



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Big Data para el análisis de sentimientos en imágenes

TÍTULO DEL TFG: Big Data para el análisis de sentimientos en imágenes

TITULACIÓN: Grado en Ingeniería Telemática

AUTOR: Marc Ballesteros Díez

DIRECTOR: Esther Salamí San Juan

FECHA: 12 de septiembre del 2017

Título: Big Data para el análisis de sentimientos en imágenes

Autor: Marc Ballesteros Díez

Director: Esther Salamí San Juan

Fecha: 12 de septiembre del 2017

Resumen

Durante los últimos años se está escuchando mucho los términos Big Data y minería de datos, por lo que en este proyecto se explicarán qué son, en qué se diferencian, su importancia en la actualidad y se verán alguna de las herramientas que se implementan con estos dos términos.

También, estos últimos años se está incrementando el uso de las redes sociales, dónde los usuarios dan su opinión sobre cualquier tema y además pueden colgar cualquier archivo multimedia, ya sea un video o una imagen. Las grandes compañías son capaces de hacer cualquier tipo de predicción y extraer información de cualquier texto que los usuarios hayan escrito, mediante el uso del Big Data y la minería de datos. Pero, ¿se podría extraer información de las imágenes que suben los usuarios para así saber su opinión sobre un tema?

En este proyecto, se realizará una página web, que simule una red social, dónde mediante el uso de hashtags y de imágenes se estime el sentimiento que provoca un tema en concreto. Para ello, primero será necesario realizar un análisis de las imágenes, concretamente aquellas en las que se detecten caras, para determinar su sentimiento. Este análisis se realizaría mediante el uso de una tecnología de Big Data y de minería de datos, con la cual la aplicación obtendrá un patrón con el que pueda determinar el sentimiento de la cara. El análisis de los datos que se extraigan de las imágenes seguirá un conjunto de pasos que se realizan en todo proyecto de minería de datos, gracias a estos pasos se podrán realizar modificaciones en los datos o en la configuración de la herramienta de Big Data y de minería de datos para mejorar los resultados. En cuanto a la aplicación web, se realizará un diseño, y se explicarán las tecnologías que serán necesarias para su creación.

Finalmente, se verá el resultado final, dónde se mostrará la aplicación web con la extracción del sentimiento de la imagen, consiguiendo así alcanzar el objetivo principal del proyecto.

Title: Big Data for sentiment analysis in images

Author: Marc Ballesteros Díez

Director: Esther Salamí San Juan

Date: September 12 th 2017

Overview

During the last years the terms Big Data and data mining are being listened to a lot, so in this project they will be explained that they are, in which they differ, their importance at present and will see some of the tools that are implemented with these two terms.

Also, in recent years is increasing the use of social networks, where users give their opinion on any subject and also can upload any multimedia file, either a video or an image. Big companies are able to make any kind of prediction and extract information from any text that users have written, using Big Data and data mining. But, could it extract information from the images that users upload to know their opinion on a topic?

In this project, a web page will be created, which simulates a social network, where through the use of hashtags and images, the page can know the feeling that causes a specific topic. To do this, it will be necessary to perform an analysis of the images, specifically those in which faces are detected, to determine their feelings. This analysis would be done using a technology of Big Data and data mining, with which the application will obtain a pattern with which can determine the feeling of the face. The analysis of the data extracted from the images will follow a set of steps that are carried out in any data mining project, thanks to these steps, it can be possible made modifications to the data or the configuration of the Big Data tool and data mining to improve results. As for the web application, a design will be made, and will explain the technologies that will be necessary for its creation.

Finally, you will see the final result, where the web application will be displayed with the extraction of the image sentiment, thus achieving the main objective of the project.

ÍNDICE

| | |
|---|-----------|
| CAPÍTULO 1. INTRODUCCIÓN | 1 |
| 1.1 Motivación | 1 |
| 1.2 Objetivos | 1 |
| 1.3 Organización del documento | 2 |
| CAPÍTULO 2. BIG DATA Y DATA MINING..... | 3 |
| 2.1 ¿Qué es el Big Data?..... | 3 |
| 2.2 ¿Qué es el Data Mining?..... | 3 |
| 2.3 Data Mining VS Big Data..... | 6 |
| 2.4 ¿Qué los hace tan importantes?..... | 7 |
| 2.5 Herramientas de Big Data y Data Mining. | 7 |
| CAPÍTULO 3. ANÁLISIS DE SENTIMIENTOS..... | 11 |
| 3.1 Tecnologías..... | 11 |
| 3.2 Definición del problema..... | 13 |
| 3.3 Preparar los datos | 13 |
| 3.4 Explorar y modificar los datos..... | 18 |
| 3.5 Generar, explorar y evaluar modelos | 22 |
| 3.6 Implementar y ejecutar modelos..... | 33 |
| CAPÍTULO 4. APLICACIÓN WEB | 34 |
| 4.1 Tecnologías..... | 34 |
| 4.2 Diseño | 37 |
| 4.3 Implementación y resultado | 40 |
| CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO..... | 45 |
| BIBLIOGRAFÍA | 48 |

CAPÍTULO 1. INTRODUCCIÓN

1.1 Motivación

En los últimos años, se está haciendo mucho el uso del Big Data y la minería de datos para todo tipo de cosas: para saber los gustos de las personas, agruparlas y así poder dar recomendaciones según los gustos que tengan en común; también para la estimación de precios, a partir de los datos que se han ido acumulando, se es capaz de analizar cuanto podría valer un producto según unos campos que se introduzcan; y también en el análisis del texto, donde las máquinas a día de hoy son capaces de saber si lo que se está diciendo es positivo o negativo, si está bien escrito o no, y hasta son capaces de traducir el texto. Pero, ¿qué ocurre con el análisis en imágenes?

Hoy en día las grandes compañías como Microsoft, con Microsoft Cognitive Service [1], y Google, con Google Cloud Vision [2], están dedicando muchos esfuerzos al análisis de sentimientos a partir de imágenes. Durante este proyecto, se trabajará en la clasificación de imágenes según si expresan sentimiento de felicidad o tristeza, con el objetivo de introducirnos en el proceso de Big Data y minería de datos con imágenes al igual que se realiza con los textos o los precios.

Desde hace unos cuantos años, se está haciendo uso exhaustivo de las redes sociales, como Facebook, Twitter e Instagram, donde la gente cuelga sus fotos y dan su opinión de distintos temas. Para muchas empresas podría resultar atractivo conocer la opinión que tienen una persona sobre algún tema a partir de las imágenes que cuelgue. En este proyecto se intentará extraer información de los sentimientos que tengan las imágenes que contengan caras, y a su vez conseguir saber la opinión que tengan los usuarios de una página Web sobre algún tema en concreto.

1.2 Objetivos

A grandes rasgos, el objetivo principal del proyecto es el desarrollo de una aplicación web que utilice tecnologías actuales de Big Data para el análisis de imágenes. Concretamente la aplicación permitirá subir una imagen y nos dirá si una persona que aparece en la imagen expresa un sentimiento de alegría o de tristeza.

Para ello, se han planteado los siguientes objetivos más específicos:

- Comprender qué son y para qué se utilizan los términos Big Data y Data Mining.
- Analizar las diferentes tecnologías existentes e implementar al menos una para el análisis de imágenes.

- Realizar un estudio de las imágenes y conseguir extraer información de ellas.
- Conseguir predecir los sentimientos de las caras a partir de la información que se haya extraído en el estudio de estas.
- Utilizar técnicas de Big Data y minería de datos para el desarrollo de una aplicación Web.
- Conseguir realizar análisis de los sentimientos que provoca un tema en una página Web, con la ayuda de hashtags y de las imágenes.

1.3 Organización del documento

El resto del documento se divide en cuatro partes:

- Primero, en el capítulo Big Data y Data Mining, se explicará qué son, para qué sirven, en qué se diferencian las dos, la importancia que tienen y por último se podrán ver algunas de las tecnologías que las implementan.
- Después, en el capítulo de análisis de sentimientos, se expondrán las tecnologías que se utilizarán para su realización, y se seguirán los pasos que todo proyecto basado en la minería de datos (Data Mining), deben de cumplir mostrando cada uno de los pasos las cosas que se debieron realizar para la consecución de la extracción del sentimiento de las caras que aparecerán en las imágenes.
- A continuación, en el capítulo de la aplicación web, se mostrarán las tecnologías que se utilizarán, el diseño que deberá seguir la página, y por último su implementación y resultado final, dónde se verá cómo ha quedado la página.
- Finalmente, se realizarán las conclusiones que se hayan extraído a lo largo del proyecto, y se propondrán posibles mejoras para futuros proyectos.

CAPÍTULO 2. BIG DATA Y DATA MINING

En los últimos años se está oyendo mucho el termino Big Data y Data Mining (o minería de datos en español), cuya implementación se está realizando en todo tipo de campos, como por ejemplo en la medicina, la economía o en la industria del motor, entre otros. Por eso, en este apartado, se explicará que son y que es lo que los hace importante hoy en día.

2.1 ¿Qué es el Big Data?

El termino Big Data hace referencia a la cantidad de datos, los cuáles superan la capacidad del software, provocando que no puedan ser capturados, administrados y procesados en poco tiempo. Con el paso de los años esto se ha ido incrementando, cada vez hay más dispositivos en la red y cada vez se transmiten más datos, por lo tanto, se está almacenando un gran volumen de datos [3].

Dentro del término Big Data, podemos encontrar tres tipos:

- Datos estructurados: Son aquellos que tienen bien definidos la longitud de su campo, además de su tipo. Estos son los datos que se almacenan en tablas.
- Datos no estructurados: Son aquellos datos que son almacenados tal y como fueron almacenados, es decir, en este caso los datos carecen de un formato específico.
- Datos semiestructurados: Son los datos que se almacenan sin tener los campos determinados, esto se puede hacer gracias a que contienen algún marcador que permiten separar los distintos elementos.

2.2 ¿Qué es el Data Mining?

El Data Mining o en español la minería de datos es el proceso de extraer información de grandes conjuntos de datos. Esto lo consigue, gracias a la implementación de análisis matemáticos para deducir los patrones y tendencias que existen en los datos. Estos patrones, normalmente, no se pueden extraer mediante un análisis tradicional, ya que estos datos pueden tener relaciones complejas o porque hay una gran cantidad de datos [4].

La realización de la minería de datos sigue un proceso, entre los cuáles se encuentran la definición del problema hasta la solución de este mediante el uso de un modelo. Este proceso sigue los siguientes pasos:

- **Definir el problema:** Este primer paso, consiste en la definición del problema y el estudio de distintas formas de utilización de los datos para dar una respuesta al problema.

- **Preparar los datos:** En este segundo paso, se seleccionarán y limpiarán los datos que se analizarán.

- **Explorar y modificar los datos:** El tercer paso, consiste, como su nombre indica, en la exploración y modificación de los datos. En este paso, se realizarán diferentes estudios de los datos que se hayan analizado. En este paso, se pueden observar si a los datos seleccionados se les debe realizar alguna transformación para mejorar el resultado.

- **Generar modelos:** En el cuarto paso, se generará el modelo o modelos de data mining. En este paso se deberán de seleccionar que columnas de datos se van a utilizar. La generación de modelos permitirá obtener un patrón de los datos, el cual se deberá entrenar para poder extraer la precisión de este modelo. Este patrón puede variar según que datos se hayan elegido, que algoritmo se ha escogido y cómo se haya configurado el algoritmo.

- **Explorar y validar modelos:** El penúltimo paso, consiste en la exploración de los modelos que se han generado y comprobar la eficacia de estos. Antes de implementar el modelo, se aconseja probarlo para evaluar si funciona correctamente. Si en la evaluación del modelo se obtiene un mal resultado, se deberá repetir alguno de los pasos anteriores.

Para validar los modelos que se hayan creado, y así poder escoger el más adecuado se encuentran tres tipos de medidas [5]:

- **Precisión (Accuracy):** Esta medida indica la exactitud del resultado con los atributos que se le hayan proporcionado.

- **Confiabilidad:** Esta medida evalúa la manera en la que un modelo se comporta con un conjunto de datos diferentes, es decir, un modelo será confiable si se genera el mismo tipo de predicciones independientemente de los datos que se le pasen.

- **Utilidad:** Esta medida incluye diferentes métricas que le permiten indicar si el modelo está proporcionando información útil.

- **Implementar y actualizar modelos:** Este último paso, consiste en la implementación del modelo que mejor resultado se haya obtenido en el anterior paso, en un entorno de producción.

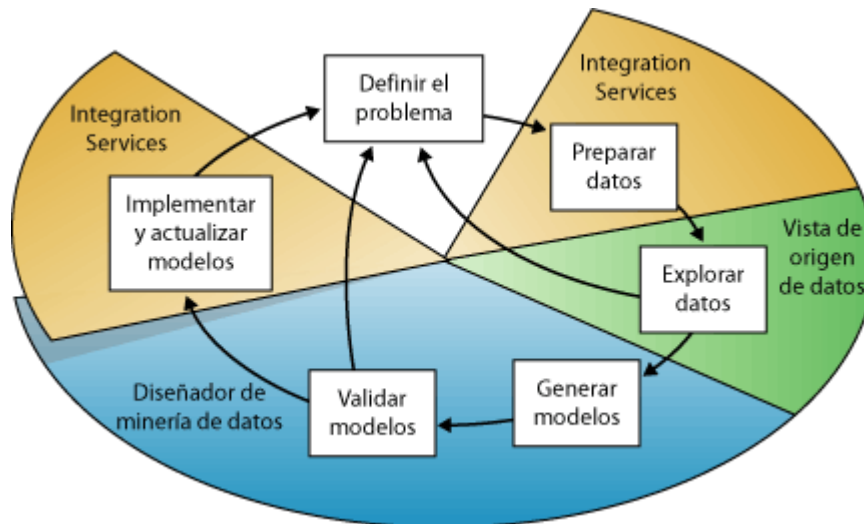


Fig 2.1 Proceso del Data Mining. Fuente: [https://msdn.microsoft.com/es-es/library/ms174949\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ms174949(v=sql.120).aspx)

Cómo se puede ver en la figura 2.1, el proceso de la minería de datos es cíclico, lo cual significa que es un proceso dinámico e iterativo. Este proceso nos permite, en caso de que los resultados obtenidos en alguno de los pasos no sean buenos, poder volver a un paso anterior para mejorarlos.

Para la realización de los modelos descritos en los pasos anteriores, se realizan distintas técnicas de minería de datos, que básicamente son algoritmos matemáticos, más o menos sofisticados, que se aplican al conjunto de datos para la obtención de patrones [6].

Estas técnicas están divididas en dos grupos, dependiendo de cómo estén almacenados los datos, estos dos grupos son [7]:

1. **Supervisado:** En este grupo se encuentran todas las técnicas que permiten realizar predicciones de los datos que se han almacenado, esto se consigue gracias a que se relacionan todos los datos almacenados (features) con un campo especial, llamado etiqueta (label). Un ejemplo de data mining supervisado se puede encontrar en una empresa, que desea saber si un proceso de producción va a fallar, por lo que tendrá un conjunto de datos de cuándo todo ha funcionado correctamente, con una etiqueta dónde se indica su correcto funcionamiento, y otro conjunto de datos con toda la información de cuándo ha ocurrido un fallo con una etiqueta indicándolo, esto permitirá que cuando se reciban el conjunto de datos que se está produciendo en esos momentos, saber a partir de los datos ya obtenidos saber si ese proceso va a fallar o no. Dentro de este grupo encontramos las siguientes técnicas:

- **Regresión lineal:** Se trata de un modelo matemático utilizado para aproximar la relación de la dependencia entre una variable dependiente Y, las variables independientes X y un término aleatorio [8].

