# UNIVERSIDAD AUTÓNOMA DE MADRID

## ESCUELA POLITÉCNICA SUPERIOR


**Grado en Ingeniería Informática**

# TRABAJO FIN DE GRADO

## DESARROLLO DE UN SISTEMA DE ANÁLISIS DE IMÁGENES USANDO INFORMACIÓN DE REDES SOCIALES

**Santiago Gómez Aguirre**
**Tutor: Antonio González Pardo**
**Ponente: David Camacho Fernández**

**Mayo 2018**

# DESARROLLO DE UN SISTEMA DE ANÁLISIS DE IMÁGENES USANDO INFORMACIÓN DE REDES SOCIALES

**AUTOR: Santiago Gómez Aguirre**
**TUTOR: Antonio González Pardo**

**Grupo Applied Intelligence & Data Analysis (AIDA)**
**Dpto. Ingeniería Informática**
**Escuela Politécnica Superior**
**Universidad Autónoma de Madrid**
**Mayo de 2018**

# Resumen

Las redes sociales se han convertido hoy en día en una de las mayores fuentes de información pública que existen. Dada la cantidad de usuarios que diariamente utilizan este tipo de plataformas (como Facebook, Twitter, o Instagram), hacen que estas sean muy interesantes desde el punto de vista del análisis de datos. El objetivo de este Trabajo Fin de Grado (TFG) consiste en la construcción de un sistema capaz de extraer y analizar el contenido de una de estas redes sociales. De una manera más concreta, vamos a utilizar la red social Instagram, debido a que ha sido una de las que más crecimiento ha experimentado en los últimos años. El sistema desarrollado estará formado principalmente por tres elementos clave: un *crawler*, que permitirá acceder al contenido de Instagram en base a las opciones que le asignemos; el uso de la API Vision de Google Cloud, que permitirá extraer información más detallada de las imágenes analizadas; y un tercer módulo de análisis de esa información para poder extraer conocimiento de las imágenes. Es necesario destacar que, aunque ya existan trabajos que nos proporcionan estadísticas con información obtenida de esta red social, no hay muchos que lo lleven a cabo analizando el contenido de las fotografías de los usuarios o los *hashtags*.

# Palabras clave

Facebook, Twitter, Instagram, *crawler*, API Vision, Google Cloud, *hashtag*s

# Abstract

Nowadays, Social Networks (SN) have become one of the largest public information sources that exists. Given the number of users who use this type of platforms (such as Facebook, Twitter, or Instagram), makes SN a very interesting domain to perform data analysis tasks. The objective of this Degree's Final Project (DFP) is the development of a system able to extract and analyze the content of one of these social networks. More concretely, we are going to use the social network Instagram because it is the one with the biggest growth in last years. The system is mainly composed by three key elements: a crawler, which allows access to Instagram content based on the options that are given; the use of Google Cloud Vision API, that provides detailed information about the analyzed images; and a third analysis module that extracts knowledge from the images. It is neccesary to emphasize that, although there are already works that provide statistics with information obtained from this social network, there are not many that carry it out by analyzing the content of the user´s or hashtags´s photographs.

# Keywords

Facebook, Twitter, Instagram, *crawler*, API Vision, Google Cloud, *hashtag*s

## *Agradecimientos*

Este trabajo está dedicado a mi familia, en especial a mis padres por aguantarme y apoyarme durante todos estos años y los que están por venir. También me gustaría agradecérselo a todos mis amigos que han estado ahí durante toda la carrera, tanto en los buenos momentos como en los malos. Por último, una mención especial a Antonio por toda la ayuda y guía ofrecida en la realización de este trabajo.

# INDICE DE CONTENIDOS

# INDICE DE FIGURAS

# 1 Introducción

Instagram es una red social creada por Kevin Systrom y Mike Krieger que permite a sus usuarios subir fotografías y vídeos. Desde que fuera lanzada al mercado en 2010 para los sistemas iOS, ha ido incorporando nuevas opciones y se ha expandido a otros sistemas, como por ejemplo a Android en 2012, que le ha valido para ser una de las redes sociales más utilizadas actualmente, con más de 800 millones de usuarios activos en un mes [1]. Esto sumado a la gran variedad de perfiles de dichos usuarios, la convierte en una gran fuente de información pública. Actualmente se espera que Instagram siga experimentando un gran crecimiento debido a su sencillez y a su capacidad de innovación, en comparación con otras redes sociales más tradicionales.

Este trabajo pretende desarrollar un sistema capaz de recolectar la información que se encuentra en dicha red social, las fotografías. En ellas se puede encontrar una gran cantidad de datos, aunque encapsulados en un formato visual que nos dificulta las tareas de análisis en un primer momento sin ningún tipo de conversión. Para ello, es necesaria la utilización de tecnología de reconocimiento de imágenes que nos permita transformar esos datos a un formato mucho más manejable y que facilite las tareas de análisis. Por esta razón, en este proyecto se emplea una API REST que ha desarrollado Google, llamada API Vision [6], que permite llevar a cabo la clasificación de imágenes y la detección de diversos contenidos en su interior.

Con el análisis de los datos resultantes se puede obtener información muy variada tanto a nivel de un solo usuario como de un colectivo de usuarios más amplio, pudiendo llegar a clasificarlo por sectores geográficos, edades, etc. Esto permite tener una amplia gama de usos que afectan a diferentes niveles sociales, por ejemplo: detectar patrones en un usuario, realizar análisis de mercados que permitan llevar a cabo campañas de marketing más efectivas, etc. Para llevar a cabo estos usos se suelen aplicar técnicas de *clustering*, ya que realizan de forma automática las agrupaciones o clústeres de elementos en función de una medida de similitud entre ellos.

En este proyecto se aplican esas técnicas mediante el algoritmo DBSCAN [4] para realizar una breve demostración de los posibles usos que se pueden llevar a cabo con la información recolectada. La razón por la que se ha escogido el algoritmo DBSCAN es porque realiza el análisis basándose en la densidad de los clústeres, debido a esto resulta más preciso para este tipo de tareas que otros algoritmos de su estilo.

Es importante remarcar una serie de problemas que se nos plantean a la hora de realizar este trabajo. El primero es que trata de un proyecto muy ampliable, por lo que es necesario acotarlo. Al ser necesaria la utilización de técnicas de aprendizaje automático (incluidas en la API Vision de Google Cloud) para cotejar la información encapsulada en las fotografías, los datos obtenidos no están garantizados que sean completamente certeros. El manejo de los datos recolectados para llevar a cabo las técnicas de *clustering* requiere la realización de métricas que se ajusten al objetivo deseado en cada análisis, por lo que se ha optado por realizar una métrica muy sencilla para la demostración con el algoritmo DBSCAN. Por último, el proyecto siempre va a mantener una dependencia del *crawler* que se emplea para la recolección de datos y de la API Vision de Google Cloud para su correcto funcionamiento.

## 1.1 Motivación

Las redes sociales en la actualidad son unas de las mayores fuentes de información pública a las que se puede acceder, por lo que su recolección y análisis de datos permite obtener una amplia gama de usos. La motivación principal de este proyecto ha sido el poder realizar un sistema que permita a un usuario llevar a cabo esa recolección y análisis fácilmente, de esta forma luego puede hacer uso de esos datos cómodamente y aplicarlos en función de sus intereses.

Se ha escogido Instagram como red social de recolección de datos ya que muestra una gran tasa de crecimiento actualmente y se estima que continúe con esa progresión a medio plazo. En cuanto al análisis, se ha optado por realizarlo sobre las fotografías dado que es el material principal en el que se basa Instagram y del que más datos se pueden obtener. Para ello, se ha hecho uso de la potente API Vision que ha desarrollado Google y que nos ha permitido clasificar estos datos en etiquetas, ya que son el formato que mejor nos permite obtener la información de una manera generalizada.

## 1.2 Objetivos

Los objetivos del sistema desarrollado son:

- **Extracción de información de la red social.** Extraer las fotos de Instagram mediante el uso de un *crawler*.

- **Almacenamiento de fotografías.** Almacenar localmente las fotos obtenidas de la extracción ordenadamente en función del modo de descarga seleccionado (usuario o *hashtag*).

- **Extracción de etiquetas.** Analizar las fotos mediante la API Vision de Google para obtener todas las etiquetas que contienen.

- **Ordenación de las etiquetas**. Ordenar las etiquetas de cada conjunto de fotos analizado para facilitar su uso posterior, en un archivo estructurado.

- **Análisis de las etiquetas utilizando DBSCAN**. Utilizar el archivo resultante de la ordenación de las etiquetas de un conjunto de fotos para realizar *clustering* mediante el algoritmo DBSCAN.

