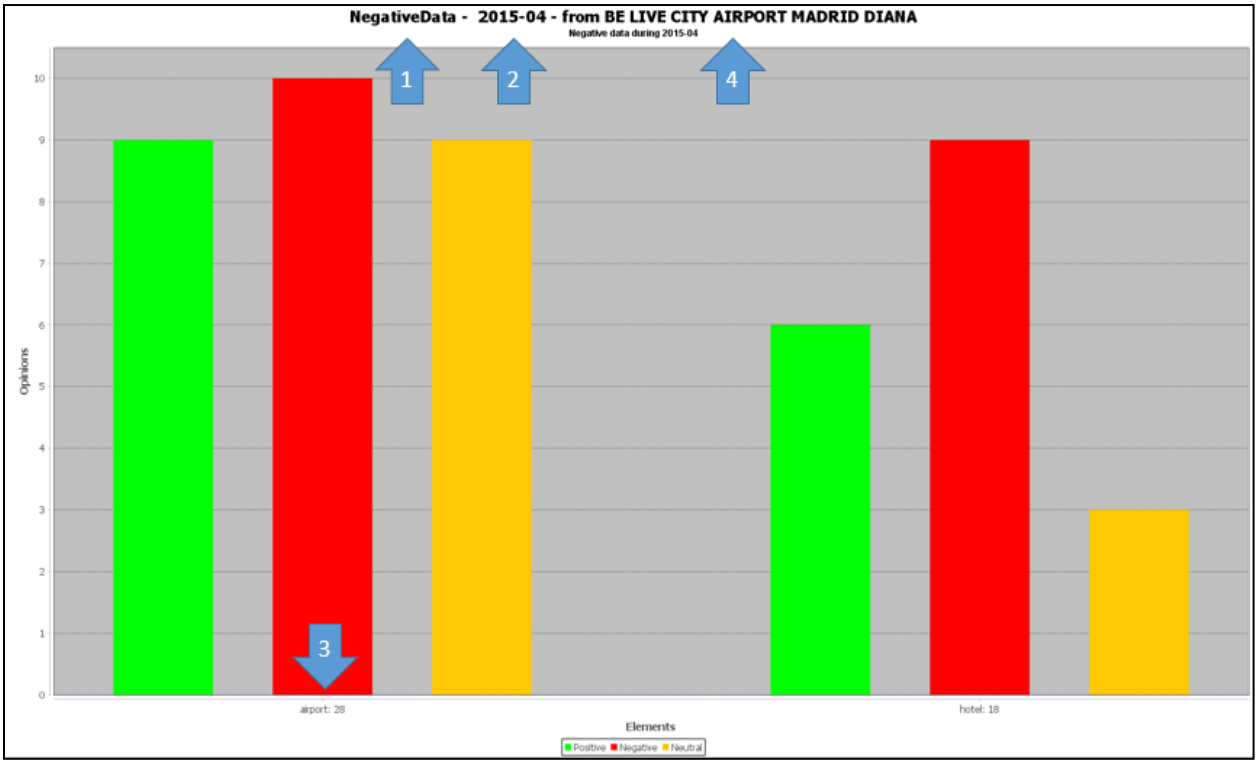


Ilustración 46 – Ejemplo de gráfica NegativeData

6.3.3.3 RateData

Este modelo de gráfica no utiliza los mismos datos de sentimiento que las anteriores, sino que muestra las valoraciones generales de los usuarios, agrupadas por mes. En la Ilustración 47 se muestra un ejemplo. Su leyenda es la siguiente:

1. Tipo de Gráfica
2. Establecimiento

Ilustración 47 – Ejemplo de gráfica RateData

6.4. Resumen del Bloque

Como último **Bloque** de este Proyecto Fin de Carrera, es el que presenta los resultados del resto del sistema, y es el que más puede ampliarse y adaptarse a posteriori. Debido a su potencial adaptabilidad, en base al cliente que vaya a requerirlo, deberá ser capaz de generar otros tipos de gráficas y modelos. Dado que se solicitarán datos actualizados, le obligará a utilizar los bloques anteriores continuamente.

Su potencial, no obstante, está ligado a las posibilidades que ofrezca la librería gratuita y, por otro lado, a satisfacer las necesidades del cliente, generando

aquellas gráficas que aporten una información valiosa y desechando aquellas carentes de valor o redundantes.

Considerando el resumen del Bloque Análisis, donde se hablaba de la subjetividad de los resultados, es en este punto donde, al disponer de unas gráficas que sintetizan cientos de opiniones en unos datos sencillos de leer, y cuya validez es posible evaluar, puede considerarse que los datos se tornan objetivos. Si se desearan unas gráficas mucho más objetivas, debería proveerse al sistema del mayor número de datos de entrada posibles, con el fin de generar gráficas con un mayor número de puntos, permitiendo por tanto hacer objetivos los datos parciales.

El filtrado inicial que realiza el **Bloque Representación**, suprimiendo muchas *SentimentDataRow* es, por tanto, absolutamente necesario para desechar los errores generados por el **Bloque Análisis**, los cuales son comprensibles teniendo en cuenta la subjetividad propia del análisis de sentimiento. La capacidad para comparar características y establecimientos quedará por tanto ligada a disponer de los datos de entrada apropiados.

Finalmente, como prueban las gráficas *RateData*, el potencial de este Bloque no está limitado solamente a los datos extraídos del **Bloque Análisis**, ya que la posibilidad de representar casi cualquier dato de forma directa e intuitiva permite obtener mucho más conocimiento de datos que, inicialmente, no tendrían una gran utilidad.

Capítulo 7: Resultados

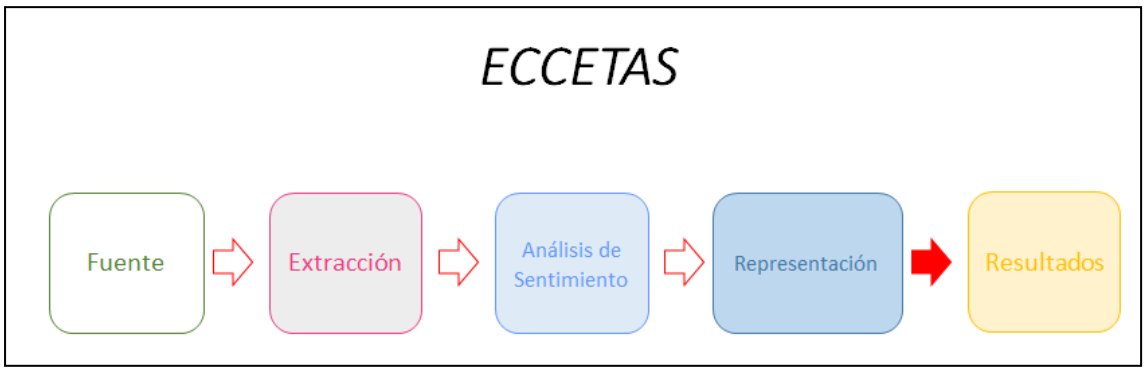


Ilustración 48 – Eccetas: Resultados

7.1. Trazas

Como se ha visto en las sucesivas etapas de este Proyecto, cada bloque genera su propio resultado parcial, siendo la suma de estos el verdadero producto final del sistema. A continuación, se representarán las distintas trazas pertenecientes a cada uno de los Bloques.

Por practicidad, se considerará que se disponen de las licencias adecuadas de Import.IO y de AlchemyAPI, por lo que el proceso no se verá ralentizado por las limitaciones de las versiones gratuitas. En cada caso, no obstante, se hará mención a dichas limitaciones, con fines instructivos.

Finalmente, para comprender mejor la traza, se procede a mostrar los distintos tipos de objetos que se encontrarán en ella mediante la Ilustración 49:

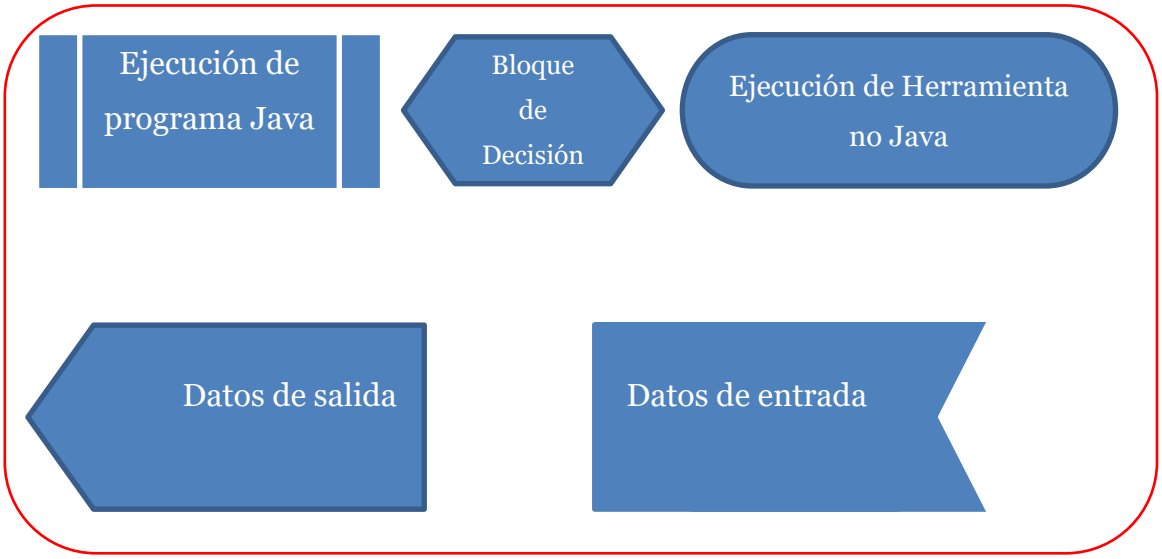


Ilustración 49 – Bloques de Traza

7.1.1. Bloque Extracción

En el caso del **Bloque Extracción**, puede considerarse que hay dos tipos posibles de traza, dependiendo de si se trata del **Caso Específico** (se buscan los datos de 1 solo establecimiento) o el **Caso General** (se buscan todos los establecimientos posibles en un idioma y país determinado).

En esencia, en el **Caso General** se pueden dar todos los pasos posibles, mientras que en el **Caso Específico** solo se realiza la ruta más corta, ejecutando uno solo de los *Extractores* de Import.IO.

Como añadido, en el **Caso Específico** se realiza la extracción de la valoración general, que servirá como resultado adicional del sistema. No se realiza en el Caso General puesto que en ese caso, la extracción de la valoración general se realiza directamente desde las herramientas de gestión de Big Data.

7.1.1.1 Limitaciones

En el **Caso General**, no se ejecutan automáticamente ninguno de los *Extractores*, siendo necesario ir paso a paso. Además, en el caso del *Extractor Main*, se ejecutará un número de veces concreto, con el fin de no saturar las peticiones de entrada.