- **Árboles de decisión:** Se trata de un modelo de predicción, el cual es utilizado para la inteligencia artificial y el análisis predictivo. Este modelo construye un diagrama a partir de los datos que se le hayan pasado, que permite representar y categorizar una serie de condiciones.
 - **Modelos estadísticos:** Se trata de una expresión en forma de igualdad o ecuación, empleado en los diseños experimentales o regresión, que permite saber los factores que modifican una variable.
2. **No supervisado:** En este grupo se encuentran aquellas técnicas que permiten agrupar el conjunto de datos sin necesidad de etiquetas como en el supervisado, con el fin de explorarlos para encontrar alguna estructura o forma de agruparlos. Un ejemplo de este tipo se encuentra en las recomendaciones de cualquier página web, donde se almacenan los gustos de cada usuario, y se realiza una búsqueda de similitudes entre los gustos tuyos y los de otros usuarios, si encuentra alguna similitud se mostrarán aquellos gustos que han tenido los usuarios más similares a ti. Las técnicas de data mining son:
- **Agrupamiento o Clustering:** Esta técnica consiste en disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes.
 - **Reglas de asociación:** Esta técnica se utiliza para el descubrimiento de hechos que ocurren en común dentro de un determinado conjunto de datos.

2.3 Data Mining VS Big Data

Durante este apartado, se han explicado estos dos términos, pero como se ha podido observar, estos dos términos se parecen bastante, aunque los dos son cosas diferentes, por lo que en este apartado se explicará que es lo que los hace que sean diferentes.

El término Big Data es utilizado para referirse a un conjunto de datos de gran tamaño. Estos conjuntos de datos superan el manejo de los datos utilizado en los primeros tiempos.

En cambio, el término Data Mining es el proceso de extraer información de los datos almacenados, por lo tanto, se refiere a las operaciones que implican búsquedas sofisticadas, las cuáles devuelven resultados concretos y específicos [9].

Por lo tanto, el término Big Data, sería el hecho de almacenar grandes cantidades de datos, lo que provoca que los ordenadores tengan problemas para manejarlos, provocando un incremento de tiempo para obtener resultados, y el término de minería de datos sería lo que permite extraer información de estos datos mediante el uso de algoritmos matemáticos.

2.4 ¿Qué los hace tan importantes?

Una vez ya sabemos que son, podemos entrever por qué es tan importante su uso. Como se ha explicado, durante todos estos años se ha ido incrementando el volumen de datos de una manera exponencial. Esto provoca que la cantidad de datos almacenados llegue a superar los peta bytes. Por lo tanto, si se quiere extraer información se necesita una gran cantidad de tiempo. Así pues, esto permitiría obtener resultados en un menor tiempo, permitiendo así poder extraer información de esos datos.

También permite a las empresas detectar sus fallas, ayudándolas así a no perder beneficios ni reputación. Esto permite que las operaciones y movimientos de los negocios sean más inteligentes, consiguiendo que los clientes estén contentos con su servicio.

Las empresas consiguen beneficio de las siguientes formas:

- Reducción de costes: El Big Data permite a las empresas reducir costes, a la hora de almacenar grandes cantidades de datos, y les permite realizar de manera más eficiente el negocio.
- Mejor toma de decisiones: El Big Data les permite la posibilidad de analizar el conjunto de datos inmediatamente.
- Nuevos productos y servicios: Las empresas son capaces de crear nuevos productos gracias a la capacidad de medir las necesidades de los clientes.

2.5 Herramientas de Big Data y Data Mining.

Para la realización del proyecto será necesario la utilización de una herramienta de las herramientas que se describirán a continuación, cada una de estas herramientas tiene un uso y unas funcionalidades diferentes, por lo que, en este apartado, se verán las distintas ventajas y desventajas de algunas de ellas.

- **MapReduce**: Se trata de un cálculo de álgebra lineal, implementado inicialmente por Hadoop, creado por Doug Cutting [10].

- Ventajas:
 - Automatiza aspectos de paralelismo y tolerancia a fallos.
 - Permite que el programador tenga que ocuparse de la funcionalidad que se requiera.
 - Factorización del código.
- Desventajas:

- Modelo forzado para cierto tipo de aplicaciones, no todos los datos pueden extraerse con MapReduce, por lo que será necesaria la implementación de otro algoritmo.
- Por cada funcionalidad que se realiza por MapReduce se incrementa el tiempo unos 10 segundos.

- **Weka:** Se trata de un software libre para la minería de datos, escrito en Java y desarrollado por la Universidad de Waikato [11].

- **Ventajas:**
 - Al tratarse de un software libre, permite el fácil acceso a los usuarios.
 - Contiene una gran cantidad de algoritmos, para la realización de modelado y procesamiento de los datos.
 - Al tratarse de un software, el cual incluye una interfaz, permite a los usuarios realizar análisis teniendo poco conocimiento sobre la materia.
 - Al estar escrito en Java, permite su funcionamiento en la mayoría de plataformas.
- **Desventajas:**
 - Los resultados pueden perder comprensibilidad, debido a la utilización de métodos de combinación de modelos.
 - Al tratarse de un software, las modificaciones de los parámetros de los modelos pueden verse limitados a los que deje el programa.

- **Spark:** Se trata de una herramienta de código abierto cuyo desarrollo fue realizado por la universidad de California. La base del código fue donado a Apache Software Foundation, quienes lo mantienen a día de hoy. Esta herramienta proporciona una interfaz que permite la realización de clusters con paralelismos de datos y tolerancia a fallos [12].

- **Ventajas:**
 - Permite la realización de análisis sofisticados de datos.
 - Incluye algoritmos de Machine Learning, para poder realizar predicciones.
 - Permite la posibilidad de realizar análisis en tiempo real.
 - Es 100 veces más rápido que Hadoop, debido a que trabaja en memoria y no en disco.
 - Es de código abierto, lo que permite que la propia comunidad mantenga y mejore el código.
 - Permite su uso en varios lenguajes de programación: Java, R, Python y Scala.

- Es flexible, permite la elección de escoger que datos permanecen en memoria y cuales en disco.
 - Combina varias plataformas, estas son:
 - Spark SQL: Permite la realización de consultas SQL.
 - Spark Streaming: Permite gestionar grandes volúmenes de datos en tiempo real, estos datos son analizados según van entrando.
 - MLlib (Machine Learning): Contiene algoritmos para la realización del Machine Learning.
 - GraphX: Es un framework que permite el procesamiento gráfico de los datos.
 - Desventajas:
 - Consume una gran cantidad de memoria.
 - Requiere de una gran utilización de recursos.
- **Hadoop**: Se trata de un framework de software libre, que permite la realización de aplicaciones distribuidas. Este framework pertenece a Apache, y está construido mediante la programación en Java [13].
- Ventajas:
 - Las tareas son tratadas independientemente, lo cual permite manejar fácilmente los fallos.
 - Permite el almacenamiento de grandes cantidades de información.
 - Se trata de una tecnología altamente escalable, esto es debido a que permite su crecimiento añadiendo nuevos nodos.
 - El almacenamiento de los datos es de bajo coste, debido a que no almacena los datos en filas y columnas, si no que asigna los datos a través de distintas computadoras.
 - Permite ejecutar procesamientos y análisis muy rápidos.
 - Es tolerante a fallos, por lo que si un equipo cae siempre habrá una copia disponible.
 - Desventajas:
 - Los datos mediante el uso de esta tecnología están comprometidos, debido a que Hadoop tiene desactivada dicha funcionalidad al tener cierta complejidad.

- No es una tecnología para utilizarla con pocos volúmenes de datos, ya que carece de la capacidad de soportar eficientemente la lectura aleatoria de archivos pequeños.

- **Storm:** Se trata de una plataforma de software libre, en el cual permite la distribución en tiempo real de la computación del sistema [14].

- **Ventajas:**

- Es de código abierto, por lo que cualquier usuario puede hacer uso de forma sencilla, además está siendo mantenido por la comunidad de usuarios.
- Permite la realización de análisis de datos en tiempo real.
- Es escalable y tolerante a fallos, lo cual permite que los datos puedan ser procesados fácilmente.
- Se puede implementar con varios lenguajes de programación, tales como Java, C#, Python, Scala, Perl o PHP.
- Gestiona los datos con una latencia baja.

- **Desventajas:**

- A diferencia de Spark no permite la generación de gráficos.
- Su uso y gestionamiento es complicado.
- Puede transmitir un mismo mensaje dos veces, lo que puede afectar el escenario.
- En caso de necesitar recoger los datos en lotes o mediante acceso SQL es mejor la utilización de Spark.

CAPÍTULO 3. ANÁLISIS DE SENTIMIENTOS

Uno de los objetivos de este proyecto es ser capaz de extraer información de las imágenes. La información que se intentará extraer serán los sentimientos de las caras. Concretamente, este proyecto se centrará en intentar predecir si una cara está feliz o triste. Para ello se realizarán técnicas de data mining, y como se ha descrito anteriormente, se deberán de seguir una serie de pasos para la obtención e implementación del modelado de los datos.

Los pasos que se seguirán son:

- Definición del problema.
- Preparar los datos.
- Exploración y modificación de los datos.
- Generar, explorar y evaluar modelos.
- Implementar y ejecutar los modelos.

A continuación, se verán las tecnologías que se utilizarán para realizar el análisis de las caras y después cómo se han ido realizando cada uno de los pasos, durante la realización del proyecto, para conseguir predecir el sentimiento de las caras que aparecen en las imágenes.

3.1 *Tecnologías*

Para poder realizar un análisis de las caras, será necesaria la implementación de al menos una tecnología de procesamiento de imágenes, concretamente de caras, y al menos una tecnología de data mining, para poder clasificar las caras según su sentimiento.

Para poder realizar el procesamiento de las imágenes, será necesario la utilización de alguna tecnología que permita saber si existe alguna cara en la imagen y dónde está situada, y también alguna otra que permita extraer información de los puntos más relevantes del rostro. Para conseguir esto, se utilizaron las siguientes dos tecnologías:

- **JQuery.faceDetection** [15]: Esta tecnología, permite detectar en una imagen la presencia de caras, por lo que, en caso de detectar una cara, se obtendrá la posición X e Y en la imagen, y el ancho y alto que ocupa en ella.
- **Cmltrackr** [16]: Esta tecnología permite ajustar modelos faciales y obtener las posiciones de 71 puntos representativos de los rasgos faciales de una imagen.

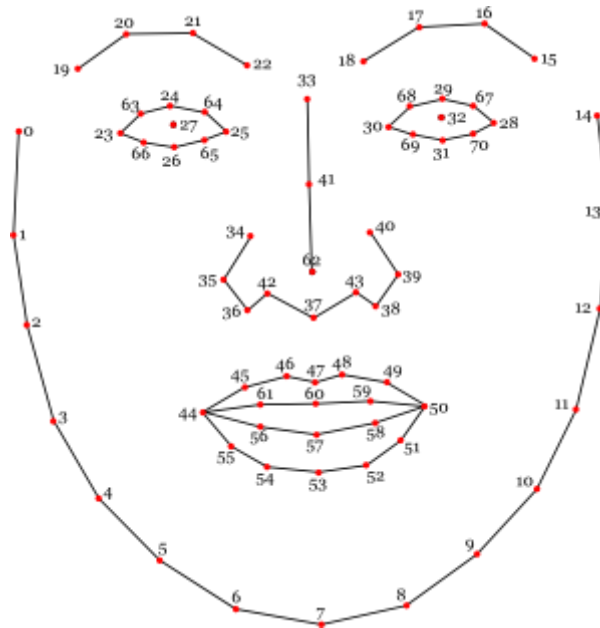


Fig. 3.1 Puntos de los rasgos de la cara que detecta la librería Cmltrackr.

Fuente: <https://github.com/auduno/cmltrackr>

Cómo se puede ver en la figura 3.1, esta tecnología devolverá los 71 puntos que se pueden apreciar en la imagen. Esto permitirá poder almacenar los datos de los puntos para posteriormente poder predecir si la imagen está triste o contenta.

En cuanto a la tecnología de data mining, alguna de las cuáles se han visto en el anterior capítulo, se utilizarán las siguientes:

- **Weka:** Al ser una tecnología que permite realizar análisis mediante el uso de una interfaz sencilla, tal y cómo se ha explicado en el capítulo anterior, la cual no necesita ser programada cómo ocurre con las otras, permitiendo, mientras se programa la tecnología que se implemente en la aplicación final, realizar el estudio de los datos.

- **Spark:** Finalmente se decidió utilizar Spark cómo la tecnología que se implementará en la aplicación final, debido a que es más rápido que otras tecnologías, que se han visto en el anterior capítulo, además de que implementa algoritmos de Machine Learning, los cuales son útiles para poder realizar predicciones, que al fin y al cabo es lo que se pretende conseguir, a partir de los datos que se extraigan de las imágenes, conseguir predecir el sentimiento de las caras. Y otro motivo por el que se ha escogido esta tecnología, en vez de utilizar las otras que se han explicado, o implementar Weka en la aplicación final, es que la aplicación que se va a implementar

pretende simular un entorno real, el cual tendrá un gran volumen de datos, por lo que esta tecnología es más adecuada que Weka, que es más para realizar estudios y comportamientos de los datos de manera rápida, y que las otras ya que incluye los otros dos motivos explicados.

3.2 Definición del problema

La aplicación tendrá que ser capaz de extraer los sentimientos de las caras a partir de las imágenes que los usuarios vayan subiendo a la página. Por lo tanto, el problema principal al que se deberá hacer frente será cómo extraer y representar la información de la imagen de forma adecuada para poder entrenar el clasificador y generar un modelo a utilizar para la predicción en futuras imágenes.

3.3 Preparar los datos

En este segundo proceso para el análisis de los datos, se deberá comprobar el correcto funcionamiento de las librerías utilizadas para extraer y representar las expresiones de las caras. Se realizará también una limpieza de los datos almacenados, ya que estos pueden provocar ruido, el cual hará que empeoren los resultados finales, dificultando al clasificador que sepa decidir que sentimiento es.

Lo primero que se realizó fue comprobar que el detector de caras, la librería JQUERY.faceDetection, detectara correctamente las caras que aparecían en las imágenes. Esta librería permite saber en qué posición de la imagen se encuentra la cara, y el ancho y alto que ocupa en ella. Esto permitirá recortar las caras de las imágenes de forma conveniente, para que la librería Cmltrackr pueda extraer los rasgos de las mismas. En nuestro caso es importante que las caras recortadas estén a una misma escala. Aquí se vio un primer problema. La librería indicaba donde estaba la cara, pero al recortar la imagen con la posición y el ancho de la cara, también se recortaba parte de ella, lo que provocaba que después no se pudiesen detectar correctamente los puntos de las caras. En la figura 3.2, se puede ver un ejemplo de lo explicado.



Fig. 3.2 Ejemplo de una cara recortada (imagen de la izquierda). Fuente: <https://www.thub.com.au/jobs/employer/my-departments>. Ejemplo una cara recortada a partir de la librería JQUERY.faceDetection (imagen de la derecha).