- **Validación de las etiquetas.** Calcular el Coeficiente de Silhouette [14] con los resultados obtenidos de aplicar el algoritmo DBSCAN para determinar la validez de los datos.

## *1.3 Organización de la memoria*

La memoria consta de los siguientes apartados:

- **Estado del Arte (Apartado 2)**. En este apartado se introduce el estado del arte, así como la tecnología y elementos empleados en este trabajo.

- **Diseño y desarrollo del sistema (Apartado 3)**. Describe la arquitectura y el funcionamiento del sistema desarrollado.

- **Pruebas y resultados (Apartado 4)**. Muestra las pruebas realizadas al sistema para comprobar su correcto funcionamiento y los resultados obtenidos acompañados de imágenes ilustrativas.

- **Conclusiones y trabajo futuro (Apartado 5)**. Incluye un pequeño resumen del trabajo realizado acompañado de observaciones que se han realizado durante el desarrollo de todo el trabajo; así como unas sugerencias de mejora y ampliación del sistema presentado para futuros desarrollos.

# 2 Estado del arte

En este apartado se explica en qué consiste la Minería de Datos, o *Data Mining*, ya que este trabajo se incluye dentro de este campo de las ciencias de la computación. También se comentan las distintas fases del proyecto con la tecnología y técnicas empleadas en cada caso.

## *2.1 Minería de Datos*

La Minería de Datos es "el proceso de descubrir nuevas y significativas correlaciones, patrones y tendencias mediante la selección de grandes cantidades de datos almacenados en repositorios, utilizando tecnologías de reconocimiento de patrones, así como técnicas estadísticas y matemáticas" [9]. Este proceso puede dividirse principalmente en las siguientes etapas: selección y extracción de datos; preprocesamiento de datos; generación y aplicación del modelo de análisis; y evaluación del modelo.

En el proceso de la Minería de Datos, de forma general, los **datos** forman la materia prima bruta, de manera que, cuando el usuario les atribuye algún significado especial pasan a convertirse en **información**. Con esa información los especialistas buscan o elaboran un modelo para que el resultado de la interacción de ambos represente un valor agregado al cual denominamos **conocimiento** [20].

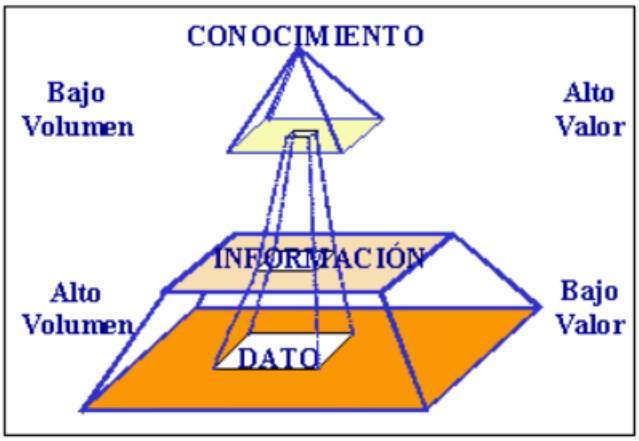


**Figura 1. Relación entre dato, información y conocimiento [20]**

### *2.1.1 Selección y extracción de datos*

Consiste en seleccionar y obtener el conjunto de datos con el que se va a trabajar. En nuestro proyecto, estos datos los obtenemos de la red social Instagram, donde los usuarios comparten imágenes y vídeos. Este contenido se puede clasificar en función del usuario que lo comparte o según los *hashtags* (etiquetas) que se le añaden.

Para poder obtener estos datos en nuestro trabajo hemos empleado un tipo de programa que se denomina *crawler*. Estos programas inspeccionan la World Wide Web haciendo uso de su estructura de grafo para moverse a través de las páginas y conectarse a un servicio web o API con el objetivo de extraer información de manera automática [10]. El *crawler* que hemos empleado es InstaLooter [16], el cual nos permite descargar cualquier fotografía o vídeo asociado a un perfil o *hashtag* de Instagram sin necesidad de acceder a la API de Instagram.

Existen otras aplicaciones para poder realizar la recolección de los datos de esta red social, como por ejemplo el uso de la API de Instagram (Instagram API Platform) [11], aunque actualmente se encuentra en proceso de deprecación y se fomenta el uso de la nueva API Graph (Instagram Graph API) [12] de la misma organización. El inconveniente de este tipo de API es que están ligadas a restricciones de acceso a los datos de forma abierta y además presentan restricciones a la hora de acceder a la información en forma de tiempos de espera.

### *2.1.2 Preprocesamiento de datos*

Esta etapa consiste en preparar el conjunto de datos inicial, obtenido de la extracción, para la aplicación de los distintos modelos de análisis que mejor se adapten al problema y datos que se manejan. De esta forma se pueden evitar problemas de pérdida de información, valores atípicos o errores en la clasificación de los datos.

En este proyecto el conjunto de datos inicial está encapsulado en las imágenes que se extraen de Instagram, por lo que el preprocesamiento de los datos se lleva a cabo mediante el análisis de estas imágenes con la API Vision de Google Cloud [6]. Esto permite obtener, entre otra información, las etiquetas que definen cada imagen. Seguidamente, se han procesado esas etiquetas hasta convertir cada imagen en un vector numérico que indica si dicha imagen posee las diferentes etiquetas o no. De esta forma el conjunto de datos queda lista para su análisis posterior con un modelo de análisis determinado.

### *2.1.3 Generación y aplicación del modelo de análisis*

En esta fase se pueden emplear distintas técnicas que se distinguen entre predictivas, descriptivas y auxiliares [13]. Las técnicas **predictivas** se caracterizan por especificar el modelo para los datos en base a un conocimiento teórico previo, después debe contrastarse el proceso de minería de datos antes de aceptarlo como válido. A este tipo de técnicas pertenecen, por ejemplo, las redes neuronales, los arboles de decisión, algoritmos genéticos, técnicas bayesianas, etc.

En cuanto a las técnicas **descriptivas**, se caracterizan por no suponer ningún modelo previo para los datos, ya que se crean partiendo del reconocimiento de patrones. A este tipo, pertenecen las técnicas de *clustering* (que son las que se emplean en este proyecto), técnicas de asociación y dependencia, las técnicas de reducción de la dimensión y de escalamiento multidimensional, etc.

Tanto las técnicas predictivas como descriptivas están enfocadas al descubrimiento del conocimiento embebido de los datos. Por último, las técnicas **auxiliares** son herramientas

de apoyo más superficiales y limitadas que se basan en técnicas estadísticas descriptivas, consultas e informes, y se enfocan en general hacia la verificación.