7.1.1.2 Caso General

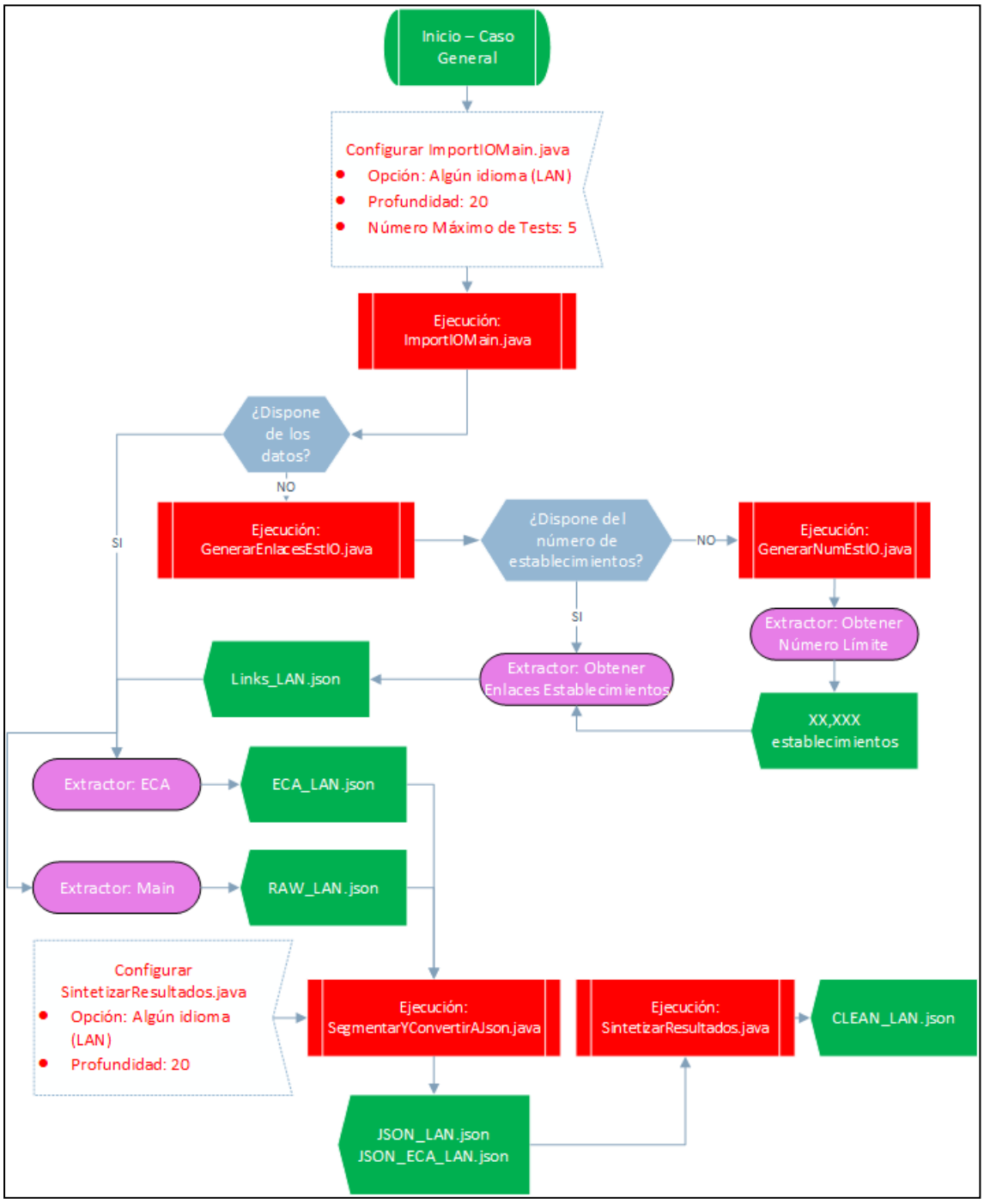


Ilustración 50 – Traza del Caso General del Bloque Extracción

7.1.1.3 Caso Específico

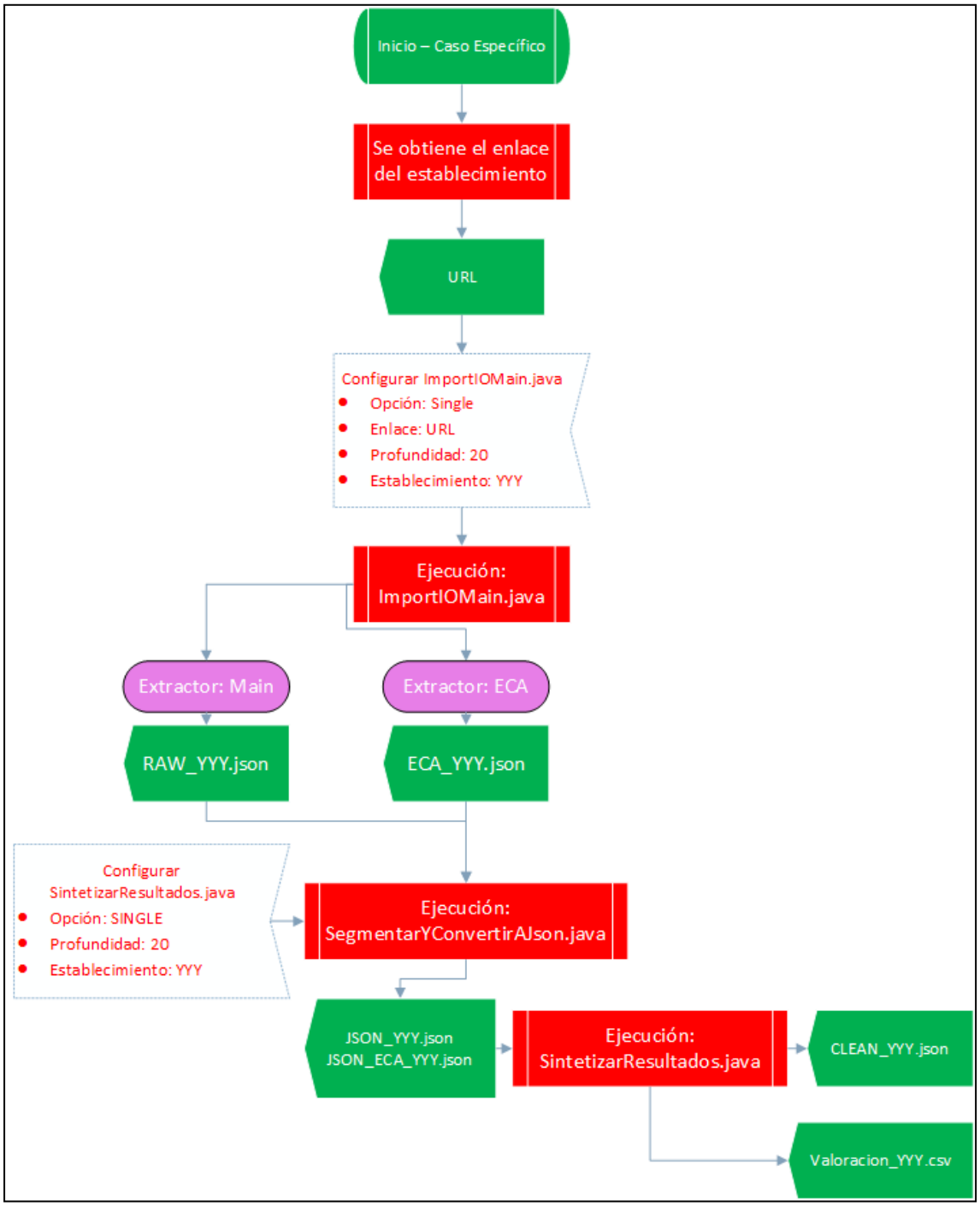


Ilustración 51 – Traza del Caso Específico del Bloque Extracción

7.1.2. Bloque Análisis

Este Bloque trabaja establecimiento por establecimiento (no tendría sentido analizar datos sin denotar claramente a qué establecimiento pertenecen), por lo que bien sea por un **Caso Específico** o por un **Caso General**, se utiliza un solo fichero de entrada.

En este caso, se presume que se dispone de los datos extraídos de un **Caso Específico** (la otra opción hubiera sido disponer de un archivo CSV extraído de la DB de Hadoop).

El fichero **DATA** resultante de este Bloque, si bien ya permite un cierto análisis de datos, incluye mucha información extra innecesaria para los resultados a obtener finalmente, debido a la naturaleza subjetiva del análisis de sentimiento. El siguiente Bloque será el que ejecute los códigos que le darán sentido, en la forma de los archivos **COMP**.

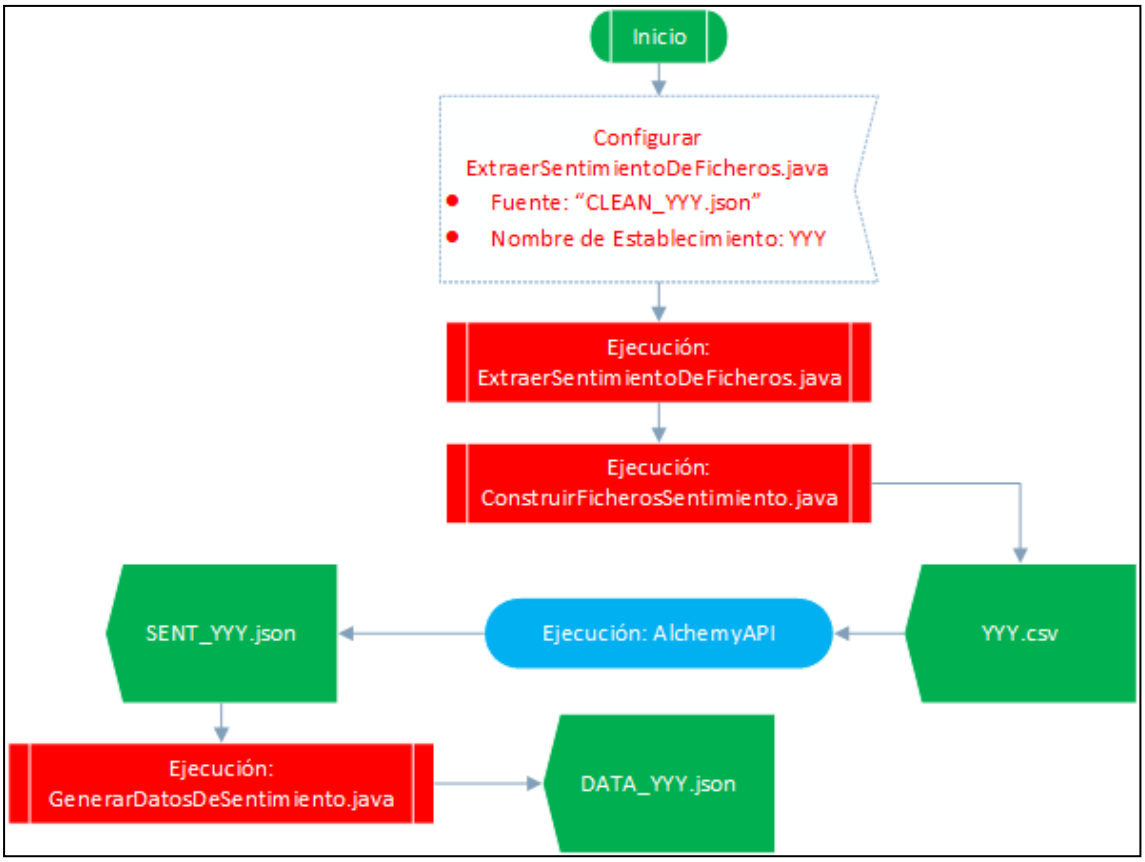


Ilustración 52 – Traza del Bloque Análisis

7.1.2.1 Limitaciones

Este Bloque es el que más limitaciones posee, ya que la licencia gratuita solo permite 1000 peticiones diarias. Estas peticiones se dividen a la mitad si se desea adquirir sentimiento. Por tanto, a la hora de convertir el archivo fuente en un fichero CSV compatible, deberá fragmentarse en archivos de un tamaño reducido, de forma que no se alcance el límite diario de peticiones.

Por tanto, se deberá ejecutar la división de *ConstruirFicherosSentimiento*, en función de que sea un JSON o un CSV, para después ejecutar 1 vez al día *ExtraerSentimientoDeFicheros*.

En la Ilustración 53 se presenta un ejemplo del trabajo que se ha realizado en este Proyecto Fin de Carrera para generar la base de datos utilizada de pruebas.

7.1.3. Bloque Representación

Para el correcto funcionamiento de este Bloque, habrá que considerar una base de datos que almacene los datos obtenidos del **Bloque Análisis**, así como los datos de valoración general adicionales. Por tanto, el primer paso será realizar una base de datos apropiada para después lanzar el menú de opciones al usuario, como se ve en la Ilustración 54.

Es de destacar la opción 6, que filtra la base de datos, modificando el valor mínimo con el que se transforma un fichero **DATA** en un fichero **COMP**. El valor mínimo es el mínimo número de comentarios que deberá haberse realizado sobre un elemento para ser considerado analizable. Aumentar este valor provocará que el sistema ofrezca muchos menos resultados, pero dispondrán de un mayor nivel de detalle y precisión.