En la imagen de la izquierda se puede apreciar la imagen original, mientras que en la segunda la imagen ya recortada. Cómo se ve la cara es detectada correctamente, pero no está recortada de forma adecuada para pasársela a Cmltrackr, ya que no es capaz de mostrar todo el rostro. Por lo tanto, se realizaron un conjunto de cálculos para conseguir aumentar la ventana de corte, de forma que apareciese el rostro completo en todas las imágenes. Lo primero que se tuvo que realizar fue cambiar la posición x e y, a una posición menor a la que estaba, y después aumentar el ancho y alto de la cara para evitar que recorte la cara.

$$x = \text{Max}(0, X_{\text{cara}} - 0.4 * \text{AnchoCara}) \quad (3.1)$$

$$y = \text{Max}(0, Y_{\text{cara}} - 0.4 * \text{AltoCara}) \quad (3.2)$$

Las dos funciones que se pueden presenciar encima se utilizaron para solucionar el problema anterior, por lo que se pensó en cambiar la coordenada x e y que devuelve la librería JQuery.faceDetection. Estas coordenadas indicarán al servidor la posición de la imagen por donde ha de empezar a recortar. Por lo tanto, se decidió pasar la coordenada x a una posición más a la izquierda y la posición y a una más arriba de forma que se asegurase que la cara por la parte de la izquierda no fuese recortada. Así pues, tal y como se ve, para obtener la posición x, se ha de realizar el máximo entre la posición inicial 0, y la resta entre la coordenada x devuelta por la librería (X_{cara}) menos el 40% del ancho que devuelve esta misma librería (AnchoCara). Realizando esto se pretende agrandar un 40% la superficie que se recortará de la imagen, en caso de que no se pueda agrandar un 40% por el lado izquierdo, se deberá

recortar la superficie por la posición inicial de la imagen, cuyo valor corresponde al 0 que podemos ver en la fórmula, de esta forma, realizando el máximo, se asegura que nunca el valor de x sea negativo. Para la obtención de la coordenada Y se realizará lo mismo, lo único, que cambia es que ahora hay que colocar la coordenada Y que da la librería (Y_{cara}) y restarle el 40% del alto de la cara ($AltoCara$).

$$\text{Ancho} = \text{Min}(\text{AnchoImagen} - 1, X_{cara} + 1.4 * \text{AnchoCara}) \quad (3.3)$$

$$\text{Alto} = \text{Min}(\text{AltoImagen} - 1, Y_{cara} + 1.4 * \text{AltoCara}) \quad (3.4)$$

Para conseguir recortar correctamente la parte derecha de la cara, se deberá realizar otras dos funciones, esta vez para aumentar el ancho y alto de la superficie. Como hemos visto con las anteriores funciones, se pretende aumentar la superficie de recorte en un 40%, como se han cambiado las coordenadas x e y , el ancho y alto que proporciona la librería ya no será el mismo, ya que si se usa el mismo se estaría recortando aun peor la imagen. Así que, para conseguir recortar la imagen bien, se deberá aumentar tanto el ancho como el alto. Esto se consigue mediante las dos ecuaciones que se pueden observar encima. Para conseguir el ancho, se realiza el mínimo entre la resta del ancho de la imagen antes de recortar (AnchoImagen) menos 1, lo que devuelve el máximo valor de la coordenada x de la imagen, es decir, que no se deberá recortar la imagen más allá de ese punto, y la suma entre la coordenada x (X_{cara}), que indica la posición x donde la librería ha detectado el inicio de la cara más el inverso de un 40% por el ancho de la cara, con lo que se pretende obtener el mínimo valor que hay que aumentar el ancho de la imagen recortada para que se obtenga un 40% más de superficie de la cara, que la que devolvía la librería. Para obtener el alto, se realizará lo mismo, y es el mínimo entre la resta del alto de la imagen sin recortar (AltoImagen) menos uno y la suma entre la coordenada y donde la librería indica que se inicia la cara (Y_{cara}) y el inverso del 40% entre el alto de la cara (AltoCara).

Realizando este conjunto de operaciones se espera conseguir que todas las imágenes muestren el rostro entero.



Fig. 3.3 Cara recortada correctamente.

En la anterior imagen se puede ver el resultado de estas operaciones, dónde se puede apreciar cómo la imagen ha sido recortada correctamente, permitiendo obtener el rostro.

Por otro lado, para que todas las imágenes tengan el mismo tamaño, se fijó el alto de las imágenes a 240 píxeles, Por lo tanto, fue necesario, antes de guardarlas, escalarlas a dicho tamaño.

Lo segundo que se realizó fue la comprobación de que las caras que se estaban almacenando en el fichero CSV, estaban bien procesadas por la librería Cmltrackr, ya que si se guardan los puntos de las imágenes que se han detectado erróneamente, estos pueden afectar posteriormente al clasificador, debido a que se le estaría añadiendo ruido innecesario, por lo tanto, en este primer paso se le deberá de pasar las mejores imágenes que se detecten.



Fig 3.4 Caras correctamente detectadas.

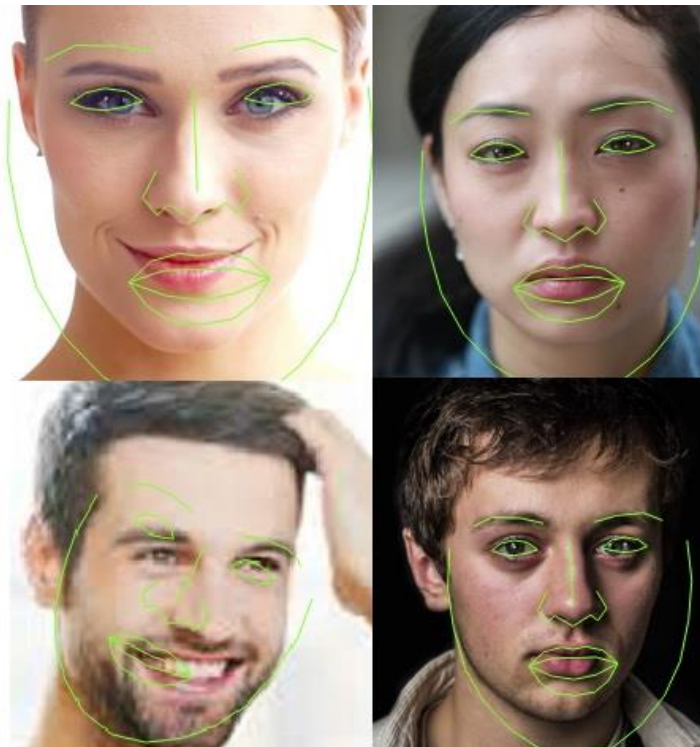


Fig. 3.5 Caras mal detectadas.

En las dos anteriores imágenes se pueden ver diferentes ejemplos de caras, y con una línea verde, los diferentes puntos de la cara. En cada uno de los ejemplos se muestran cuatro caras, dos de hombres y dos de mujeres, con sentimiento de felicidad y de tristeza, que son de momento los dos sentimientos que la página intentará adivinar.

Se puede apreciar como en la primera imagen las cuatro caras están perfectamente detectadas, mientras en el segundo grupo, las líneas de expresión están erróneas. A esto nos referíamos anteriormente, si se hubiesen añadido en el fichero CSV estas cuatro imágenes, esto provocaría que el clasificador que se acabe utilizando, realice un tratamiento errado de las imágenes, clasificándolas erróneamente en la aplicación final, debido a que la precisión del clasificador sería realmente mala al haber añadido ruido. Si nos fijamos detalladamente en las imágenes erróneas, se puede ver como en el caso de las caras tristes, la librería detecta que la boca está abierta, cuando no lo está, igual que pasa en el ejemplo de la chica que está sonriendo. Por lo tanto, el clasificador podría detectar esa abertura de la boca como que la persona está riéndose, o en caso de que en un futuro se acaben implementando otras emociones, se podría llegar a pensar que el sentimiento que hay en esa imagen es de sorpresa u otro sentimiento, por eso estas imágenes finalmente no se han utilizado.

Al final, estos fallos al recoger los puntos de la cara han provocado que se tengan un total de 120 imágenes para entrenar el clasificador, de las cuales 60 son de caras tristes y las otras 60 de caras felices.

3.4 Explorar y modificar los datos

Una vez ya se han seleccionado los datos y se han limpiado posibles alteraciones de ellos se realizará una exploración y modificación de estos. Es decir, partiremos del conjunto inicial de 71 puntos proporcionados por Cmltrackr y a partir de ahí miraremos si podemos mejorar la forma de representar dicha información y en definitiva el resultado de la clasificación.

Durante una primera prueba, los resultados obtenidos por todos los clasificadores al obtener todos los puntos de la cara tal cual cómo vienen en las imágenes, estaban alrededor del 50% con todos los clasificadores que se pensaron utilizar (Naive Bayes, Decision Tree y Random Forest), por lo que se decidió buscar otras formas de representar los datos. Cómo veremos más adelante, algunas permitieron obtener mejores resultados y otros incluso los empeoraron.

Como se ha comentado anteriormente, se detectarán por cada una de las caras 71 puntos, estos puntos indicarán las posiciones de los rasgos más importantes del rostro. Por lo tanto, antes de empezar a mirar que datos nos serán útiles y cuáles no, se deberán realizar diferentes formas de analizarlos. Al encontrar estos malos resultados, se pensaron seis formas diferentes de poder representar los datos, a partir de los 71 puntos comentados previamente.

La primera forma, se trata de almacenar los datos tal y cómo se reciben, teniendo en cuenta que, por cada punto, se guardará su coordenada X y su coordenada Y. La segunda forma que se pensó consiste en normalizar los puntos, de manera que todos los puntos tuviesen la misma referencia en todas las imágenes, esto se consigue gracias a que el valor de los puntos estaría comprendido entre 0 y 1. Para conseguir esta normalización, se deberá encontrar el punto central de cada una de las caras.

Las siguientes cuatro formas que se pensaron tienen la misma relación. El objetivo de estas formas es mirar si a través de las distancias de los puntos se podía representar mejor los sentimientos. El primer intento fue mirar la distancia de la coordenada x e y con la nariz, dándonos la tercera forma de guardar los datos. La cuarta forma seguía el ejemplo anterior, de mirar las distancias con la nariz, pero a diferencia de la anterior, esta vez se realizaría la distancia entre puntos, y no entre coordenadas por separado. Por último, encontramos las dos últimas maneras que se pensaron para coleccionar los datos, que al igual que las dos anteriores estaban basadas en las distancias de sus coordenadas y la otra de sus puntos, pero en vez de apuntar a la nariz, esta vez se miraría de guardar la distancia con el centro de la cara.

Por lo tanto, nos quedarían estas seis formas de almacenar los datos:

- Guardar las coordenadas absolutas x e y de cada uno de los puntos de la cara (142 atributos).
- Guardar las coordenadas x e y normalizadas, de forma que todos los valores estuviesen contenidos entre 0 y 1 (142 atributos).
- Guardar las distancias de los puntos con el centro de la cara (71 atributos).
- Guardar las distancias entre los puntos de la cara y la punta de la nariz (71 atributos).
- Guardar la distancia en x y la distancia en y de cada punto con el centro de la cara (142 atributos).
- Guardar la distancia en x y la distancia en y de cada punto con la punta de la nariz (142 atributos).

Los resultados obtenidos con estas seis formas almacenar los datos fueron bastante malos, daban unos valores cercanos al 50% de precisión, lo cual vendría a ser lo mismo que realizar la clasificación del sentimiento de la cara

aleatoriamente, además de que no se ha mejorado el resultado obtenido al guardar los puntos tal cual están en las imágenes. Estos resultados son malos por dos razones principalmente, la primera es que se tienen muy pocas muestras, concretamente 60 de cada tipo y la segunda razón se debe a como se estaban guardando los puntos de la cara.

Hasta ahora, se han considerado todas las caras como si estuviesen en la misma posición, es decir, no se tuvo en cuenta en ningún momento la inclinación de la cara. Esto provoca que, en las dos primeras formas de medir los puntos, la posición de éstos, por mucho que se normalicen estarán en un lugar distinto, en el caso de las otras formas de guardar los puntos, esto no afectaría debido a que éstos solo están guardando la distancia con un punto de referencia, por lo que por mucho que la cara este inclinada el valor de estos no se vería afectado.

Por lo tanto, se necesitará realizar un conjunto de cálculos para que las caras estén en vertical, y no inclinadas.



Fig. 3.6 Cara con el rostro inclinado (imagen de la izquierda). Fuente: <https://twitter.com/carlamaney?lang=es>. Cara con el rostro recto (imagen de la derecha). Fuente: <https://es.scribd.com/article/336317996/Tim-Ferriss-If-Youre-Not-Happy-With-What-You-Have-You-Might-Never-Be-Happy>.

En las dos imágenes que hay arriba, se tiene un ejemplo claro de lo que se está explicando, donde se ven a dos personas, un hombre y una mujer que aparentemente están contentos. El clasificador, obtendría los puntos, pero como se puede apreciar, las posiciones de la cara no están colocadas de la misma forma, la cara de la mujer esta inclinada mientras que la del hombre permanece recta. Por lo tanto, las posiciones de los puntos de la cara no estarían colocadas en la misma posición, dificultando al clasificador la elección

del sentimiento de cada rostro. Es por eso por lo que se requiere de un conjunto de pasos para solucionar este problema.

Para solucionar este problema fue necesario programar en el código un conjunto de funciones para así poder colocar todas las caras rectas [17].

En la figura 3.1 se puede apreciar los puntos que se obtienen por cada una de las caras. Para encontrar el ángulo de inclinación de las caras se tendrá en cuenta los puntos 62 y 33, correspondientes al perfil de la nariz, concretamente la coordenada x de cada uno de ellos, por lo que si las coordenadas x de los dos puntos son exactamente iguales el ángulo de inclinación equivaldrá a 0. En caso de que sean diferentes se realizará la siguiente operación:

$$\text{Ángulo nariz} = \tan^{-1} \left(\frac{y_{33} - y_{62}}{x_{33} - x_{62}} \right) \quad (3.5)$$

Para obtener el ángulo que forma la nariz con la vertical, se realizará el arco tangente de la resta entre la coordenada y del punto 33 menos la del punto 62 dividido entre la resta entre la coordenada x del punto 33 menos la del punto 62. Esta fórmula devolverá el ángulo en radianes. A este resultado, se le deberá de sumar $\pi/2$ en el caso de que el ángulo sea negativo, y en caso contrario se deberá restar $\pi/2$, para obtener el ángulo con la perpendicular. A continuación, se explicará lo que se acaba de realizar utilizando las dos imágenes anteriores.

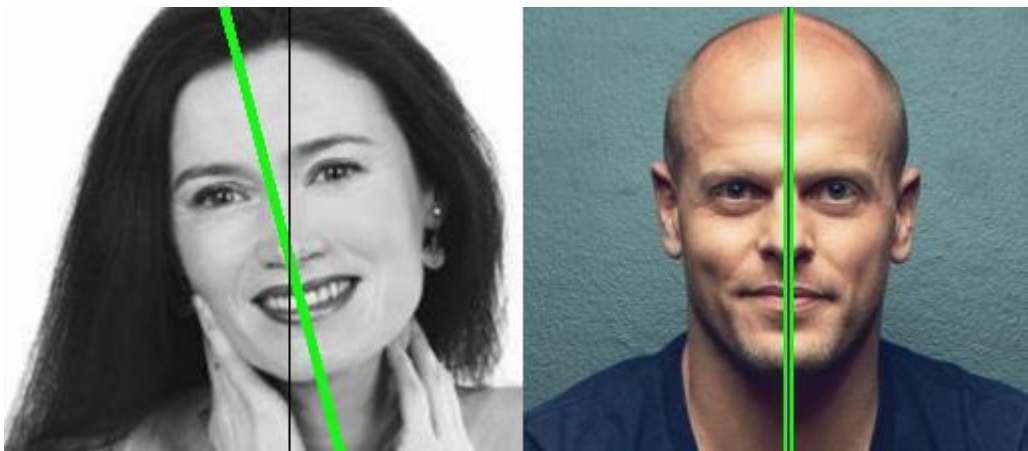


Fig 3.7 Corrección ángulo de inclinación.