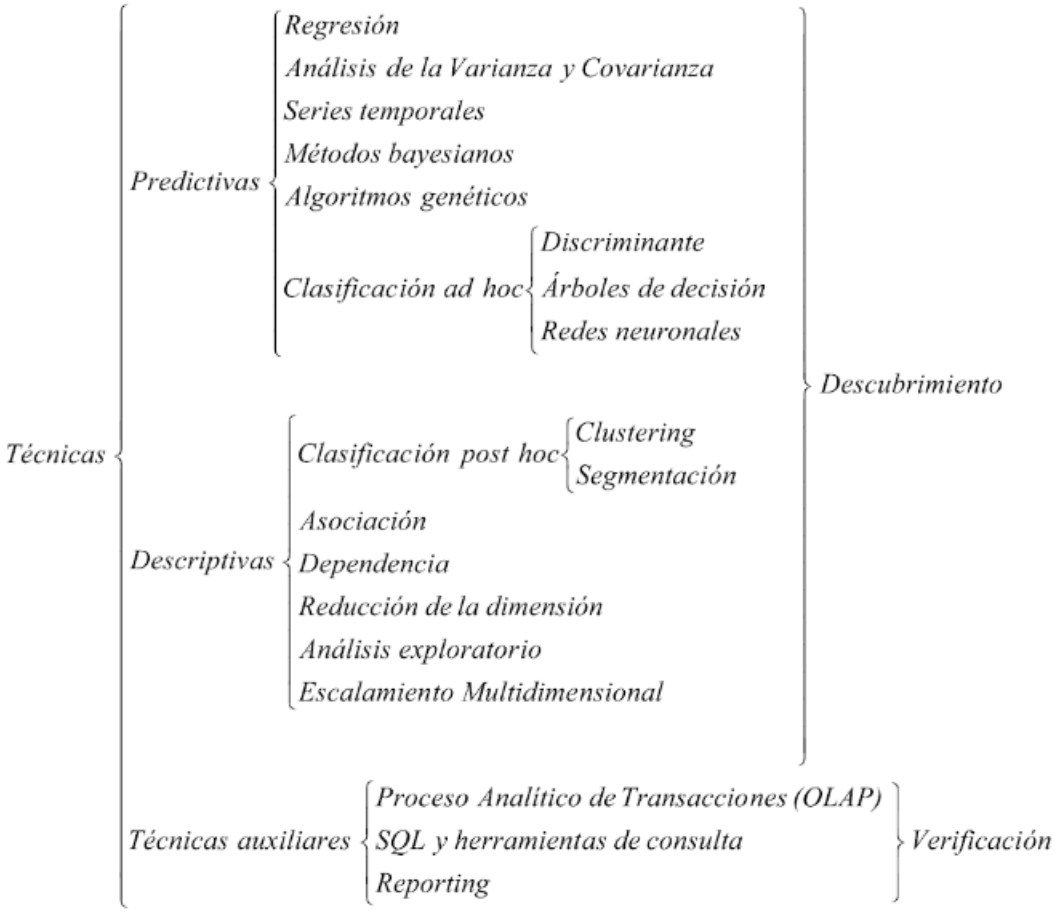


**Figura 2. Técnicas de Data Mining [13]**

### *2.1.4 Evaluación del modelo de análisis*

La evaluación de los modelos, en general, puede ser muy variada. En este trabajo se emplean técnicas de *clustering*, las cuales se caracterizan por ser especialmente sensibles a los parámetros de entrada, por lo que es difícil definir cuando el resultado de un agrupamiento es aceptable. Para la valoración de este tipo de modelos existen técnicas de validación que se pueden clasificar en dos categorías distintas: validación externa y validación interna. La principal diferencia que existe entre ellas es la utilización de información externa para la validación, es decir, información que no es producto de la técnica de *clustering* empleada [15].

Las técnicas de **validación externa** miden la calidad del agrupamiento contando inicialmente con información externa que es usada para escoger un algoritmo de *clustering* óptimo sobre un conjunto de datos específico.

Las técnicas de **validación interna** miden el *clustering* basándose únicamente en la información de los datos. Evalúan la calidad de la estructura del agrupamiento sin necesidad de información ajena al propio algoritmo y su resultado. Las métricas de validación interna

pueden emplearse para seleccionar el mejor algoritmo de *clustering*, así como el número óptimo de clústeres sin ningún tipo de información externa.

En este proyecto empleamos para la validación del clustering una técnica de validación interna que se denomina **Coeficiente de Silhouette** [14]. Esta técnica emplea una métrica de validación basada en los conceptos de cohesión y separación. La **cohesión** hace referencia a la cercanía de cada miembro del clúster con el resto de los miembros de dicho clúster, mientras que la **separación** se centra en la distancia que separa a cada clúster del resto de clústeres.

El Coeficiente de Silhouette para un agrupamiento se calcula con el siguiente procedimiento:

$$SC = \frac{1}{N} \sum_{i=1}^{N} s(x)$$

Donde *s(x)* es el Coeficiente de Silhouette para un punto *x*:

$$s(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}$$

Y donde:

- $a(x)$ es la cohesión; la distancia promedio de *x* a todos los demás puntos en el mismo clúster.
- $b(x)$ es la separación; la distancia promedio de *x* a todos los demás puntos en el clúster más cercano.

El valor de *s(x)* puede variar en el intervalo [-1, 1], en el cual cada valor indica:

- -1 = mal agrupamiento
- 0 = indiferente
- 1 = bueno

Con todo esto, lo que se pretende en la fase de análisis es la maximización de esta métrica.

## *2.2 API Vision de Google Cloud*

La API Vision es una herramienta desarrollada por Google capaz de analizar el contenido de las imágenes mediante el encapsulado de potentes modelos de aprendizaje automático en una API REST, la cual, emplea operaciones HTTP POST para realizar el análisis de las imágenes que se envían en la solicitud [6].

Las principales características con las que cuenta son:

- **Detección de etiquetas**. Detecta distintos tipos de elementos presentes en la imagen y los categoriza extrayendo a su vez el porcentaje de predominio de cada uno de ellos.

- **Detección de contenido explícito y logotipos**. Detecta contenido inapropiado y logotipos de productos famosos.

- **Detección de puntos de referencia**. Detecta estructuras artificiales y naturales famosas.

- **Detección de caras**. Detecta las caras que aparecen en la imagen y extrae los atributos faciales, emociones o prendas, aunque no ofrece reconocimiento facial.

- **Reconocimiento óptico de caracteres (OCR)**. Detecta y extrae texto de una imagen en diversos idiomas.

- **Detección de atributos de la imagen**. Detecta diversos atributos, como por ejemplo el color dominante, y ofrece sugerencias de recorte de la imagen.

- **Detección web**. Busca imágenes similares en Internet.

Otras herramientas muy similares a la que empleamos en este trabajo son: Cognitive Services [7] y Caffe [8]. La primera ha sido desarrollada por Microsoft y es la competencia más directa que recibe la API Vision de Cloud debido a la gran similitud en el servicio que ofrecen. La segunda herramienta es un entorno de trabajo de *Deep Learning* desarrollado por investigadores de la Universidad de Berkeley, que supondría la alternativa *open source* a la API Vision de Cloud.

Para este sistema hemos elegido API Vision debido a que la librería de cliente que ofrece simplifica la construcción y el envío de solicitudes. Además, su uso e integración es muy sencillo y está desarrollado para una gran variedad de lenguajes de programación.

## *2.3 Clustering*

El Clustering es "el subcampo del Aprendizaje Automático No Supervisado que tiene como objetivo dividir los conjuntos de datos no etiquetados en grupos consistentes, o clústeres, en función de algunas características desconocidas que comparten" [2].

Los problemas a los que se aplica el Clustering se caracterizan por dos conceptos clave diferentes: **la similitud métrica** y **el proceso de agrupamiento** [3]. El primero está relacionado con el estudio de las diferentes métricas que miden la similitud entre las distintas instancias del conjunto de datos. Dentro de este concepto se pueden encontrar dos enfoques diferentes: algoritmos de *clustering* basados en la **distancia** o en la **densidad**. El objetivo del primero es minimizar la suma de las distancias cuadradas medidas por las distancias Euclídea, de Minkowski o Mahalanobis, entre otras. El objetivo de los algoritmos de *clustering* basados en la densidad es agrupar las instancias del conjunto de datos en función de la compactación entre los elementos de los clústeres resultantes.

El segundo concepto clave es el proceso de agrupamiento, para el cual, también hay dos enfoques diferentes dependiendo de cómo son construidos los distintos clústeres: **agrupamiento jerárquico** y **agrupamiento particional**. El agrupamiento jerárquico intenta crear una jerarquía de clústeres y cuenta con dos métodos diferentes de realizar esta tarea:

1. **Aglomerativo**. Inicialmente, el algoritmo maneja N clústeres diferentes compuestos por un solo elemento del conjunto de datos. El algoritmo intenta juntar los diferentes clústeres hasta que todos los elementos pertenecen al mismo clúster. Esta técnica recibe el nombre de acercamiento ascendente, o *bottom-up*.

2. **Divisivo**. Al contrario que el anterior, es una técnica de acercamiento descendente, o *top-down*. Sólo hay un clúster que contiene todos los elementos del conjunto de datos. En este caso, el algoritmo intenta dividir el clúster hasta que obtenga un clúster por cada elemento del conjunto de datos.

En cualquier algoritmo de *clustering*, dado un conjunto de datos compuestos por N elementos, se crean K clústeres ($N \geq K$) de tal manera que cada clúster contenga al menos un elemento, y cada elemento está asignado a un solo grupo. Los algoritmos más famosos son *K-means* [17], *K-medoids* [18], y el algoritmo *Expectación-Maximización* (EM) [19]. Sin embargo, en este trabajo vamos a emplear el algoritmo DBSCAN [4] dado que se adapta mejor a la métrica actual del sistema desarrollado y no nos presenta tantos problemas como *K-means*.

## *2.4 DBSCAN*

El algoritmo DBSCAN es un algoritmo de *clustering* basado en la densidad espacial, que define los clústeres como áreas de alta densidad de puntos separadas por áreas de baja densidad de puntos o ruido. Dado que esta visión es muy genérica, los clústeres encontrados por DBSCAN pueden tener cualquier forma, a diferencia del algoritmo *K-means* que asume que los clústeres tienen forma convexa.

El componente principal de DBSCAN son los puntos, donde cada uno representa una muestra del conjunto de datos. Por lo tanto, un clúster es un conjunto de puntos denominados **puntos núcleo** cercanos entre sí en una medida de distancia determinada. Los puntos que no forman parte de un clúster, al no cumplir el requisito determinado por la medida de distancia, se denominan **ruido**.