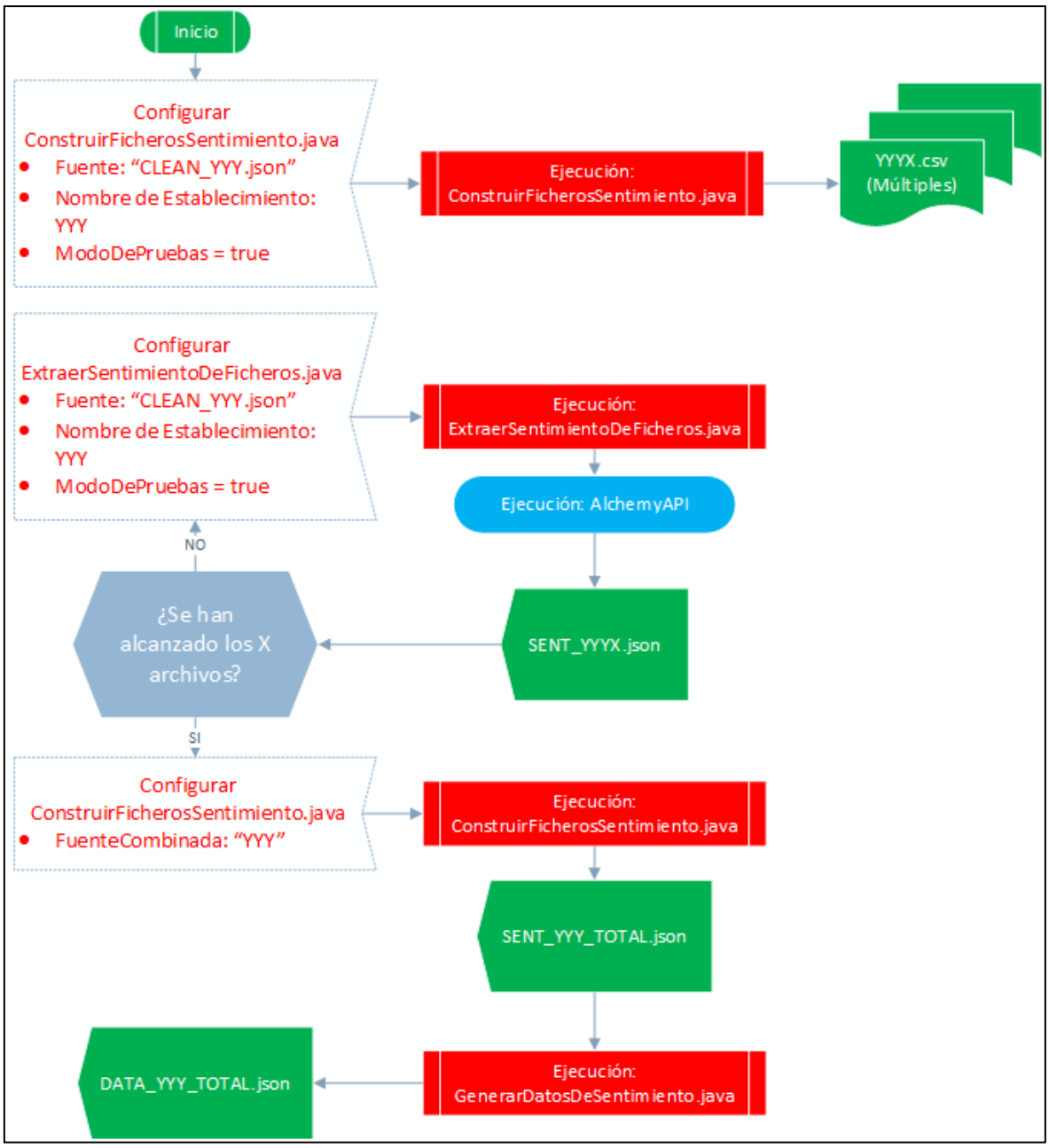


Ilustración 53 – Traza del Bloque Análisis con las limitaciones por licencia

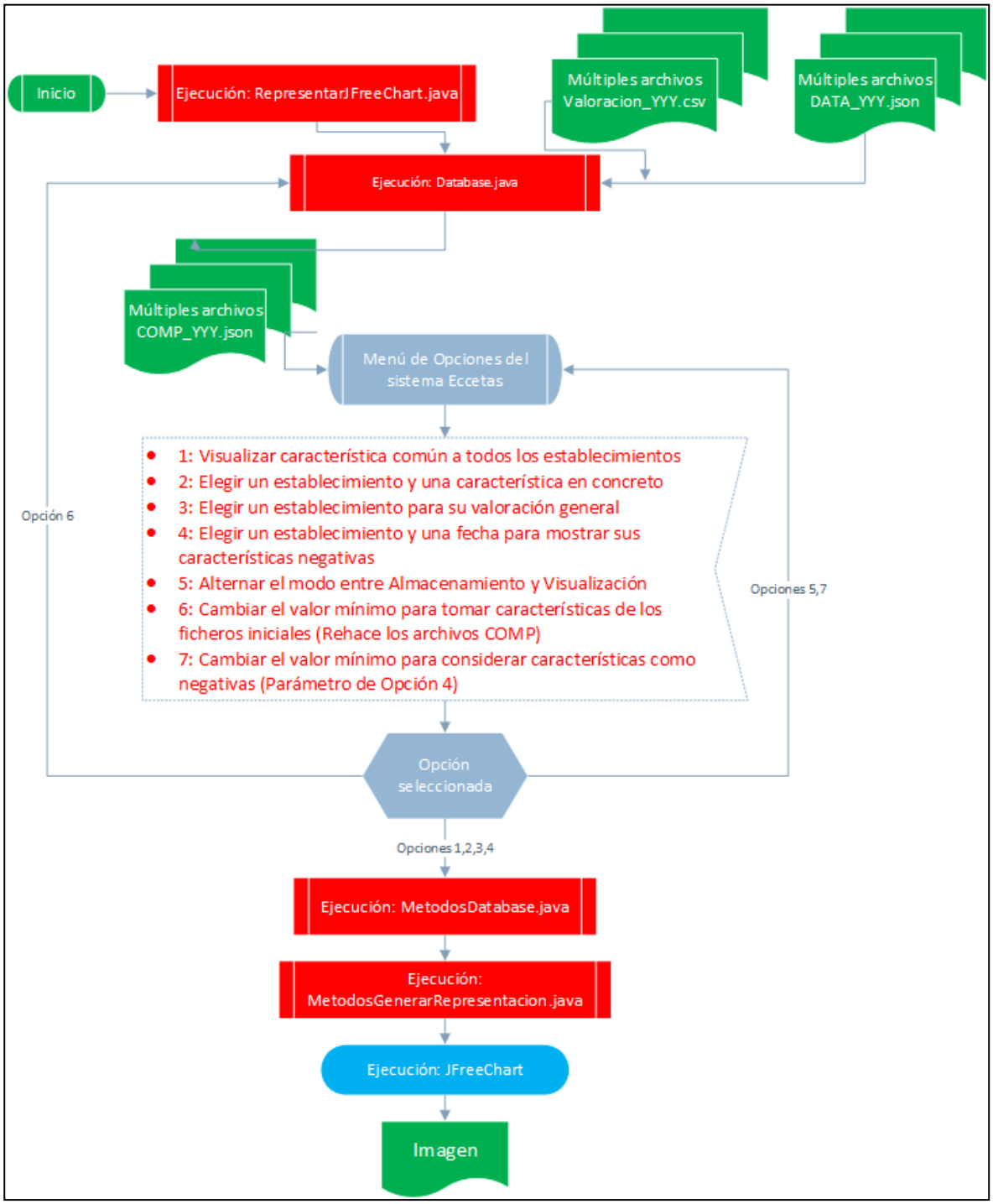


Ilustración 54 – Traza del Bloque Representación

7.1.4. Tipos de Resultados

El resultado principal de Eccetas son los gráficos que genera la suma de los tres bloques mencionados anteriormente. Sin embargo, hay toda una serie de resultados parciales, los cuales se exponen a continuación:

7.1.4.1 Datos "CLEAN"


```
{
    "nombre_establecimiento": "Hotel Europa",
    "estrellas": "3",
    "ubicacion": "Carmen, 4, Madrid City Centre, 28013 Madrid",
    "pais_establecimiento": "Spain",
    "nombre": "Victoria",
    "pais": "United Kingdom",
    "comentarios": "5",
    "fecha": "2016-05-23",
    "detalles": "* Leisure trip * Couple * Double or Twin Room * Stayed 3 nights * Submitted via mobile",
    "valoracion": 8.3,
    "texto_neg": "It could get noisy at night when you're trying to sleep, but that wasn't really a problem for us.
    There was also a lot of work going on Sunday morning, again meaning lots of noise.",
    "texto_pos": "Excellent location, right inbetween Sol and Grand Via. Great value for money."
}
```


Ilustración 55 – Ejemplo de clase DataIO, elemento del fichero CLEAN

- Resultante del **Bloque Extracción**.
- Aporta información general, pero dado que pueden ser miles de comentarios como el mostrado, no resulta fácil de analizar.
- Fácilmente almacenable, y preparado para guardarse en un entorno Hadoop.
- Provee de más información de la utilizada. Por ejemplo, podrían filtrarse los comentarios en función del país de quien lo escribe, o almacenar su valoración general para mostrarla gráficamente.

7.1.4.2 SentimentDataRow

- Objeto Java, representado en la Ilustración 56.
- Resultante del **Bloque Análisis**.
- Aporta información específica sobre una característica de un establecimiento concreto. Su información es interesante, pero es necesario saber cuál *SentimentDataRow* tiene datos relevantes y cual no (en función de su elemento *Quantity*, por ejemplo)
- Almacenable, pero su lista de resultados parciales puede ser complicada de guardar.


```
{
    "name": "Hotel Europa",
    "element": "location",
    "quantity": 314,
    "positive": 292,
    "negative": 7,
    "neutral": 15,
    "rate": "Very Positive",
    "list": [{
        "sentiment": 1.0,
        "fecha": "2016-05-21"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-05-03"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-14"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-10"
    }, {
        "sentiment": 1.0,
        "fecha": "2016-04-10"
    }]
}
```


Ilustración 56 – SentimentDataRow

7.1.4.3 Imágenes

- Resultantes del **Bloque Representación,** mostradas por las Ilustraciones 57, 58, y 59.
- La información que aportan depende del tipo de gráfico y la opción seleccionada. Actualmente, permite:
    - Característica específica de un establecimiento concreto.
    - Característica específica de todos los establecimientos.
    - Todas las características negativas de un establecimiento en función de una fecha indicada.
    - La media de la valoración general sobre un establecimiento en función de los mensajes de los usuarios.
- Muy fácilmente almacenable al tratarse de simples imágenes en formato PNG.