Como vemos en la imagen de arriba, se ha trazado una línea negra, que indica la vertical, pasando por la punta de la nariz. Tal y cómo puede ver en la imagen de los puntos se trata del punto 62 y se ha trazado otra línea, esta de color verde, para indicar la inclinación de la cara. Esta segunda línea pasa por el punto 62 y 33 de la cara. Por lo tanto, en la imagen de la derecha el ángulo de la nariz sería 0, mientras que, en el caso de la mujer, se puede apreciar una cierta inclinación, que habrá que corregir, mediante la fórmula comentada anteriormente.

Para acabar de corregir la cara, se deberán rotar todos los puntos a partir del ángulo obtenido, para ello se realizarán las siguientes dos operaciones:

$$X = \cos \alpha * (x_i - x_{62}) + \sin \alpha * (y_i - y_{62}) + x_{62} \quad (3.6)$$

$$Y = \cos \alpha * (y_i - y_{62}) + \sin \alpha * (x_i - x_{62}) + y_{62} \quad (3.7)$$

Las anteriores dos funciones se deberán realizar para cada uno de los puntos de la cara. Para obtener la coordenada x rotada se deberá realizar el coseno del ángulo (α) por la resta de la coordenada x que se desea rotar (x_i) menos la coordenada x de la punta de la nariz (x_{62}), a esto habrá que sumarle el seno del ángulo por la resta entre la coordenada y (y_i) del punto que se desea rotar menos la coordenada y de la nariz, y finalmente se le deberá de sumar la coordenada x de la punta de la nariz. Para obtener la y, se deberá de realizar el coseno del ángulo por la resta entre la coordenada y que se desea rotar menos la coordenada y de la nariz, a esto se le sumará el seno del ángulo por la resta entre la coordenada x del punto que se desea rotar menos la coordenada x de la nariz y finalmente a todo esto se le sumará la coordenada y de la nariz. Estas dos ecuaciones permiten girar los puntos, debido a que se tratan de funciones circulares. Estas funciones toman un punto como central, que sería el punto 62 de la cara, y las rota el ángulo deseado para cada uno de los puntos.

Una vez ya se ha realizado la modificación de los datos, se podrá realizar el siguiente paso, la generación, exploración y evaluación de modelos.

3.5 Generar, explorar y evaluar modelos

En este apartado, se ha decidido juntar el paso de generación de modelos y la exploración y evaluación de estos en un único paso, para simplificar la

explicación, además de que estos dos pasos realmente se podrían considerar cómo uno solo.

Ahora que ya se han limpiado los datos, y se han modificado para obtener un mejor resultado, se procederá a crear cada uno de los modelos.

Cómo se ha comentado a lo largo del proyecto, se pretende saber el sentimiento de las caras que hay en las imágenes, por eso los modelos que se creen deberán ser clasificadores, esto permitirá al programa obtener un patrón con el cual, según los valores de los puntos, sepa si la cara tiene un sentimiento de felicidad o de tristeza.

Los clasificadores que se escogieron fueron los siguientes:

- Naive Bayes.
- Decision Tree.
- Random Forest.

Una vez ya se saben los clasificadores que se van a implementar, se realizará un análisis para saber cuál de ellos y que forma de obtener los datos finalmente se escogerá. Para poder realizar este análisis se requerirá el software Weka, esto nos permitirá obtener los resultados de manera rápida. A los clasificadores se les entrenará con el 70% de las imágenes, y se realizará una validación con el 30% restante, es decir, se entrenará el clasificador con 84 imágenes y se realizará el testeo del modelo que se genere con las 36 imágenes restantes.

Los resultados obtenidos son los que se pueden apreciar en la tabla que hay a continuación:

Tabla 3.1 Tabla con la precisión obtenida de los datos para cada uno de los clasificadores.

| | Naive Bayes | J48 (Decision Tree) | RandomForest |
|--|-------------|---------------------|--------------|
| Cara sin normalizar | 54.7% | 57.5% | 53.3% |
| Diferencia de puntos con la nariz | 55.2% | 43.3% | 50% |
| Diferencia de puntos con el centro de la cara | 55.2% | 48.4% | 51.7% |
| Diferencia de coordenadas x e y con la nariz | 55.2% | 54.7% | 49.2% |
| Diferencia de coordenadas x e y con el centro de la cara | 52.8% | 52.8% | 46.7% |
| Cara normalizada | 51% | 59.2% | 65.1% |

Como se puede ver, los resultados no acaban de ser buenos en la mayoría de casos, a pesar de todas las modificaciones de los datos que se han realizado. En la mayoría de casos la tasa de aciertos ronda el 50% por lo que sería cómo si se predijesen los sentimientos de las imágenes aleatoriamente, estos malos resultados pueden ser debidos a las pocas imágenes con las que se cuenta para entrenar al clasificador, y al funcionamiento de la librería de Cmltrackr, lo que hace pensar, en un futuro, en mejorar la parte de procesado de caras. El mejor resultado que se obtiene en el caso de las caras almacenadas con los puntos normalizados, con una tasa de aciertos del 65,1%, para el clasificador Random Forest.

Los resultados obtenidos tienen sentido, en cuanto al valor obtenido, ya que cómo se había comentado en el anterior paso, cuando se estaban rotando las caras, esta modificación de los datos no iba a cambiar los resultados obtenidos en los datos que se almacenaban las distancias, debido a que por mucho que se rotasen las caras, las distancias con los puntos de estas iba a ser el mismo, además de que tiene sentido que sea el peor resultado, ya que cada cara puede tener una altura y una anchura diferente, y el hecho de estar triste o contento, no se puede saber solo sabiendo las distancias, ya que estas pueden no ser lo suficientemente diferentes para que el clasificador sepa cómo se deben de clasificar. En cuanto a las otras dos formas, que son con las que se han obtenido los mejores resultados, se puede diferenciar más si la cara esta triste o contenta, debido a que se están guardando todos los puntos de la cara, con lo cual se le estaría dando más información que pasándole solo las distancias, permitiendo saber cómo afecta la anchura, la altura, y cómo estén colocados los puntos, y el hecho de normalizar estos puntos es lo que permite

mejorar el resultado, ya que colocaría todos los puntos en una posición similar, provocando que el clasificador pueda encontrar diferencias si el valor es muy diferente.

Una vez, ya se ha obtenido el resultado de cada uno de los clasificadores y se ha elegido cuál será el que se acabe implementando, se deberá modificar para obtener el máximo resultado posible.

El clasificador Random Forest, permite modificar dos parámetros, el número de árboles y la profundidad máxima del árbol [18]. Por lo tanto, se deberá realizar un análisis de los resultados que se obtengan al modificar estos valores.

Durante las pruebas al cambiar el valor de la profundidad máxima del árbol se descubrió que el máximo valor que permitía era 30, y que la modificación de este no mejoraba los resultados, sino que permanecían iguales. Además, este parámetro mejora el resultado cuanto mayor es la profundidad, si la profundidad máxima es 30, y por lo que se ha podido saber al realizar las pruebas es el valor que tiene por defecto, poco más se va a poder mejorar el resultado, ya que no se podrá incrementar el valor. Por lo tanto, el único valor que se podrá cambiar será el número de árboles.

Para saber cuál es el valor idóneo de árboles que deberá tener el clasificador se deberá comprobar la precisión de este por cada uno de los valores que se le introduzca. Para ello se realizará un estudio de la precisión, tal y cómo se verá en el gráfico que se mostrará a continuación de este párrafo. Los valores que tendrá el número de árboles estarán comprendidos entre 10 y 100, dando saltos de 10 en 10.

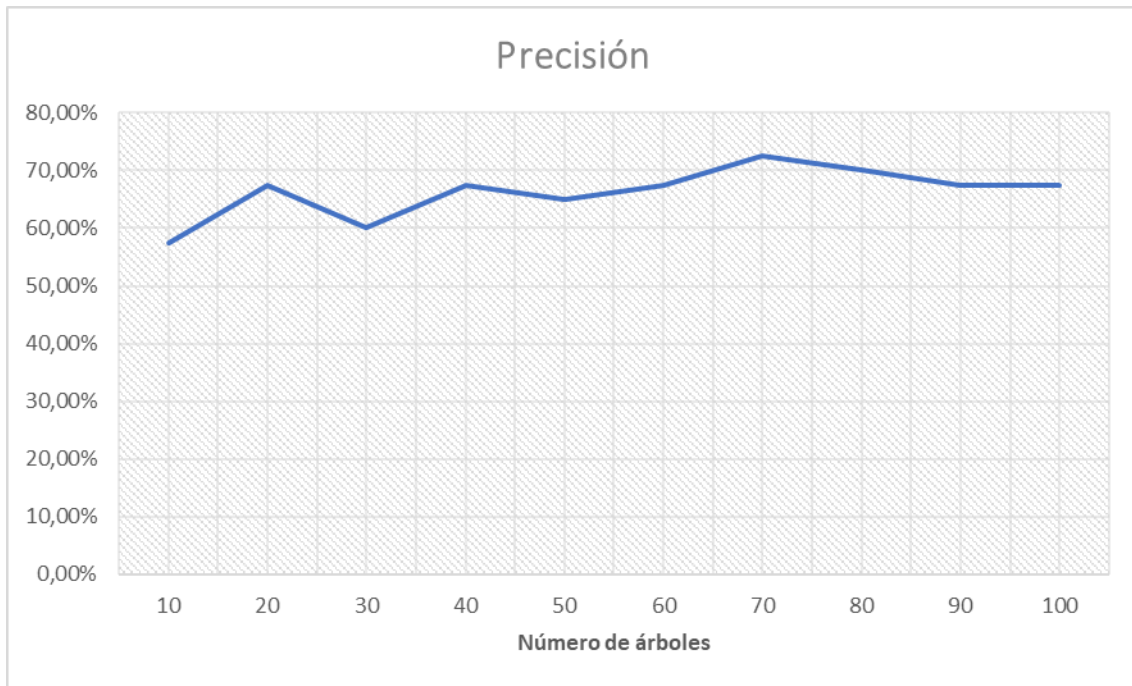


Fig. 3.8 Precisión del clasificador según en el número de árboles.

En la gráfica anterior se puede ver por cada uno de los valores que se le ha asignado al parámetro que permite cambiar el número de árboles, la precisión del clasificador. Se puede apreciar cómo a medida que se va incrementando el número de árboles, la precisión del clasificador se incrementa también, aunque se ven caídas en la precisión cuando se tienen 30 árboles y también cómo decrece un poco al tener 50. La máxima precisión que se obtiene es de un 72.5%, y se obtiene con 70 árboles, a partir de aquí la precisión decae hasta llegar a un 67.5% con 100 árboles. Como se puede ver se ha conseguido mejorar el clasificador modificando el número de árboles, pasando de tener un 65.5% al 72.5% con 70 árboles.

Para finalizar la mejora del clasificador, se deberán de escoger los puntos de la cara que mayor importancia tengan para el clasificador, por lo que se requerirá la realización de otro estudio para seleccionar los puntos de la cara con los que el clasificador tenga una mayor precisión. Spark, permite la obtención de los datos que más influyen al escoger un sentimiento u otro, por lo tanto, lo primero que se deberá realizar es la obtención de los puntos de la cara según su importancia.

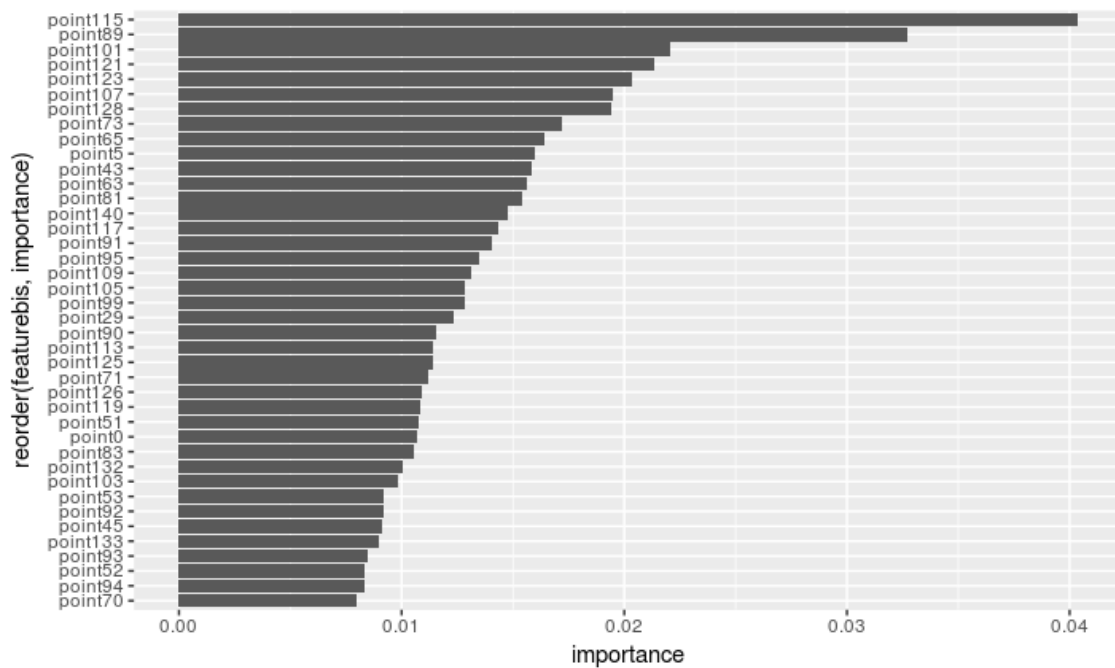


Fig. 3.9 Ordenación de los atributos según su importancia.

En la imagen anterior, se puede ver cada uno de los puntos que se han guardado y su importancia. Cada punto está ordenado según los proporciona la librería Cmltrackr, donde los números pares son las coordenadas x's y los impares son las coordenadas y's. Al descomponer los puntos en sus coordenadas x e y provoca que se nos almacenen 142 puntos, en vez de los 71 que proporciona. Por lo tanto, según este gráfico el punto 115, que es el que mayor importancia tiene, correspondería a la coordenada y del punto 57, que, si se observa en la imagen de los puntos, se puede ver que corresponde al labio inferior. El hecho de que este punto sea uno de los más relevantes tiene sentido, puesto que el estar contento o triste hace que este punto, concretamente la coordenada y, esté en una posición más arriba o más abajo de la cara. Lo mismo sucede con los dos puntos siguientes, estos dos puntos corresponden a las coordenadas x de los puntos 44 y 50, que mirando dónde se encuentran se puede ver que son los extremos de la boca, lo cual tiene su lógica, ya que cómo pasaba antes, el hecho de estar triste o contento hace que estos puntos estén más arriba o más abajo. El cuarto punto más importante, corresponde a la coordenada y del punto 60, que está situado en la parte central del labio superior de la boca.

Si se presta atención, se puede ver cómo los 5 puntos que mayor importancia tienen corresponden a números impares, y cómo se ha explicado estos puntos son coordenadas y's de la cara, y de los 20 que mayor importancia tienen, 2 son coordenadas x's, por lo tanto, la conclusión que se puede extraer de este análisis es que para el clasificador son más importantes las coordenadas y's que las x's, lo cual tiene sentido, puesto cómo se ha explicado en el anterior párrafo, el estar contento o triste afecta en la parte vertical de la cara, más que

en la parte horizontal. Una segunda conclusión que se puede extraer de este gráfico, es que para el clasificador, los 6 puntos más importantes, corresponden a coordenadas y's de la boca, por lo que tal y cómo se explicaba, si se analiza el comportamiento cuando una persona está feliz o está triste, el comportamiento del labio es uno de los más importantes, ya que en caso de que se esté contento, la parte central del labio baja y los extremos suben, mientras que si se está triste pasa lo contrario, la parte central sube y los extremos decrecen, por lo tanto tiene sentido que los puntos más importantes estén en la boca, concretamente en las coordenadas y's.