Para definir la **densidad** con la que trabaja el algoritmo a la hora de formar los clústeres es necesario determinar dos parámetros: una ***épsilon*** ($\varepsilon$) positiva y el número mínimo de puntos necesarios para formar un cluster, ***minPtos***.

El funcionamiento del algoritmo consiste en seleccionar un punto arbitrario del conjunto de datos, comprobar si hay una cantidad de puntos mayor o igual que *minPtos* a una distancia *épsilon* de dicho punto, y en caso de cumplirse esa condición se consideran a todos esos puntos como parte de un clúster. A continuación, se expande ese grupo mediante la comprobación anterior y tomando como referencia los nuevos datos incluidos en el cluster. Si para ellos también se cumple con otros puntos, el clúster va creciendo. Este procedimiento se repite para todos los puntos del conjunto de datos, por lo que los puntos que no cumplen la condición no pertenecen a ningún clúster y se consideran ruido [4, 5].

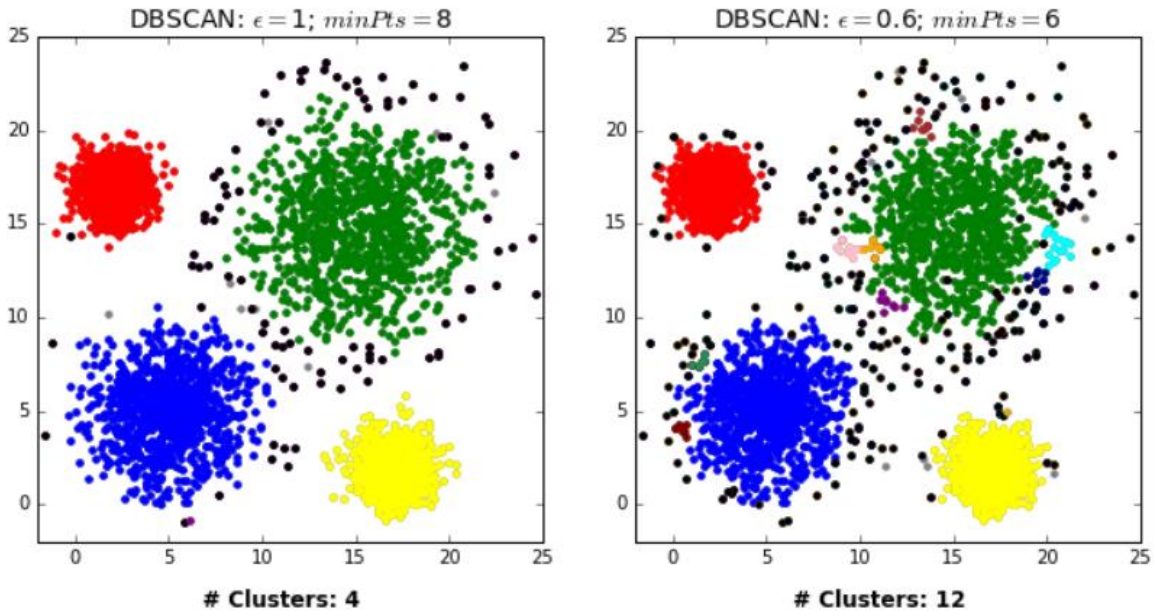


**Figura 3. Ejemplo DBSCAN con distintos *épsilon* y *minPtos* [2]**

# 3 Diseño y desarrollo del sistema

Este capítulo describe el sistema desarrollado durante este trabajo y cuyo objetivo es extraer y analizar información de la red social Instagram para obtener conocimiento de ella.

## *3.1 Arquitectura*

La estructura del sistema desarrollado se divide en cinco procesos:

1. **Extracción de imágenes.** Extrae un conjunto de imágenes de Instagram pertenecientes a un usuario o a un *hashtag*, haciendo uso del *crawler* InstaLooter, y lo almacena localmente.

2. **Procesamiento de imágenes**. Toma un conjunto de imágenes almacenadas previamente y lo procesa a través de la API Vision para obtener los datos de cada imagen.

3. **Preprocesamiento de los datos**. Este módulo tiene como objetivo construir el vector de características que define a cada imagen del *dataset*. Entonces, utiliza las etiquetas recibidas de API Vision para crear el vector de manera similar al procesamiento de texto con *Bag of Words* [22].

4. **Proceso de análisis**. Aplica el algoritmo DBSCAN sobre los vectores de características construidos por el módulo anterior.

5. **Valoración de los resultados**. Lleva a cabo la validación de los resultados del algoritmo DBSCAN mediante el Coeficiente de Silhouette.

Como se puede observar en la **Figura 4**, el sistema está compuesto por tres módulos independientes entre sí: **Extracción**, **Preprocesamiento** y **Análisis**. El primero está formado por el programa *crawler* InstaLooter y la red social Instagram por lo que se encarga de llevar a cabo las tareas de extracción de contenido. El segundo está compuesto por la API Vision de Google Cloud y los mecanismos de preprocesamiento de los datos para encargarse de este tipo de procesos. El último lo componen el algoritmo de *clustering* DBSCAN y el método de validación del Coeficiente de Silhouette, por lo que se encarga de llevar a cabo las tareas de valoración de los resultados.

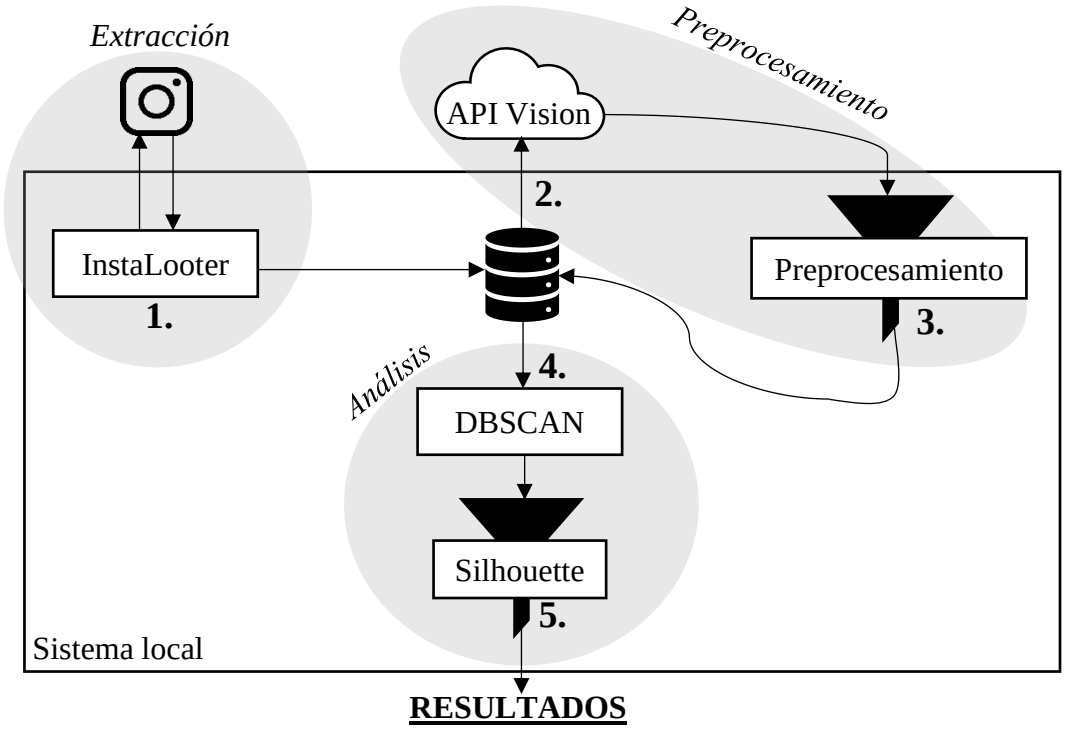


**Figura 4. Diseño del sistema**

### *3.1.1 Extracción de imágenes*

La red social Instagram consta de una política de privacidad que permite el libre acceso a la información que contiene. Esto nos permite utilizar programas como InstaLooter para acceder a dicho contenido y extraerlo.

En este proceso se introduce el usuario o *hashtag* del que se quiere obtener el lote de imágenes. Además, el programa InstaLooter nos proporciona un gran número de opciones que se pueden emplear para filtrar el contenido que se desea extraer. En este sistema solo se han incorporado las siguientes opciones: extraer todo el contenido, extraer un número determinado de imágenes y extraer el contenido nuevo o que no se haya extraído anteriormente. El resto de las opciones se han considerado irrelevantes para la funcionalidad del sistema que se ha desarrollado.