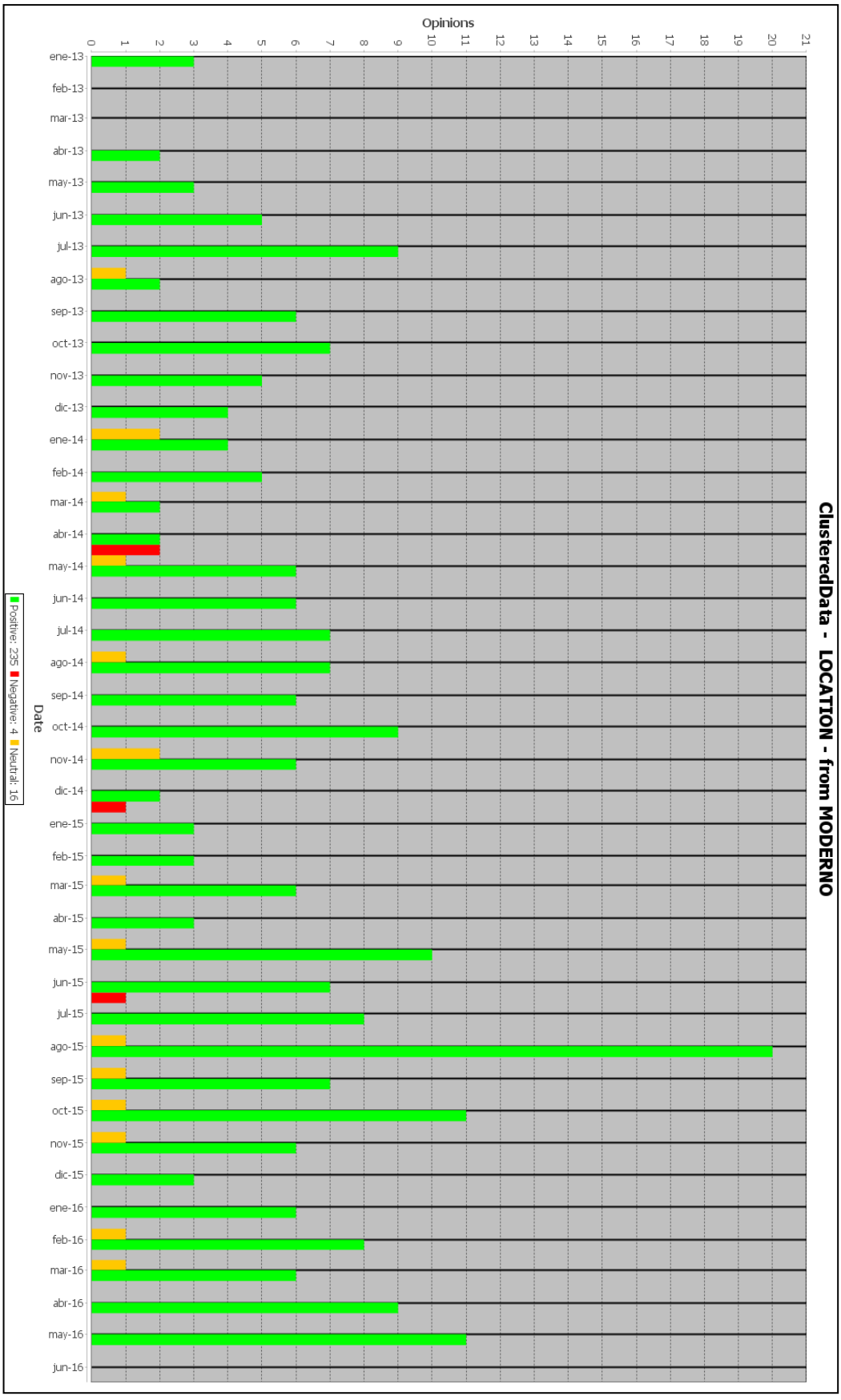


Ilustración 57 – Gráfica ClusteredData

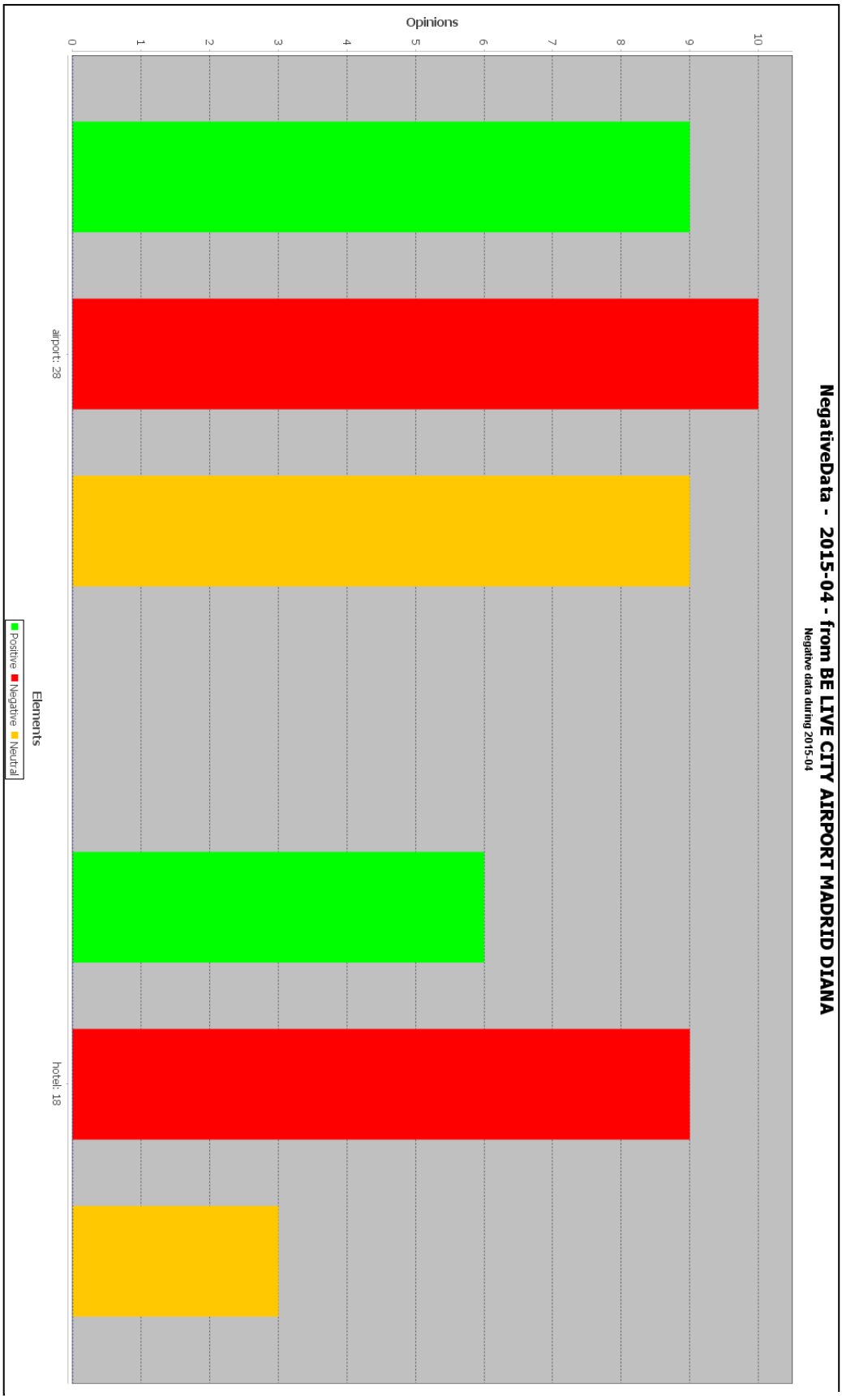


Ilustración 58 – Gráfica NegativeData

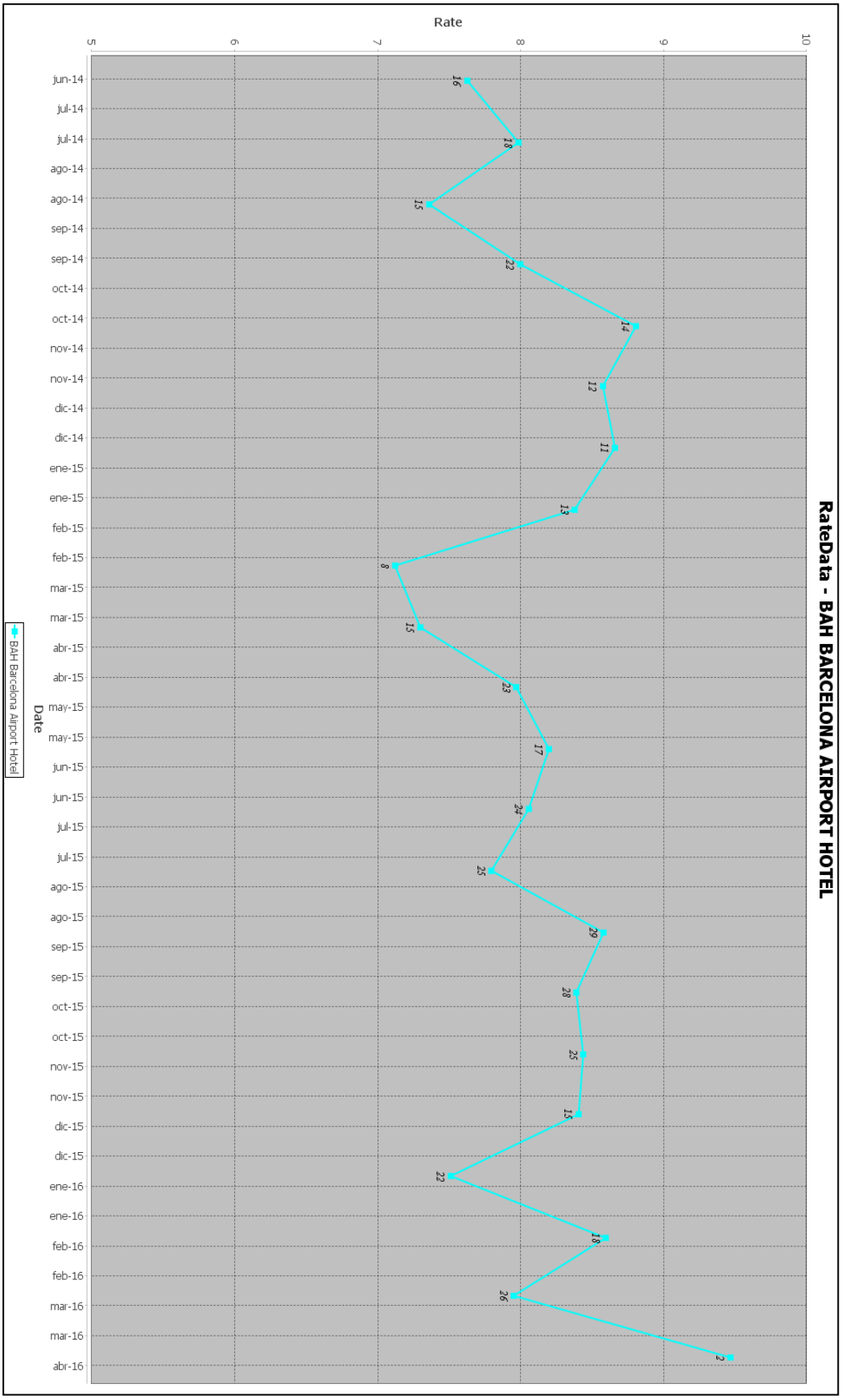


Ilustración 59 – Gráfica RateData

7.1.4.4 Datos "Valoración"

- Resultante del **Bloque Extracción**.
- Aporta las valoraciones generales de cada comentario junto a su fecha correspondiente.
- Fácilmente almacenable, y preparado para guardarse en un entorno Hadoop.
- De por si no aporta demasiada información, pero es muy fácil generar gráficos con esta clase de datos.

```
"fecha","valoración"
"2017–01–30","8.8"
"2017–01–28","7.5"
"2017–01–27","8.3"
"2017–01–21","6.7"
"2017–01–20","6.7"
"2017–01–18","10.0"
```

Ilustración 60 – Ejemplo de datos Valoración

7.2. Análisis de los Resultados

Se decide responder a la siguiente pregunta con el fin de analizar apropiadamente los resultados:

¿Qué puede ofrecer este sistema?

En términos generales, el sistema ofrece al usuario final fundamentalmente gráficas. Estas gráficas son completamente modificables, tanto en la forma de representación de datos como en los valores que muestran. Con las gráficas que se han expuesto anteriormente, por ejemplo, se puede obtener:

7.2.1. Característica específica de un establecimiento concreto

- Conocer el **estado** de la característica. Ejemplos concretos podrían ser: ¿Está bien considerada la habitación? ¿Hay algún problema con la comida? Si una característica posee unos resultados neutrales, será necesario analizarla.
    - Por ejemplo, se presenta la Ilustración 61, donde se analiza la característica "**personal**" del establecimiento **Ciutat del Prat**. Como puede verse, el análisis es fundamentalmente positivo.

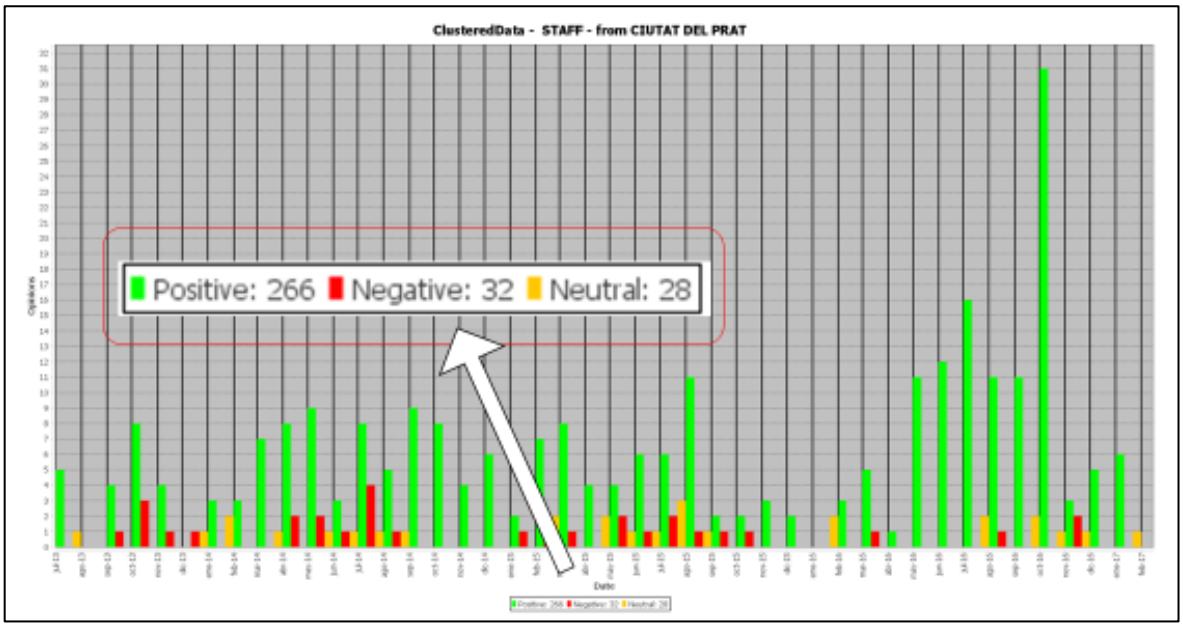


Ilustración 61 – Gráfica ClusteredData "Personal" de Ciutat del Prat

- Estudiar el **histórico** de las opiniones y conocer si dicha característica ha empeorado o mejorado últimamente. De esa forma, por ejemplo, podrían considerarse si otros factores tuvieron que ver al respecto, como el empeoramiento de dicha característica al mismo tiempo que existía un empleado ineficiente al cargo. O, por ejemplo, si las habitaciones son peor valoradas durante el invierno porque no aíslan el calor.
    - En este caso, se analiza la característica "**restaurante**" del establecimiento **Barcelona Airport Hotel**. El histórico de opiniones mostrado en la Ilustración 62 demuestra que, a pesar de tener un estado mayoritariamente negativo, las opciones positivas han ido aumentando, por lo que hay una tendencia a la mejora.
- En este sistema, las características que aparecen son aquellas con un **mayor número de valoraciones** por parte de los clientes. De esta forma, es posible saber qué características del establecimiento son más populares, sea positivamente o negativamente.
    - La Ilustración 63 muestra las características más comentadas de sus respectivos establecimientos. En esencia, hay elementos comunes, como la **localización** o la **habitación**, pero por ejemplo en los

establecimientos cercanos a **aeropuertos**, este suele ser muy mencionado.

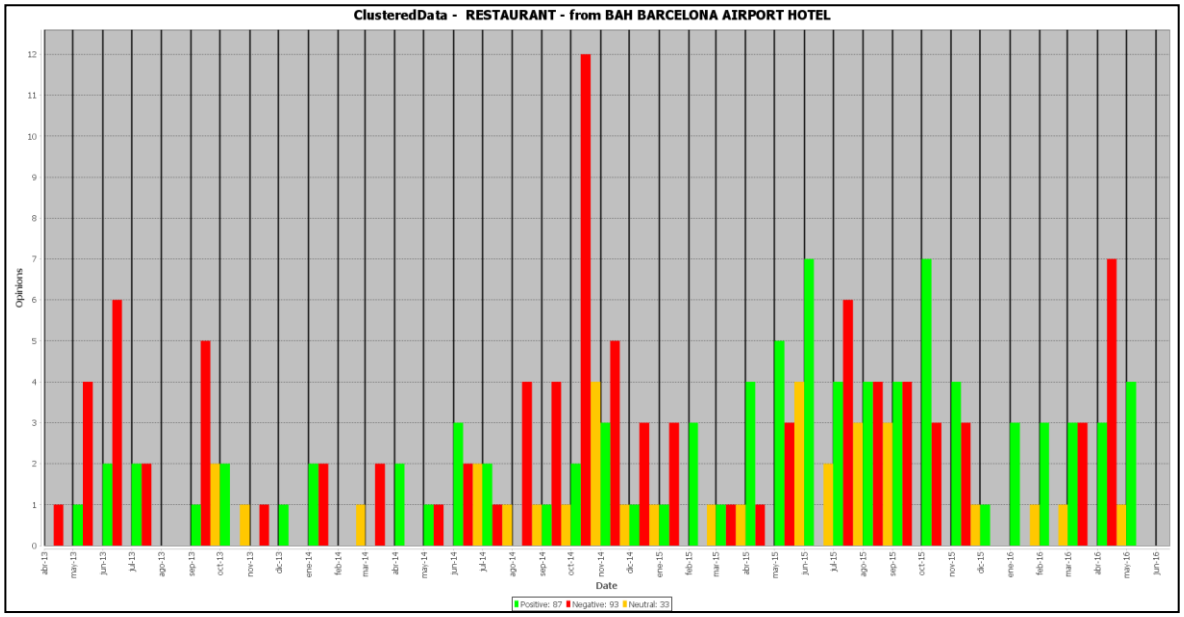


Ilustración 63 – Gráfica ClusteredData "Restaurante" de Barcelona Airport Hotel

Establecimiento: Hotel Europa
Elementos visualizables:
0: location
1: room
2: staff
3: hotel
4: madrid
5: restaurant

Establecimiento: Ciutat del Prat
Elementos visualizables:
0: airport
1: room
2: hotel
3: staff
4: shuttle
5: breakfast
6: location
7: restaurant
8: shuttle service
10: bed

Establecimiento: BAH Barcelona Airport Hotel
Elementos visualizables:
0: airport
1: room
2: hotel
3: staff
4: shuttle
5: location
6: bed
7: breakfast
8: restaurant
9: food
10: shuttle service
11: service
12: shuttle bus
13: time
14: bar
15: bathroom

Ilustración 62 – Características más comentadas junto a sus establecimientos

7.2.2. Característica específica de todos los establecimientos.

- Puede verse si la característica en cuestión está en un mejor o peor lugar que en relación a la **competencia**, aunque será necesario considerar si realmente es equitativo. Comparar la calidad de un establecimiento de máxima categoría con la de uno mucho menor no aporta realmente información; esta clase de comparaciones puede ajustarse en futuras iteraciones del sistema, en función de la base de datos.
- La comparativa en función de la competencia no es aplicable simplemente al **estado**, sino también al **histórico** y a los elementos en **común** que une a todos los establecimientos, como en el caso anterior.

Las Ilustraciones 64, 65, 66 y 67 representan la opinión sobre la característica "**habitación**" de los establecimientos:

- **Be Live City Airport Madrid Diana.**
- **Ciutat Del Prat.**
- **Barcelona Airport Hotel.**
- **Air Rooms Barcelona Airport By Premium Traveler.**

Como puede verse, la comparativa genera el vector de fechas conveniente; de este modo el eje X es idéntico en todos los casos. Así mismo, un vistazo rápido permite obtener datos interesantes, como:

- Ningún establecimiento posee un % bajo de opiniones negativas.
- El primer establecimiento tiene unas opiniones más negativas que el cuarto.
- No hay demasiada información analizable del tercero comparado con el resto.

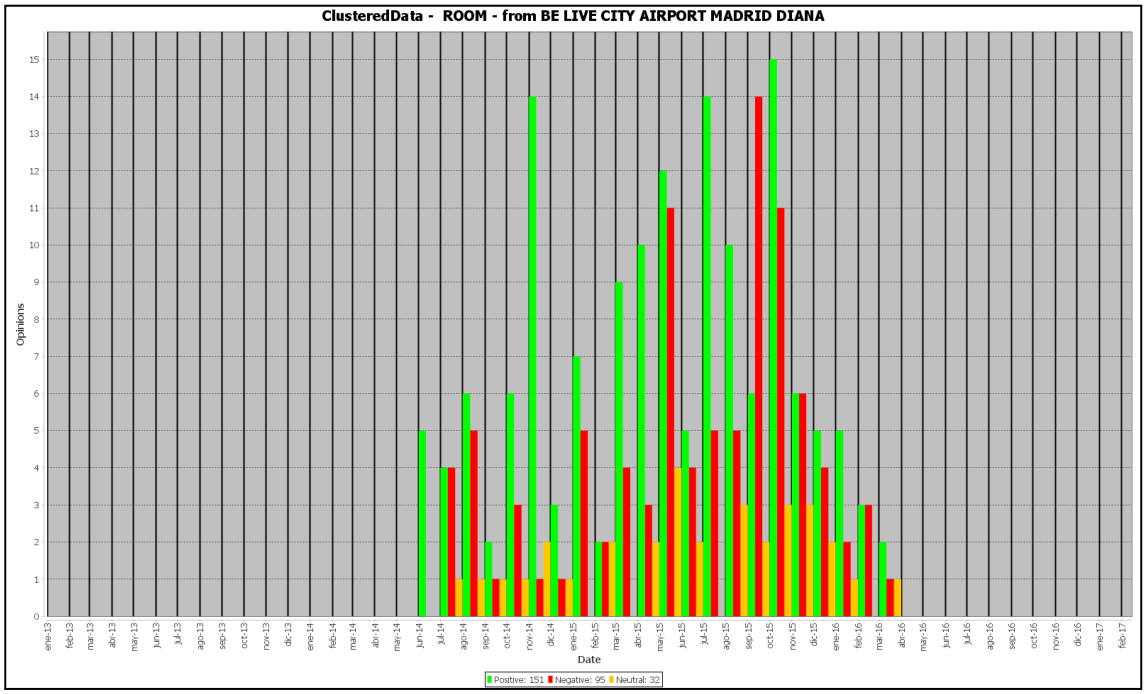


Ilustración 64 – Gráfica ClusteredData "Habitación" de Be Live City Airport Madrid Diana