Para finalizar este paso, se realizará un análisis para obtener la mayor precisión que se pueda obtener del clasificador, quitándole, por cada prueba, aquellos puntos que sean menos importantes. También se deberá realizar esto con diferentes configuraciones para asegurarse de que se está escogiendo el número correcto de árboles para el clasificador, ya que este parámetro se había obtenido con todos los puntos y al modificar los datos que se le pasan la precisión puede cambiar. Por lo tanto, se realizará el análisis para los tres mejores casos, es decir, para el caso de 70 árboles, cuyo valor es el mejor, ya que la precisión obtenida es de un 72.5%, el caso de 80 árboles, cuya precisión es de un 70% y por último con 20 árboles, con el que se ha obtenido un 67.5% de precisión.

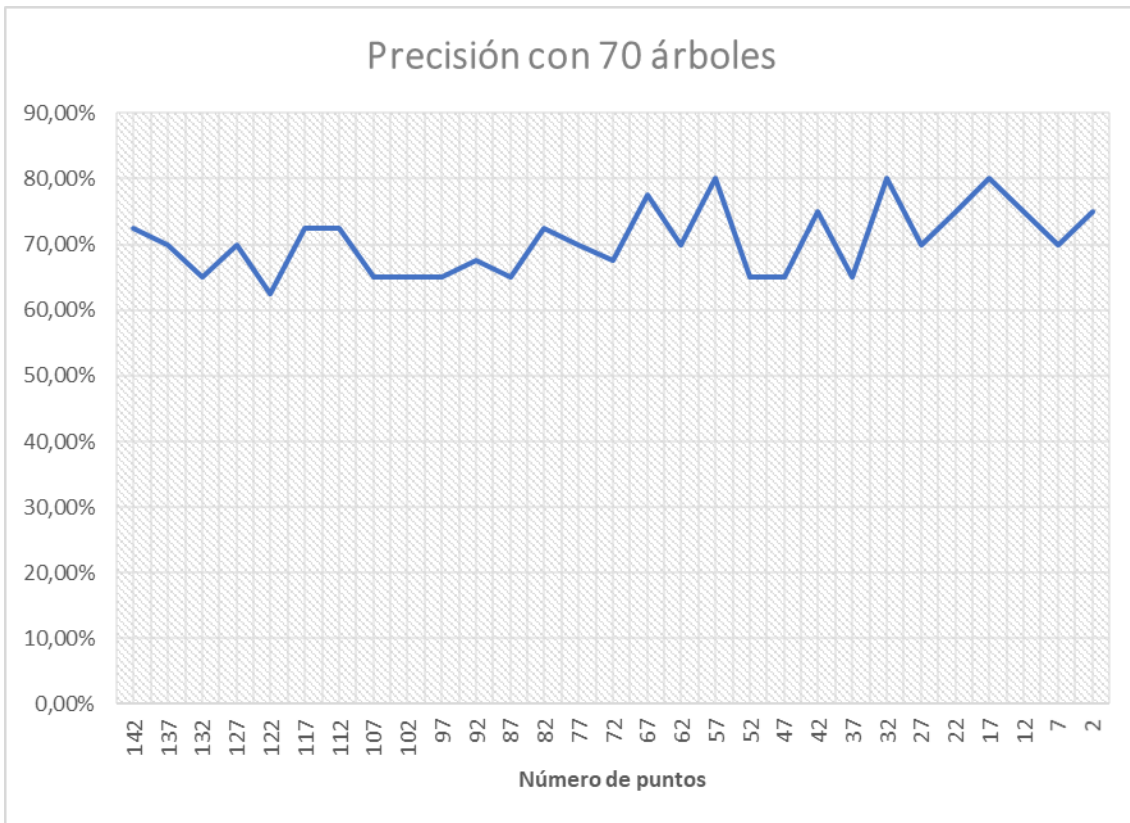


Fig. 3.10 Precisión del clasificador para 70 árboles según se le va quitando los atributos menos importantes.

En este gráfico aparece la precisión obtenida utilizando un menor número de atributos, eliminando progresivamente aquellos de menor importancia. En esta primera prueba, el clasificador ha sido configurado con 70 árboles. Como se ha visto, anteriormente, se había obtenido por orden de importancia los puntos de la cara, por lo tanto, se fue eliminando por cada prueba aquellos puntos que eran menos importantes, de esta forma se pretendía mejorar la precisión del clasificador. Si se observa el gráfico, se ve cómo esto se consigue a medida que se van quitando puntos, pero también hay veces que empeora el resultado obtenido con todos los puntos de la cara. Esto ocurre sobre todo al principio. Para esta configuración, se obtuvo un resultado máximo de un 80% de precisión en tres ocasiones, cuando se seleccionaban los 57, 32 y 17 mejores puntos de la cara. Este resultado es bastante bueno, en relación a la configuración inicial, donde se obtuvo una precisión de un 65.1%. Por lo que se ha conseguido aumentar la precisión un 14.9%.

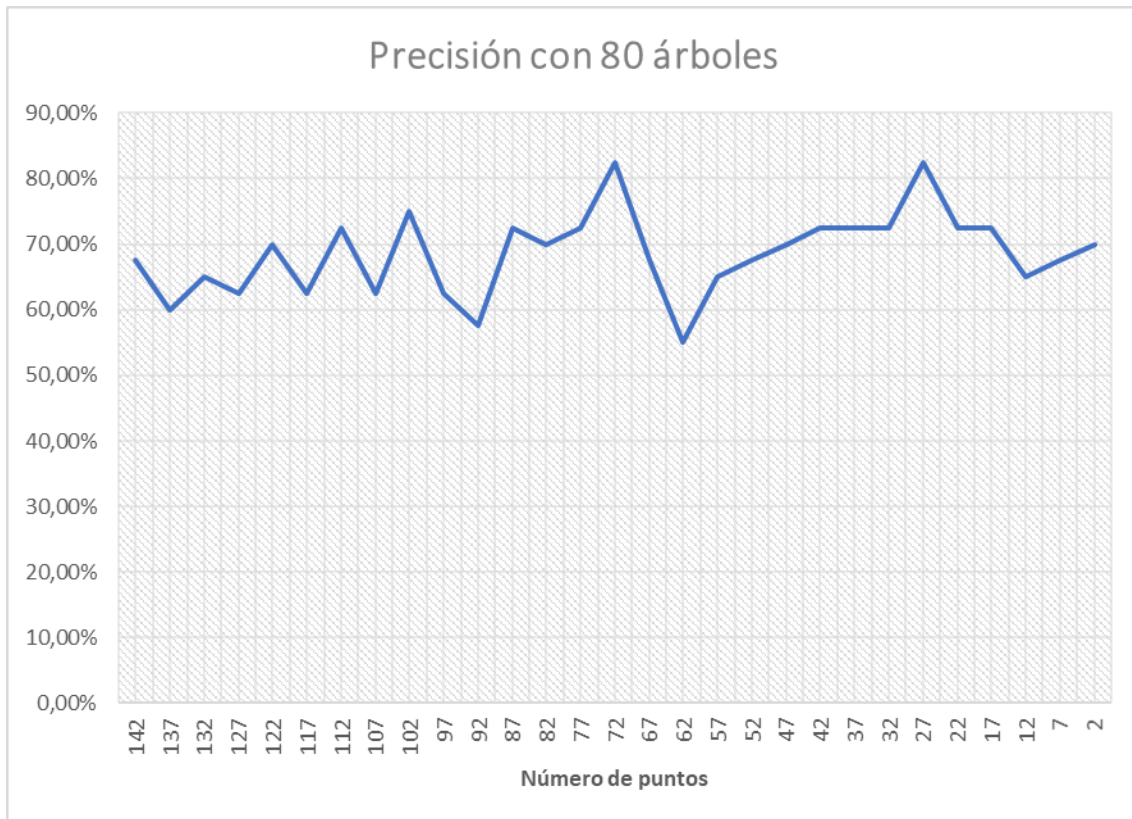


Fig. 3.11 Precisión del clasificador para 80 árboles según se le va quitando los atributos menos importantes.

En este otro gráfico se puede ver la variación de la precisión del clasificador según se le van quitando los puntos menos importantes. Esta vez, el clasificador se ha configurado con 80 árboles, que es con el que se ha obtenido el segundo mejor porcentaje de precisión. En este segundo caso, se pueden apreciar cómo los resultados son peores a los obtenidos con 70 árboles, por lo que esta configuración en el clasificador quedaría descartada. Según se van quitando los puntos los resultados empeoran, y no empiezan a mejorar hasta que quedan los 52 puntos más relevantes. Se puede ver un máximo con los 22 puntos que mayor importancia tienen, cuya precisión, en este punto, es de un 77.5%. Por lo tanto, esta configuración quedaría descartada, ya que estaría por debajo del 80% obtenido con 70 árboles.

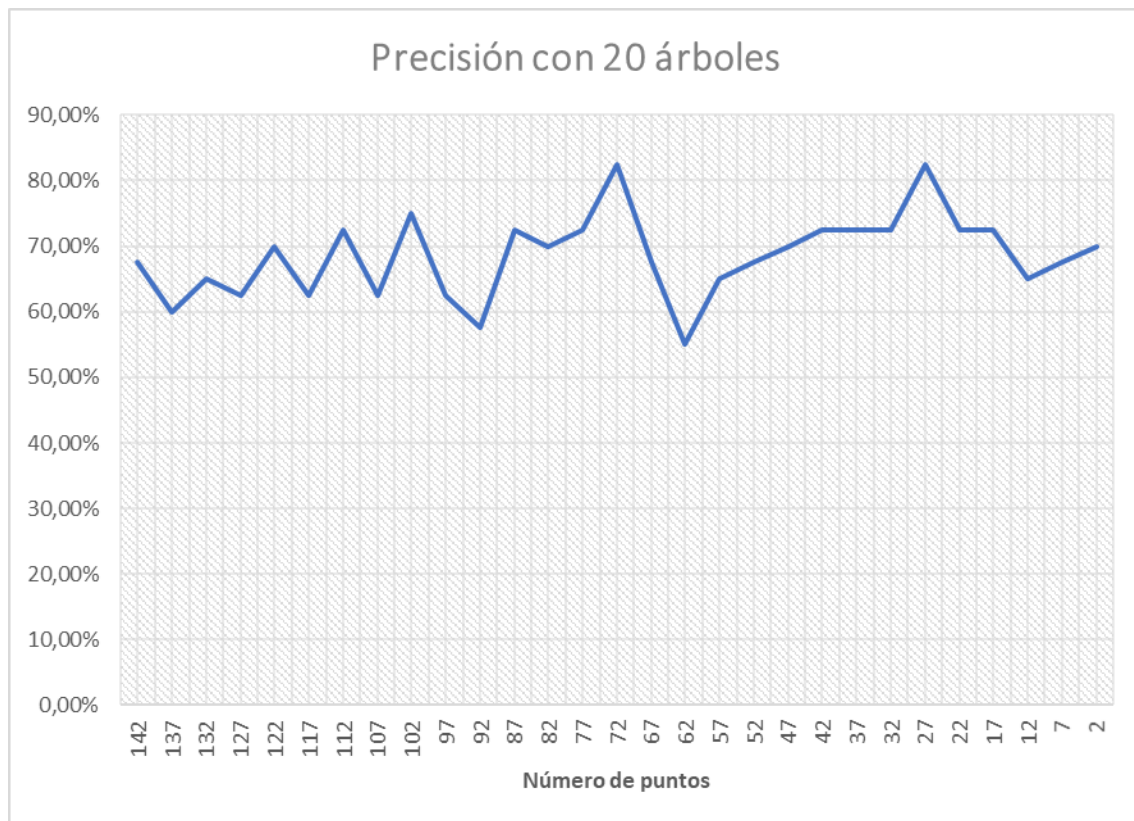


Fig. 3.12 Precisión del clasificador para 20 árboles según se le va quitando los atributos menos importantes.

Finalmente, en este último gráfico, aparece la precisión según se le van quitando los puntos menos importantes para una configuración del clasificador con 20 árboles, cuya precisión de un 67.5% fue la tercera mejor que se obtuvo. Si se observan los resultados obtenidos se puede ver cómo la precisión va fluctuando, teniendo subidas y bajadas, hasta llegar a una precisión de un 82.5% para los 72 puntos más importantes, a partir de aquí decrece hasta llegar a un mínimo de un 55% de precisión para los 62 puntos más importantes y finalmente se vuelve a obtener una precisión de un 82.5% para los 27 puntos más importantes. Este resultado de un 82.5% de precisión del clasificador es un resultado bastante bueno, el cual se ha conseguido incrementar un 17,4% el valor obtenido antes de seleccionar los puntos y configurar el clasificador.

En los tres casos, se puede ver cómo el hecho de quitar los puntos que tienen menor importancia no tiene por qué mejorar la precisión del clasificador, cuando parecería más lógico que el resultado obtenido por cada caso, fuera un incremento de la precisión, hasta llegar a un máximo, y a partir de ahí, el resultado debería ir decreciendo, debido a que el clasificador cada vez tendría menos información. Una posible explicación sería que, Spark realice algún tipo de agrupación de los puntos, por lo tanto, si se elimina un punto que aparentemente por sí solo no es importante, podría estar afectando a esta agrupación y cómo resultado provocando esta fluctuación de la precisión.

Finalmente, viendo los resultados obtenidos por cada uno de los casos, se puede decir que el mejor caso ocurre cuando se configura el clasificador con 20 árboles, dónde se ha obtenido un 82.5% de precisión para los 72 y 27 puntos más importantes. Por lo que solo queda por escoger si seleccionar los 72 o los 27 puntos más importantes. Al final, se escogió el caso de los 27 puntos más importantes, ya que se obtuvo el mismo resultado sin necesidad de pasarle tanta información. Por lo tanto, al elegir estos 27 puntos, el clasificador estaría seleccionando los puntos que se podrán ver en las imágenes que aparecerán a continuación.

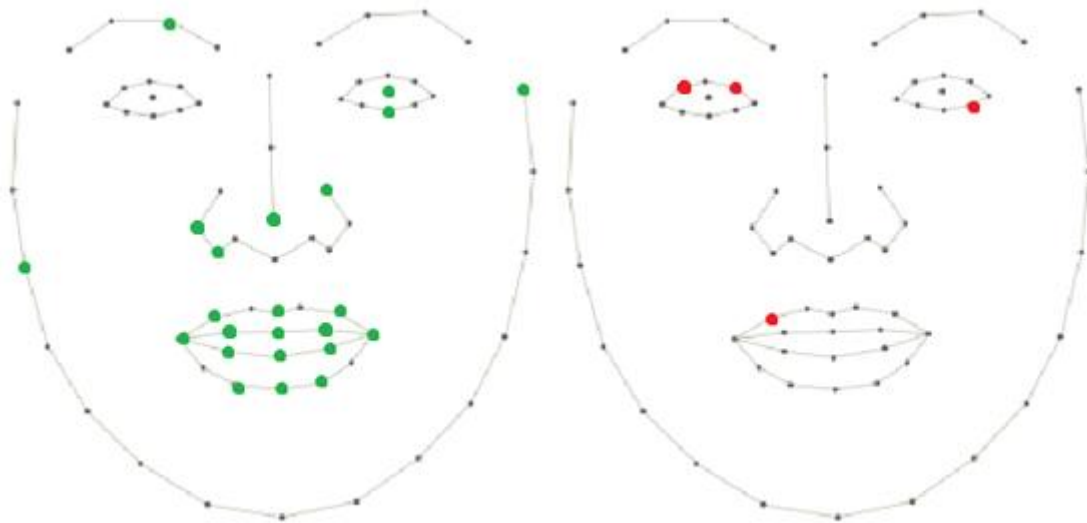


Fig. 3.13 En el rostro de la izquierda aparecen las coordenadas y 's de la cara que se han seleccionado, en el rostro de la derecha aparecen las coordenadas x 's seleccionadas.