### *3.1.2 Procesamiento de imágenes*

El conjunto de imágenes extraído en el proceso anterior se almacena localmente para poder realizar su preprocesamiento mediante la librería de la API Vision de Google Cloud. Esta librería de cliente es proporcionada por Google para simplificar el proceso de construcción y envío de solicitudes, así como el de recibir y analizar respuestas. Para poder hacer uso de esta tecnología es necesario registrar el proyecto en la plataforma Google Cloud Platform Console [21]. Esta plataforma, enfocada en la tecnología de la nube, permite la administración de recursos y datos de manera centralizada, además de la utilización de las distintas soluciones que ofrece.

Una vez registrado el proyecto en la plataforma obtenemos las credenciales necesarias para utilizar la API Vision. Para ello, se selecciona un conjunto de imágenes y se va realizando el análisis individual de cada una de ellas mediante solicitudes a la API, utilizando los mecanismos que nos facilita la librería. Como resultado del análisis obtenemos las etiquetas que definen cada una de las imágenes para su posterior preprocesamiento.

### *3.1.3 Preprocesamiento de los datos*

Este proceso se lleva a cabo automáticamente después de realizar el procesamiento de las imágenes. Las distintas etiquetas extraídas del análisis de todas las imágenes se emplean para construir la representación de cada imagen. Para ello, se utiliza el modelo de representación **Bolsa de Palabras** (*Bag of Words*) [22], que se caracteriza por su facilidad de uso y su eficiencia computacional.

La tarea de preprocesamiento de estos datos en el sistema desarrollado consiste en utilizar el conjunto con todas las etiquetas extraídas para comprobar cuales están presentes en la representación de cada imagen. De esta manera obtenemos como resultado un archivo de texto por cada conjunto de imágenes que se analiza, el cual contiene el conjunto de etiquetas extraídas y una representación de cada imagen con unos y ceros en función de las etiquetas que contenga, siguiendo la representación de Bolsa de Palabras y con la misma dimensión del conjunto de etiquetas. Finalmente, este proceso termina con el almacenamiento local del archivo generado para poder utilizarlo en el módulo siguiente.

### *3.1.4 Proceso de análisis*

Este proceso es el encargado de llevar a cabo las técnicas de *clustering* sobre los archivos que contienen los datos preprocesados. Para ello hace uso del algoritmo DBSCAN, que nos permite llevar a cabo los análisis basándose en la densidad de los clústeres que se forman. Mediante esta técnica, podemos obtener el conocimiento que se esté buscando de cada conjunto de imágenes que analicemos, para posteriormente validarlo. Dependiendo de los parámetros *épsilon* y *minPtos* que se introduzcan para ejecutar el algoritmo DBSCAN los resultados sufrirán variaciones, en cambio, siempre se obtendrá el mismo resultado al emplear el mismo *dataset* junto a los mismos valores para *épsilon* y *minPtos*.

El resultado del *clustering* varía significativamente para un mismo *dataset* si cambian los valores de *épsilon* y *minPtos*, por lo que la definición del valor de estos dos parámetros es una tarea clave para el correcto funcionamiento del sistema.

### *3.1.5 Valoración de resultados*

Una vez obtenidos los resultados del análisis se ejecuta automáticamente este proceso que realiza la validación de dichos resultados. Para llevar a cabo esta tarea de validación se emplea la técnica del Coeficiente de Silhouette, la cual, utiliza una métrica que estudia la cohesión y separación de los clústeres resultantes de realizar el *clustering*. Como resultado de aplicar esta técnica obtendremos un valor dentro del intervalo [-1, 1]. La interpretación de este resultado sería la siguiente:

- Si el valor obtenido es cercano a ‘-1’ esto significa que los resultados del análisis no reflejan un buen agrupamiento, por lo que se podría considerar que los parámetros empleados para realizar el *clustering* no son los adecuados.

- Cuando obtenemos un valor cercano a ‘0’ significa que el resultado del análisis no está aportando un buen grado de conocimiento final.

- Si el valor obtenido es cercano a ‘1’ se puede considerar que los resultados obtenidos son buenos y aportan conocimiento.

Por todo esto, el objetivo será maximizar el valor del Coeficiente de Silhouette, lo que significa que los clústeres resultantes están bien formados.

## *3.2 Desarrollo*

El desarrollo del sistema propuesto se ha realizado en el lenguaje Python debido a su versatilidad y se ha desarrollado en forma de aplicación de consola. Esta aplicación consiste en un menú principal que contiene las diferentes opciones que ofrece el sistema desarrollado, tal y como se muestra en la **Figura 5**.

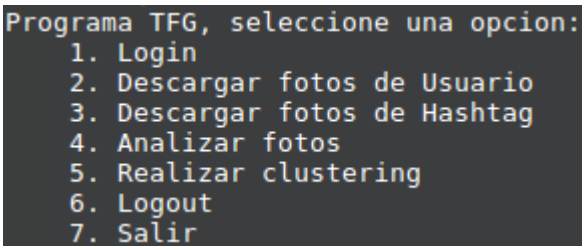


**Figura 5. Menú Principal**

Las opciones **Login**, **Logout** y **Descargar fotos** pertenecen al módulo de Extracción. Este módulo esta implementado de forma que el usuario primero introduce por teclado los datos necesarios para cada operación y posteriormente realiza la llamada al programa InstaLooter. Siempre que se realiza una extracción de contenido se crea un directorio llamado *data* en la misma dirección donde se ejecuta el sistema. Dentro de este directorio se van creando subdirectorios con el nombre del usuario o del *hashtag* cuyo contenido se está extrayendo. De esta manera quedan almacenados localmente para posteriores análisis o actualizaciones del contenido.

La opción de **Analizar fotos** pertenece al módulo de Preprocesamiento. Para el correcto funcionamiento de este módulo, es necesario cargar previamente en una variable de entorno la dirección del archivo JSON que contiene las credenciales para usar la librería de la API Vision, que obtenemos de nuestro proyecto en la plataforma de Google Cloud. Después el usuario introduce el nombre del subdirectorio de la carpeta *data* que desea analizar, lo cual forma un conjunto de imágenes que se van analizando individualmente para extraer todas las etiquetas existentes en el subdirectorio.

Las etiquetas se extraen empleando las funciones que nos proporciona la librería de cliente de la API Vision y son almacenadas en diferentes *arrays*, uno general que contiene todas las etiquetas del lote y el resto que son las etiquetas que tiene cada imagen.

Una vez analizado todo el contenido del directorio, se realiza el preprocesamiento de los datos. Para ello, se van recorriendo todos los *arrays* de las fotos comprobando qué etiquetas contienen del conjunto de etiquetas global. Por cada etiqueta que contiene se evalúa con el valor '1' y en caso contrario '0'. Finalmente, se obtiene como resultado un archivo de texto formado por el conjunto de etiquetas global y el de cada imagen acompañado de su respectivo identificador. Este archivo queda almacenado localmente dentro del directorio *results*, que al igual que el directorio *data*, se crea en la misma dirección donde se ejecuta el sistema.

Un ejemplo de la funcionalidad de este módulo de Preprocesamiento sería el presentado por la **Figura 6**. Obtenemos dos imágenes: una que contiene un rectángulo y un círculo; y otra que contiene un círculo y un triángulo. Se procesan las imágenes mediante la API Vision y se obtienen las etiquetas del conjunto de imágenes. Después se realiza el preprocesamiento para formar los vectores de características de cada imagen y almacenar todo el contenido en un archivo de texto que se almacena en el directorio *results*.