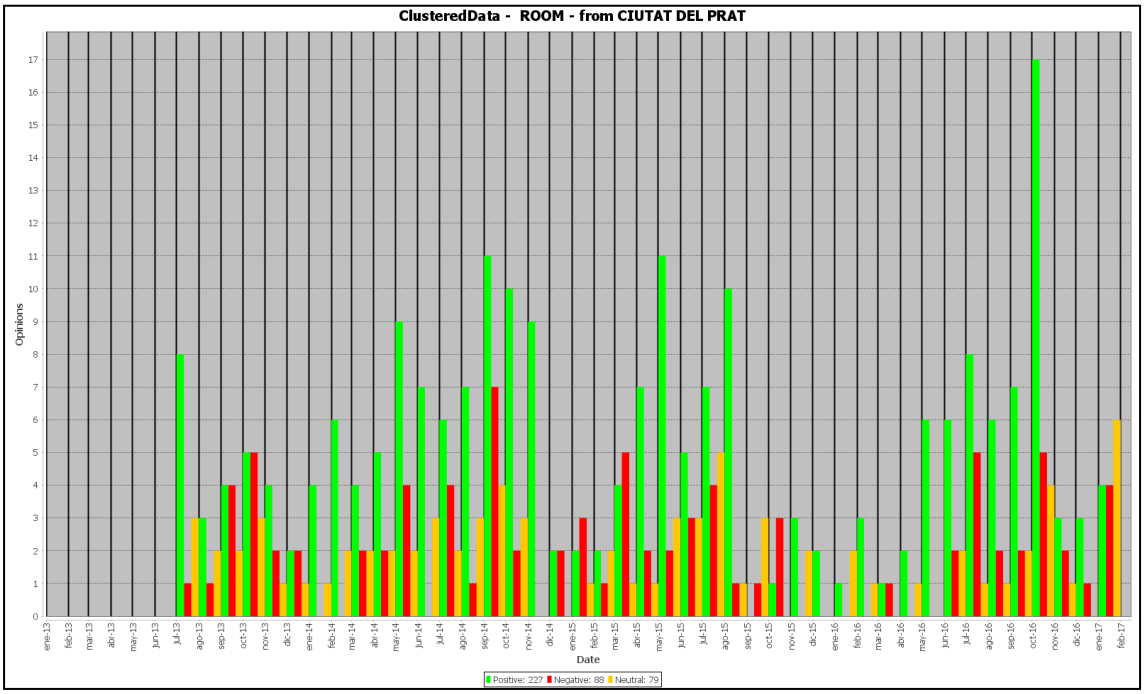


Ilustración 65 – Gráfica ClusteredData "Habitación" de Ciutat del Prat

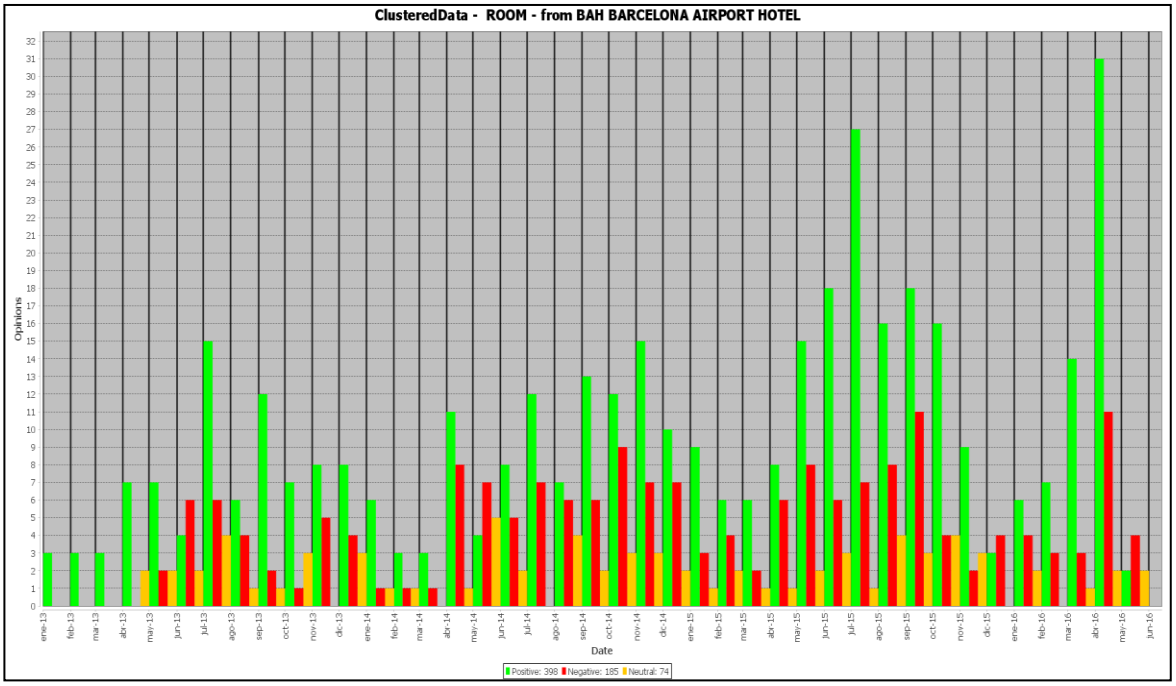


Ilustración 67 – Gráfica ClusteredData "Habitación" de Barcelona Airport Hotel

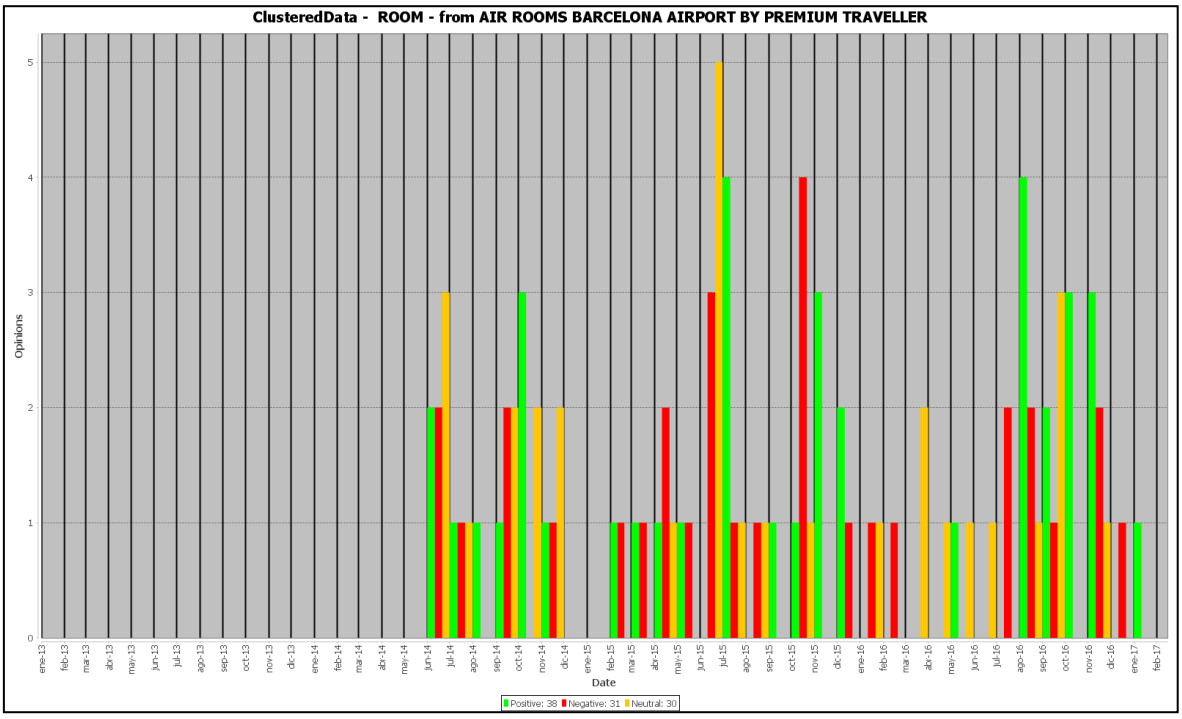


Ilustración 66 – Gráfica ClusteredData "Habitación" de Air Rooms Barcelona Airport By Premium Traveller

7.2.3. Características negativas de un establecimiento en función de una fecha indicada.

Esta modalidad está pensada para el caso en el que se quieran detectar qué características de un establecimiento no son correctas, en función de un **mes y año** concretos. Ello permite ver cuáles son las características que no funcionan en un tiempo determinado, así pudiendo conocer las áreas que requieren revisión sin necesidad de estudiar todas las características que ofrecen las modalidades anteriores.

El primer resultado que se ofrece es la **lista de meses**, en función del año indicado, que poseen datos negativos. Se considerará negativo siempre que la suma de las opiniones neutrales y negativas sea superior a las positivas, y haya más de 10 opiniones en ese mes. Este último número puede aumentarse en pro de obtener una mayor precisión en la representación, aunque sacrificando muchos potenciales resultados.

Por tanto, como se observa en la Ilustración 68, solo será posible obtener imágenes si los datos superan las condiciones indicadas anteriormente, ya que en dicho caso se verán reflejadas como opción a elegir en la columna de la derecha.

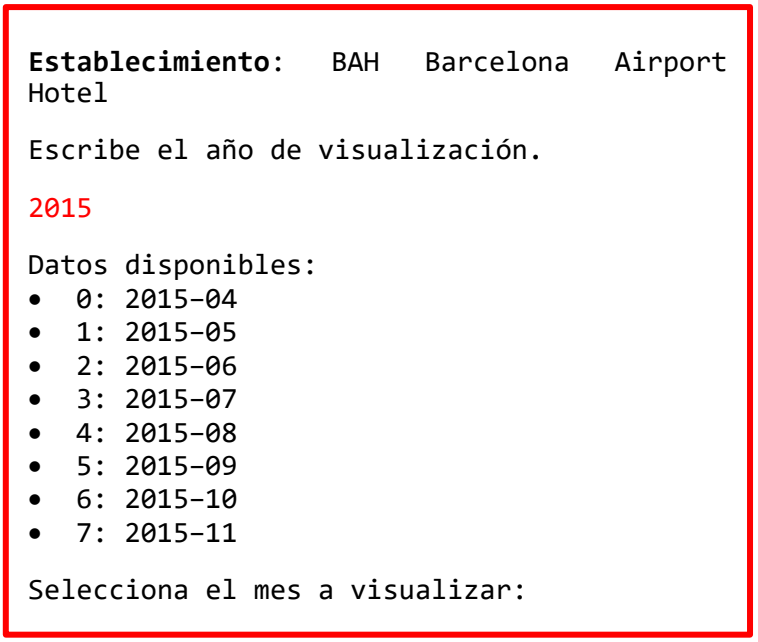


Ilustración 68 – Lista de opciones de visualización

La Ilustración 69 es un ejemplo del resultado de escoger uno de los resultados; en este caso, la característica "**aeropuerto**" del **Barcelona Airport Hotel** es bastante neutral, por lo que posiblemente haya algún aspecto a considerar con respecto a lo que debería ser un establecimiento situado al lado de un aeropuerto, mientras que los datos sobre "**restaurante**", aunque escasos, son directamente negativos.

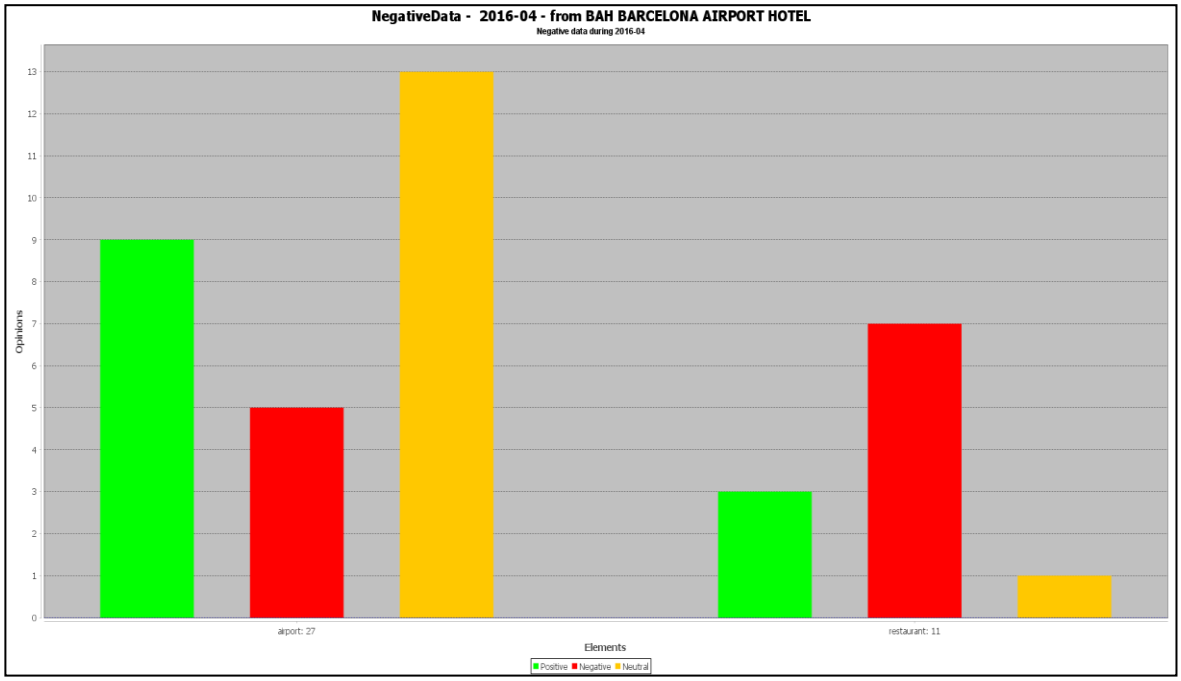


Ilustración 69 – Gráfica NegativeData 04–2016 de Barcelona Airport Hotel

7.2.4. Media de la valoración general sobre el establecimiento.

Esta modalidad secundaria se realizó como ejemplo de la utilidad de extraer más elementos que los comentarios de la fuente. Recoge mes a mes las valoraciones generales que los clientes otorgaron en sus respectivos comentarios, y genera con ellos una **gráfica de valores**. Dicha gráfica permite, por tanto, visualizar un histórico y conocer muy fácilmente si algún aspecto en general no ha sido satisfactorio en un mes y año determinados, y por tanto decidir utilizar los otros tipos de resultado, que ofrecen un análisis más exhaustivo.