En las dos imágenes que se pueden ver encima, aparecen los puntos que se han seleccionado para que el clasificador escoja los sentimientos de las caras. En el rostro de la izquierda, los puntos de color verde son las coordenadas y 's y en la imagen de la derecha, los puntos de color rojo son las coordenadas x 's de la cara. Por lo tanto, se puede apreciar cómo para el clasificador son más importantes para clasificar las caras, las coordenadas y 's, ya que, de los 27 puntos, 23 lo son. También se puede ver cómo de los 27 puntos, 15 están situados en la boca. Por lo tanto, estos resultados tienen sentido, tal y cómo se ha comentado antes, para saber si una persona está contenta o triste, hay que fijarse en los labios, ya que si en ellos hay una sonrisa significa, que la persona está contenta, y de lo contrario significaría que esta triste.

3.6 *Implementar y ejecutar modelos*

Una vez ya se ha creado y analizado el modelo, solo queda un último paso, la implementación y ejecución de modelos.

En este último paso, se deberá de implementar el modelo escogido en el anterior paso en un entorno de producción. Por lo tanto, se deberá de implementar en la aplicación Web, que se explicará más adelante, el modelo generado con Spark, con el clasificador Random Forest, configurado con 20 árboles, y con los 27 puntos más importantes.

CAPÍTULO 4. APLICACIÓN WEB

Una vez ya se tiene el modelo y se ha realizado el análisis de los datos, se realizará la aplicación. El objetivo de esta aplicación será simular una red social, cómo por ejemplo Twitter y Facebook, donde los usuarios serán capaces de escribir mensajes, o tweets en el caso de Twitter, en los cuáles se le podrá añadir una imagen al mensaje, además de permitir el uso de hashtags tal y cómo hacen las dos redes sociales comentadas.

Hoy en día se pueden ver cómo la gente usa las redes sociales para dar una opinión sobre cualquier tema, utilizando hashtags, por lo que utilizando la red social que se pretende crear, se pueda a partir de las imágenes que se añadan a la página, saber cómo está afectando ese tema en concreto a los usuarios que utilicen esta página. Esto permitirá a las empresas, por ejemplo, saber si la gente está contenta o a disgusto con sus productos, o en el caso de unas elecciones, permitiría saber si la gente está de acuerdo con un partido político o no, o se podría implementar en otros usos, ya que el uso de los hashtags, puede ser para referenciar cualquier cosa.

Para la realización de esta página Web, será muy importante definir las tecnologías que se van a implementar, el diseño que deberá tener y por último su implementación y resultado final.

4.1 Tecnologías

Cómo se ha explicado, se realizará una aplicación Web, la cual tiene que permitir simular una red social, por lo tanto, cómo toda página Web, se deberá implementar tecnologías para el lado cliente (front-end) y para la parte del servidor (back-end).

Las tecnologías que se implementarán en la parte back-end de la aplicación son las siguientes:

- **NodeJS:** Se trata de un entorno de ejecución para JavaScript, que utiliza un modelo de operaciones E/S sin bloqueo y orientado a eventos [19]. Esta tecnología es adecuada para páginas de video en streaming, aplicaciones de una única página, aplicaciones que necesiten recibir datos en tiempo real y API's basadas en JSON [20]. NodeJS permite manejar una gran cantidad de peticiones por segundo gracias a que ejecuta las operaciones de manera asíncrona. Al realizar las distintas funciones que pueda tener la aplicación al

mismo tiempo permite realizar las peticiones en un menor tiempo de respuesta que otros entornos, cómo por ejemplo una aplicación realizada con Java.

Por lo tanto, esta tecnología será útil para la aplicación Web que se pretende realizar, ya que, cómo se ha explicado se tratará de una red social, por lo tanto, el servidor ha de ser capaz de manejar varias peticiones por segundo, debido a que se espera que, como ocurre en todas las redes sociales, la utilizarán un gran número de usuarios, y las peticiones que se realicen estarán en formato JSON, por lo que es más cómodo si tanto en el lado cliente cómo en el servidor el lenguaje está escrito con JavaScript y que la página Web que se utilizará será una página única, por lo que es otro punto a favor de la utilización de esta tecnología. Además, que gracias a la utilización de esta tecnología permitiría la implementación de video en streaming, que es una de las funcionalidades que utilizan varias redes sociales cómo Facebook.

- **MongoDB:** Se trata de una base de datos NoSQL, que permite ser más ágil y escalable. Esta tecnología permite modificar los esquemas a medida que las aplicaciones van evolucionando. Otra ventaja que tiene es la tolerancia a fallos, lo que permite tener fiabilidad y además flexibilidad operativa [21].

La implementación de MongoDB, permitirá que la aplicación sea tolerante a fallos. Además de que gracias a su escalabilidad permitirá modificar los distintos esquemas, ya que en esta primera versión de la página se han tenido en cuenta solo la inclusión de imágenes en los mensajes, pero a lo mejor en un futuro, se desea añadir videos a los mensajes, o añadir ciertos datos a los usuarios, esto no se conseguiría con una base de datos SQL, las cuales son más restrictivas a posibles modificaciones de cada una de sus tablas.

- **EclairJS:** Se trata de una tecnología que permite la utilización de Spark en JavaScript creada por IBM. Esta tecnología cuenta con un componente cliente y otro componente servidor. La parte cliente de EclairJS permite la utilización de la API de Spark en NodeJS y JavaScript, mientras que el componente que actúa como servidor es responsable de exponer el modelo de Apache Spark a JavaScript, permitiendo aprovechar las capacidades de computación de Spark [22].

En la figura 4.1 se puede ver el funcionamiento de esta tecnología, dónde en Web Browser, estarían todas funcionalidades de la parte front-end. En la siguiente capa, estaría la parte back-end de la aplicación; en esta capa se encuentra el componente cliente de EclairJS, que permitirá programar Spark en NodeJS. Mientras que en las siguientes capas se encontraría los componentes de servidor de EclairJS, dónde están todas las funcionalidades que permite Spark. Es decir, que para que funcione Spark en NodeJS, se deberán programar todas las funcionalidades, y cuando se ejecute el código, esas líneas de código que se hayan escrito realizarán peticiones al componente servidor de EclairJS, para ejecutar el código que permita realizar las

funcionalidades de Spark. Para el componente servidor de EclairJS, será necesaria la implementación de otra tecnología, concretamente Docker, ya que es donde estarán todas las funcionalidades de Spark.

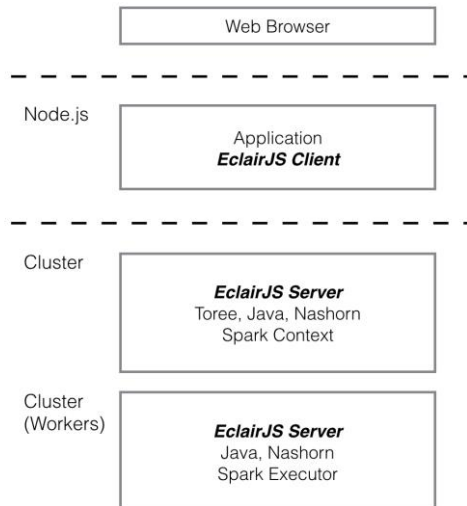


Fig. 4.1 Componentes de EclairJS. Fuente: <https://eclairjs.github.io/>.

Cómo se ha explicado, esta tecnología permitirá la utilización de Spark con NodeJS, ya que Spark no permite la utilización de sus funcionalidades con JavaScript. Sabiendo que NodeJS permite una mayor velocidad de respuesta y que Spark es una de las tecnologías de BigData más rápidas, IBM pensó en que NodeJS pudiese utilizar Spark con la ayuda de esta tecnología. Por lo tanto, se decidió implementar esta tecnología, ya que es la única que permitía la utilización de Spark en NodeJS.

- **Docker:** Esta tecnología es de código abierto y su funcionalidad es automatizar el despliegue de aplicaciones dentro de contenedores de software [23][24], es decir, permite ejecutar aplicaciones sin necesidad de instalar software adicional o permite ejecutarlas en un sistema operativo que no se permitía su ejecución. Su uso en la aplicación se debe al componente servidor de EclairJS, esta tecnología permitirá tener instalado Spark, sin necesidad de instalarla, ni de instalar software adicional que dificulte su instalación por temas de incompatibilidades de versiones. Por lo tanto, el componente cliente de EclairJS realizará las peticiones sobre esta plataforma, donde estará el componente servidor de EclairJS.

Las tecnologías que se implementarán en la parte front-end de la aplicación son las siguientes:

- **AngularJS:** Se trata de un framework de JavaScript de código abierto, mantenido por Google, que permite la creación de aplicaciones web de una sola página [25] [26]. Esta tecnología permitirá que el paso de una página a otra sea cómo si se estuviera en la misma, evitando tiempos de carga, esto permite una mayor rapidez a la hora de mostrar las páginas, ya que no ha de cargar todas las librerías y estilos cada vez que se pasa de una página a otra, esto sólo ocurre la primera vez que se carga la página.

- **JQuery:** Se trata de un software libre y de código abierto que permite simplificar la manera de interactuar con los documentos HTML, pudiendo modificar el código, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX [27] [28]. El uso de esta tecnología tiene que ver con las tecnologías utilizadas para el reconocimiento y de detección de caras, es decir, para JQuery.faceDetection y Cmltrackr, ya que su funcionamiento requiere de esta tecnología.

4.2 Diseño

Una vez ya se han explicado qué tecnologías se van a implementar en la aplicación, se deberá explicar el diseño de la aplicación Web. Cómo ya se ha dicho, se desea crear una página Web que simule una red social, la cual permita a los usuarios crear mensajes, en los cuales podrán subir imágenes, y con las imágenes, que contengan una cara, poder analizar el sentimiento y poder saber la opinión de la gente a partir de la imagen.

Lo primero que se deberá diseñar de la página, será la estructura de las bases de datos. A continuación, se explicará detalladamente su estructura:

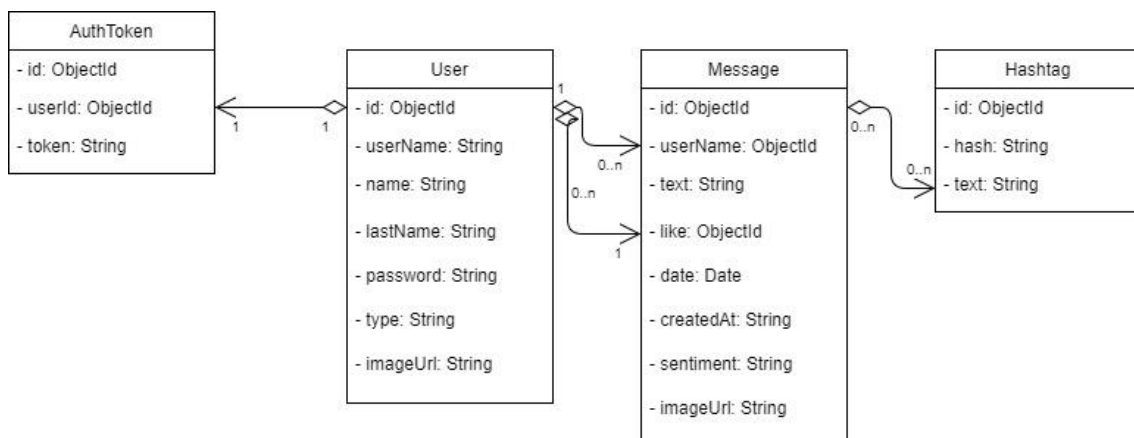


Fig. 4.2 Diagrama de clases UML.

En la figura 4.2, se puede apreciar la estructura que tendrá la base de datos de la aplicación mediante el uso de un diagrama de clases. Para permitir que haya usuarios se ha creado la clase User, que contendrá un identificador de usuario (id), el cual permitirá diferenciar saber el usuario sin necesidad de saber el nombre de este, el Nick del usuario (userName), al igual que pasa con el identificador, el valor de este campo será único, es decir, no se podrá haber más de un usuario con el mismo Nick, este campo permitirá a los usuarios iniciar sesión, también se incluye el nombre (name) y el apellido del usuario (lastName), con los cuales los usuarios indicarán su nombre y apellidos, en caso de ser un usuario, y en caso de ser una empresa el nombre de esta. Para que los usuarios puedan tener un inicio de sesión estos, además del Nick de usuario, han de tener un campo para escribir su contraseña (password), por lo tanto, se añadió este cuyo valor se guardará mediante un hash, de esta forma se evitará que mirando las bases de datos alguien pueda saber la contraseña de un usuario. Por último, se encuentran los campos de tipo de usuario (type) y de imagen de perfil (imageUrl), el primero tendrá dos posibles valores, o bien es un usuario (user) o es una empresa (business), dependiendo del tipo de usuario sea podrá tener unas funcionalidades o no, en cuanto al campo imageUrl, se trata de la imagen de perfil, dónde en caso de que no se inserte ninguna imagen al registrarse, se asignará una imagen por defecto. A excepción del identificador, que se trata de un objeto de id, los demás campos serán cadenas de texto (string).

Cuando los usuarios inicien sesión se les asignará un token, el cual les permitirá mantener la sesión abierta y no tener que estar registrándose cada dos por tres. Esto se consigue con la clase AuthToken. En esta clase, a parte de un identificador (id), se tiene el token, cuyo valor es una cadena de texto (string), y el identificador de usuario (userId), cuyo valor es un objectId, ya que el valor de este campo tiene relación con el campo id de la clase User. La relación entre estas dos tablas es una relación 1 a 1, debido a que un usuario solo podrá tener un token por sesión. Cuando el usuario cierre la sesión, este se borrará, y en caso de que se inicie sesión en otro explorador o PC, el token será modificado de forma que si vuelve al otro explorador o PC, al hacer cualquier funcionalidad, se le pedirá que inicie la sesión. Esta clase permitirá autenticar el usuario.

Cada usuario podrá crear tantos mensajes cómo desee, por lo tanto, la clase User tendrá una relación "1 a muchos" con la clase Message. Esta clase, permite la creación de mensajes. En ella encontramos, al igual que en las demás clases, un identificador de mensaje (id), el cual permitirá diferenciar el mensaje de los otros. El campo userName, será el identificador del usuario que haya creado el mensaje, permitiendo la relación entre las dos clases, es por esto que el valor de este campo es un objectId. Al tratarse de un mensaje, este campo ha de tener un campo que sea de texto (text), el valor de este campo será una cadena de texto (string). En estos mensajes, los usuarios, al igual que ocurre en las redes sociales más conocidas, podrán dar me gusta (like), cuyo valor será un array de identificadores de usuarios, de esta forma se asegurará de que un usuario no dé me gusta a un mensaje más de una vez, por lo tanto,

habrá una relación 1 a muchos, donde un mensaje podrá tener muchos usuarios que puedan dar me gusta al mensaje. También se han añadido dos campos de fecha, el primero (date), es la fecha en la que se ha creado el mensaje con el formato de fechas con el que guarda MongoDB, mientras que el segundo, cuyo valor es un string, es la fecha de forma que cualquier usuario pueda entender cuando se creó. Por último, encontramos el campo sentiment, el cual será una cadena de texto (string), y permitirá indicar el sentimiento de la imagen que tenga ese mensaje, cuyo campo es imageUrl, cuyo valor es un string e indicará donde está situada la imagen en el servidor.