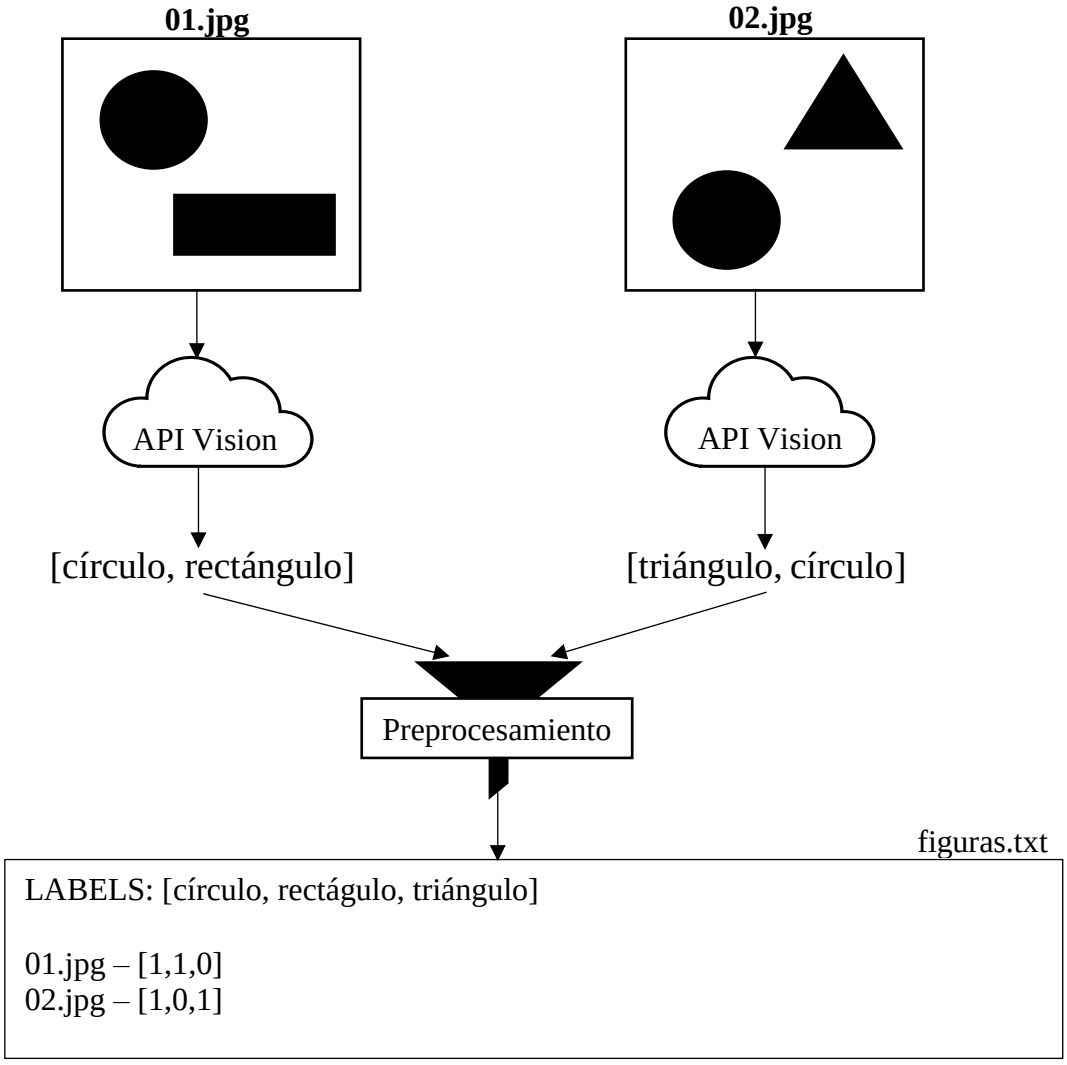


**Figura 6. Ejemplo de la funcionalidad del módulo de Preprocesamiento**

La opción de **Realizar clustering** pertenece al módulo de Análisis. Para llevar a cabo las técnicas de *clustering* de este módulo hacemos uso de las herramientas desarrolladas en Python que se ofrecen en scikit-learn [23] dado que todas están enfocadas al Aprendizaje Automático y son *open source*. En específico, hacemos uso del módulo *sklearn.cluster*, ya que es el que contiene la implementación del algoritmo DBSCAN, y de *sklearn.preprocessing*, para estandarizar los datos de entrada que se le pasan al algoritmo.

El funcionamiento de esta opción consiste en que el usuario introduzca el nombre del archivo con los datos preprocesados del conjunto de imágenes que se quiere analizar para que el sistema los introduzca en el algoritmo DBSCAN. En este trabajo, para probar la funcionalidad del sistema se ha elaborado una simple lógica que permita detectar las fotos más similares entre sí dentro del conjunto de imágenes mediante el algoritmo DBSCAN.

Después de estandarizar los datos para introducirlos correctamente en el algoritmo, este nos proporcionará un resultado que analizaremos empleando el método del Coeficiente de Silhouette. Esta metodología de validación se encuentra implementada dentro del módulo *sklearn* lo que simplifica notablemente el desarrollo del sistema.

Finalmente, se obtiene un valor que estará dentro del intervalo [-1, 1] y el usuario tendrá que interpretar si el resultado es óptimo o es necesario realizar algún cambio en los parámetros del algoritmo empleado.

# 4 Pruebas y resultados

En este apartado se detallan las pruebas realizadas para comprobar el correcto funcionamiento del sistema a través de la aplicación de consola. También se exponen los resultados obtenidos tras ejecutar el sistema con usuarios y *hashtags* reales.

El tipo de pruebas realizadas son pruebas unitarias en cada módulo que compone el sistema. No se han llevado a cabo pruebas de integración dado que no existe un único proceso continuo que una todos los módulos, sino que cada uno funciona independientemente del resto.

## *4.1 Pruebas del módulo de Extracción*

En este módulo se encuentran las opciones del menú principal de Login, Logout y Descargar fotos de Usuario o de Hashtag. Para poder realizar las pruebas de este módulo, el equipo de AIDA nos proporciona una cuenta de usuario de Instagram que podemos utilizar.

### *4.1.1 Prueba Login*

Introducimos los datos necesarios que nos solicita el sistema, en este caso, el nombre de usuario de nuestra cuenta de Instagram y su contraseña. Una vez introducidos los datos, el programa InstaLooter realiza la instrucción correspondiente y nos muestra un mensaje con el resultado de la operación.

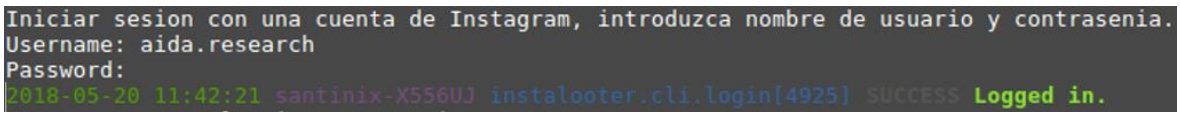


**Figura 7. Prueba Login**

### *4.1.2 Prueba Descargar fotos de Usuario*

El sistema nos solicita introducir el nombre del usuario del que vamos a extraer el contenido, en este caso, hacemos las pruebas sobre la cuenta de Instagram cuyo nombre de usuario es *thesantinix*. Una vez realizado este paso se elige una de las opciones que nos ofrece el sistema:

- **Descargar todas las fotos**. Extraemos todo el contenido de la cuenta *thesantinix*, en este caso formado por 14 fotos.

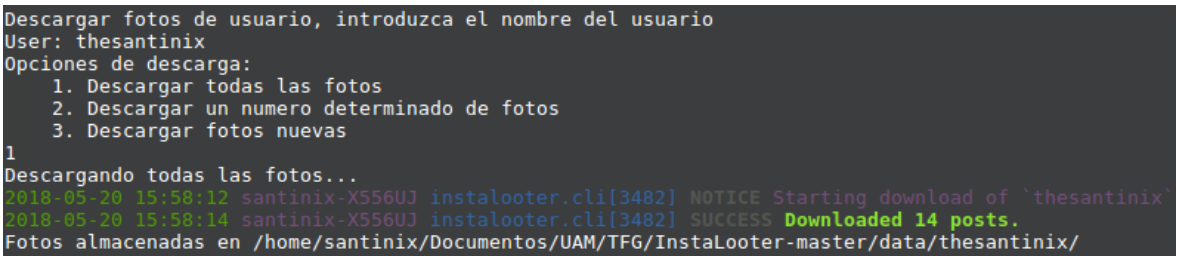


**Figura 8. Prueba Extracción de Usuario Completo**

- **Descargar un número determinado de fotos**. En esta opción el sistema solicita que se introduzca el número de imágenes que se quieren extraer, por lo que probamos a realizar la extracción de 8 imágenes de la cuenta *thesantinix*.

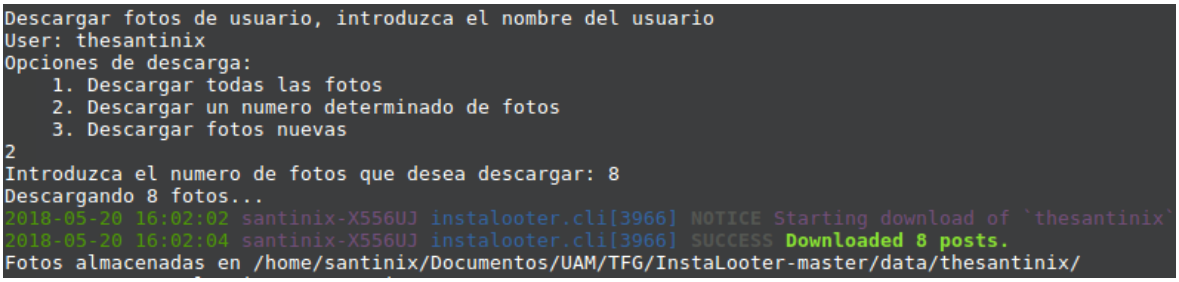


**Figura 9. Prueba Extracción de Usuario con Número**

- **Descargar fotos nuevas**. Extraemos las imágenes de la cuenta *thesantinix* que no estén ya almacenadas localmente en nuestro sistema. En este caso, previamente llevamos a cabo la eliminación de las últimas 4 imágenes pertenecientes al perfil objetivo en nuestro almacenamiento local, por lo que el sistema extrae nuevamente esas imágenes.