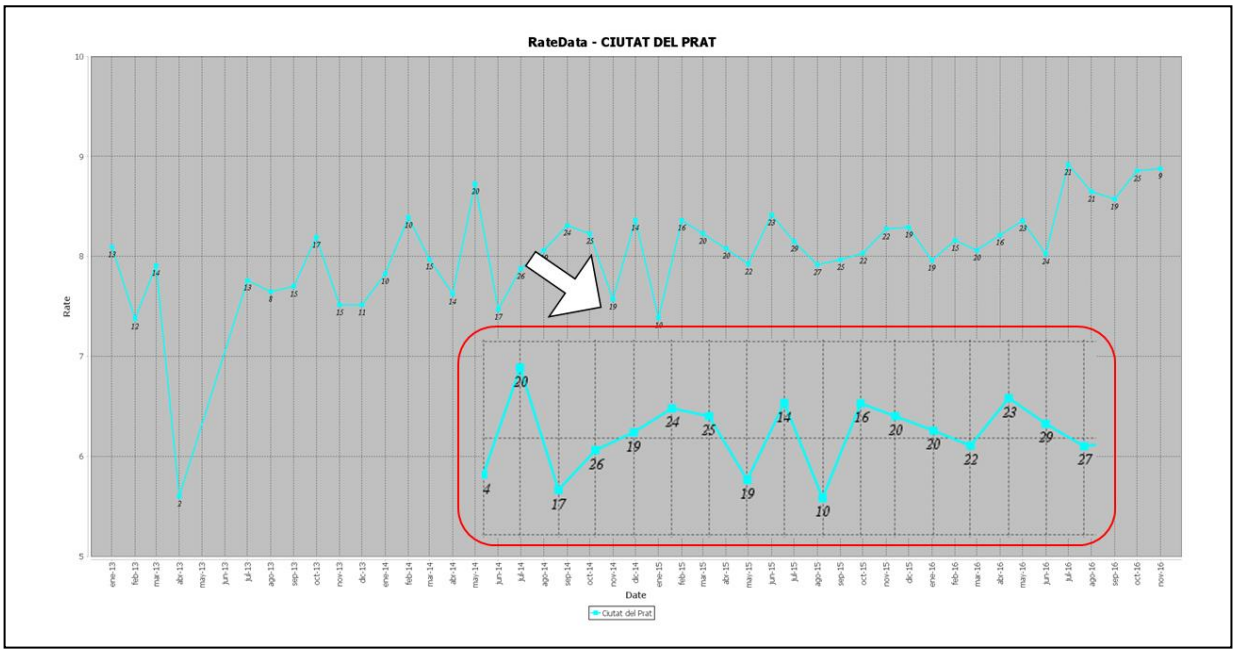


Ilustración 70 – Gráfica RateData de Ciutat del Prat

En la Ilustración 70, se muestra la **valoración** general del establecimiento **Ciutat del Prat**. La valoración se recoge mes a mes, mostrándose en la media que se ve en la imagen, de 5 a 10. El número que acompaña a cada valor indica el número de comentarios que se ha considerado en ese mes.

Capítulo 8: Conclusiones

8.1. Introducción

Como se ha podido observar a lo largo de este Proyecto Fin de Carrera, las nuevas ramas de la tecnología relacionada con los datos tienen el potencial para cambiar sustancialmente el panorama de negocio actual. Desde mejorar cualitativamente las metodologías de trabajo comunes, hasta abrir nuevos nichos de negocio, basándose en el hecho de que las nuevas herramientas proporcionan un conocimiento que, de otra forma, no hubiera sido posible obtener.

El potencial de la **Minería de Datos** es sencillamente abrumador. La cantidad de información y conocimiento que permite obtener de forma relativamente sencilla y rápida se vuelve un arma indispensable en las empresas que ven sus increíbles prestaciones. La capacidad de analizar datos prácticamente en tiempo real, o de establecer patrones de datos útiles con terabytes de información puede dar una ventaja significativa a la hora de competir en el mercado laboral. Junto a este concepto, se han de considerar las herramientas de soporte que ha establecido Big Data, así como las bases de datos no relacionales, utilizar un clúster en paralelo de sistemas en lugar de servirse de un solo servidor, etc... Que son algunas de las formas en las que se puede utilizar la minería de datos apropiadamente.

Por otra parte, la capacidad del **Análisis de Sentimiento** para valorar las opiniones humanas también da paso a numerosas e interesantes opciones. A pesar de la naturaleza conflictiva de las decisiones humanas, la habilidad de poder filtrar y analizar estas valoraciones personales y obtener información de ello se traduce en que las redes sociales son, por ejemplo, unas valiosísimas fuentes de datos. Por ejemplo, en un TFT [Poner enlace] presentado hace unos años, se ilustró el significativo poder que aporta utilizar una herramienta de análisis de sentimiento sobre Twitter, siendo capaces de extraer el sentimiento ante los términos que se desearan analizar.

Finalmente, la apropiada inclusión de estas tecnologías en herramientas que un usuario pueda utilizar con la mayor facilidad posible resulta indispensable, considerando la complejidad general que conllevan. Esto es, independientemente de la dificultad que entrañe la tecnología o la herramienta, es necesario mantener siempre una interfaz sencilla y práctica, permitiendo su utilización y desarrollo.

8.2. **Conclusiones**

Las conclusiones se presentarán tomando como base los objetivos propuestos en el anteproyecto de este Proyecto Fin de Carrera, para pasar a indicar cuáles han sido cumplidos y en qué forma:

8.2.1. Estudio del estado del arte.

Considerando la gran variedad de las herramientas aptas para el caso práctico del extractor y comparador, se deberán buscar aquellas que ofrezcan una mejor funcionalidad, bien por su rapidez, su robustez o su flexibilidad.

Si bien no se ha hecho un extenso análisis de todas las herramientas posibles, como estaba inicialmente planteado, sí se han encontrado las herramientas adecuadas que han permitido la síntesis del sistema Eccetas. Separar la búsqueda y estudio de las herramientas en los Bloques que se han comentado previamente resultó ser muy buena opción de planteamiento, ya que no era necesario buscar herramientas multipropósito, sino resolver una serie de problemas concretos.

8.2.2. Análisis y selección de herramientas especializadas.

Dado que el presente proyecto tiene su núcleo en el análisis de sentimiento, será necesario considerarlo a la hora de realizar la selección de herramientas, también teniendo en mente las áreas de extracción y representación de datos.

Respecto a Big Data, se escogió **Hadoop** y su *Toolbox*. Si bien hay otros conjuntos de herramientas similares, este resultó ser muy intuitivo y cómodo de utilizar, así como notablemente extendido. El hecho de poder acceder a los datos directamente mediante un navegador web facilitó enormemente el manejo de los archivos que se generaron a posteriori. Por tanto, dentro de la *Toolbox* destacaron particularmente *Ambari* (Permite el manejo desde un navegador web), *Hive* (Base de Datos) y *HDFS* para el manejo de ficheros.

Por otro lado, **Import.IO** se escogió por su potencia, efectividad y capacidad de análisis. En las primeras etapas del Proyecto Fin de Carrera, poseía una interfaz distinta, y la herramienta no estaba limitada, lo que hizo posible extraer una gran cantidad de datos. Actualmente, la herramienta es incluso más efectiva, pero no permite la creación de scripts de forma gratuita. Aun así, sigue siendo una opción más que razonable, como se mostró en su correspondiente bloque.

Por otra parte, **AlchemyAPI** responde a la necesidad de analizar texto plano de forma efectiva y automática. Que dispusiera incluso de librerías para varios lenguajes de programación solo la hizo una opción más atractiva, así como el hecho de permitir realizar hasta 1000 operaciones de forma gratuita diarias, lo cual permitía utilizarla para este Proyecto. Otras opciones estaban limitadas en función del día, o no proveían de resultados tan claros.

Finalmente, **JFreeChart** fue empleado por la enorme oferta de gráficas posibles que permitía generar. Aún con una leve dificultad inicial para integrar los datos provenientes de las otras herramientas al formato apropiado, esta resultó ser la herramienta gráfica más efectiva para los propósitos de este Proyecto.

8.2.3. Estudio, recolección y procesado de los datos disponibles en las redes sociales.

Se estudiarán las páginas web de establecimientos turísticos con el fin de tomar ideas iniciales de cuál será la forma más apropiada de obtener el conjunto de datos. Ello dependerá también de las herramientas escogidas en el paso anterior.

Como se comprobó, en función de la estructura de los datos que se presenta será más o menos viable acceder a los datos de forma automática, y en ese caso, es necesario adaptar el extractor de datos a la fuente de origen. La información general puede resumirse en:

- Valoración general
- Fecha

- Comentario

Si bien algunas páginas web también disponen de APIs privadas que permiten una mejor interacción con ellas, habitualmente el uso de estas API está reglado a un acuerdo comercial, o no permiten un uso académico de estos, por lo que finalmente se tomó la opción de **Booking**, debido a que era posible extraer todos los comentarios sin necesidad de inscripciones o el uso de su API.

Para esta tarea, **Hadoop** fue una herramienta muy útil, permitiendo almacenar la mayor parte de los datos almacenados en los archivos extraídos por **Import.IO** en su base de datos, la cual permitía un estudio directo de estos. Si bien esto incluyó algunos requerimientos a la hora de generar el extractor, estos fueron menores en comparación con las ventajas que ofrecía.

8.2.4. Diseño del extractor y comparador.

Disponiendo de una idea de los datos y las herramientas apropiadas, se procederá a la generación del sistema completo, que enlace las herramientas entre sí y permita un uso sencillo por parte del usuario.

Como se ha expuesto a lo largo del presente Proyecto Fin de Carrera, el sistema de Eccetas es una combinación de código Java y las tres herramientas anteriormente mencionadas, y que permite tanto extraer la información como hacer un análisis comparativo de esta, por lo que considera que este objetivo está cumplido.

Inicialmente, se pretendía dar un énfasis mayor al Caso General del Bloque Extracción, con la idea de poder extraer la información de todos los establecimientos del país en múltiples idiomas, e incluso de distintos países, permitiendo así elaborar una enorme base de datos que diera lugar a una efectiva comparativa. A lo largo del Proyecto, el Caso Individual (creado inicialmente solo para pruebas) resultó ser una opción mucho más razonable, de cara al interés del usuario, quien al menos inicialmente, deseará saber su propia valoración, y no la de su entorno.

8.2.5. Utilización del extractor y comparador y análisis de los resultados obtenidos.

Disponiendo de los datos y el sistema completo, se procederá a la obtención de resultados fiables y su representación, de forma que se verifique la utilidad de todo el sistema.

El capítulo de Resultados contiene varios ejemplos que prueban que este objetivo ha sido cumplido. Es posible ampliarlo y mejorarlo, realizando análisis más exhaustivos y obteniendo unos parámetros más ajustados, pero el concepto inicial está superado, incluso habiendo obtenido otros resultados, como:

- La posibilidad de analizar la valoración general de los comentarios.
- Buscar los conceptos negativos de un establecimiento.
- Generar directamente las imágenes de los resultados en archivo, con el fin de extraer todos los resultados cómodamente.

8.3. **Líneas futuras**

- Es posible ampliar la base de datos de forma muy significativa de las siguientes formas:
    - Añadiendo nuevos idiomas (actualmente solo se encuentran el inglés, el alemán y el español).
    - Añadiendo nuevas páginas web (requiere hacer unos cambios en el **Bloque Extracción** y en el Bloque Análisis, pero permitiría ampliar la información de forma muy significativa).
    - Incorporando más países.
- En el caso del **Bloque Análisis**, actualmente se encuentra limitado al idioma inglés. En caso de no poder utilizar más idiomas, una opción sería incorporar un paso anterior que convierta los comentarios al idioma inglés. Se sacrificaría alguna objetividad, pero la cantidad de información a ganar daría lugar a gráficas muy completas.
- Es posible mejorar las funciones del **Bloque Representación**, adaptándolo a los intereses del cliente, suprimiendo o añadiendo más opciones al menú

general de representación. Por ejemplo, cabe añadir la opción de hacer una comparativa solamente con los establecimientos más cercanos (basándose en la dirección y el código postal), o filtrar los resultados en función del país de residencia del autor de los comentarios.

- Finalmente, este sistema dispone de un sencillo menú de consola con el que el usuario puede interactuar. Una opción mucho más razonable y útil sería generar un entorno web que, enlazado con un servidor donde se encuentre almacenada la base de datos, permita su interacción, con usuarios de acceso y la posibilidad de solicitar información adicional, con un coste apropiado.

8.4. **Comentario Final**

Es un hecho que la extracción y el procesado de datos es un área que resulta de gran interés en el Internet de las Cosas, y que solo puede haber progreso y avance tanto en las herramientas que se generan como en los conceptos que estas utilizan. Mejorar la velocidad de procesado, facilitar el aprendizaje en las herramientas, utilizar los resultados para corregir y evolucionar dichas herramientas... Si bien la automatización, que son otra parte fundamental al tratar con tantos datos, puede resultar también peligrosa para los servidores de datos (Un exceso de peticiones provocaría un bloqueo o caída del servidor, potencialmente arruinando todas las ventajas de la automatización).

Por otra parte, la extracción de información no tiene utilidad sin una idea de uso. Sin ella, la información de la que se disponga no aportará conocimiento alguno, al no poder generar un patrón de datos. Este aspecto es mucho más creativo que tecnológico, si bien la mejora en los sistemas propiciará el nacimiento de nuevas ideas, potencialmente permitiendo un extenso crecimiento en el área.

En los sucesivos capítulos de este Proyecto Fin de Carrera, se han llegado a algunas conclusiones específicas, pero en términos generales, lo que se puede extraer de Eccetas y el sistema que ofrece, es que las nuevas tecnologías siguen siendo dependientes del factor humano, y de la necesaria información que este provea. El análisis de sentimiento o la minería de datos no tienen razón de ser sin el aspecto humano, bien sea para ser estudiado, o para ser cuantificado y analizado.

Referencias Bibliográficas

[1] Wikipedia, “Scientia potentia est.” [Online]. Available: https://en.wikipedia.org/wiki/Scientia_potentia_est. [Accessed: 07-May-2017].

[2] ITU, “ITU releases 2014 ICT figures,” *International Telecommunication Union*, 2014.

[3] “World Statistics - International Statistics,” 2015. [Online]. Available: http://world-statistics.org/index-res.php?code=IT.NET.USER.P2?name=Internet users (per 100 people)#top-result. [Accessed: 07-May -2017].

[4] “Minería de datos (data mining). ¿Qué es? ¿Para qué sirve? (1ª parte) (DV00105A).” [Online]. Available: http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=252:mineria-de-datos-data-mining-ique-es-ipara-que-sirve-1o-parte-dv00105a&catid=45:tendencias-programacion&Itemid=164. [Accessed: 07-May-2017].

[5] S. Chalmers, C. Bothorel, and R. Picot-Clemente, “Big Data-State of the Art,” *People*, vol. 3, p. B4, 2013.

[6] I. Herrero, “El análisis de sentimiento de texto en las redes sociales - BiblogTecarios,” 2016. [Online]. Available: http://www.biblogtecarios.es/inmaherrero/el-analisis-de-sentimiento-de-texto-en-las-redes-sociales/. [Accessed: 07-May-2017].

[7] “Java Resources for Students, Hobbyists and More | go.Java | Oracle.” [Online]. Available: https://go.java/index.html. [Accessed: 07-May-2017].

[8] C. Paramio, “El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional,” 2011. [Online]. Available: https://www.genbetadev.com/bases-de-datos/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional. [Accessed: 07-May-2017].

[9] K. Sitto and M. Presser, *Field Guide to Hadoop*. O’Reilly Media, 2015.

[10] M. S. Brown, “(For Dummies) Meta S. Brown-Data Mining For Dummies-Wiley Publishing Inc. (2014).pdf.” 2014.

[11] “Import.io | Extract data from the web.” [Online]. Available: https://www.import.io/. [Accessed: 07-May-2017].

[12] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New Avenues in Opinion Mining and Sentiment Analysis,” *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 15-21, Mar. 2013.