Finalmente, habrá una clase Hashtag, de esta forma se sabrán los distintos hashtags que los usuarios hayan creado, y ver los mensajes que hay con ese hashtag. Al igual que en todas las clases, esta contendrá un identificado de hashtag (id), el nombre del hashtag (hash), cuyo valor será un string, y, por último, un campo donde irán todos los identificadores de los mensajes (text), creando una relación muchos a muchos con la clase Messages.

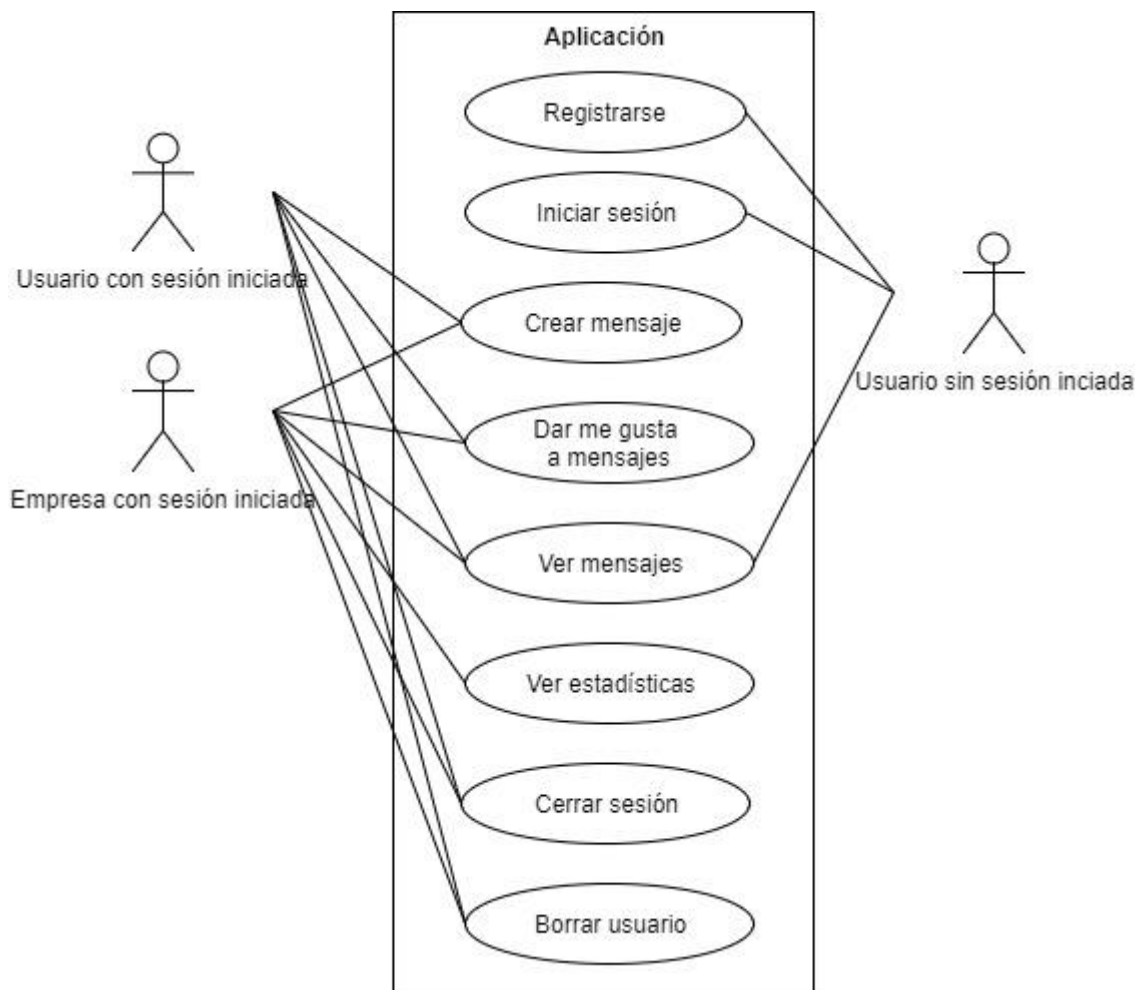


Fig. 4.3 Diagrama de casos de uso de la aplicación

En la figura 4.3, se puede apreciar un diagrama de casos de uso de la aplicación, en este diagrama se explican las diferentes acciones que podrán realizar cada uno de los usuarios que utilicen la página web.

En esta aplicación se espera que haya tres actores, los cuáles son usuarios sin sesión iniciada, usuario con sesión iniciada y el último sería un usuario de tipo empresa con sesión iniciada. Por lo tanto, un usuario que no haya iniciado sesión podrá ver los mensajes que los usuarios hayan escrito, podrá registrarse en la página introduciendo los datos que se precisen para el tipo de usuario que sea y también podrá iniciar sesión, una vez inicie sesión pasaría a uno de los otros dos casos posibles que haya, dependiendo de si el usuario es de tipo empresa o no.

Una vez, el usuario haya iniciado sesión, encontramos los otros dos casos, en el caso de que el usuario no sea una empresa, podrá ver los mensajes que los demás usuarios hayan escrito, además de poderles dar me gusta, permitiendo así realizar una de las funcionalidades más famosas que tienen todas las redes sociales. Otra de las funcionalidades que podrá realizar un usuario con sesión iniciada será la posibilidad de poder escribir mensajes, en los cuáles podrá añadir hashtags y añadir una imagen, con la que, en caso de que haya una cara, permitirá saber el sentimiento de esta. Por último, el usuario tendrá la posibilidad de borrar su cuenta, para así borrar todos los datos que haya podido dar para utilizar la aplicación o podrá cerrar la sesión.

Finalmente encontramos el último caso de uso, provocado por un usuario que haya iniciado sesión y sea de tipo empresa. Este usuario podrá ver los mensajes, dar me gusta, crear mensajes y además ver las estadísticas que estos mensajes proporcionen, es decir, a partir de los hashtags que hayan escrito los usuarios, y las imágenes que haya en los mensajes con esos hashtags, las empresas serán capaces de ver el sentimiento que tiene la imagen en ese mensaje, y además saber la cantidad de mensajes que hay con cada uno de los sentimientos, permitiendo así saber cómo está afectando ese tema a los usuarios. Finalmente, al igual que se podía hacer un usuario con sesión iniciada, las empresas podrán borrar la cuenta y cerrar la sesión.

4.3 Implementación y resultado

Finalmente, después de haber especificado las tecnologías que se van a implementar y el diseño que deberá de tener la página web, se deberá realizar la implementación de ambas cosas y ver el resultado final.

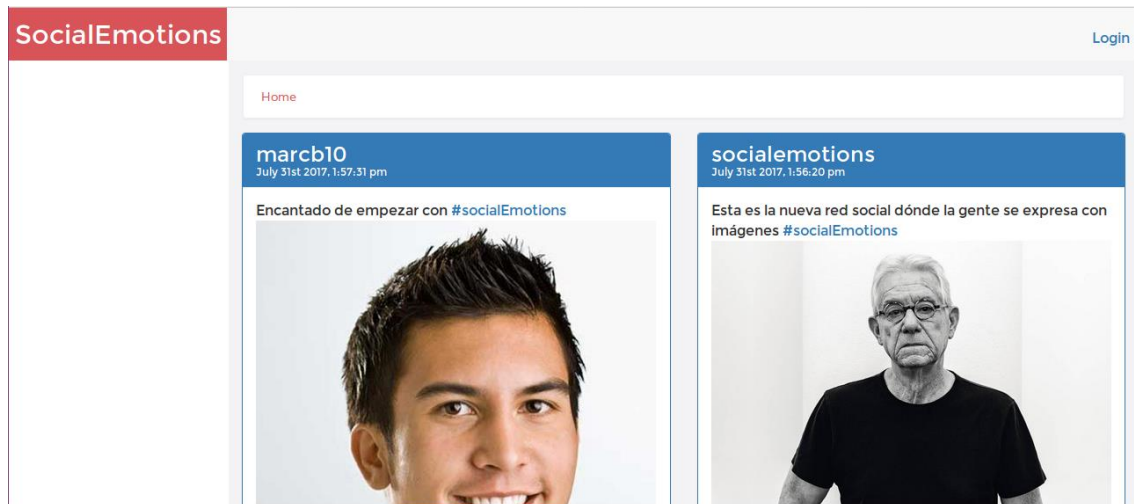


Fig. 4.4 Página principal de la aplicación.

En la figura 4.4, se puede ver la página principal de la aplicación dónde aparecerían los mensajes que los usuarios hayan ido escribiendo. Estos mensajes irán ordenados de los más recientes a los más antiguos (más adelante se hablará con más detalle de los mensajes). También se puede apreciar cómo en la parte superior izquierda de la página estaría el nombre de la página, que en este caso es "SocialEmotions". En la parte superior derecha, se encuentra el botón de Login, con el cuál los usuarios podrán registrarse o iniciar sesión.

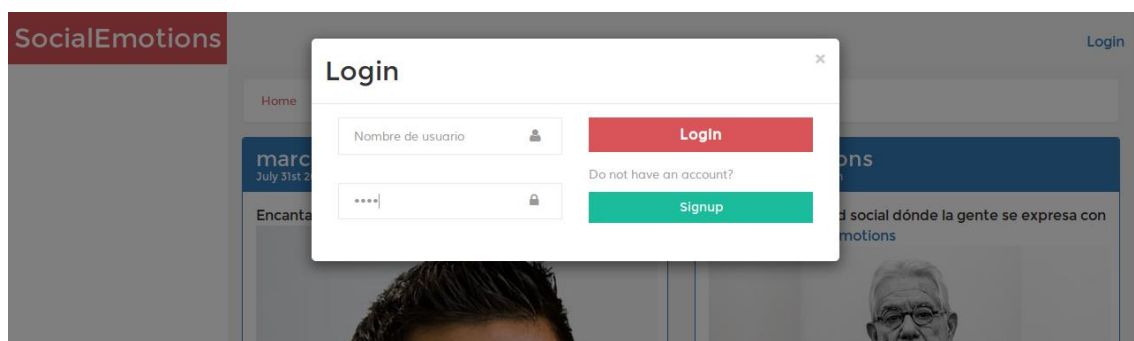
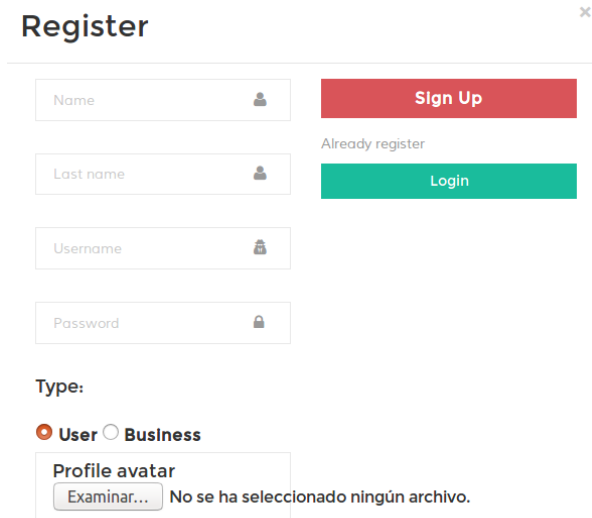


Fig. 4.5 Página principal con formulario para iniciar sesión.

Al apretar el botón de login, que había en la parte superior derecha, se abrirá una ventana, encima de la página principal, tal y como se aprecia en la figura 4.5, dónde aparecen dos campos de texto, el primero para introducir el nombre de usuario (username) y el segundo, para introducir la contraseña, que cómo se puede ver en la imagen, todo lo que se escriba estará oculto para proporcionar seguridad. En la parte derecha, se nos da dos opciones, o bien

iniciamos sesión, una vez introducidos los dos campos anteriores, apretando el botón login, o bien se crea un usuario nuevo, apretando al botón signup.

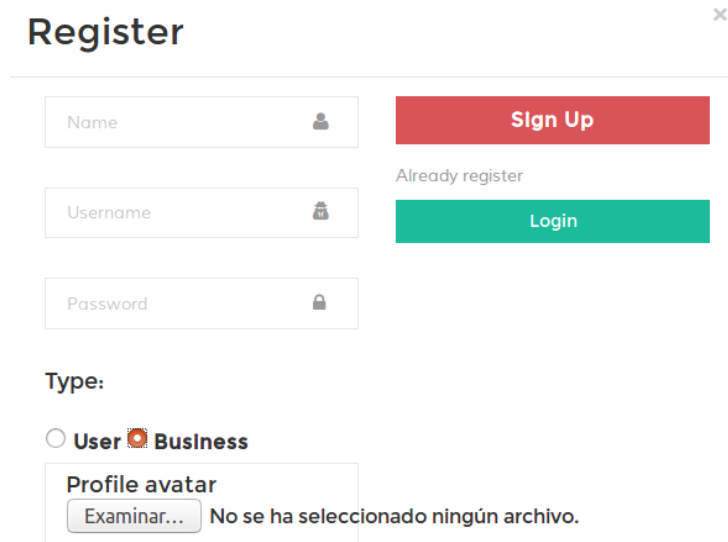


The image shows a 'Register' form with the following elements:

- Title:** Register (with a close button 'x' in the top right corner).
- Form Fields:**
 - Name:** Input field with a person icon.
 - Last name:** Input field with a person icon.
 - Username:** Input field with a person icon.
 - Password:** Input field with a lock icon.
 - Type:** Radio buttons for **User** (selected) and **Business**.
 - Profile avatar:** A section with a button labeled 'Examinar...' and the text 'No se ha seleccionado ningún archivo.'
- Buttons:**
 - Sign Up:** A red button located to the right of the Name and Last name fields.
 - Login:** A green button located to the right of the Username and Password fields.
- Text:** 'Already register' is displayed above the Login button.

Fig. 4.6 Ventana para registrar usuario.

Al apretar el botón signup, aparece en la misma ventana que había antes, el siguiente formulario, este formulario es para el caso de que sea un usuario y no una empresa. En este formulario, se pedirá que el usuario introduzca su nombre, su apellido, un nombre de usuario y una contraseña, en caso de que lo desee el usuario podrá añadir una imagen apretando el botón que hay en la parte inferior, debajo de profile avatar. Si el usuario no utiliza una imagen propia, su imagen de perfil será una imagen por defecto.



The screenshot shows a 'Register' window with a close button (x) in the top right corner. It contains three input fields: 'Name' with a person icon, 'Username' with a lock icon, and 'Password' with a lock icon. To the right of these fields are two buttons: a red 'Sign Up' button and a green 'Login' button. Below the input fields, there is a section labeled 'Type:' with two radio buttons: 'User' (selected) and 'Business'. Below this is a 'Profile avatar' section with an 'Examinar...' button and the text 'No se ha seleccionado ningún archivo.'

Fig. 4.7 Ventana para crear un usuario de tipo empresa.

En caso de que el usuario que se desee crear sea el de una empresa, las opciones que se dan son las mismas, a excepción de que en el formulario no está la opción para introducir los apellidos, ya que, siendo una empresa, no es útil este campo, solo con el nombre y un nombre de usuario ya habría suficiente. En ambos casos, tanto para registrar un usuario, como para registrar una empresa, se dan dos opciones, o bien se registra, apretando el botón que tiene como texto Sign Up, o bien se vuelve a la ventana de inicio de sesión apretando el botón de login.