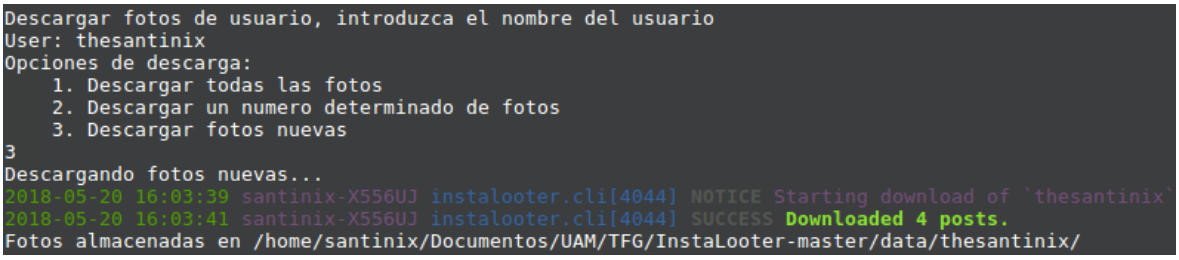


**Figura 10. Prueba Extracción de Usuario con Imágenes Nuevas**

Por último, el programa muestra un mensaje con el resultado de la operación y las imágenes extraídas. También se indica la dirección del subdirectorio en el que quedan almacenadas localmente las imágenes.

### *4.1.3 Prueba Descargar fotos de Hashtag*

Las pruebas de este proceso son muy similares a las del punto anterior. El sistema nos solicita introducir el nombre de un *hashtag* del que queremos extraer el contenido y después seleccionamos una de las opciones que nos ofrece el sistema:

- **Descargar todas las fotos.** Para esta prueba hemos elegido un *hashtag* poco utilizado, como por ejemplo *padsoft*, dado que la gran mayoría tienen una gran cantidad de contenido y no nos es necesario para elaborar esta comprobación en la que se extrae la totalidad del contenido.

```
Descargar fotos de un hashtag, introduzca el nombre del hashtag
Hashtag: padsoft
Opciones de descarga:
    1. Descargar todas las fotos
    2. Descargar un numero determinado de fotos
    3. Descargar fotos nuevas
1
Descargando todas las fotos...
2018-05-20 16:09:08 santinix-X556UJ instalooter.cli[4341] NOTICE Starting download of `padsoft`
2018-05-20 16:09:10 santinix-X556UJ instalooter.cli[4341] SUCCESS Downloaded 7 posts.
Fotos almacenadas en /home/santinix/Documentos/UAM/TFG/InstaLooter-master/data/padsoft/
```

**Figura 11. Prueba Extracción de Hashtag Completo**

- **Descargar un número determinado de fotos**. En esta opción el sistema vuelve a solicitar introducir el número de imágenes que se quieren extraer. En este caso, realizamos la prueba con cualquier otro *hashtag* ya que no necesitamos extraer todo el contenido. Por ello, elegimos el hashtag *playa* y extraer 20 imágenes.

```
Descargar fotos de un hashtag, introduzca el nombre del hashtag
Hashtag: playa
Opciones de descarga:
    1. Descargar todas las fotos
    2. Descargar un numero determinado de fotos
    3. Descargar fotos nuevas
2
Introduzca el numero de fotos que desea descargar: 20
Descargando 20 fotos...
2018-05-20 16:06:36 santinix-X556UJ instalooter.cli[4171] NOTICE Starting download of `playa`
2018-05-20 16:06:43 santinix-X556UJ instalooter.cli[4171] SUCCESS Downloaded 20 posts.
Fotos almacenadas en /home/santinix/Documentos/UAM/TFG/InstaLooter-master/data/playa/
```

**Figura 12. Prueba Extracción de Hashtag con Número**

- **Descargar fotos nuevas**. Extraemos las imágenes del *hashtag playa* que no estén almacenadas localmente a partir de la imagen más reciente que se tenga almacenada. En este caso se añaden 18 nuevas imágenes desde la última que se almacenó.

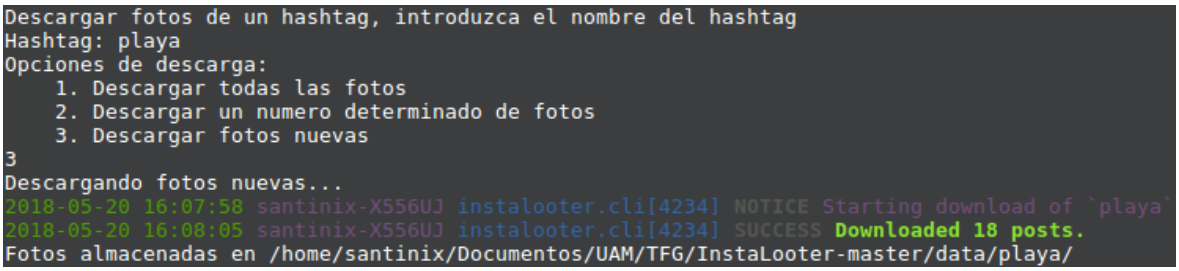


**Figura 13. Prueba Extracción de Hashtag con Imágenes Nuevas**

Por último, y al igual que en el punto anterior, el programa muestra un mensaje con el resultado de la operación, las imágenes extraídas y la dirección del subdirectorio donde quedan almacenadas localmente.

### *4.1.4 Prueba Logout*

El programa InstaLooter realiza la instrucción del cierre de sesión de la cuenta con la que se realizó la prueba de Login previamente y nos muestra un mensaje con el resultado de dicha operación.

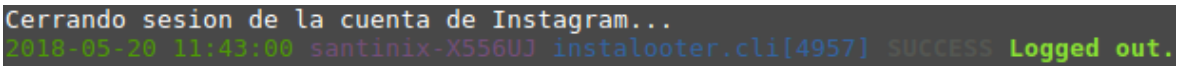


**Figura 14. Prueba Logout**

## *4.2 Pruebas del módulo de Preprocesamiento*

En este módulo se encuentra la opción de Analizar fotos, por lo que vamos a llevar a cabo la prueba con las imágenes extraídas de la cuenta *thesantinix* para comprobar su correcto funcionamiento. El sistema nos solicita introducir el nombre del subdirectorio donde se encuentra el conjunto de imágenes que se quiere analizar. Después, el sistema realiza el procesamiento con la API Vision y el preprocesamiento de los datos resultantes. El sistema nos muestra mensajes del proceso que se está llevando a cabo y la dirección en la que se almacena localmente el archivo resultante.

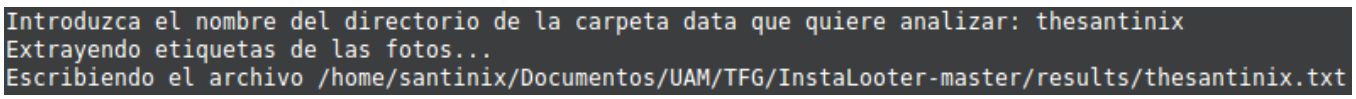


**Figura 15. Prueba Preprocesamiento**

## *4.3 Pruebas del módulo de Análisis*

En este módulo se encuentra la opción de Realizar clustering, por lo que vamos a probar a analizar con DBSCAN un archivo con los resultados del preprocesamiento de un conjunto de mil imágenes extraídas del *hashtag playa*. Al ser un conjunto tan grande de imágenes se han obtenido un total de 1293 etiquetas distintas, por lo que el vector de características de cada imagen tiene esa misma dimensión. Esto hace que la tarea de representación gráfica se complique mucho y por lo tanto se ha optado por no llevar a cabo dicha representación. El objetivo de esta prueba es analizar que imágenes son más parecidas entre sí, es decir, a cuáles les coinciden el mayor número de etiquetas, y viendo los distintos resultados que se obtienen al probar con distintos parámetros de entrada. Al emplear el algoritmo DBSCAN los parámetros de entrada son *épsilon* y *minPtos*. En cuanto al resultado, valoramos el resultado obtenido del Coeficiente de Silhouette para cada variación de los análisis.

| ***épsilon*** | ***minPtos*** | **Coeficiente de Silhouette** |
|---|---|---|
| 0.3 | 5 | 0.085 |
| 0.4 | 5 | 0.104 |
| 0.5 | 15 | 0.502 |
| 0.5 | 6 | 0.672 |
| 0.6 | 15 | 0.683 |
| 0.7 | 15 | 0.701 |
| 0.8 | 20 | 0.701 |
| 0.6 | 7 | 0.714 |
| 0.7 | 10 | 0.722 |
| 0.8 | 10 | **0.741** |

**Tabla de Resultados de la Prueba de Análisis**

Como se puede observar, los resultados obtenidos varían en función de la distancia (*épsilon*) y el número de puntos (*minPtos*) que debe incluir para considerarse miembro de un clúster. La variación de estos parámetros tiene que buscar siempre acercarse lo máximo posible a el valor ‘1’ del Coeficiente de Silhouette, ya que esto significa que un mayor número de puntos se habrán clasificado en un clúster y por la tanto, más información útil aportarán al análisis.