[13] M. Ogneva, “How Companies Can Use Sentiment Analysis to Improve Their Business (vendor),” 2010. [Online]. Available: http://mashable.com/2010/04/19/sentiment-analysis/#eVCqsvJDn5qT. [Accessed: 07-May-2017].

[14] “AlchemyLanguage | IBM Watson Developer Cloud.” [Online]. Available:

https://alchemy-language-demo.mybluemix.net/. [Accessed: 07-May-2017].

[15] "JFreeChart." [Online]. Available: http://www.jfree.org/jfreechart/. [Accessed: 07-May-2017].

[16] T. White, "Hadoop: The definitive guide 4th Edition," *Online*, vol. 54, p. 258, 2012.

[17] "Hadoop Tutorial - YDN," *Yahoo! Developer Network*. [Online]. Available: https://developer.yahoo.com/hadoop/tutorial/module2.html. [Accessed: 07-May-2017].

[18] "Download Apache Hadoop Sandbox on Virtual Machine or Azure | Hortonworks." [Online]. Available: https://es.hortonworks.com/products/sandbox/. [Accessed: 07-May-2017].

[19] "Hive Plays Well with JSON | Mawazo." [Online]. Available: https://pkghosh.wordpress.com/2012/05/06/hive-plays-well-with-json/. [Accessed: 07-May-2017].

[20] "How-to: Use a SerDe in Apache Hive - Cloudera Engineering Blog." [Online]. Available: http://blog.cloudera.com/blog/2012/12/how-to-use-a-serde-in-apache-hive/. [Accessed: 07-May-2017].

[21] Stanford NLP Group, "The Stanford Natural Language Processing Group." [Online]. Available: https://nlp.stanford.edu/software/tagger.shtml. [Accessed: 07-May-2017].

[22] GNU, "GNU General Public License v3.0," 2007.

Códigos

9.1. Introducción

Este capítulo sirve para enumerar y ofrecer un breve resumen de los distintos algoritmos que se han desarrollado para la constitución del sistema Eccetas.

Se han omitido de la siguiente lista aquellos algoritmos provisionales o desarrollados a modo de prueba (si bien en los dos primeros bloques existe un modo de pruebas incorporado en los propios algoritmos para la correcta generación de datos de este proyecto). Se han dividido en los respectivos Bloques de diseño, omitiéndose la descripción de los códigos cuya única función es describir las clases Java necesarias.

9.2. Códigos Implementados

9.2.1. Extracción

9.2.1.1 GenerarNumEstIO.java

Código que obtiene el número de establecimientos totales que existen en el país fijado y en el idioma seleccionado, llamando a un determinado script de **Import.IO**. Con ese número, genera la lista de enlaces que requiere el siguiente código para funcionar.

Limitaciones: Al ser una única llamada al *Extractor*, no está limitado.

9.2.1.2 Extractor Número_establecimientos

Este programa de **Import.IO** analiza una página web de Booking y extrae el número de establecimientos totales.

9.2.1.3 GenerarEnlacesEstIO.java

Código que analiza todos los enlaces proveídos por el software anterior con otro script de **Import.IO** y genera la lista de enlaces a comentarios de establecimientos que utilizará el resto del Bloque.

Limitaciones: Si el modo de pruebas está activo, solo llama 3 veces al script, independientemente del número de resultados posibles.

9.2.1.4 Extractor Enlaces comentarios

Este programa generado en **Import.IO** permite obtener los enlaces de comentarios de los distintos establecimientos de Booking, filtrados por país. Al solo mostrar 15 resultados por página, será necesario ejecutar este Extractor múltiples veces.

9.2.1.5 GenerarEcaIO.java

Código que genera un objeto ECA (Establecimiento, País, Dirección), llamando al correspondiente script de **Import.IO**, necesario para el siguiente programa.

Limitaciones: Al ser una única llamada al *Extractor*, no está limitado.

9.2.1.6 Extractor ECA

Programa de **Import.IO** que devuelve el nombre, el país y la dirección (incluye código postal) del enlace al establecimiento indicado.

9.2.1.7 ImportIOMain.java

El código principal del Bloque utiliza todos los códigos mencionados anteriormente (el último solamente en los casos individuales) para generar el archivo principal de datos *RAW*, utilizando otro script de **Import.IO** y guardando todos los resultados en un fichero compatible con **Hadoop**.

Limitaciones: Si el modo de pruebas está activo, solo llama 3 veces al *Extractor Main* y al *Eca*, independientemente del número de resultados posibles.

9.2.1.8 Extractor Main

Programa de **Import.IO** que devuelve:

- Nombre del autor del comentario
- Comentario (partes positiva y negativa)
- Fecha del comentario
- País de procedencia del autor
- Número de comentarios realizados en Booking
- Detalles (parámetro de Booking que indica el nº de noches, si iba solo o en pareja, etc)
- Valoración general

Dado que devuelve un máximo de 75 resultados, en casos donde existan más será preciso ejecutar este programa más de una vez.

9.2.1.9 SegmentarYConvertirAJson.java

Permite adaptar el resultado *RAW* anterior a un formato *JSON* válido que puedan leer otros elementos del Bloque y que detecte si el fichero es demasiado grande para dividirlo en archivos que sí sean procesables.

9.2.1.10 SintetizarResultados.java

Aplica al fichero anterior *JSON* una serie de condiciones y transformaciones para corregir posibles errores de escritura o formato, de forma que genera un fichero *CLEAN* formado con los nuevos objetos DATAIO, cuya cabecera es el objeto ECA mencionado anteriormente. Este fichero es compatible con **Hadoop**, así como con el siguiente Bloque.

También extrae aparte las valoraciones generales y sus fechas, formando los archivos *Valoración*, que podrán utilizarse directamente en el último Bloque de este sistema.

9.2.2. Análisis

9.2.2.1 ConstruirFicherosSentimiento.java

Este programa posee como objetivo principal convertir la fuente de datos que se le indique en un archivo compatible con la herramienta **AlchemyAPI** para obtener el análisis de sentimiento. Por tanto, bien provenga la fuente de **Hadoop** (fichero CSV con los datos buscados), o del **Bloque Extracción** (fichero JSON con todos los datos), el programa copiará, filtrará y formateará convenientemente los datos para el siguiente código.

Limitaciones: Con el modo de pruebas activo, se dividirá la fuente de datos en ficheros de un tamaño compatible con el límite diario que exige la herramienta. Una vez analizadas todas las partes, el modo de pruebas permite combinarlas en un solo elemento, que será compatible con el último código de este Bloque.

9.2.2.2 ExtraerSentimientoDeFicheros.java

Este programa toma el fichero que contenga los datos y el nombre del establecimiento del usuario, y genera un archivo SENT con el nombre del establecimiento y una lista de SentimentElement (elemento, fecha, sentimiento), que se procesarán en el siguiente bloque. Estos “elementos” son las posibles características del establecimiento que se está analizando.

Limitaciones: Si el modo de pruebas está activo, el sistema reconoce que no analiza un solo archivo sino múltiples, por lo que analizará el fichero parcial y generará un archivo SENT parcial. Será necesario volver al anterior programa una vez se realice el análisis de todos los ficheros parciales.

9.2.2.3 GenerarDatosDeSentimiento.java

Este programa, similar al proceso que realiza *SintentizarResultados.java*, toma los datos SENT previos y genera una serie de nuevos objetos SentimentDataRow, que se almacenarán en el fichero DATA. Estos objetos se generan mediante la agrupación de los distintos SentimentElement mencionados anteriormente, así como del resto de procedimientos explicados en el Bloque Análisis.

Este programa también dispone de la capacidad de filtrar los ficheros DATA en los ficheros COMP, que serán en última instancia los que utilizará el bloque siguiente. Los ficheros COMP son los SentimentDataRow de los ficheros DATA cuyo atributo Quantity supere el umbral dictado por el usuario.

9.2.3. Representación

9.2.3.1 Database.java

Este programa genera la base de datos que se utilizará en el programa principal de este Bloque. Utiliza los ficheros DATA que se le indiquen, permitiendo generar los respectivos ficheros COMP. Así mismo, también carga los archivos Valoración generados por el primer Bloque.

9.2.3.2 MetodosDatabase.java

Este código trabaja sobre la base de datos, extrayendo los datos que sean necesarios, actualizándolos o filtrándolos. Incluye:

- Analizar todos los ficheros COMP y mostrar qué características son comunes.
- Extraer todos los SentimentDataRow de la base de datos en función de la característica indicada.
- Extraer todas las características de la base de datos en función de una fecha indicada.
- Extraer las fechas comunes de toda la base de datos de cada elemento (Genera el vector de fechas que engloba toda la base de datos)

- Extraer la fecha común de un elemento de un solo establecimiento.
- Obtener las SentimentDataRow cuya característica haya sido valorada como negativa.
- Generar la valoración general de toda la base de datos.

9.2.3.3 MetodosGenerarRepresentacion.java

Este código permite generar correctamente los Chart y los Dataset necesarios para JFreeChart, concretamente:

- Genera el Chart y Dataset "LastData", que muestra las características negativas en función de una fecha.
- Genera el Chart y Dataset "ClusteredData", que muestra los resultados positivos, negativos y neutrales de una característica.
- Genera el Chart y Dataset "RateData", que muestra la valoración general del establecimiento, agrupando las valoraciones en meses.

9.2.3.4 RepresentarJFreeChart.java

El programa principal del Bloque, utiliza todos los códigos anteriores para proveer al usuario de las gráficas que son el resultado final de todo el sistema Eccetas. Cargando la base de datos y utilizando los métodos generados para ella, permite al usuario mostrar las gráficas que desee, modificar la base de datos, o cambiar parámetros del programa (valor Quantity para filtrar la base de datos, alternar entre guardar las gráficas o solo representarlas, etc).

9.2.4. Almacenamiento

9.2.4.1 Script para ficheros RAW

Este script, que se utiliza en **Hive**, permite incorporar a la base de datos que se encuentra en **Hadoop** los ficheros RAW generados por el **Bloque Extracción**.

9.2.4.2 Script para ficheros Clean

De forma análoga al anterior apartado, también se dispone de otro script para añadir los ficheros Clean. De esta forma, podrán analizarse o filtrarse convenientemente para el **Bloque Análisis**, particularmente si son el resultado del **Caso General**.

9.2.4.3 Script para ficheros DATA

Este script permite almacenar los SentimentDataRow agrupados en un fichero DATA. Para poder controlar los resultados parciales que contiene es necesario crear una estructura adicional que dificulta notablemente su uso, pero que permite su correcto almacenamiento.

Pliego de Condiciones

10.1. Introducción

El documento en el que se hallan las cláusulas que regulan los derechos, responsabilidades, garantías mutuas y obligaciones entre los diferentes agentes asociados al proyecto, es conocido como el pliego de condiciones. Recoge las exigencias de índole técnica y legal que han de regir la ejecución del proyecto, teniendo efectos vinculantes.

Por tanto, el siguiente apartado se ha dividido en:

- **Condiciones técnicas** – Aquellas condiciones necesarias para la correcta instalación y ejecución de los programas desarrollados en el proyecto. Se comentan los recursos materiales y herramientas necesarias.
- **Condiciones legales** – Aquellas condiciones o pautas legales en relación al desarrollo, aplicación y distribución del proyecto. Se tratan los derechos de autor, la licencia, las restricciones o garantías, y otros temas de interés.

10.2. Pliego de condiciones técnicas

10.2.1. Requisitos mínimos

10.2.1.1 Recursos Hardware

- Procesador > i5. Se recomienda un procesador i7.
- Memoria RAM > 8Gb. Se recomiendan 16 Gb.
- Monitor o pantalla. Se recomienda una resolución superior a 1024x728.
- Teclado y ratón.
- Conexión a Internet.

Varios de los requisitos anteriores se cubren mediante un ordenador portátil o de sobremesa que disponga de las características mencionadas.

10.2.1.2 Recursos Software

- Sistema operativo mínimo: Microsoft Windows XP.
- Versión mínima herramienta software: Eclipse 4.2 Juno. (Verificado su funcionamiento en la versión 4.6 Neón).
- Adobe Acrobat Reader para leer los ficheros complementarios del directorio del DVD.
- Base de datos incluida en el DVD.
- Toolbox Hadoop HDP versión 2.4.
- VirtualBox versión 5.1.8.
- Librerías Java:
    - Commons–IO–2.4.
    - Gson–2.6.2.
    - Java–sdk–3.0.0–RC1–jar–with–dependencies (Librería de AlchemyAPI).
    - Jcommon–1.0.23.
    - Jfreechart–1.0.1.9.
    - Opencsv–3.7.
    - Standford–postagger–3.6.0.
        - Slf4j–api.
        - Slf4j–simple.
- REST API de Import.IO.
- Se recomienda la instalación de un procesador de texto y hojas de cálculo, en caso de que sea necesario revisar los datos parciales generados por el código.

10.2.2. Instalación y ejecución del software

El software desarrollado a lo largo de este Proyecto Fin de Carrera puede englobarse en dos áreas distintas:

10.2.2.1 Ejecución del código Eccetas.

El proyecto Java puede importarse al Workspace de cualquier entorno Eclipse, siempre que se cumplan los requisitos hardware y software mencionados anteriormente. No es preciso obtener las librerías Java indicadas, puesto que estas ya se encuentran insertadas en el proyecto.

Como se ha expresado en anteriores capítulos, los códigos se encuentran separados en sus respectivos Bloques, y podrán ejecutarse de forma separada, siendo el bloque Representación el principal para el usuario.

El presente proyecto ya incorpora una serie de archivos iniciales que funcionan como base de datos, por lo que no es necesario utilizar la Toolbox de HDP para hacer pruebas iniciales. En caso de suprimirlos o modificarlos, el sistema no dispondrá de datos que presentar y provocará un mal funcionamiento en el código.

10.2.2.2 Instalación y ejecución de la máquina virtual para utilizar Hadoop.

Desde la página web de Hortonworks se ofrece una solución gratuita que adjunta las instrucciones adecuadas para ejecutar la máquina virtual, y cargar sobre esta el Toolbox HDP. Ejecutada, podrá accederse desde un navegador web tanto a HDFS, que almacena los datos, como Hive, que permite realizar búsquedas en formato SQL.

10.3. Pliego de condiciones legales

10.3.1. Concesión de licencia

Este Proyecto Fin de Carrera es propiedad de la Universidad de las Palmas de Gran Canaria y cualquier usuario debe estar de acuerdo en obligarse por los términos y condiciones establecidas en esta licencia del proyecto aceptando todas sus cláusulas. El uso de los distintos programas o ficheros, o de una copia en cualquier ordenador o computadora sólo podrá darse bajo la autorización expresa del autor, el tutor del proyecto y de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria.