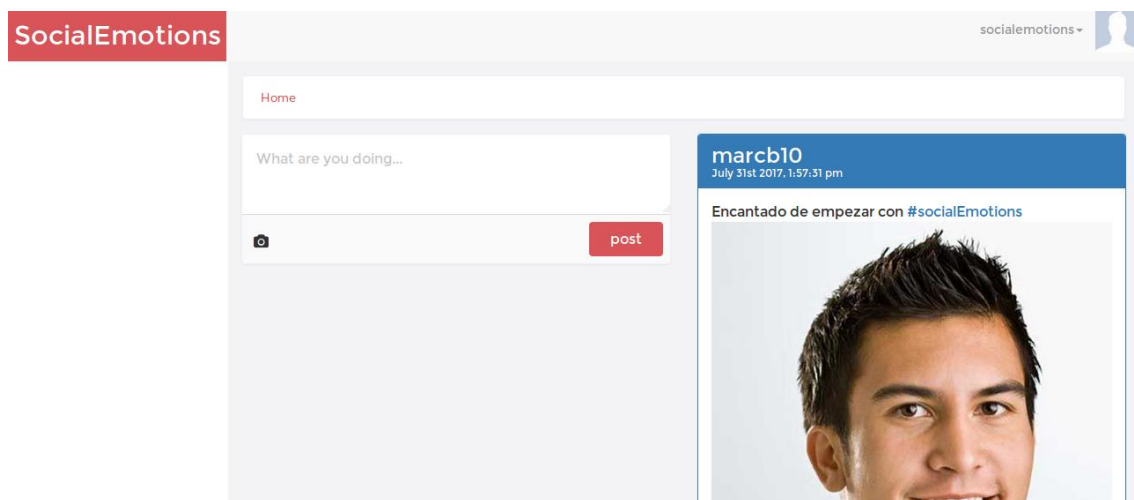


Fig. 4.8 Página principal con sesión iniciada.

Al iniciar sesión, como se puede apreciar en la figura 4.8, aparte de los mensajes, se permite la posibilidad de añadir un mensaje nuevo. Aquí se podrá añadir el texto, junto con los hashtags que el usuario desee utilizar. En la parte inferior, apretando al icono de la cámara se podrá añadir una imagen al mensaje. Finalmente el usuario publicará el mensaje apretando el botón post. También se puede ver cómo ha cambiado la parte superior derecha: dónde antes se tenía el botón de login, ahora, aparece el nombre de usuario y la imagen de perfil, que en este caso es una por defecto.

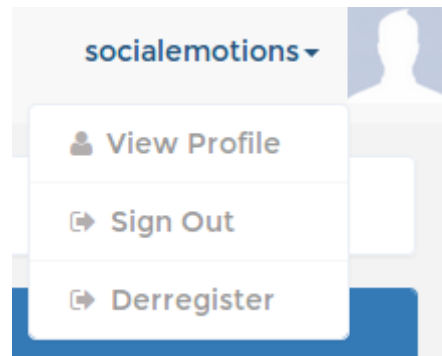


Fig. 4.9 Menú de opciones de usuario.

Al hacer click sobre la foto o el nombre de usuario, se permitirá las funcionalidades que se pueden ver en la figura 4.9. Se puede ver cómo los usuarios tendrán un menú dónde se les permitirá ver su perfil, cerrar la sesión o borrar la cuenta.

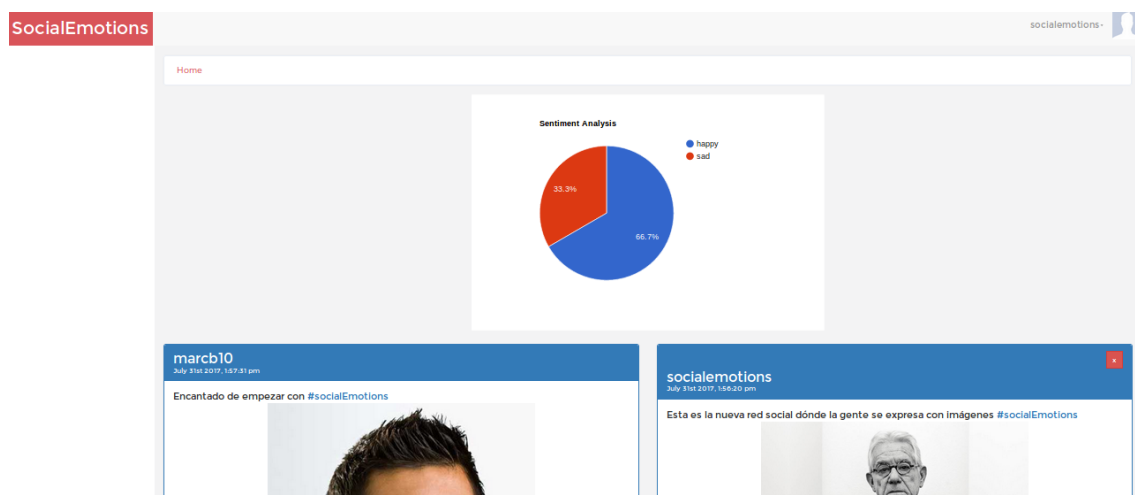


Fig. 4.10 Página de mensajes con el hashtag seleccionado.

Cuando se aprieta encima de un hashtag, la página muestra todos los mensajes que contienen ese hashtag: además, si el usuario es de tipo empresa, le aparecerá encima un gráfico donde pondrá el porcentaje de mensajes que tienen un sentimiento en concreto. En el caso de la figura 4.10, se puede ver, que para el hashtag socialEmotions, el 66.7% de los mensajes están contentos y el 33.3% restante están tristes. Con este análisis las empresas podrían sacar información de cómo está afectando un tema en concreto a partir de un hashtag.



Fig. 4.11 Formato de los mensajes.

En la figura 4.11, se puede ver un ejemplo de mensaje, dónde aparecería el texto, con los hashtags que se hayan escrito, y la imagen que se haya añadido. En caso de que el usuario que tenga la sesión iniciada sea el propietario de la imagen, le aparecería en la parte superior derecha del mensaje un botón para eliminarlo, si no es el propietario del mensaje, este botón no aparecerá. Debajo del contenido del mensaje, se permitirá a los usuarios dar me gusta al mensaje, y poder ver la cantidad de gente a la que le gusta el mensaje. Por último, en caso de que sea un usuario de tipo empresa, debajo de cada mensaje, le aparecerá en la parte inferior izquierda el sentimiento, en caso de que tenga, del mensaje.

El intercambio de páginas de la aplicación ocurrirá cómo si de una página se tratase, por lo que disminuye el tiempo de carga al solo tener que mostrar el código HTML de esa página y de los datos obtenidos por las peticiones que se realicen al abrir la nueva página web.

CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO

Una vez finalizado el proyecto, cabe decir que se han conseguido alcanzar todos los objetivos que se habían propuesto al principio del proyecto.

En este proyecto se han podido ver qué es tanto el Big Data y el Data Mining, cómo también algunas de las tecnologías que los implementan. Estas tecnologías, aunque ya consiguen predecir los valores con un gran porcentaje, todavía les falta un poco para ser altamente efectivas, ya que, aunque se lleva implementando desde hace algunos años, las tecnologías de Big Data son bastante recientes, las cuales, les faltan cosas por mejorar.

Otra cosa que se ha visto en este proyecto, son los pasos que se han seguido en un proyecto de minería de datos, dónde cada paso es importante, y los cuales, permiten, si el resultado no es bueno o no es el esperado, volver hacia atrás permitiendo realizar tantas pruebas como se desee. Cabe decir, que la realización de estos pasos es muy importante, a la hora de obtener una buena predicción, ya que, con un mal análisis de los datos, los resultados serían pésimos, cómo se ha podido ver en los primeros puntos del análisis que se ha realizado en este proyecto. Durante el análisis de los sentimientos de las caras, se han podido ver cómo se han tenido que buscar soluciones para evitar un mal resultado de los datos, aunque a veces, si hay una mala base, puede dificultar la mejora de estos. Un ejemplo en este proyecto de esto se encuentra en un mal procesador de caras, lo cual hace pensar que se debería mirar de mejorar en un futuro. A medida que se va obteniendo unos resultados buenos, se ha podido comprobar cómo no hay que quedarse con el primer resultado que se encuentre, sino que es muy importante, realizar distintas pruebas, con el fin de comprobar que ese resultado tiene un mayor margen de mejora, y sobre todo ajustar correctamente el clasificador para aumentar la precisión.

Cuando se analizaban las caras, se pudo comprobar la dificultosa tarea que lleva detrás para conseguir unos resultados más que correctos, lo que lleva a pensar el trabajo que habrán tenido que realizar las grandes compañías que lo implementan como Google o Microsoft. Esta complejidad es debido, a que primero de todo, se han de tener todas las caras normalizadas, si no están normalizadas, no se obtendrían unos buenos resultados, ya que, estas estarían en posiciones diferentes, y otro punto bastante importante es que todas las caras han de estar en la misma posición, el hecho de que una cara este inclinada y las otras no, puede desvirtuar el resultado final. A pesar de estas complicaciones, la precisión que se ha obtenido es de un 82.5%, que es un resultado bastante bueno, si se tienen en cuenta los problemas con el detector de caras y a la poca cantidad de imágenes.

Uno de los mayores problemas que ha habido durante la realización del proyecto, es la utilización de la tecnología EclairJS. Al ser relativamente nueva, es muy propensa a errores, que los usuarios a día de hoy no han encontrado solución, lo cual hace que sea dificultosa su programación, y que conlleve una

gran pérdida de tiempo en intentar dar solución al error. Cabe destacar que uno de los mayores inconvenientes que tienen las tecnologías de Big Data y Data Mining es que no tienen una forma de depurar los errores, lo cual dificulta todavía más la resolución de problemas. Por último, se ha de decir que la documentación proporcionada por EclairJS es bastante escasa, y además su lectura es dificultosa.

Durante la realización de la aplicación, ha sido muy importante tener definido un diseño sólido, y, sobre todo, que permita añadir posibles mejoras en un futuro, sin un buen diseño, no se podría haber alcanzado el resultado que se deseaba.

De cara a un futuro, a parte de las cosas que se han ido diciendo durante la realización del proyecto, cómo por ejemplo, la mejora del procesador de imágenes, se podrían realizar algunas más, las cuales se explicarán a continuación:

- Ampliar el número de sentimientos que puedan aparecer en la cara. Es decir, actualmente sólo se mira de predecir que las caras estén o tristes o contentas, pero no se mira si la cara esta enfadada o el sentimiento que tiene esta es de sorpresa.

- Mirar de implementar el análisis de caras en video. El hecho de analizar las caras en un video permitiría obtener más información que en el caso de extraerlo de una imagen.

- Poder realizar análisis de los usuarios. Es decir, poder saber si una persona sufre de depresión o no, a partir de lo que publique en la red social. Para esto, se debería añadir la posibilidad de analizar textos.

- Realizar una implementación de video en streaming, con lo cual se podría implementar un análisis de los datos en tiempo real.

BIBLIOGRAFÍA

- [1] Microsoft Cognitive Services [en línea]. Microsoft. [Fecha de consulta: 23 de febrero de 2017]. Disponible en:
<https://azure.microsoft.com/es-es/services/cognitive-services/>
- [2] API Vision de Google Cloud [en línea]. Google. [Fecha de consulta: 23 de febrero de 2017]. Disponible en: <https://cloud.google.com/vision/?hl=es>
- [3] Big Data [en línea]. Wikipedia. [Fecha de consulta: 28 de mayo de 2017]. Disponible en: https://es.wikipedia.org/wiki/Big_data
- [4] Conceptos de minería de datos [en línea]. Microsoft. [Fecha de consulta: 28 de mayo de 2017]. Disponible en:
[https://msdn.microsoft.com/es-es/library/ms174949\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ms174949(v=sql.120).aspx)
- [5] Validar modelos de minería de datos [en línea]. Microsoft. [Fecha de consulta: 28 de mayo de 2017]. Disponible en:
[https://technet.microsoft.com/es-es/library/ms174493\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms174493(v=sql.105).aspx)
- [6] Minería de datos [en línea]. Wikipedia. [Fecha de consulta: 28 de mayo de 2017]. Disponible en: https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos
- [7] Andrés González. Conceptos básicos de Machine Learning [en línea]. Cleverdata. [Fecha de consulta: 28 de mayo de 2017]. Disponible en:
<http://cleverdata.io/conceptos-basicos-machine-learning/>
- [8] Regresión lineal [en línea]. Wikipedia. [Fecha de consulta: 29 de mayo de 2017]. Disponible en: https://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal
- [9] What is the difference between Big Data and Data Mining? [en línea]. Techopedia. [Fecha de consulta: 29 de mayo de 2017]. Disponible en:
<https://www.techopedia.com/7/29678/technology-trends/what-is-the-difference-between-big-data-and-data-mining>
- [10] MapReduce [en línea]. Wikipedia. [Fecha de consulta: 2 de junio de 2017]. Disponible en: <https://es.wikipedia.org/wiki/MapReduce>
- [11] Weka (aprendizaje automático) [en línea]. Wikipedia. [Fecha de consulta: 2 de junio de 2017]. Disponible en:
[https://es.wikipedia.org/wiki/Weka_\(aprendizaje_autom%C3%A1tico\)](https://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico))
- [12] Spark [en línea]. Apache. [Fecha de consulta 22 de febrero de 2017]. Disponible en: <https://spark.apache.org/>
- [13] Hadoop [en línea]. Wikipedia. [Fecha de consulta: 22 de febrero de 2017]. Disponible en: <https://es.wikipedia.org/wiki/Hadoop>

- [14] Storm [en línea]. Apache. [Fecha de consulta 22 de febrero de 2017]. Disponible en: <http://storm.apache.org/>
- [15] FaceDetection [en línea]. Jaysalvat. [Fecha de consulta: 26 de abril de 2017]. Disponible en: <http://facedetection.jaysalvat.com/>
- [16] Cmltrackr [en línea]. Github. [Fecha de consulta: 26 de abril de 2017]. Disponible en: <https://github.com/auduno/clmtrackr>
- [17] Adrián Rosebrock. Facial landmarks with dlib, OpenCV, and Python [en línea]. Pyimagesearch. [Fecha de consulta 8 de abril de 2017]. Disponible en: <http://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
- [18] Ensembles [en línea]. Apache. [Fecha de consulta: 12 de mayo de 2017]. Disponible en: <https://spark.apache.org/docs/latest/ml-lib-ensembles.html>
- [19] NodeJS [en línea]. Nodejs. [Fecha de consulta: 25 de febrero de 2017]. Disponible en: <https://nodejs.org/es/>
- [20] NodeJS Tutorial [en línea]. Tutorialspoint. [Fecha de consulta: 12 de julio de 2017]. Disponible en: <https://www.tutorialspoint.com/nodejs/index.htm>
- [21] MongoDB [en línea]. Mongoddb. [Fecha de consulta: 25 de febrero de 2017]. Disponible en: <https://www.mongodb.com/>
- [22] EclairJS [en línea]. Github. [Fecha de consulta: 25 de febrero de 2017]. Disponible en: <https://eclairjs.github.io/>
- [23] Docker [en línea]. Docker. [Fecha de consulta: 27 de febrero de 2017]. Disponible en: <https://www.docker.com/>
- [24] Docker (Software) [en línea]. Wikipedia. [Fecha de consulta: 12 de julio de 2017]. Disponible en: [https://es.wikipedia.org/wiki/Docker_\(software\)](https://es.wikipedia.org/wiki/Docker_(software))
- [25] AngularJS [en línea]. Angularjs. [Fecha de consulta: 25 de febrero de 2017]. Disponible en: <https://angularjs.org/>
- [26] AngularJS [en línea]. Wikipedia. [Fecha de consulta: 12 de julio de 2017]. Disponible en: <https://es.wikipedia.org/wiki/AngularJS>
- [27] JQuery [en línea]. Jquery. [Fecha de consulta: 26 de abril de 2017]. Disponible en: <https://jquery.com/>
- [28] JQuery [en línea]. Wikipedia. [Fecha de consulta: 12 de julio de 2017]. Disponible en: <https://es.wikipedia.org/wiki/JQuery>