En esta tabla se ha resaltado el mejor valor obtenido que se corresponde con una configuración de **épsilon**=0.8 y formando clusters que contengan mínimo 10 puntos (**minPtos=10)**. Como se puede observar, el resultado del clustering tiene un coeficiente de Silhouette de 0.741, el cual, además de estar cercano al ‘1’, es el valor más alto obtenido para las diferentes configuraciones.

# 5 Conclusiones y trabajo futuro

## *5.1 Conclusiones*

En este trabajo se ha implementado un sistema que extrae y analiza información de la red social Instagram. Para llevar a cabo la extracción de contenido de la red social se ha empleado un *crawler* llamado InstaLooter que no hace uso de la API de Instagram y por lo tanto permite no estar sujeto a restricciones. En cuanto a la parte de análisis, se ha utilizado la API Vision de Google Cloud para el procesamiento del contenido debido a la gran cantidad de posibilidades que ofrece y la facilidad de uso e integración de su librería de cliente. A parte se ha desarrollado un módulo que emplea técnicas de *clustering* para hacer una simple demostración de las posibilidades que puede ofrecer el sistema implementado. En este caso, se ha hecho uso del algoritmo DBSCAN y se ha empleado la métrica de validación del Coeficiente de Silhouette para valorar los resultados obtenidos.

El sistema se ha diseñado y desarrollado teniendo en mente que en un futuro pueda ser integrado en otro sistema más potente y con una interfaz de usuario propia. Por ese motivo, para realizar las pruebas únicamente se ha desarrollado una simple aplicación de consola. Como resultado de la realización de dichas pruebas podemos concluir que, por un lado, el rendimiento de la extracción de contenido ha resultado ser bastante superior al esperado inicialmente en la planificación del trabajo, debido al uso de un *crawler* que no emplea la API de Instagram y no está sujeto a las restricciones de extracción de contenido. Por otro lado en cambio, las pruebas realizadas en la parte de análisis con las técnicas de *clustering* nos hace concluir que es necesario encontrar una forma de representación gráfica óptima para el modelo de análisis empleado debido a las dimensiones que maneja.

## *5.2 Trabajo futuro*

Este sistema presenta un gran abanico de posibilidades de ampliación al formar parte de un campo tan extenso como es el de la Minería de Datos. Los módulos en los que más se podrían ampliar o mejorar los procesos del sistema son en los de Preprocesamiento y Análisis.

La API Vision de Google Cloud posee muchas opciones para el procesamiento de las imágenes. En este trabajo solo hemos extraído de dicho procesamiento las etiquetas que las definen, pero también se puede tener en cuenta el campo del porcentaje de fiabilidad para poder realizar *clustering* difuso; o ampliar las técnicas de análisis incluyendo otras características como por ejemplo la detección de caras para el análisis de emociones, la detección de textos para localizar documentos, el color predominante de las imágenes, la detección de logotipos, etc.

En cuanto a la parte del preprocesamiento de los datos, sigue el método de representación de *Bag of Words*, el cual se caracteriza por su gran sencillez, pero esto hace que el módulo de clustering trabaje con una matriz de gran dimensionalidad donde muchos de los valores son ‘0’ (es una matriz *sparse*). Por lo que esta representación es bastante ineficiente. Debido a esto, una buena opción de mejora o ampliación futura sería la elaboración de una representación más óptima y mecanismos de preprocesamiento más potentes en función del análisis que se quiera llevar a cabo.

Por último, también se podría ampliar enormemente el módulo de análisis, incorporando otros algoritmos de *clustering* más novedosos como podrían ser algoritmos de *clustering* bio-inspirado.

# Referencias


1. "Estadísticas de redes sociales 2018: Usuarios de Facebook, Twitter, Instagram, YouTube, LinkedIn, Whatsapp y otros". Disponible en: http://www.juancmejia.com/marketing-digital/estadisticas-de-redes-sociales-usuarios-de-facebook-instagram-linkedin-twitter-whatsapp-y-otros-infografia/. Último acceso en: 13-Mayo-2018

2. "Clustering with Scikit with GIFs". Disponible en: https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/. Último acceso en: 14-Mayo-2018

3. Gonzalez-Pardo *et al.*, "ACO-based clustering for Ego Network analysis", Future Generation Computer Systems 66, pp. 160 – 170, 2017.

4. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", 1996

5. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python", JMLR 12, pp. 2825-2830, 2011

6. "API Vision de Cloud". Disponible en: https://cloud.google.com/vision/?hl=es. Último acceso en: 14-Mayo-2018

7. "Directorio de Cognitive Services" Disponible en: https://azure.microsoft.com/es-es/services/cognitive-services/directory/vision/. Último acceso en: 14-Mayo-2018

8. "Caffe". Disponible en: http://caffe.berkeleyvision.org/. Último acceso en: 14-Mayo-2018

9. D. T. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining". John Wiley and Sons, 2005

10. "Los Crawlers o recolectores de la Web". Disponible en: http://sistemasinformaticos1213.blogspot.com.es/2013/05/los-crawlers-o-recolectores-de-la-web.html. Último acceso en: 15-Mayo-2018

11. "Instagram API Platform". Disponible en: https://www.instagram.com/developer/. Último acceso en: 15-Mayo-2018

12. "Instagram Graph API". Disponible en: https://developers.facebook.com/docs/instagram-api. Último acceso en: 15-Mayo-2018

13. C. Pérez López y D. Santín González, "Minería de Datos: Técnicas y Herramientas". Editorial Paraninfo, 2007

14. P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, Volumen 20, pp. 53-65, 1987

15. E. León Guzmán, “Métricas para la validación de Clustering”. Disponible en: http://www.disi.unal.edu.co/profesores/eleonguz/cursos/md/presentaciones/Sesion13_validacion_Clustering.pdf. Último acceso en: 16-Mayo-2018

16. “InstaLooter”. Disponible en: http://instalooter.readthedocs.io/en/latest/index.html. Último acceso en: 17-Mayo-2018

17. D. MacKay, “Information Theory, Inference and Learning Algorithms”, Cambridge University Press, 2003

18. L. Kaufman, P. J. Rousseeuw, “Finding Groups in Data: An Introduction to Cluster Analysis”, Vol. 344, John Wiley and Sons, 2009

19. A. P. Dempster, N. M. Laird, D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm”, Journal of the Royal Statistical Society, Series B (Methodological), Vol. 39, No. 1, pp. 1-38, 1977

20. L. C. Molina, “Data Mining: Torturando a los datos hasta que confiesen”, Universitat Oberta de Catalunya, 2002

21. “Google Cloud Platform Console”. Disponible en: https://cloud.google.com/cloud-console/?hl=es. Último acceso en: 19-Mayo-2018

22. F. Camastra, J. A. Hernandez, P. Kokol, J. Wang, and S. Zhu, “Bag-of-Words Representation in Image Annotation: A Review”, ISRN Artificial Intelligence, 2012

23. “scikit-learn”. Disponible en: http://scikit-learn.org/stable/index.html. Último acceso en: 23-Mayo-2018

# Glosario

| | |
|---|---|
| **API** | Application Programming Interface |
| **DBSCAN** | Density-Based Spatial Clustering of Applications with Noise |
| **EM** | Expectation Maximization |
| **HTTP** | Hypertext Transfer Protocol |
| **JSON** | JavaScript Object Notation |
| **OCR** | Optical Character Recognition |
| **REST** | Representational State Transfer |