10.3.2. Derechos de autor

Este proyecto y su documentación asociada están protegidos por las leyes de propiedad intelectual que le sean aplicables así como las disposiciones de los tratados internacionales. Por consiguiente, el usuario deberá utilizar el material relativo al proyecto como cualquier producto protegido por derechos de autor. Sin embargo, se permitirá que el usuario pueda realizar una copia de los códigos fuente de programación y de la documentación del proyecto siempre que exista una autorización previa del autor, el tutor del proyecto y de la Escuela de Ingeniería de Telecomunicación y Electrónica perteneciente a la Universidad de Las Palmas de Gran Canaria.

10.4. **Restricciones**

El usuario no podrá realizar ingeniería inversa, de compilación o desensamblado del proyecto. El usuario podrá transferir el programa a un tercero bajo la autorización del autor al tutor o de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria siempre que no posea copia del mismo. Asimismo, la copia transferida podrá incluir posibles actualizaciones que las pueda acompañar.

10.5. **Garantía**

El autor y el tutor garantizan que las muestras y ficheros asociados al proyecto funcionarán correctamente en el momento de su uso. Análogamente, se

garantiza que el soporte en el cual estén grabados los distintos programas no contendrá defectos en el momento de la adquisición del mismo.

La única excepción de lo dispuesto en el párrafo anterior es que los programas están creados sin garantías de ninguna clase. El autor y el tutor no aseguran, garantizan o realizan ninguna declaración respecto al uso o los resultados derivados de la utilización de los ficheros o de la documentación asociada al proyecto.

10.5.1. Limitación de responsabilidad

En ningún caso serán el autor, el tutor o la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria responsables de los perjuicios directos, indirectos, incidentales o consiguientes, gastos, lucro cesante, pérdida de ahorros, interrupción de negocios, pérdida de información comercial o de negocio, o cualquier otra pérdida que resulte del uso o de la incapacidad de usar los ficheros o la documentación del proyecto. El usuario conoce y acepta que los derechos de licencia reflejan esta asignación de riesgo como el resto de cláusulas y restricciones. Asimismo, el autor y el tutor de este proyecto rechazan cualquier otra garantía que no haya sido indicada anteriormente.

10.6. **Otros**

En el supuesto de que cualquier disposición de esta licencia sea declarada total o parcialmente inválida, la cláusula afectada será modificada convenientemente de manera que sea ejecutable una vez modificada, plenamente eficaz, permaneciendo el resto de este contrato en vigencia y regido por las leyes de España.

Finalmente el usuario acepta la jurisdicción exclusiva de los tribunales de este país en relación con cualquier disputa que pudiera derivarse de la presente licencia.

Presupuesto

11.1. Declaración jurada

Don Néstor Marín Siruela, autor del presente Proyecto Fin de Carrera,

DECLARA QUE:

El proyecto Fin de Carrera con título "Extractor y Comparador de Características de Establecimientos Turísticas empleando Análisis de Sentimientos con Big Data", realizado a petición de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria y en un periodo de 12 meses, tiene un coste total de **TREINTA Y TRES MIL SETECIENTOS SESENTA Y CUATRO EUROS CON SETENTA Y OCHO CÉNTIMOS (33764,78 €)** correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

Firmando la presente para que así conste a los efectos oportunos.

Autor del Proyecto:

Néstor Marín Siruela

Las Palmas de Gran Canaria, a 07 de Mayo de 2017

11.2. Desglose del presupuesto

El presupuesto del proyecto realizado se ha generado según los precios de mercado actual, y de las indicaciones del COIT (Colegio Oficial de Ingenieros de Telecomunicación) y de la AEIT (Asociación Española de Ingenieros de Telecomunicación), a efectos de visado.

Los conceptos en los que se reparte el presupuesto son los siguientes:

1. Amortización del inmovilizado material.
    a. Amortización del material hardware empleado.
    b. Amortización del material software empleado.
2. Trabajo tarifado por tiempo empleado.
3. Redacción del proyecto.
4. Derechos de visado del COIT.
5. Gastos de tramitación y envío.
6. Aplicación de impuestos.

11.2.1. Amortización del inmovilizado material

Se trata de la corrección de valor por la depreciación del inmovilizado material del proyecto realizada de acuerdo con un plan sistemático definido con anterioridad.

Normas de valoración

El inmovilizado material se ha registrado a su coste de adquisición. No se han capitalizado ningún tipo de gastos financieros. No se han incurrido en gastos de reparación, conservación o mantenimiento.

A esta categoría pertenece la amortización del hardware y del software empleado en la realización del Proyecto Fin de Carrera. El sistema de amortización empleado ha seguido el método lineal, que distribuye el coste de los activos entre los años de vida útil de los mismos de forma constante. Asimismo, dichos recursos están ubicados en la categoría de “Útiles, herramientas y equipos para tratamiento de la información, sistemas y programas informáticos” de las tablas de amortización propias del I.R.P.F.

La amortización se practicará elemento por elemento, si bien cuando se trate de elementos patrimoniales integrados en el mismo grupo de la tabla de amortización, la amortización podrá practicarse sobre el conjunto de ellos, siempre que en todo momento pueda conocerse la parte de la amortización correspondiente a cada elemento patrimonial.

Asimismo, los elementos de inmovilizado material nuevos cuyo valor unitario no exceda de 601,01 euros, podrán amortizarse libremente, hasta el límite de 3.005,06 euros anuales.

Según la tabla "*Útiles, herramientas y equipos para tratamiento de la información, sistemas y programas informáticos*", el número de años mínimos y máximos de amortización en esta categoría es de entre 2.5 y 5 años, por lo que para este proyecto, se ha estipulado en 3 años para cada uno de los elementos de tipo hardware empleados.

Teniendo en cuenta que la duración del proyecto ha sido aproximadamente de 12 meses, y sabiendo que el cálculo del coste de amortización se constituye en un periodo de 3 años, los costes de amortización de la mayoría de los recursos utilizados se calcularán para el primer año.

Finalmente, la cuota de amortización anual será calculada haciendo uso de la Ecuación 1:

$$Cuota\ de\ amortización\ anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Nº\ de\ años\ de\ vida\ útil}$$

Ecuación 1 – Cuota de amortización anual

Donde el valor residual es el valor de cada uno de los elementos en cuestión después de su vida útil, teniendo en cuenta los índices de depreciación actual.

11.2.2. Amortización del material hardware

Puesto que la elaboración del proyecto ha precisado de 12 meses de trabajo y el cálculo del coste de amortización se estipula en un periodo de tres años, los

costes serán calculados como los derivados del tiempo de utilización que se ha requerido por cada uno de los elementos hardware.

En la siguiente tabla se muestra la relación del elemento hardware con su valor de adquisición, su valor residual y el coste de amortización finalmente obtenido.

Elementos Hardware	**Coste**	**Valor residual (3 años)**	**Amortización**
Ordenador de sobremesa. Procesador AMD FX–8350, 16Gb RAM, 1Tb, 2 monitores tipo LED.	1050 €	0	350 € (1 año)
TOTAL			**350 €**

Tabla 1 – Amortización del material hardware

Por tanto, el coste total del hardware empleado en el proyecto asciende a la cantidad de TRESCIENTOS CINCUENTA EUROS.

11.2.3. Amortización del material software

De forma análoga, en este apartado se procederá a calcular la amortización del inmovilizado material de tipo software del proyecto.

Elementos Software	**Coste**	**Valor residual (3 años)**	**Amortización**
Paquete Microsoft Office 365 Pro Plus.	155 €	0	51,67 € (1 año)
Sistema Operativo Microsoft Windows 10 Pro, 64 bits.	279 €	0	93 € (1 año)

Eclipse Java Juno 4.2	0€	0	0€
Notepad++ 7.3.3	0€	0	0€
Google Drive	0€	0	0€
Toolbox Hadoop Hortonworks 2.4	0€	0	0€
TOTAL			**144,67 €**

Tabla 2 – Amortización del material software

Por tanto, el coste total del software empleado ha sido de CIENTO CUARENTA Y CUATRO CON SESENTA Y SIETE CÉNTIMOS.

11.2.4. Trabajo tarifado por tiempo empleado

Normas de valoración

La aproximación del importe de las horas empleadas en la realización del presente proyecto con respecto a los honorarios finales estimados, está indicada en la fórmula de recomendación del COIT mostrada en la Ecuación 2:

$$H = 74{,}88 \cdot H_N \cdot C_T + 96{,}72 \cdot H_E \cdot C_T \; euros$$

Ecuación 2 – Fórmula de recomendación del COIT

En donde:

- H son los honorarios por tiempo.
- H_N son las horas trabajadas dentro de la jornada laboral.
- H_E son las horas especiales trabajadas.
- C_T es un factor de corrección en función del número de horas trabajadas.

Asimismo, el valor del factor de correlación según el número de horas trabajadas vendrá dado por la Tabla 3:

HORAS TRABAJADAS	FACTOR DE CORRECCIÓN *CT*
Hasta 36 horas	1,0
De 36 a 72 horas	0,90
De 72 a 108 horas	0,80
De 108 a 144 horas	0,70
De 144 a 180 horas	0,65
De 180 a 360 horas	0,60
De 360 a 540 horas	0,55
De 540 a 720 horas	0,55
De 720 horas a 1080 horas	0,45
Más de 1080 horas	0,40

Tabla 3 – Factor de corrección de los honorarios

Para la realización del presente proyecto, considerando que no se ha realizado una dedicación completa, se estima que se ha trabajado 1200 horas laborales (5 horas x 5 días x 4 semanas x 12 meses) y 0 horas especiales. Por tanto, se considerará **0,4** como factor laboral.

De igual forma, habrá que descontar un número de horas asociadas a los periodos vacacionales o festivos de Navidades, Carnaval y Semana Santa que han afectado a la jornada laboral anteriormente definida. Como consecuencia, y teniendo en cuenta que dichos periodos vacacionales han acumulado un total de 168 horas para este caso, la cantidad total de horas efectivas trabajadas en jornada laboral ha sido de: 1200 – 168 = 1032 horas.

Por lo tanto, los honorarios asociados al total de horas trabajadas en este proyecto han sido:

$$H = 74{,}88 \cdot 1032 \cdot 0{,}40 + 0 \cdot 72 \cdot 0 = 30910{,}46\ €$$

Ecuación 3 – Fórmula del trabajo tarifado por tiempo empleado

Obteniéndose una tarifa final por tiempo de ejecución de TREINTA MIL NOVECIENTOS DIEZ EUROS, CON CUARENTA Y SEIS CÉNTIMOS.

Siendo el desglose por tiempo y fase del proyecto el que se muestra en la Tabla 4:

Descripción	Tiempo	Coste/mes	Importe
Formación	2	2575,87€	5151,74 €
Documentación	2		5151,74 €
Diseño	3		7727,61 €
Desarrollo	5		12879,35 €
TOTAL	**12 meses**		**30910,46 €**

Tabla 4 – Trabajo tarifado por tiempo empleado desglosado

11.2.5. Derechos de visado

El COIT establece que para la redacción de proyectos y trabajos en general, los derechos de visado se calculan de acuerdo con la siguiente expresión:

$$V = 0{,}006 \cdot P \cdot C$$

Ecuación 4 – Fórmula de derechos de visado

Donde P representa el presupuesto total y C es el coeficiente reductor en función de dicho presupuesto.

El presupuesto total se obtiene de la suma de las secciones anteriores correspondientes al trabajo tarifado por tiempo empleado, la amortización del inmovilizado material y la redacción del proyecto, como se observa en la Tabla 5:

CONCEPTO	COSTE
Recursos Hardware	350 €
Recursos Software	144,67 €
Trabajo tarifado por tiempo empleado	30910,46 €
TOTAL	**31405,13 €**

Tabla 5 – Costes de las herramientas y del tiempo empleado

En función del presupuesto obtenido, se extrae el valor del coeficiente reductor del presupuesto C, que según el COIT, para presupuestos de más de 30.050€ y menos de 90.150€ viene definido con un valor de 0,8.

Por tanto:

$$V = 0{,}006 \cdot 31405{,}13 \cdot 0.8 = 150{,}74\ € \ (Ec.5)$$

Ecuación 5 – Coste total de los derechos de visado

Finalmente, los costes por derecho de visado del proyecto ascienden a CIENTO CINCUENTA EUROS CON SETENTA Y CUATRO CÉNTIMOS.

11.2.6. Gastos de tramitación y envío

Los gastos de tramitación y envío son fijos y se estipulan por el COIT en 6,01€ por cada documento en un visado digital.

Por ello, los gastos de tramitación y envío ascienden a SEIS EUROS Y UN CÉNTIMO.

11.2.7. Presupuesto antes y después de impuestos

Sumando todos los conceptos calculados hasta el momento, se obtiene el total del presupuesto previo a la aplicación de impuestos, como se muestra a continuación en la Tabla 6.

CONCEPTO	COSTE
Amortización del material hardware	350 €
Amortización del material software	144,67 €
Trabajo tarifado por tiempo	30910,46 €

empleado	
Derechos de visado del COIT	150,74 €
Gastos de tramitación y envío	6,01 €
TOTAL (Antes de impuestos)	**31555,87 €**
Aplicación de Impuestos (7% de IGIC)	2208,91 €
TOTAL (Después de impuestos)	**33764,78 €**

Tabla 6 – Presupuesto total del proyecto

El presupuesto calculado antes de impuestos asciende a TREINTA Y UN MIL QUINIENTOS CINCUENTA Y CINCO EUROS CON OCHENTA Y SIETE CÉNTIMOS, y el presupuesto después de impuestos es un total de TREINTA Y TRES MIL SETECIENTOS SESENTA Y CUATRO EUROS CON SETENTA Y OCHO CÉNTIMOS.

Las Palmas de Gran Canaria, a 07 de Mayo de 2017

Fdo: Néstor Marín Siruela

Anexo 1.

Contenido del DVD–R

12.1. Introducción

Junto con esta memoria se adjunta un DVD–R en el que se recopila el trabajo realizado a lo largo de este Proyecto Fin de Carrera.

El contenido de este DVD–R es el siguiente:

- Memoria en formato PDF.
- Funciones y scripts implementadas en Java.
- Scripts implementados en Hive.
- Base de datos con todos los elementos extraídos.

12.2. Descripción del contenido

Se ha organizado el contenido en las siguientes carpetas: Memoria, Eccetas, Scripts Hive, Base de Datos.

- **Memoria**: Documentación de la memoria de este proyecto. También se incluye la bibliografía, el pliego de condiciones, el presupuesto y los anexos.
- **Scripts Hive**: Contiene unos breves scripts que prueban la forma en la que es posible importar los resultados parciales de Eccetas en la base de datos de Hadoop.
- **Base de Datos**: Contiene todos los datos que se han extraído, ordenados en función de ser datos de un idioma completo, o datos de un establecimiento. En este último caso, además, su información estará contenida en una carpeta propia.
- **Eccetas**: Esta carpeta consta del proyecto Java, preparado de forma que pueda implementarse directamente a un entorno para su ejecución. Dentro, contiene las librerías necesarias para su ejecución, así como varios elementos de la base de datos para su prueba directa, no siendo necesario añadir ningún archivo.